THE UNIVERSITY *of* EDINBURGH

Computational Methods for the Quantification of RNA Transcript Abundance and Messenger RNA Regulation

Samuel Joseph Haynes

Doctor of Philosophy

School of Biological Sciences
University of Edinburgh

2023

To Angela, Wayne, Charlotte,
and all the fools who call me friend.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification.

Samuel Joseph Haynes

2023

Acknowledgements

This entire body of work would have been impossible without the tireless support of past and present members of the Wallace lab. In a literal sense, this work would not have existed without the experiments conducted with expertise and insight by Jamie and Weronika on top of their own projects during a period of global turmoil. I will be eternally grateful and I want you to know how exceptional you are. Although having never worked in the lab, I have to trust the work was done in yeast.

I would like to thank the students I've mentored over the years for their incredible work and for reminding me of how cool science is: Abhishak Jain, Hanqin Du, Jingqi Chong, and Xuejia Ke. Also, the long list of lab members who have created a warming and humorous community over the years: Clémence, Savvas, Ania, Louis, Apple, Amy, Domenico, Flic, Liz, and Rachael. Laura and Rosey who guided me through my crash course in molecular biology from the first day I entered the office. Marah for always finding time to listen and for providing ample opportunities to consume alcohol.

I would also like to thank my co-supervisor Guido Sanguinetti for patiently answering my questions during the early days and for helping to push it over the line at the end. And finally, my supervisor Edward Wallace for gambling on a stellar astrophysicist as his first PhD student in an experimental fungal transcriptomics lab. I hope it paid off.

An honourable mention has to go to my parents and sister – and their pets that I am highly allergic to – who have had to endure me during this time. Your whisky and treats are in the post. I will leave the allocation to Hector. As for all the friends and flatmates – and half of the baristas in the Lothian area – who have had the pleasure of my company over the last four years, you're welcome.

Abstract

Experiments investigating the regulation of RNA transcripts have been revolutionised by technology developed over the last 40 years. The data acquired from these experiments have revealed novel regulatory mechanisms for the localisation, degradation and modification of RNA transcripts. However, the volume and complexity of the data sets have led to an unprecedented reliance on statistical software. Inadequate analysis of data sets is contributing to the ongoing crisis around reproducing conclusions from published research. Rigorous implementation of statistical analysis software can continue to uncover novel regulatory mechanisms, but closed, obscure, and incorrect analyses will propagate the reproducibility crisis to unassailable new heights.

The objective of this research project is to develop open-source software and implement reproducible analyses to enable further exploration of regulatory mechanisms acting on RNA transcripts. This thesis focuses on the analysis of transcriptomics data sets, predominately from the model organism *Saccharomyces cerevisiae*. This first project discusses the standardisation of the analysis of qPCR data. The chapter compares the R package tidyqpcr, developed by the author, to other current software available. This case highlights how quality software supported by comprehensive documentation can improve the quality of an entire experimental assay. The next chapter showcases how the implementation of quality analysis can detect subtle interactions between regulatory motifs. The design of several reporter constructs using insights from published data sets shows how even short regulatory motifs can be affected by their overall context. The final results chapter outlines the development of a statistical software package to rigorously analyse noisy transcriptomic data from RNA-Seq assays exploring RNA localisation. The statistical software package uses a Bayesian hierarchical model of fractionation-based assays to overcome common biases in RNA-Seq data sets.

In summary, this thesis presents and implements two examples of research software that improve the reproducibility and quality of conclusions from data acquired from common experimental assays in molecular biology. The thesis also outlines how to implement open-source development practices and create inclusive documentation in an academic setting. Software developed within this framework is then used to elucidate subtle ways that cells regulate their transcriptome.

Lay Summary

A simple model for how cells regulate themselves begins with deoxyribonucleic acid (DNA) as the information storage molecule inherited over generations and ends with the proteins encoded by DNA that a cell uses to respond to its environment. Ribonucleic acid (RNA) is considered to be an auxiliary molecule that is used to facilitate the flow of information from DNA to ribosomes, the cellular machinery that creates proteins from amino acids. However, with less than 2% of the human genome encoding a protein, the model is over-simplistic as it emphasises the regulatory role of proteins over RNA. Even within this model, RNA is required to perform a multi-faceted role: the DNA template of a protein is transcribed as messenger RNA (mRNA) which transports it to a ribosome, the ribosomes themselves are predominately made of ribosomal RNA (rRNA), and the amino acids used to create proteins are carried by transfer RNA (tRNA). Beyond this model populations of non-coding RNA (ncRNA) continue to be discovered with distinct regulatory roles, including: long non-coding RNAs (lncRNA), microRNAs (miRNA), and small nuclear RNAs (snRNA).

Our understanding of the world of RNA has been revolutionised over the last 40 years by technology that has enhanced the extraction and quantification of different RNA populations. Experiments can now be designed to complete a wide range of tasks from carefully comparing specific RNA targets across large samples to exploring differences in entire populations of RNA transcripts across sub-cellular compartments. However, as the experiments have become more sensitive and the regulatory mechanisms of interest more subtle, the detection of biologically significant effects from experimental noise has become increasingly complex. Therefore, the demands on molecular biologists now include: biological knowledge and experimental skills to plan and execute an experiment; and programming and statistical skills to analyse the data they create. Biologists are equipped to meet the biological knowledge and experimental demands, but the demand for programming and statistical skills has yet to be met by sufficient training or funding. This in turn leads biologists to depend on proprietary software or to develop their own analysis scripts without the knowledge of best practices or understanding the implicit assumptions behind the methods they use. Closed, obscure, and incorrect analyses are

fueling an ongoing crisis around reproducing results in published papers.

This thesis outlines how best practices in software development and rigorous statistical analyses can contribute to more informative and reproducible experiments investigating the regulatory role of RNA. The thesis consists of three main results chapters. The first results chapter describes the development of a software package called tidyqpcr which analyses data from a key experiment in molecular biology. tidyqpcr uses comprehensive documentation and intuitive function design to empower biologists to conduct quality-controlled experiments and publish reproducible results. The second results chapter implements tidyqpcr, together with the rigorous analysis of several other experimental assays, to detect subtle interactions between short regulatory sequences within mRNA. The final results chapter introduces a novel statistical method to remove known biases in experiments designed to compare changes in RNA populations between sub-cellular compartments. This research contributes to our understanding of how cells regulate themselves through their finessed control of their RNA and provides open-source software to enable other researchers to enhance their own experiments.

Contents

Declaration iii

Acknowledgements iv

Lay Summary vi

Chapter 1: Introduction 1
1.1 Overview . . . 1
1.2 Contributions . . . 2
1.3 Thesis Layout . . . 2

Chapter 2: Preliminaries 4
2.1 Gene Expression Regulation . . . 4
2.1.1 Transcriptional regulation . . . 4
2.1.2 Post-transcriptional regulation . . . 6
2.2 Experiments Quantifying RNA Abundance . . . 8
2.2.1 qPCR . . . 8
2.2.2 Microarrays . . . 11
2.2.3 RNA-Seq . . . 12
2.3 Transcriptomic Data Analysis . . . 16
2.3.1 RNA-Seq data analysis . . . 16
2.3.2 Detecting differential expression . . . 17
2.3.3 Downstream analysis of transcriptomic data sets . . . 18
2.4 Research Software Engineering . . . 20
2.4.1 Software development practices . . . 20
2.4.2 Open source software development . . . 22
2.4.3 Software documentation . . . 22
2.4.4 Suggestions for improving software documentation . . . 24

Chapter 3: Materials and Methods 26
3.1 Statistical Methods . . . 26
3.1.1 Correlation metrics . . . 26
3.1.2 Linear regression . . . 27
3.1.3 Penalised linear regression . . . 28
3.1.4 Bayesian hierarchical models . . . 29
3.1.5 Gaussian processes . . . 29
3.1.6 Model selection . . . 30
3.1.7 Model evaluation . . . 31
3.1.8 Multiple hypothesis testing . . . 32

3.2 tidyqpcr Software Development . . . 33
3.2.1 User interviews . . . 33
3.3 Limitations of Composability of Cis-Regulatory Elements in Messenger RNA . . . 34
3.3.1 Strains and media . . . 35
3.3.2 Construction of chimeric reporter plasmids . . . 35
3.3.3 Fluorescence measurements: Plate reader analysis of strain growth and fluorescence . . . 35
3.3.4 RNA measurements: Strain growth, RNA extraction, RT-qPCR, RNA-Seq and analysis . . . 36
3.3.5 Determining 3'UTR decay motifs . . . 38
3.3.6 Design of modified 3'UTRs for testing the effects of mutated motifs . . . 40
3.3.7 Determining motif effect on abundance . . . 41
3.3.8 Predicting changes in transcript abundance from changes in half-life . . . 41

Chapter 4: tidyqpcr: Quantitative PCR Analysis in the tidyverse 43
4.1 Introduction . . . 43
4.2 Results . . . 45
4.2.1 qPCR analysis software review . . . 45
4.2.2 tidyqpcr: Quantitative PCR Analysis in the tidyverse . . . 50
4.2.3 Reviewing and improving tidyqpcr . . . 62
4.3 Conclusion . . . 67

Chapter 5: Limitations of Composability of Cis-Regulatory Elements in Messenger RNA 68
5.1 Introduction . . . 68
5.1.1 Discovering sequence determinants of gene expression . . . 68
5.1.2 From regulatory elements to modular cloning . . . 69
5.1.3 Current limitations in predicting gene expression . . . 70
5.2 Results . . . 71
5.2.1 Terminator effects on gene expression depend on cis-regulatory context . . . 71
5.2.2 Candidate cis-regulatory elements contribute to transcript decay rates . . . 74
5.2.3 Quantification of differential expression due to motif insertion or mutagenesis in multiple 3'UTRs . . . 75
5.2.4 Motif effects on gene expression depend both on terminator context and promoter pairing . . . 78
5.2.5 Inserting motifs into terminators shifts poly(A) site usage downstream . . . 83
5.3 Conclusions . . . 84

Chapter 6: DiffFracSeq: A Bayesian Model for the Detection of Differential Fractionation of Sequencing Data 89
6.1 Introduction . . . 89

6.2 Results . . . 92
6.2.1 Bayesian hierarchical model . . . 92
6.2.2 Overview of the simulated test data set . . . 93
6.2.3 Posterior checks using the simulated test data set . . . 95
6.2.4 Detecting differential fractionation using DESeq2 . . . 95
6.2.5 Detecting differential fractionation with DiffFracSeq and DESeq2 with the simulated ground truth . . . 96
6.2.6 Overview of the experimental test data sets . . . 97
6.2.7 Quantifying fractionation in the experimental data sets . . . 99
6.2.8 Extending DiffFracSeq to include a generalised linear model on transcript counts . . . 102
6.2.9 Detecting relative changes in fractionation and expression across conditions . . . 102
6.3 Conclusion . . . 105

Chapter 7: Discussion **108**
7.1 Future Work . . . 110

Appendix A **112**
A.1 tidyqpcr User Questionnaire . . . 113
A.1.1 Subject questionnaire . . . 113
A.1.2 tidyqpcr worksheet . . . 115
A.2 Software review table . . . 116

Appendix B **120**
B.1 Chapter 5 Supplementary Tables . . . 121
B.2 Chapter 5 Supplementary Figures . . . 130

Appendix C **139**
C.1 Chapter 6 Supplementary Code . . . 140

Bibliography **142**

List of Figures

2.1 Overview of key mRNA regulatory processes. 5
2.2 Key steps in a qPCR assay. 9
2.3 Comparison of RNA-Seq technologies. 13
2.4 Summary of Agile and Waterfall software development practices. 20

4.1 Block diagram showing the functionality of tidyqpcr. 51
4.2 Developing tidyqpcr using the tidyverse packages grants access to a larger ecosystem of data analysis packages. 52
4.3 tidyqpcr facilitates flexible, modular plate design. 54
4.4 Extensive vignettes teach users MIQE-compliant analysis. 55
4.5 tidyqpcr can be used to quickly analyse multi-plate, multi-target, and multi-sample qPCR assays. 59
4.6 A text cloud showing the keywords repeatedly used across the semi-structured interviews. 64

5.1 Terminator contributions to gene expression are promoter, coding sequence and length dependent. 73
5.2 A linear model of transcript half-life quantifies the effect of candidate terminator motifs on half-life. 76
5.3 Motifs inserted into RPS3 and TSA1 host terminators change transcript abundance in RT-qPCR measurements. 79
5.4 Motifs removed from PIR1 host terminators change transcript abundance in RT-qPCR measurements. 80
5.5 Promoter and terminator context alter the regulatory effect of motifs. . . . 82
5.6 Inserting motifs into RPS3 and TSA3 terminators changes 3'UTR length. 85

6.1 DiffFracSeq chapter summary. 90
6.2 Graphical representation of the DiffFracSeq Model. 93
6.3 Summary of the simulated test data set. 94
6.4 DiffFracSeq posterior predictive check 95
6.5 DiffFracSeq vs DESeq2 performance on the simulated data set. 98
6.6 Summary experimental test data sets. 99
6.7 DiffFracSeq vs DESeq2 performance on the experimental data sets. . . . 101
6.8 Graphical representation of the DiffFracSeq model with conditions. . . . 103
6.9 Detection of differential expression and fractionation across temperatures. 104
6.10 Graphical representation of the multi-fraction DiffFracSeq Model. 105

B1 Both terminator and promoter contribute to gene expression. 130
B2 Relative protein abundance correlates with relative mRNA abundance for reporter constructs with modified 3'UTRs. 131

B3 Relative positions of ATATTC, TGTAHMNTA and HWNCATTWY motifs in native 3'UTRs. 132
B4 Decapped constructs measured by 5Pseq match mature constructs measured by QuantSeq. 133
B5 High correlation in transcript counts between samples for both RNA-Seq assays. 134
B6 Poly(A) site usage for genomic PGK1, TSA1 and RPS3 terminators remains the same across samples for each RNA-Seq assay. 135
B7 Poly(A) site usage remains the same for genomic TSA1 and RPS3 terminators as for plasmid expressed WT constructs in QuantSeq and 5PSeq. . 136
B8 Construct poly(A) site usage across 5PSeq and QuantSeq. 137
B9 5PSeq data finds no detectable changes in 5'-phosphorylated intermediates between reporter constructs. 138

List of Tables

3.1 Overview of the coding and qPCR experience of tidyqpcr interviewees . . 34

4.1 Example tidyqpcr code for designing a 96-well qPCR plate for an assay with 16 target stress response genes across 6 conditions. 57
4.2 Example tidyqpcr code for reading threshold Cq values from LightCycler® qPCR machines and combining them with the designed plate plan. . 58
4.3 Example tidyqpcr code for calculating ΔCq across multiple plates and plotting summary results across target genes. 58
4.4 Function definition for the calculate ΔCq method within tidyqpcr. This example showcases the use of the `group_by` function provided by the core tidyverse package dplyr to split the Cq values by `sample_id` and apply the `calculate_normvalue` on each group separately. 60
4.5 Function documentation for the calculate ΔCq method within tidyqpcr as structured by the roxygen2 R package. 61
4.6 Function test for the calculate ΔCq method within tidyqpcr as structure by the testthat R package. 62
4.7 tidyqpcr's development included improvements from user feedback. . . . 66

5.1 Summary of the terminator library. 72
5.2 Summary of shortlisted motif characteristics. 77

6.1 DESeq2 example R code 96

B1 Tables showing changing contributions to gene expression from terminators paired with different promoters and coding sequences. 121
B2 Tables showing changing contributions to gene expression from terminators paired with different promoters and coding sequences. 122
B3 Variance explained by each type of CRE in the half-life model applied to data from Chan et al (2018). 123
B4 Table showing the DNA sequences for all mCherry reporter constructs. . . 123
B5 Table showing the DNA sequences for all mTurq reporter constructs. . . . 123
B6 Primer sequences created for all qPCR experiments. 123
B7 Selecting the HWNCATTWY motif, TTTCATTTC 124
B8 Selecting the UGUAHMNUA motif, TGTACAATA 124
B9 Table showing minimum free energies of 3'UTR constructs with inserted/deleted motifs. 125
B10 Tables showing fold changes in transcript abundance of promoter-terminator constructs with different inserted motifs. 126
B11 Tables showing fold changes in transcript abundance of promoter-terminator constructs with different deleted motifs. 127

B12 Tables showing fold changes in transcript abundance of motifs in different contexts. 128
B13 Tables showing relative usage of the two major Poly(A) sites in tRPS3 and tTSA1 terminators across constructs. 129
B14 Tables showing relative usage of the two major Poly(A) sites in tRPS3 across constructs as detected by 5PSeq. 129

Chapter 1

Introduction

1.1. Overview

Assays quantifying changes in transcript abundance with unprecedented sensitivity have rewritten our understanding of RNA regulation. Regulatory mechanisms for the localisation, degradation and modification of RNA transcripts can now be investigated across entire transcriptomes. However, as the sensitivity of experiments has increased so has the difficulty of distinguishing biologically significant effects from experimental noise. Confounding biological effects with experimental noise could undermine the progress promised by a data-rich age of molecular biology.

Science is suffering from a reproducibility crisis. In 2017, Nature surveyed over 1500 scientists and found over 70% of them tried and failed to reproduce someone else's work (Baker, 2016). The origins of the crisis come from the lack of detail in experimental protocols, obscure analysis methods, and misunderstanding of statistical tests. The level at which scientists misuse statistical tests has even led a journal to ban any reference to statistical significance (Trafimow & Marks, 2015).

The reproducibility crisis is worsened by the gap between the statistical methods developed to analyse noisy data and the implementation of these methods on biological questions. The choice of statistical method can change depending on subject, hypothesis and data quality which also contributes to the gap (Ching et al., 2018). The way data is preprocessed, the method used to deal with missing values and the software used can all drastically change results (Ioannidis et al., 2009). Meanwhile, those that bridge this gap are typically biologists with no formal software engineering training (Attwood et al., 2019) and who are unlikely to develop prototype analysis scripts into fully fledged programs (Prins et al., 2015). This leads to high duplication, poor reproducibility and slower overall progress (Lawlor & Walsh, 2015).

The use of rigorous statistical methods implemented in reproducible, open-source

research software is the only way to overcome this crisis. Furthermore, the inclusion of comprehensive documentation with research software can encourage experimental best practices and improve the reproducibility of an entire experimental assay. This thesis combines software development best practices with rigorous analysis of multiple transcriptomic assays to conduct reproducible experiments and investigate post-transcription regulatory mechanisms acting on mRNA.

1.2. Contributions

The aim of this thesis is to develop and apply computational methods to investigate the regulation of RNA abundance. The contributions of this thesis are as follows:

- The development of tidyqpcr, an open-source R package for the analysis of qPCR data. tidyqpcr contains extensive documentation and integrates with the wider tidyverse suite of data analysis packages to help users conduct reproducible, flexible, and MIQE best-practice-compliant quantitative PCR experiments. The R package is available to download and has an accompanying publication in the Journal for Open Source Software, doi:10.21105/joss.04507.

- The detection of the limitations of composability of cis-regulatory elements beyond promoter and terminator regions. Short regulatory sequences in the 3'UTR of mRNA transcripts are shown to have different contributions to gene expression depending on context. The paper is currently under review by Nucleic Acids Research with a preprint available on bioRχiv, doi:10.1101/2021.08.12.455418v2.

- The development of DiffFracSeq, a novel Bayesian statistical model that normalises bulk RNA-Seq assays exploring differential fractionation. Exploiting the physical properties of sequencing sub-fractions of a larger body DiffFracSeq can overcome issues with normalising samples that have global changes in the transcriptome. The model to available to use as an R package downloadable from GitHub, github.com/DimmestP/DiffFracSeq.

1.3. Thesis Layout

In this thesis, I outline the development of analysis software and statistical models to explore transcript localisation and the context dependence of cis-regulatory elements. Chapter 2 provides the necessary background knowledge required to understand the results of this thesis. It starts with an overview of the key mechanisms used by eukaryotic cells to regulate RNA abundance. Then, the basics of several transcriptomic assays that have enabled quantitative comparisons of RNA abundance are explained. qPCR, microarrays

and RNA-Seq are introduced with emphasis on the sources of error that can be present in these experiments. Finally, the chapter introduces software development practices that have been implemented in this work.

Chapter 3 describes the materials and methods used to complete my PhD, including: a brief overview of the statistical methods used, details of the practices followed to develop tidyqpcr, and an overview of experiments conducted by members of the Wallace lab for the results in chapter 5. Chapter 4 starts by justifying the need for a new qPCR analysis package and explaining how tidyqpcr has been designed to overcome some of the deficiencies in currently available software.

Chapter 5 introduces the concept of composability of regulatory elements in the contexts of synthetic and computational biology. Then, the changing contributions to gene expression from terminators when paired with different promoters and coding sequences are shown. The chapter ends by describing the design of constructs with short regulatory motifs inserted or removed from their terminators and showing that these motifs also have differing contributions depending on context.

Chapter 6 begins by outlying the difficulties in detecting differential fractionation using standard RNA-Seq analysis software. The Bayesian statistical model behind DiffFracSeq is then introduced and its ability to successfully detect differential fractionation is inspected using three different data sets. Finally, chapter 7 summarises the main contributions of this body of work and suggests some avenues for future research.

Chapter 2

Preliminaries

2.1. Gene Expression Regulation

Across all cells genetic information flows from DNA to RNA to proteins, the central dogma of molecular biology (Crick, 1970). Transcription from DNA to RNA and translation from RNA to protein are regulated by numerous mechanisms simultaneously to enable cells to respond to their environment. This chapter is a brief overview of the transcription and post-transcriptional regulatory mechanisms that contribute to the differential expression of mRNA transcripts with emphasis on the mechanisms present in *Saccharomyces cerevisiae*, Figure 2.1.

2.1.1. Transcriptional regulation

The creation of an mRNA transcript from a DNA template requires the completion of three key transcriptional steps: initiation, elongation and termination. Initiation of mRNA transcription consists of the RNA polymerase II binding to the DNA template upstream of the sequence encoding a protein. The region where the RNA polymerase II initiates transcription is called the promoter. In eukaryotes, DNA is wound around nucleosomes and densely packaged in several orders of chromatin structure. Therefore, the initiation of transcription requires a variety of transcription factors to aid in the unwinding of the chromatin, scanning of regions for promoters, and the binding of RNA polymerase II. Promoters consist of regulatory sequences that encourage the binding of transcription factors and can be further affected by distal regulatory regions such as enhancers (Cramer, 2019).

Once transcription is initiated the polymerase begins the sporadic process of elongation from the transcript start site. The polymerase first transcribes the 5' untranslated region (5'UTR) of an mRNA transcript, then the coding sequence for the corresponding protein, and finally the 3' untranslated region (3'UTR). For genes that contain introns, the polymerase will also transcribe the intron sequences which can occur across the nascent

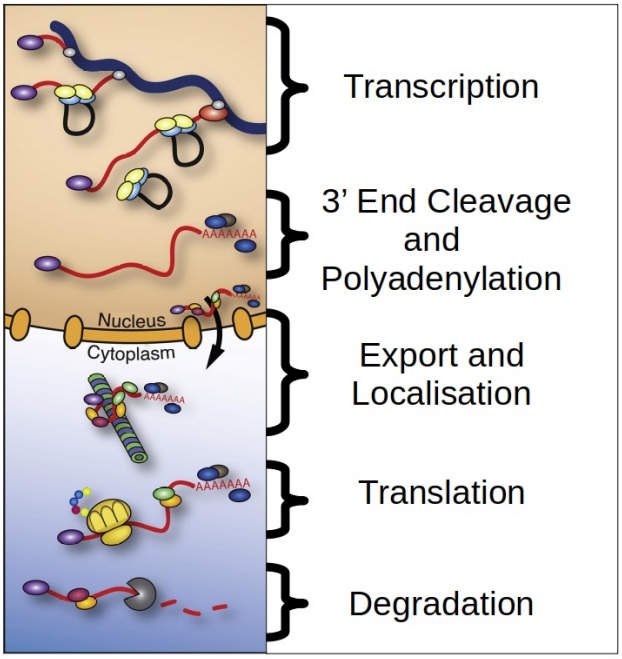


Figure 2.1: Overview of key mRNA regulatory processes. Several mechanisms act simultaneously to enable cells to respond to their environment through the regulation of mRNA transcripts. Figure adapted from Corbett, 2018

transcript but are removed from the mature mRNA transcript. The process of elongation is highly stochastic with polymerases regularly pausing and even stalling with several accessory proteins required to maintain the process. Early on in the elongation stage, the 5' end of the pre-mRNA is modified by several enzymes to form a 5'-methyl cap which inhibits degradation and aids translation. Finally, the termination of transcription remains a relatively unclear process as a distinct termination sequence has not been found. Instead, the polymerase continues to transcribe the sequence downstream of the 3'UTR. This downstream terminator region does contain sequences to recruit cleavage and polyadenylation factors. The still elongating RNA transcript is cleaved at the end of the 3'UTR as dictated by these sequences. The freely floating pre-mRNA transcript is then bound by a poly(A) polymerase that adds a tail of hundreds of adenine bases to the end of the transcript. The remaining string of RNA bound to the polymerase and DNA template is degraded by a 5' to 3' exonuclease, which is thought to dislodge the polymerase and terminate transcription (Alberts et al., 2015; Cramer, 2019). The terminator can contain multiple cleavage sites leading to transcript variants, called isoforms, with different poly(A) positions. Transcript isoforms can also be created as the promoter can contain different transcription start sites leading to transcript isoforms with different coding sequences and 5'UTRs (de Klerk &

't Hoen, 2015).

2.1.2. Post-transcriptional regulation

A variety of regulatory mechanisms act in between the transcription of a nascent mRNA transcript and its translation into a sequence of amino acids. Key tasks include: removing introns, conducting quality control and transporting transcripts around the cell. Introns are regions that are transcribed but are not in the final mature mRNA transcript. Concurrently with or immediately after transcription a group of co-functional RNA and proteins called the spliceosome remove intron segments within nascent RNA transcripts. The regions that form the mature mRNA transcript are called exons and a single transcript may consist of 10s of exons spliced together. Alternative splicing of introns and exons can theoretically produce thousands of different versions of a protein in some Drosophila genes (Wilkinson et al., 2020).

Selective degradation of low quality mRNA or transcripts that are no longer needed is a crucial post-transcriptional regulation mechanism. The importance of degradation in gene regulation is reflected in the number of redundant processes to degrade transcripts, but the majority of degradation is facilitated by the deadenylation-dependent mRNA decay pathway. This pathway starts by flagging transcripts for degradation by shortening the poly(A) tail. Then, either the 5'methyl cap is removed to enable 5'->3' degradation by the XRN1 exoribonuclease or 3'->5' degradation is initiated by the exosome attaching to the exposed 3' end. A variety of surveillance methods can trigger transcript degradation, including: non-sense mediated decay which checks for premature stop codons, non-stop mediated decay which checks for missing stop codons, and no-go mediated decay which checks for stalled ribosomes on mRNA. In response to stress or other causes of high load on the degradation machinery, P-bodies can form in the cytoplasm that are believed to facilitate degradation as they often contain deadenylation, decapping and degradation factors (Garneau et al., 2007).

Spatial regulation enables centrally transcribed mRNA transcripts to be regulated differently depending on the target location of their encoded protein. In budding yeast, the Ash1 protein represses mating-type switching, but only in daughter cells (Sil & Herskowitz, 1996). The localisation of the ASH1 transcript at the bud tip and subsequent localised translation ensures the Ash1 protein is not present in the parent cell despite being transcribed in its nucleus (Niednery et al., 2014). It is thought that co-transcriptional recruitment of She2 protein to the Ash1 transcript in the nucleus of the parent cell enables the later recruitment of cytoplasmic factors Khd1/Hek2 and Puf6, factors known for translational repression. Furthermore, the successful transport of ASH1 to the bud tip by She2-She3-Myo4 complexes depends on translational repression by Khd1 and Puf6.

Later, phosphorylation of Khd1 and Puf6 by bud-membrane-localised kinases leads to localised translational activation of the ASH1 mRNA (Paquin et al., 2007; Deng et al., 2008).

Another example where the effect of a CRE on a transcript depends on co-localisation with a regulatory kinase comes from the fungal RNA-binding protein Ssd1. Yeast cell wall proteins such as Sun4 and Tos6 are translationally repressed by Ssd1 (Jansen et al., 2009). It is thought that these transcripts are translationally activated at bud sites after the phosphorylation of Ssd1 by a localised kinase, Cbk1 (Jansen et al., 2009; Kurischko et al., 2011). There is no evidence that Ssd1 directly acts to transport RNA, so this localised activation presumably depends on the recruitment of other RNA-binding proteins to those transcripts (Hogan et al., 2008; Bayne et al., 2021), that then recruit transport machinery.

Post-transcription regulation is also known to facilitate temporal regulation in cells. Temporal regulation is common in developmental processes where the order of production of specific proteins is highly regulated. For example, in C. elegans lin-4 is a non-coding RNA gene crucial for regulating cell fates during the early stages of larval development (Wightman et al., 1993). Lin-4 is a small RNA that binds to its target mRNA lin-14 and inhibits the translation of lin-14 (Lee et al., 1993). Since lin-4 is only expressed at the end of the first larval development stage, lin-14 is only translationally inhibited at the end of stage 1, initiating the start of stage 2 (Olsen & Ambros, 1999). Similarly, to establish meiotic chromosome segregation in budding yeast, mRNA encoding cyclin CLB3 is transcribed in stage I of meiosis, but is translationally repressed until stage II of meiosis. CLB3 is translationally repressed by the RNA-binding protein Rim4. During the transition to meiosis II, Rim4 is phosphorylated which inhibits binding to CLB3 and enables CLB3 to be translated (Berchowitz et al., 2013). Therefore, post-transcription control of CLB3 by Rim4 and of lin-4 by lin-14 depends on the timing of promoter-specified transcriptional control.

2.2. Experiments Quantifying RNA Abundance

2.2.1. qPCR

Quantitative polymerase chain reaction (qPCR) is the basis of countless assays that can quantify various populations of DNA and RNA. Polymerase chain reaction (PCR) is regarded as one of the most significant methods in molecular biology as it enables the production of copies of regions of DNA. The log-linear growth of copies from PCR duplication led researchers to explore its use as an accurate method to quantify abundance (Saiki et al., 1988). After its invention in the 1980s the quantification of the rate of amplification in real time quickly followed (Holland et al., 1991), but it was not until the 2000s that biochemistry and technology matured into a reliable quantitative PCR (qPCR) method (Walker, 2002). qPCR is a relatively low-throughput quantification method when compared to the other methods described here. However, developments in microfluidics and multiplexing target probes are overcoming the bottlenecks in conducting high-throughput qPCR (Dreier et al., 2022).

The basic principle of PCR consists of the duplication of a region of DNA that is specified by two short nucleotide sequences, called primers, that are designed to be complementary to the start and the end of the region of interest. A highly thermotolerant polymerase, adapted from the bacteria *Thermus aquaticus*, is then able to complete the duplication of the region by elongation of the sequence between the two primers (Saiki et al., 1988). The duplication cycle is repeated several times leading to an exponential growth in the number of copies of the original region. The PCR polymerase must be thermotolerant as the duplication cycle is rapidly repeated by raising the temperature to melt the newly created complementary strand away from the original strand before dropping the temperature back down to enable the next round of elongation. Quantifying the rate of amplification is done by introducing dyes that only fluoresce when a region has been successfully duplicated. The fluorescence of the sample is measured as the PCR cycle is repeated to determine the exponential growth in duplicates. Quantitative PCR (qPCR) uses the amplification curve to infer the number of transcripts of the target sequence in the original sample (Holland et al., 1991).

2.2.1.1. qPCR methods: RNA vs DNA

qPCR is highly optimised for amplifying DNA fragments using engineered derivatives of the *Thermus aquaticus* polymerase (Witte et al., 2018). Therefore, to quantify RNA fragments an additional step is required to create complementary DNA (cDNA) from RNA using a reverse transcriptase. Unfortunately, this step can be a significant source of vari-

ation and has been determined to be the source of most variation between RNA samples. The variation in cDNA yield between replicates can be influenced by the choice of reverse transcriptase priming method, the original RNA target concentration and the total RNA concentration in the sample. In order for an RT-qPCR experiment to be reproducible the reverse transcriptase step must be optimised and clearly described (Ståhlberg et al., 2004).

2.2.1.2. qPCR methods: Intercalating dyes vs probe-specific dyes

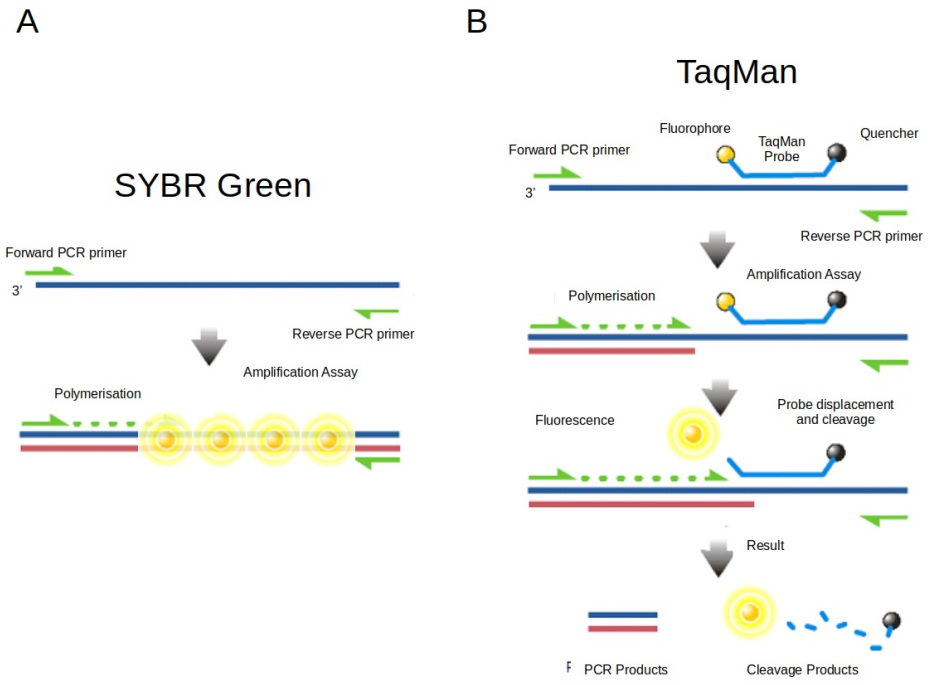


Figure 2.2: Comparison of the key steps in a qPCR assay using two different fluorescent probes. (**A**) Intercalating dyes, such as SYBR Green fluorescent probes, bind to any double-stranded DNA. (**B**) Probe-specific dyes, such as TaqMan fluorescent probes, bind to specific sequences and only fluoresce once detached from their paired quencher during elongation. Figure adapted from Wikimedia Taqman diagram.

There are two common types of fluorescent dye used to measure duplicated fragments: probe-specific dyes vs intercalating dyes. Intercalating dyes fluoresce when they bind to any double-stranded DNA species in a sample (Ihmels & Otto, 2005), Figure 2.2A. This leads to it being a cheap and relatively easy to use system, but it is highly susceptible to contamination and it is unable to distinguish between samples from different targets. Probe-specific dyes bind the fluorescent dye to an oligonucleotide probe that is designed to attach to the region of interest somewhere in between the two primers. The oligonucleotide probe has the fluorescent dye on one end and a quencher on the other. The

quencher stops any excitation emissions from the dye through fluorescence resonance energy transfer (Juskowiak, 2010). However, during elongation, when the polymerase reaches the oligonucleotide probe it is hydrolysed separating the dye from the quencher. Emissions from the fluorescent dye can then be measured and the creation of a new duplicate is detected, 2.2B. The introduction of a custom oligonucleotide probe increases the specificity of probe-specific dye methods and reduces the effects of contamination. Also, the abundance of multiple targets can be measured in the same sample by carefully designing different oligonucleotide probes with different fluorescent dyes. Unfortunately, the design and creation of custom probes cause probe-specific dye methods to be more expensive and technical (Adams, 2020). The accuracy of the cheaper intercalating dyes methods and the probe-specific dye methods is comparable, if correctly conducted (Tajadini et al., 2014). Although, the limit of detection (LOD) of low copy targets depends on protocol optimisation.

2.2.1.3. Quantifying abundance: Curve fitting vs cycle threshold

The exponential limit of the number of duplicates per cycle enables methods that compare abundance across samples. Assuming all samples reach the exponential growth stage at the same time then the difference in fluorescence at any cycle of the PCR assay is dependent only on the original copy number. However, even if the duplication is perfectly efficient, the amplification curve of duplicates per cycle is not a perfect exponential as there is a limited window through which the number of duplicates will grow exponentially. The window is defined by limitations in detecting fluorescence at low abundance and the exhaustion of resources at high abundance. The original copy number can be inferred from the fluorescence if a fluorescence threshold is set during the exponential phase and the number of cycles needed for a sample to reach it are compared. Unfortunately, this method assumes both that each sample reaches the exponential phase at the same time and that each cycle doubles the number of duplicates perfectly for each sample (VanGuilder et al., 2008). An alternative method fits a sigmodal curve to the amplification curve and uses this model to deduce the cycles required to reach the threshold. The additional fitting can account for differences in the times to reach the exponential growth phase between samples and can directly account for deviations in perfect duplication (Swillens et al., 2008).

2.2.1.4. Quantifying Abundance: Relative vs absolute

Multiple methods exist for converting cycle threshold measurements, Cq, into quantitative values for sample abundance whilst accounting for experimental and technical noise. First, qPCR experiments can be designed to measure the relative change in abundance across samples. Relative abundance measurements depend on the determination of genes that have constant expression across all samples/conditions. Any change in the gene(s) of interest across samples can then be detected by comparing ΔCq or the expressions relative to the set of constantly expressed genes. Normalising the fluorescence to genes with constant expression minimises batch effects introduced by sample preparation, reverse transcription and reactants. Alternatively, the absolute number of copies of a target in a sample can be estimated. Absolute quantification of a target requires a preliminary experiment where known initial quantities of the gene of interest are measured with qPCR. Several amplification curves for the gene of interest, with gradually increasing copies of the gene of interest, are measured to create a collection of standard curves. Next, the sample from the primary experiment is measured with qPCR and its amplification curve is compared to the standard curves. The absolute copy number of the gene of interest in the experimental sample can then be interpolated (Wong & Medrano, 2005; VanGuilder et al., 2008).

2.2.2. Microarrays

Microarrays facilitated the creation of some of the first high-dimensional data sets in transcriptomics. In the 1980s an assay was published to simultaneously determine multiple specific cell surface antigens through the use of a matrix of antibodies fixed to a glass slide (Chang, 1983). The opportunity to quantify multiple characteristics of a sample using the same chip led other labs to explore attaching oligonucleotides to a slide, inventing microarrays (Schena et al., 1995). The technology enabled the abundance of thousands of genes to be measured simultaneously unlocking genome-wide studies of gene expression regulation across conditions (Gasch et al., 2000). The assay also benefited from the high-quality sequencing of the genomes of multiple species as oligonucleotide probes could be designed to investigate any regions of interest (Lander et al., 2001).

Microarrays consist of a glass or silicon substrate with spots of DNA printed in a regular grid. Each DNA spot is a complement to a different target which fluoresces when bound with the target. In assays to determine differential expression, two colour microarrays are used where each spot contains two fluorescent probes; one to detect the target abundance in the sample of interest and one to detect target abundance in a control or another sample of interest. A camera detects the level of fluorescence across each spot

after excitation by a laser which is used to determine the abundance of that target. The two colour microarray assay measures the fluorescence of the two fluorophores and uses the ratio to determine changes in expression. As the DNA probes have to be designed and printed onto the glass plate their complementary targets have to be decided before conducting the experiment which reduces the opportunity to discover novel regulatory elements (Schena et al., 1995). Microarrays facilitated the development of high-throughput transcriptomic experiments as RNA transcript abundance can also be investigated by introducing a reverse transcriptase step to create cDNA fragments.

2.2.3. RNA-Seq

The success of the microarray was limited by the requirement of specifying the target probes prior to the experiment. However, its densely packed array of oligonucleotides fixed to a solid surface inspired a new sequencing method. For 30 years the primary method for sequencing unknown DNA fragments was Sanger sequencing with its successful application in decoding the human genome (Lander et al., 2001). The method consists of introducing a di-deoxynucleotide triphosphate (ddNTP) version of one of the four nucleic acids which induces premature termination of elongation when the polymerase incorporates it into a DNA chain. Due to the stochasticity of elongation, the introduction of ddNTP occurs at different stages of duplication leading to the creation of a population of different lengths of copies of a target. Separating the population by weight using electrophoresis creates bands where the chosen nucleic acid has been replaced by a ddNTP version. Repeating the process by replacing each nucleic acid in turn enabled a target sequence to be decoded (Sanger et al., 1977). Similar to qPCR, Sanger sequencing can also be extended to RNA sequencing by introducing a reverse transcriptase step to make cDNA. The accuracy of the Sanger method means it remains in use today, but the cost and difficulty of scaling up the method have limited its use.

2.2.3.1. Shotgun sequencing

In the early 2000s, several companies competed to improve microarrays by overcoming the deficiency in requiring DNA targets to be defined before the experiment (Rusk & Kiermer, 2007). Solexa (now Illumina) developed adapters that could be ligated to any sample of DNA and facilitate the attachment of the DNA sample to a solid surface. Fluorescent nucleotides were created that could terminate elongation in a reversible way. These nucleotides enabled a base-pair by base-pair cycle of elongation with the identity of the last bound nucleotide being revealed by its colour. Fixing the fragments of DNA to the surface meant islands of duplicates of the original fragment would be created and

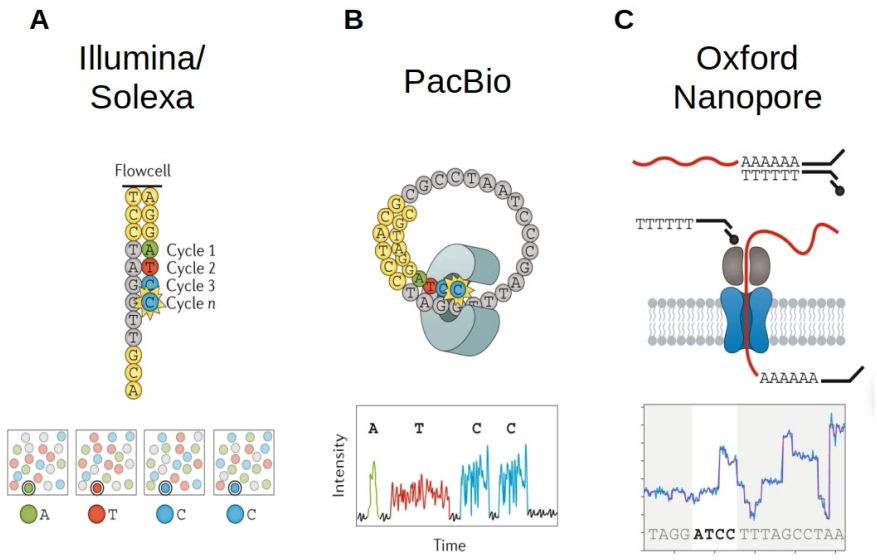


Figure 2.3: Comparison of RNA-Seq technologies. (**A**) Illumina sequencers attach short fragments to a solid surface called a flowcell. Fluorescent nucleotides create a clear fluorescent spot that identifies the last nucleotide to be attached. (**B**) PacBio sequencers enable long reads by opening up double-stranded DNA into a circle. A polymerase can loop around the circle attaching fluorescent nucleotides that identify the next nucleotide in the sequence. (**C**) Oxford Nanopore sequencers force nucleotide fragments through protein pores in a membrane. The transmembrane proteins change the current passing through them in response to the nucleotide sequence. The changes in the current can be used to determine the nucleotide sequence. Figure adapted from Stark et al., 2019

a clear fluorescent spot could be detected, Figure 2.3A. Unfortunately, the original technology only allowed accurate determination of reads with length <50 nucleotides. As this length is significantly shorter than the majority of sequences of biological interest, the preparation of samples for Solexa sequencing included a step to break samples into small fragments, coining the term shotgun sequencing. The size of the fragments is of the order of 100 nucleotides, which is still larger than the length of reads received from the sequencing machine. An extension to the single read per fragment method is the paired-end reads method. Sequencing two reads for each fragment, one at either end, enabled even more accurate detection. Receiving two reads per fragment also opened the door to exploring structural variants where fragments were copies of non-adjacent segments of the genome (Risca & Greenleaf, 2015).

2.2.3.2. Long read sequencing

Shotgun sequencing propelled molecular biology into an age of affordable, high-throughput sequencing data. However, the limit in read size from shotgun sequencing made tasks such as de novo genome assembly and the detection of structural variants difficult. Long read sequencing technology has now matured with Oxford Nanopore and PacBio offering solutions to read 1000s of nucleotides at a time. PacBio sequencer uses fluorescent nucleotides to determine sequences similar to an Illumina sequencer, but instead of binding fragments to a solid surface, they form single-stranded circles from long segments of double-stranded DNA (dsDNA). Hairpin-shaped SMRTbell adapters are attached to either end of a dsDNA segment creating a closed loop that is opened up into single-stranded circular DNA, Figure 2.3B. A polymerase is attached to the adapters which can loop around the circle producing multiple copies of both strands of the original dsDNA segment (Hu et al., 2021). An Oxford Nanopore sequencer consists of a membrane with 100s of transmembrane proteins that alter their electric resistance when deformed by nucleotides moving through them. A current passing through the transmembrane protein then produces a signal in response to the nucleotides moving through the protein, Figure 2.3C. Machine learning algorithms have been trained to convert the signals in the current into the sequences nucleotides (Jain et al., 2016).

2.2.3.3. Overview of RNA-Seq based assays

As high-throughput RNA-Seq technology has matured, assays to explore a variety of different transcriptome effects have been developed. With around 80% of the RNA in a cell being ribosomal, methods to investigate other RNA populations have been developed using enrichment, through poly(A) tail selection or ribosomal RNA depletion (Stark et al., 2019). Methods that require samples to be PCR amplified before sequencing can add unique molecular identifiers (UMI) to their transcripts to check for biases in duplication (Kivioja et al., 2011). Multiplexing methods now allow samples to be pooled together by introducing sample unique barcodes to read adapters which enables ultra-high-throughput methods with one library preparation stage (Craig et al., 2008). RNA-protein interactions can be discovered by UV cross-linking transcripts to proteins, pulling out the protein of interest, degrading the protein and analysing the remaining RNA (Granneman et al., 2009). Pulse labelling methods can uncover genome-wide transcript production and degradation by introducing a labelled nucleotide and measuring changes in the population of transcripts with that nucleotide (Chan et al., 2018). Transcript isomers created by alternative polyadenylation are uncovered by using adapters that ensure reads are anchored to the poly(A) tail (Pelechano et al., 2013). Localisation of transcripts to organelles or mem-

branes can be detected by ultra-centrifugating cell lysate and sequencing the pellet (Iserman et al., 2020). Finally, single-cell RNA-Seq technologies are unlocking new cell types and new sources of heterogeneity between homogeneous samples (Jovic et al., 2022).

2.2.3.4. Biases in RNA-Seq assays

The ubiquitous use of RNA-Seq assays across biology has led to multiple advances in its accuracy and reliability, but many well-documented biases remain. The fragmentation step of RNA-Seq methods introduces a significant gene length bias as longer genes create more fragments (Oshlack & Wakefield, 2009). GC content of genes changes the reliability of base-calling and alters read-coverage (Dohm et al., 2008). RNA-Seq data sets are also highly susceptible to batch effects with total reads per run changing by orders of magnitude (Auer & Doerge, 2010). The choice of RNA extraction and enrichment can lead to significant changes in differential expression detection in the same samples (Sultan et al., 2014). Meanwhile, poly(A) anchored reads can initiate elongation from an internal stretch of adenine instead of the 3'end tail or it can switch templates mid-elongation (Balázs et al., 2019).

2.3. Transcriptomic Data Analysis

Modern transcriptomics experiments are acquiring quality, high-throughput data sets at unprecedented scales. In 2012, the European Bioinformatics Institute (EBI) was one of the biggest biology repositories in the world with a 20Pb storage facility (EMBL-EBI, 2012). However, by 2021, the upload of new data reached 20Pb a year with the institute having to explore collaborations with Google and Amazon in order to keep up (EMBL-EBI, 2021). Also, the creation of the high-dimensional data sets with transcript abundance of thousands of genes over dozens of conditions exposed biologists to the n $<<$ p problem. The n $<<$ p problem is the low statistical power due to the small number of data points, n, compared to the number of genes and conditions, p. Here, a typical workflow for the analysis of an RNA-Seq data set is outlined to highlight the growing dependence on research software in molecular biology.

2.3.1. RNA-Seq data analysis

RNA-Seq analysis consists of three core steps: quality control, alignment and counting (Costa-Silva et al., 2023). The exact quality control steps can change significantly between types of assay. For example, the enrichment of mRNA transcripts using degradation or poly(A) anchors need to be checked for effectiveness by inspecting ribosomal RNA content or tRNA levels. Meanwhile in the case of single-cell RNA-Seq checking for cases where two or more cells may have accidentally been combined (as multiplets are a common in occurrence in many techniques) is a vital step that is not required for bulk RNA-Seq methods, (Zheng et al., 2017). However, across all methods it will be required to check for sequence amplification biases, calling quality and whether each lane has successfully detected reads with FASTQC, (Andrews, 2010). Once QC has been completed UMI-tools and cutadapt may be used to remove any adapters and UMI that have been introduced during the library preparation as these may complicate alignment to the genome (Martin, 2011; Smith et al., 2017). Also, it is common to trim the 3' end of reads as errors in nucleotide callings tend to occur at the end. If the assay also includes multiplexed samples these need to be detected and separated with a tool such as demultiplex (Laros, 2018). Once the reads have been trimmed and demultiplexed, then they need to be aligned to the genome in order to be able to determine which gene they map to. A variety of genome aligners are available depending on organism and computing infrastructure limitations. BowTie2 is an accurate aligner but it struggles to align mRNA transcripts with introns (Langmead & Salzberg, 2012). Other aligners like STAR or HISAT2 are much better at aligning reads across introns, (Dobin et al., 2013; Kim et al., 2019). It is vital that quality control steps are conducted after the alignment step with MultiQC (Ewels et al., 2016). If the vast majority of reads do not align to the genome of an organism then there could

be a contaminant present. Visualising reads on a genome browser, such as the integrated genome browser (Freese et al., 2016), is also important to check for artefacts, strandedness and poly(A) anchoring if appropriate. Once the sequences have been aligned the next step is to remove PCR duplicates if UMIs are present again using UMI-tools. If reads are aligned to exactly the same sequence and they have exactly the same UMI, then they are considered duplicates and can be flattened to just one read. Finally, with the deduplicated aligned reads fully processed featureCounts can then count the number of reads to each gene (Liao et al., 2014; Conesa et al., 2016).

2.3.1.1. RNA-Seq analysis pipelines

The complexity of analysing high-volume RNA-Seq data sets has led to the development of scalable, flexible and reproducible analysis pipelines. Assay-dependent quality control steps, from removing adapter sequences to mapping to different genome annotations, are often completed by software packages written in different scripting languages. Workflow languages, such as Nextflow, are able to integrate the inputs and outputs of software in a domain-agnostic manner (Tommaso et al., 2017). A community of bioinformaticians are bringing together standardise modules using Nextflow which can be cherry-picked to create the best pipeline for any specific RNA-Seq assay (Ewels et al., 2020). Furthermore, as differences in software versions contribute to different outcomes workflow languages are being combined with containers: such as singularity and docker (Tommaso et al., 2015).

2.3.2. Detecting differential expression

Determining changes in the expression of genes across conditions is a common RNA-Seq data analysis task, but differential expression analysis is easily confounded by RNA-Seq biases (Soneson & Robinson, 2018). RNA-Seq data sets need to be normalised to remove gene length and sequencing run-dependent biases introduced during reverse transcription and amplification. Sequencing bias can be removed by normalising to internal controls. Internal normalisation commonly consists of converting mapped reads into transcripts per million (TPM). Transcripts per million divides the number of mapped reads mapped to a gene by the length of that gene and the total number of reads mapped in that sequencing run. Therefore, it accounts for the total read variation between runs and the gene length biases. However, dividing by the total number of reads in a run introduces a dependence on the behaviour of a subset of highly expressed genes that constitute the majority of the transcriptome (Zhao et al., 2020). Alternatively, several methods have been developed to detect genes that are expressed at constant levels across all conditions, i.e. quantile

normalisation (Evans et al., 2018), median of mean ratio (Anders & Huber, 2010) and the trimmed mean of the m-values (Robinson & Oshlack, 2010). Any changes in the stable genes can then be assumed to be due to sequencing bias so normalising all other genes by the number of reads mapped to stable genes can help remove the bias.

Statistical methods to determine significant changes in expression have been developed to account for the low replication and high variability of discrete RNA-Seq data. Originally, a statistical method developed to analyse microarray experiments was applied to RNA-Seq data. Linear Models of Microarray Data (limma) was developed to detect changes in the ratios of the fluorescence of target probes corresponding to genes of interest across conditions (Smyth, 2005). Although limma has been successfully applied to several RNA-Seq experiments (Ritchie et al., 2015), the noise structure of continuous fluorescence values is fundamentally different from the integer counts of RNA-Seq data. Regression on integer data sets is more accurately modelled by discrete distributions such as a Poisson distribution (Cameron & Trivedi, 1998). However, Poisson models are limited in their ability to model noisy data as its variance must equal its mean by definition. edgeR (Robinson & Oshlack, 2010) and later DESeq (Anders & Huber, 2010) offered an alternative noise model specifically for RNA-Seq data by using a negative binomial distribution as an overdispersed Poisson count model. These methods increased statistical power despite the low number of replicates typical of RNA-Seq experiments by sharing information across genes to determine the overdispersion parameter of the negative binomial. A further improvement to modelling the dispersion of RNA-Seq data sets in DESeq2 included a regression step on the gene-wise dispersions with respect to their means. Shrinking the dispersion parameter of a gene towards the regression model trained across all genes enhanced the statistical power when detecting differential expression (Love et al., 2014).

2.3.3. Downstream analysis of transcriptomic data sets

Overcoming the n $<<$ p problem has been a fruitful task for applied statistics with robust methods being developed for sharing expression behaviour across genes and conditions (Gui et al., 2005). In investigations of linear covariates, robustness to noise and outliers can be improved by using alternative loss functions, such as the least absolute deviation, or the introduction of penalising terms, such as the L_n-norm of the regression coefficients (Wu & Ma, 2015). Reducing the dimensionality of data sets to emphasise regions of interest has also become standard through methods such as principle component analysis (Wall et al., 2005). A variety of machine learning architectures have also been successfully applied to big data across biology ranging from detecting cancer to predicting gene expression (Liang et al., 2015; Xie et al., 2017; Tang et al., 2019). However, the effec-

tiveness of an algorithm is decided by the quality of the software that implements it.

2.4. Research Software Engineering

The development of high-throughput, multi-omic experiments across the biological sciences has led to an unprecedented demand for software for research. In the late 90s less than 20% of research papers mentioned the use of software in their research, but by 2021 over 70% of publications stored on PubMed cited the use of software. Software developed specifically to answer research questions has rapidly become a cornerstone of the modern empirical method (Schindler et al., 2022). However, academia has been slow to incorporate software development practices into training programs and to create official career paths for experts in research software development. The academic position of research software engineer was only coined in the late 2000s (Prause et al., 2010) with the creation of the society of software engineers being founded in 2010.

2.4.1. Software development practices

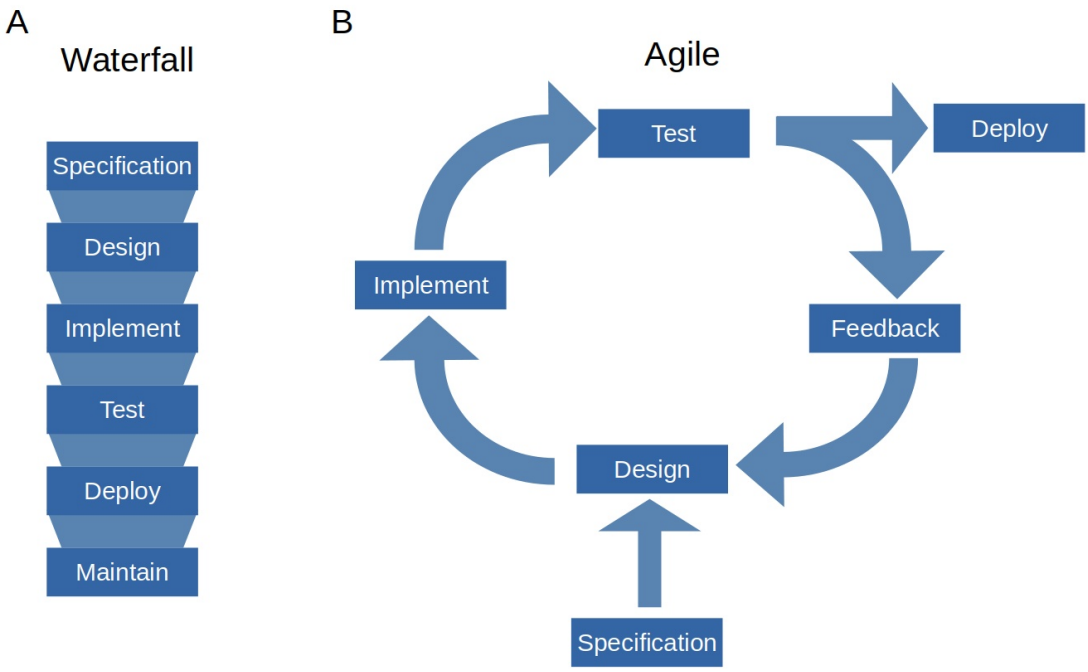


Figure 2.4: Comparison of two common software development practices. A Waterfall consists of a linear sequence of tasks each of which must be fully completed before moving on to the next. **B** Agile focuses on gaining feedback as soon as possible by quickly implementing small changes and using feedback to influence the next design stage.

The simplest software development practice that biologists can implement when writing any code is don't repeat yourself (DRY). DRY encourages programmers to write short and specific functions to solve regularly occurring tasks. Minimising the number of

repetitions helps reduce errors; which can be introduced by imperfect copying, improves readability, and enables faster debugging as compartmentalising tasks into separate functions enables testing of each function separately (Thomas & Hunt, 1999). However, DRY does have limits as the focus on general abstractions can lead to unreadable code. For example, a correctly design suite of tests is intended to help diagnose bugs during the development stage. Abstracting error messages to the shortest, most general form can invalidate their usefulness in diagnosing bugs.

Research projects that involve more extensive computational analysis can benefit from incorporating the structure given by software development methods used in industry. A common software development method is called waterfall or plan-driven development. Waterfall introduces a structure to the coding practice by outlining a series of stages that are completed linearly, Figure 2.4A. It starts with a detailed specification before moving to development and implementation. Although the traditional implementations of waterfall encourage a strict sequential structure to the completion of a software development task, modified waterfall methods enable a degree of flexibility as adjacent steps can overlap enabling some aspects of the design to change as the software is implemented (McConnell, 1996).

Agile is an alternative branch of software development methods that focuses on getting regular feedback and deviates from the waterfall ethos of leaving testing to a later stage. They prioritise creating a minimal viable product as soon as possible and testing its functionality. The specification of an agile project be altered and redefined as the project develops, Figure 2.4B. The principles behind the agile development method are outlined in the agile manifesto, agilemanifesto.org. The inclusion of agile practices in biomedical research suggests the iterative nature of exploratory research combines effectively with the flexibility of agile software development(Kane et al., 2006).

Two common agile practices in industry that have already been successfully implemented in research software development are scrum and extreme programming (Sletholt et al., 2011; Sadath et al., 2018). Scrum is the modern archetypal agile method (Schwaber & Sutherland, 2020). Instead of fixing the development schedule the project is broken into sprints. Each sprint iteratively adds some functionality which is reviewed, tested and implemented before moving on to the next. Constantly reviewing and testing the code enables programmers to catch bugs early and to receive feedback on whether the initial functionality is useful and achievable. As its name suggests extreme programming pushes the principles of agile programming to the extreme (Beck & Andres, 2004). Updates to the software are tested and implemented on a weekly basis. In addition, programmers are expected to conduct paired programming; two programmers work together at all times with only one coding while the other verbally dictates what should be added. Although

this reduces the number of lines of code written per developer, constantly reviewing each other's code reduces the number of errors which is assumed to negate any reduction in productivity.

Despite there being multiple software development methods available in industry, there remains a need for a general method that meets the demands of research software development in an academic environment, (Cereci & Karakaya, 2018). The variety of team sizes, project times scales, software development expertise and the usage of research software for exploratory analysis leads to difficulties in finding similarities between projects that can be used to build development methods (Hannay et al., 2009; Gomez-Diaz & Recio, 2019). Hybrid software development methods that combine the overarching structure of waterfall with the flexibility of agile could meet the hypothesis-driven and exploratory demands of research software development (Pathak & Saxena, 2012).

2.4.2. Open source software development

The growing demand for research to be open access has led to almost 25% of all publications on the web to be openly available in some form (Khabsa & Giles, 2014). The demand for research software that is openly available on public repositories, therefore, is also increasing. Open source research software can also have improved findability, accessibility and reproducibility. Open science research overall is linked with increased citations, funding bodies placing more weight on open access policies and open projects tend to get more coverage in the media (McKiernan et al., 2016). However, these advantages are only achievable if quality coding practices are implemented using public repositories (Prlić & Procter, 2012). A modern parable for open source software development comes from the epidemiological modelling of the spread of COVID-19 by Prof Neil Ferguson at Imperial College London. Crucial to the justification of national lockdowns to kerb the spread of COVID-19, the model was actually developed 13 years prior using undocumented, closed source C++ code. After six weeks of intense revisions and refactoring, with direct help from Microsoft and GitHub software architects, the model code is now the perfect example of open source research software github.com/mrc-ide/covid-sim.

2.4.3. Software documentation

2.4.3.1. The importance of quality documentation

The literature on developing useful and usable bioinformatics software is unanimous on the need to document how and why to use a package (da Veiga Leprevost et al., 2014; Taschuk & Wilson, 2017; Wilson et al., 2017). Furthermore, the widely popular Findable, Accessible, Interoperable and Reusable (FAIR) principles for scientific data now

has a similar set of principles for FAIR research software and documentation is at the forefront: "R1. Software is described with a plurality of accurate and relevant attributes" (Barker et al., 2022). Widely used software repositories, such as Bioconductor, demand long-form documentation outlining the decisions made in creating a package as well as how to interact with it in order for the package to be accepted (Gentleman et al., 2004). Documentation acts as "a resource for learning and a second role: as an advertisement for the software project" and the current health of a project (Geiger et al., 2018). As well as being best practice for ensuring code usability, quality documentation can also improve the quality of the code itself. Documentation of technicalities and a suitable code of conduct can help develop a community of maintainers that can fix bugs, update dependencies and add functionality together (The Turing Way Community, 2022). Open source documentation also combats unconscious knowledge as developers of the code can overlook key pieces of information for using the software that can only be rectified by new users contributing to the package (Hermann & Fehr, 2022).

2.4.3.2. Factors contributing to poor documentation

Software is published with inadequate documentation because writing software documentation is a neglected step in software development. In a 2017 GitHub survey of OSS contributors, 93% reported that “incomplete or outdated documentation is a pervasive problem” but “60% of contributors say they rarely or never contribute to documentation” (Geiger, 2017). Software documentation typically is the least credited part of software development with little time or funds allocated to its development. In industry, documentation writers are first to go in times of economic difficulty (Forward & Lethbridge, 2002). In academia, research posts are only for a few years so there is little time, or motivation, for the developer to respond to user queries (Hermann & Fehr, 2022). Simultaneously, writing software documentation requires the most diverse set of skills and experiences to enable people from different backgrounds and knowledge to engage at an appropriate level (Geiger et al., 2018).

Software documentation needs to meet multiple demands and engage users with different skill sets in order to be adequate. Previous research found common issues were based on factually incorrect statements in the documentation, sections of code/functions without any documentation at all or documentation becoming out of date with the latest package versions. Other issues discuss the difficulty at which API documentation could be found and searched at all, exactly what terms meant in specific contexts and not having quality translations of documentation in other languages. As expected, a complete lack of documentation is the most common issue but on the other extreme is dense, unintelligible

documentation that is difficult to maintain and search (Aghajani et al., 2019).

2.4.4. Suggestions for improving software documentation

Understanding the purposes of different types of documentation can help improve the overall quality of research software documentation. Previous studies have recognised three categories of documentation: documentation of decisions, what problem does this software solve and why was this particular method chosen to solve it; documentation of product, what is contained within this software implementation and how do users interact with it; and documentation of technicalities, how did the developers create the software and how can maintainers contribute to it. Any software intended to be shared contains some product documentation, but few research software projects outline technical details and fewer still mention any decisions made in development (Geiger et al., 2018).

Documentation methods need to be developed to structure the writing of documentation to meet the needs of multiple users and tasks. Diataxis is a framework for creating documentation using its two axes of knowledge: theory vs practice and acquisition vs application. They separate software documentation into four rough types: Tutorials, practical, and application knowledge; How-tos, practical, and acquisition knowledge; references, theoretical, and acquisition knowledge; and explanations, theoretical, and application knowledge, diataxis.fr. Following a systematic approach to developing software documentation helps projects cover the range of needs of documentation users from first-time users to regular maintainers.

The solutions to improving research software documentation target the three main causes: lack of understanding of how to document software, loss of focus on the audience and lack of time allocated to writing documentation, (Rios et al., 2020). Developers of research software need to be taught the pedagogy of software documentation and the tools available to support documentation. Institutes such as the Software Sustainability Institute and the Turing Institute are supporting training and learning resources, but little is mentioned in formal data analysis training. Similar to the frameworks developed for software development, documentation frameworks need to be popularised to acknowledge the continued effort required to keep documentation relevant, accurate and searchable.

Researchers and software developers need to be rewarded for creating usable and documented software packages. Recognising and correctly citing the use of software should be as important as citing research papers. Long-lasting code requires long-term funding which needs to be supported by suitable grants judged on appropriate criteria (Goble, 2014). Funding bodies and journals have acknowledged across the board that research data needs to be FAIR. The FAIR principles for research software need to be incorporated into funding decisions to reward those who create usable software (Hong

et al., 2022). Finally, encouraging open source research development will improve documentation as both benefits from contributions from a diverse and inclusive community of maintainers (Strasser et al., 2022).

Chapter 3

Materials and Methods

3.1. Statistical Methods

The list of statistical methods described has been curated to cover the methods implemented in this thesis. A comprehensive overview of methods to explore data in the $p >> n$ regime, where p is the number of features and n is the number of data points, is beyond the scope of this section. This section draws broadly from several sources: the Elements of Statistical Learning (Hastie et al., 2009), Bayesian Data Analysis (Gelman et al., 2014) as well as relevant review papers (Wu & Ma, 2015; Greener et al., 2021). A more representative view of available statistical methods can be found in the original literature.

The list also reveals a pragmatic approach to the frequentist vs Bayesian debate on statistical inference. Philosophically, the idea that there is an objective truth to be found for any inference task, which motivated the methods within the frequentist ideology, has been unhelpful at best in the pursuit of scientific knowledge. The Bayesian method of incorporating prior knowledge into a model and exploring uncertainty in your results by sampling from a posterior distribution offers a highly applicable structure for compartmentalising sources of error. However, the efficiency at which frequentist methods can be applied and the quality of available software implementations using them means they are useful tools in high-throughput data analysis. Their use as methods to highlight fruitful avenues for further exploration is distinct from their misuse as arbitrators of truth.

3.1.1. Correlation metrics

The initial exploration and quality checking of a data set typically includes the determination of a linear dependency between variables. The Pearson correlation coefficient, R, is a metric for determining positive, negative or uncorrelated linear dependencies between two

variables. The correlation coefficient between two random variables x and y is defined as

$$R = \frac{\sum_i (x_i - \mu_x)(y_i - \mu_y)}{\sigma_x \sigma_y}$$

where μ_x, μ_y are the mean values for x and y and σ_x, σ_y are the standard deviations of x and y. Alternative correlation metrics have been developed that enable the exploration of non-linear monotonic relationships between variables, such as the Spearman's rank correlation coefficient.

3.1.2. Linear regression

Predicting observations from linear combinations of variables, or combinations of transformed variables, is the most studied model in statistics as well as being the starting basis of many non-linear models. A linear model with input vector $X^T = (x_1, x_2, ..., x_p)$ has the form

$$f(X) = \beta_0 + \sum_j x_j \beta_j$$

where β_j are the coefficients of interest to be determined. Assuming Gaussian noise with constant variance, σ^2, on the observation, y,

$$y = f(X) + \varepsilon \qquad \varepsilon \sim N\big(0, \sigma^2\big)$$

we get the likelihood of getting this observation given the predictor variables from

$$L(\beta, \sigma^2; X, y) = (2\pi\sigma^2)^{-1/2}\, exp\Big(-\frac{1}{2\sigma^2}(y - f(X))^2 \Big).$$

The task of linear regression is to find the values of β that maximise the likelihood function $L(\beta, \sigma^2; X, y)$. In frequentist statistics, the objective is to find the point values of maximum likelihood denoted $\hat{\beta}$. The most common method to determine $\hat{\beta}$ is to minimise the residual sum of least squares (RSS) over N observations,

$$RSS(\beta) = \sum^{N} (y - f(X))^2.$$

Writing the above in matrix form, with **y** being the vector of observations and **X** the matrix of predictor values for each observation,

$$RSS(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta),$$

differentiating with respect to β and setting to zero gives the $\hat{\beta}$ that minimises the RRS

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

It can be proved that the $\hat{\beta}$ that minimises the RSS maximises $L(\beta, \sigma^2; X, y)$ as defined above, (Hastie et al., 2009).

3.1.3. Penalised linear regression

In standard linear regression, the introduction of more predicting variables will always increase accuracy on a training set as the model begins learning patterns in the noise. As biological data sets often contain multiple possible predictors and are based on stochastic processes that are inherently noisy a model needs to select biologically relevant predictors. Penalised linear regression enables variable selection by introducing additional terms to the likelihood that penalise the inclusion of predictors. The penalty acts on the coefficients of all predictors creating penalised coefficients, $\hat{\beta}$, with general definition

$$\hat{\beta} = argmin_{\beta}\{L(\beta; x, y) + pen_{\lambda}(\beta)\}.$$

The penalty function $pen_{\lambda}(\beta)$ has a parameter λ that can be optimised to increase the penalty of adding coefficients and reduce the complexity of the model. A common penalty function is the L_n norm acting on the magnitudes of the coefficients

$$pen_{\lambda}(\beta) = \lambda \sum_j |\beta_j|^n.$$

The L_1 and L_2 norms are regularly implemented with the corresponding penalised regression methods called lasso and ridge regression (Hoerl & Kennard, 1970; Tibshirani, 1996). The choice of norm does have a significant effect on the regression with the L_1 norm able to set penalised coefficients exactly to zero, but the L_2 norm able to deal with collinearity in a more intuitive way by pulling the coefficients of collinear terms to the same value rather than arbitrarily setting some to zero. Furthermore, work to create a compromise between the two norms has created the elastic-net penalty

$$\lambda \sum_j (\alpha|\beta_j|^2 + (1-\alpha)|\beta_j|)$$

where α is an additional parameter to be optimised (Zou & Hastie, 2005). This version attempts to combine with variable selection properties of the L_1 norm with the collinearity properties of the L_2 norm.

3.1.4. Bayesian hierarchical models

The Bayesian view of probability is that it represents a reasonable expectation of an event given what we know (Cox, 1946). The fundamental basis of a Bayesian model is Bayes' Theorem

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)},$$

which states that the probability of getting certain parameter values given the data, $P(\theta|D)$, is equal to the likelihood of getting this data given the parameters, $P(D|\theta)$, multiplied by the probability distribution over all possible values of the parameter, $P(\theta)$, divided by the probability of all possible data points, $P(D)$. $P(\theta|D)$ is known as the posterior distribution, $P(\theta)$ is the prior distribution and $P(D|\theta)$ is the likelihood function. The theorem was published by Reverend Thomas Bayes in 1763, but the majority of the modern interpretation of Bayesian statistics was developed independently by Pierre-Simon Laplace from 1774 onwards.

Bayesian hierarchical models are designed if the prior distribution itself contains parameters, ϕ,

$$P(\theta, \phi|D) = \frac{P(D|\theta, \phi)P(\theta|\phi)P(\phi)}{P(D)}.$$

The higher tiered priors may act across multiple θ enabling information to be shared across data points to counter the $p >> n$ problem. As an example, consider a Bayesian implementation of linear regression. Instead of finding the optimum value of the coefficients, $\hat{\beta}$, we are interested in the posterior distribution given the training data, $P(\beta|D)$. We can use the same likelihood function, but we need to define a prior distribution on the values of $P(\beta)$. We can recreate the feature selection properties of lasso regression if we use double-exponential distribution centred on zero for the values of β. The data, through the likelihood function, must then shift the probability mass above or below zero to suggest non-zero β values. Alternatively, a hyperparameter can be introduced to the double-exponential distribution to select a bias term other than zero. The hyperparameter could be trained across all terms in the linear model possibly learning that most βs are actually 1.

3.1.5. Gaussian processes

Gaussian processes are a highly applicable tool for Bayesian inference. Gaussian processes are an extension of a multivariate normal distribution to infinite dimensions. It is a collection of random variables with any finite subset having a joint Gaussian distribution (Rasmussen & Williams, 2005). The collection is indexed by a variable typically representing time as Gaussian processes were originally developed to filter and smooth

noise time-series data. A Gaussian process is fully defined by a mean function, $m(t)$, and co-variance function, $k(t,t')$, of a real function $f(t)$

$$f(t) \sim GP\big(m(t), k(t,t')\big)$$

$$m(t) = \mathbb{E}[f(t)] \qquad k(t,t') = \mathbb{E}[(f(t) - m(t))(f(t') - m(t'))].$$

Unlike linear regression, Gaussian processes act on function space rather than the weight space of the β coefficients. This means a Gaussian process is able to approximate practically any function and is far more flexible when fitting data that is non-linear and/or correlated. The definition of the covariance function can give the Gaussian process a variety of useful properties and defines the prior distribution in the Bayesian paradigm. The squared exponential is a common covariance function,

$$k(t,t') = exp\big(-\frac{1}{2}|t-t'|^2\big)$$

which is infinite differentiable leading to a very smooth Gaussian process. Akin to any Bayesian method, test data points, $f(t_*)$, can be sampled by conditioning the joint Gaussian prior distribution on any given training observations, $f(t)$,

$$f(t_*) \mid t_*, t, f(t) \;\sim\; N\Big(k(t_*,t)\; k(t,t)^{-1}\; f(t), k(t_*,t_*) - k(t_*,t)\; k(t,t)^{-1}\; k(t,t_*)\Big)$$

.

3.1.6. Model selection

Rigorous assessment criteria are needed to select the best model between a group with different sets of predictors or with different penalty terms, i.e. λ. Ideally, the model with the lowest prediction error for all possible data would be selected. However, since any data set is a subset of all possible data the prediction error of a model can only be approximated. K-fold cross-validation is a popular method to approximate the prediction error by splitting the available data into K equal-sized groups and calculating the mean prediction error when each of the groups in turn is excluded from the train set and used as the validation set

$$CV(\hat{f}) = \frac{1}{K}\sum_i L(y_i, \hat{f}^{-k(i)}(x_i))$$

where $\hat{f}^{-k(i)}(x_i)$ is the model trained without the i^{th} group. The model with the lowest cross-validation error is then selected.

Alternatively, the Akaike information criterion (AIC) and the Bayesian information criterion (BIC) can be used to assess models instead of the approximate prediction

error.

$$AIC = -\frac{2}{N}\, log\, L(y, f(x)) + 2\,\frac{d}{N},$$
$$BIC = -2\, log\, L(y, f(x)) + d\, log(N)$$

where is d the degrees of freedom, typically the number of predictors (Akaike, 1971; Schwarz, 1978). The AIC is derived from the extension of the maximum likelihood principle over the parameters space to include uncertainty over the dimensionality of the parameters, not just the values of the parameters. Borrowing from information theory, the distance between the ideal model and a candidate model can be measured with the Kullback–Leibler divergence. The AIC is the asymptotically unbiased estimate of the discrepancy in the Kullback–Leibler divergence between models of different dimensionality.

In the Bayesian paradigm, if we have the posterior probability of two different models, $Pr(f_a|X,Y)$ and $Pr(f_b|X,Y)$ we can calculate the posterior odds

$$\frac{Pr(f_a|X,Y)}{Pr(f_b|X,Y)}.$$

Model f_a would be selected if the posterior odds > 1, else f_b is selected. However, using Bayes theorem

$$\frac{Pr(f_a|X,Y)}{Pr(f_b|X,Y)} \propto \frac{Pr(X,Y|f_a)}{Pr(X,Y|f_a)}$$

which is known as the Bayes factor. Under some approximations of normality of $Pr(X,Y|f_a)$ it can be shown that

$$log\, Pr(X,Y|f_a) \approx log\, Pr(X,Y|f_a) - \frac{d}{2}\, log(N)$$

which is $-1/2 \times BIC$. Comparing the BIC values of multiple models is approximately equivalent to comparing the posterior odds of the models. It is important to note that both AIC and BIC contain approximations that do not hold in all cases. A suitable example of model selection with AIC and BIC is with nested models, i.e. when one model contains a subset of predictors contained in the other model.

3.1.7. Model evaluation

Once a model has been selected, its effectiveness at predicting the observed data can be evaluated. The coefficient of determination, R^2, represents the fraction of the total variance in the observed data that is explained by the model. R^2 is a common goodness-of-fit metric for linear models and generally defined as

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

$$SS_{res} = \sum_i (y_i - \hat{y}_i)^2$$

$$SS_{tot} = \sum_i (y_i - \bar{y})^2,$$

where y_i is an observed value, $\bar{y}$ is the mean of the observed values, and $\hat{y}_i$ is the predicted value. In linear regression, the coefficient of determination can also be defined as the square of the correlation coefficient. This limits the value of R^2 to between 0 and 1. Although in the general case, the coefficient of determination can be negative, i.e. when $SS_{res} > SS_{tot}$.

3.1.8. Multiple hypothesis testing

Testing whether an experimental result is statistically significant given some approximate model of the process creating the data is a mainstay of modern research. The decision on where to place the boundary on what is or is not significant is ultimately decided by the researcher's concern about Type I errors; falsely declaring a result significant when it actually arose from the variation in the data, and Type II errors; falsely declaring a result insignificant when it is in fact unexpected. As high-throughput experiments enable researchers to test thousands of hypotheses simultaneously the susceptibility to Type I errors increases dramatically as more outliers are expected to be detected. For example, in a frequentist manner, define a result to be significant if there is less than 5% chance a result like it, or more extreme than it, occurs given the null hypothesis is true. If we test 100 results using this method, we would expect 5 results to exceed this threshold even if all the results are insignificant, each being Type I errors. If we test 10,000 results then 500 Type I errors are expected. Two common methods to account for the increase in Type I errors when testing multiple hypotheses are the Bonferroni correction and the False Discovery Rate (FDR) (Bonferroni, 1936; Benjamini & Hochberg, 1995a). The Bonferroni correction scales the threshold, α, at which a result is considered significant by the number of tests being conducted, N,

$$\alpha_{Bon} = \frac{\alpha}{N}.$$

The correction is known to be conservative leading to more Type II errors and lower statistical power. The FDR is the ratio of Type I errors, V, to total number of results

called significant, S,

$$FDR = E\left(\frac{V}{S}\right).$$

The FDR method attempts to keep the FDR constant by changing the threshold of significance given the total number of tests to be conducted and the number of results already tested, j,

$$\alpha_{FDR} = \alpha\frac{j}{N}.$$

The FDR method is implemented by ranking the results according to their p-value and comparing the p-value to a scaled α_{FDR}.

3.2. tidyqpcr Software Development

tidyqpcr was developed using the open software development best practices taught by the eLife Innovative Leaders Program elifesciences.org/labs. Following the Mozilla Open Leaders resources, we developed user personas to describe the types of users that would want to use our software. We created a development pathway with clear targets for the minimal viable product and subsequent updates to the MVP mozillascience.github.io. The development followed the Extreme programming paradigm with continuous integration and testing using GitHub Actions. Updates were small and often, sometimes paired programming was implemented to develop significant updates. All updates were first added to individual branches before being pulled into the protected main branch after code review. Unit testing was implemented using the testthat R package with a minimal test coverage of 80% (Wickham, 2011). A full code review was completed following the Google literature google.github.io/eng-practices/. Function documentation was created using the software package roxygen2 whilst the detailed examples were created using Rmd vignettes (Wickham et al., 2021; Wickham & Seidel, 2022). The code was developed using git version control. A permissive licence, Apache-2.0, is used to share the code enabling users to modify the source code to their own problems. Decision-making was conducted in an open manner using GitHub issue tickets and significant software changes were denoted by GitHub tags to enable users to revert to older versions. Infrastructure to encourage inclusive and encouraging inputs from new contributors was provided by introducing a code of conduct created by rOpenSci.

3.2.1. User interviews

We explored how users interacted with tidyqpcr by conducting 6 semi-structured, exploratory interviews. The interviewees consisted either of academic colleagues who were known to have conducted qPCR previously or were intending to conduct research in the near future, 3.1. User interviews for tidyqpcr were conducted and recorded using zoom.

Position	Coding Experience	qPCR Experience
Undergraduate student	Novice Python user	No prior experience
Senior post-doctoral research assistant.	Intermediate R user	Conducting qPCR for >10 years
Research assistant	Intermediate R user	Conducted 1000s of qPCR experiments
PhD student	Confident R user	Conducted several qPCR experiments
PhD student	Novice R user	Conducted several qPCR experiments
PhD student	Intermediate Python user	Conducted several qPCR experiments

Table 3.1: Overview of the coding and qPCR experience of tidyqpcr interviewees

The audio was then transcribed using otter.io rather than the default zoom transcriber as otter.io allows you to introduce specialist vocabulary (such as tidyqpcr, TaqMan, ect.) to aid accuracy. Once the transcripts were available they were split into two halves: the semi-structured interview and the user task. The semi-structured interview and the user task sections were combined for all the user interviews. Each combined section was then analysed separately. The text mining R packages tm and pluralize were used to preprocess the transcripts and extract frequently used words (Feinerer et al., 2008; Rudis & Embrey, 2020). The text was pre-processed by removing whitespace, numbers and common words; converting all plural nouns to singular; and changing all letters to lowercase. The frequency of each word was then counted and any generic words regularly occurring within the transcript were removed, i.e. that, like and thing. Finally, the word frequency matrix was used to create the text cloud with the R package wordcloud (Fellows, 2018).

3.3. Limitations of Composability of Cis-Regulatory Elements in Messenger RNA

I did not conduct any of the experimental assays discussed in this thesis. The experiments were done by Jamie Auxillos and Weronika Danecka with help from Abhishek Jain and Clemence Alibert. I did contribute to the planning of the qPCR, construct design, plate reader and RNA-seq assays conducted in the lab. I outline the protocols here for completion.

3.3.1. Strains and media

Saccharomyces cerevisiae strain BY4741 (*MATa his3*Δ1 *leu2*Δ0 *met15*Δ0 *ura3*Δ0) was used as the wild-type strain in this study, and the host for all yeast plasmid transformations. For all quantitative assays, plasmid-transformed strains were grown in synthetic complete medium without uracil (SC-Ura), containing 0.69% yeast nitrogen base without amino acids and with ammonium sulfate (Formedium, Norfolk, UK), 0.193% amino acid drop-out supplement mixture (Formedium, Norfolk, UK) and 2% glucose. To prepare BY4741 for transformation, we grew it in YPDA medium, containing 2% peptone, 1% yeast extract, 2% glucose and 0.004% adenine.

3.3.2. Construction of chimeric reporter plasmids

All fluorescence reporter plasmids were constructed by Golden Gate assembly using the YeastFab system as described in (Garcia-Ruiz et al., 2018). Promoters, coding sequences and terminators were either amplified from the yeast genome or synthesised by a commercial vendor (IDT) and then cloned into a parts accepting plasmid (HcKan_P for promoters, HcKan_O for coding sequences and HcKan_T for terminators) by Golden Gate assembly using Bsa1-HFv2 (NEB). A detailed protocol for Golden Gate assembly is available at protocols.io, doi:10.17504/protocols.io.bkqrkvv6. Using these parts libraries, the promoters, coding sequences and terminators were assembled together into the transcription unit acceptor plasmid (POT1-ccdB) by Golden Gate assembly using Esp3I (NEB); these are low-copy centromeric plasmids with URA3 selection marker, as described in (Garcia-Ruiz et al., 2018). Plasmid inserts were confirmed by Sanger sequencing (MRC PPU DNA Sequencing and Services, Dundee). DNA sequences used in this study are summarised in Supplementary Tables B4 and B5. Assembled plasmids were transformed into yeast BY4741 using lithium acetate transformation (Gietz & Woods, 2002), and selected in SC-URA agar plates to isolate successful transformants.

The mCherry coding sequence is as used in (Garcia-Ruiz et al., 2018), which in turn was amplified from the mCherry sequence in (Sharon et al., 2012). The mTurquoise2 coding sequence is as used in (Lee et al., n.d.).

3.3.3. Fluorescence measurements: Plate reader analysis of strain growth and fluorescence

Yeast with plasmids were grown in a 96-well deep well plate (VWR) containing 100μl of SC-Ura medium with 2% glucose and grown for ~12 hours at 30°C in a shaking incubator set at 250 rpm. The next day, the cultures were diluted to an OD of 0.2. For each sample, 3 technical replicates of 200μl were transferred to a 96-well black microtiter plate (Corning) and grown according to the protocol described in (Lichten et al., 2014). The Tecan

Infinity M200 series plate reader was set at the temperature of 29.9 (range of 29.4-30.4°C) with linear shaking (6 mm amplitude at 200-220 rpm). OD measurements were carried out at an absorbance wavelength of 595 nm with a measurement bandwidth of 9 nm with 15 reads. mCherry fluorescence measurements were carried out with an excitation wavelength at 585 nm and an emission wavelength of 620 nm (excitation bandwidth of 9 nm and emission bandwidth of 20 nm) with the gain set at 100. mTurquoise2 fluorescence measurements were carried out with an excitation wavelength at 434 nm and an emission wavelength of 474 nm (excitation bandwidth of 9 nm and emission bandwidth of 20 nm) with the gain set at 60.

Plate reader data were analysed using omniplate software (Swain et al., 2016). Omniplate accounts for autofluorescence and fits a model to the time series data using Gaussian processes to infer the time of maximum growth rate for each well. We minimised growth-dependent effects by using the fluorescence at maximum growth rate for all of our protein fluorescence experiments. Each fluorescence measurement was also normalised by OD to remove dependency on cell number, so every protein fluorescence measurement is recorded as fluorescence per OD at max growth rate. A detailed protocol for setting up and conducting the plate reader assay is available at protocols.io, https://dx.doi.org/10.17504/protocols.io.bbicikawdoi:10.17504/protocols.io.bbicikaw. The log2 fold change in fluorescence per OD at max growth rate with respect to the tPGK1 construct of each promoter set were deduced using a linear model with terminators as predictors. p-values were calculated using t-tests and converted into p.adj values using the FDR (Benjamini & Hochberg, 1995b).

3.3.4. RNA measurements: Strain growth, RNA extraction, RT-qPCR, RNA-Seq and analysis

Yeast with plasmids were grown in a 24-well deep well plate (4titude) containing 1.5 ml of SC-Ura for ~20 hours at 30°C in a shaking incubator set at 250 rpm. The next day, the OD was diluted to a starting OD between 0.15-0.2 in a 12-column deep well reservoir plate (4titude) to a total volume of 7 ml. Diluted cultures were grown at 30°C in a shaking incubator set at 90 rpm to an OD of 0.5-0.7 then pelleted by centrifugation. Pelleted cells in the plate were stored at -80°C.

To extract RNA, we adapted a silica column DNA/RNA extraction protocol from Zymo Research (Irvine, California, USA). The pelleted cells were thawed and individually resuspended in 400 μl of RNA binding buffer (Zymo), then transferred to 2 ml screw-cap tubes containing zirconia beads, lysed using the Precellys Evolution homogeniser then pelleted by centrifugation for 1.5 minutes. The supernatant was transferred to a Zymo Spin IIICG column (Zymo) and then centrifuged. The flow through was mixed with 1

volume of ethanol then transferred to a Zymo Spin IIC column (Zymo) and centrifuged. This flow through was discarded and 1 volume of DNA/RNA Prep buffer (Zymo) was added and then centrifuged. The column was washed with 700 μl of Zymo DNA/RNA Wash buffer (Zymo) and then centrifuged. The column was washed a second time, but with 400 μl of Zymo DNA/RNA Wash buffer (Zymo). The column was centrifuged once more to remove residual wash buffer in the column. Lastly, 30 μl of nuclease free water was added to the column and then eluted. All centrifugation steps in the RNA extraction protocol were carried out at 12,000g for 1 minute unless otherwise stated. A detailed protocol for yeast growth and RNA extraction is available at protocols.io, doi:10.17504/protocols.io.beetjben.

The quantity and quality of the RNA were measured using both a spectrophotometer (DS-11, DeNovix, Wilmington, Delaware, USA) and Fragment Analyser (Agilent). 4 μg of RNA was treated with DNAse1 (Thermo) and then inactivated using the RapidOut DNA removal kit (Thermo) according to the manufacturer's protocol. 2.5 μl of Random primer mix (NEB) was added to the mixture and then separated into 2 PCR tubes (one for -RT and one for +RT) then denatured at 70°C followed by cooling on ice. Reverse transcription (RT) master mix was prepared, containing 2 μl of First Strand synthesis buffer, 0.75 μl of 10mM dNTP mix, 1.5 μl of nuclease free water, 0.25 μl of RNase inhibitor and 0.5 μl of SuperScript IV Reverse Transcriptase (Invitrogen) per reaction. 5 μl of the RT master mix was added to the denatured RNA and then incubated at 25°C for 5 minutes then 55°C for 1 hour. The cDNA was diluted with 200 μl of nuclease free water.

Target cDNAs were measured by quantitative PCR with Brilliant III Ultra-Fast SYBR Green qPCR master mix (Agilent) using a Lightcycler 480 qPCR machine (Roche). We measured all +RT reactions in technical triplicate, and negative control -RT samples using one replicate. We used the manufacturer's software to calculate the quantification cycle (Cq) for each individual well using the fit points method and exported both raw fluorescence and Cq data. All primer sets were thoroughly validated by serial dilution and by confirming amplicon size. Sequences are available in Supplementary Table B6.

The RT-qPCR data was analysed using our tidyqpcr R package version 0.3. For each biological replicate, ΔCq values were calculated by normalising the median mCherry Cq values by the median Cq values of the three reference genes (RPS3, PGK1 and URA3). For the constructs with motif insertions in terminators, ΔΔCq values were calculated by normalising mCherry ΔCq by that of control construct mod_NNN strains (with the corresponding promoter) for tRPS3 and tTSA1 constructs. For the constructs with motif deletions in terminators, ΔΔCq values were calculated by normalising mCherry ΔCq by that of the WT terminator (with the corresponding promoter) for tPIR1 constructs. Complete scripts for qPCR analysis, quality control, and figure generation are available online

at github.com/DimmestP/chimera_project_manuscript/.

RNA-seq libraries were prepared using QuantSeq 3' mRNA-Seq Library Prep Kit REV for Illumina (Lexogen Gmbh, Vienna Austria). See doi:10.1016/bs.mie.2021.03.020. 500 ng of RNA (not treated with DNaseI) was used as input and the manufacturer's protocol was followed without modifications. The number of amplification cycles was determined using PCR Add-on Kit for Illumina (Lexogen). The quality of the library was measured using Fragment Analyzer NGS Fragment Kit (1-6000bp) (Agilent). Pooled libraries were then sequenced using NextSeq 500/550 (Illumina) with paired-end reads using Custom Sequencing Primer to obtain 3'-end reads.

5PSeq libraries were prepared as described in (Zhang & Pelechano, 2021) with modifications to the reverse transcription step: anchored oligo(dT) was used instead of oligo(dT) to allow for sequencing of 3'-ends and random primers were not used. The library was sequenced using NextSeq system (Illumina) with paired-end reads.

RNA-Seq alignment and quality control were conducted using a pipeline available online at github.com/DimmestP/nextflow_paired_reads_pipeline, written in Nextflow (Tommaso et al., 2017). Quality control was conducted with FASTQC and MultiQC reports (Ewels et al., 2016) and adapters were removed with Cutadapt (Martin, 2011). Alignment was conducted with HISAT2 (Kim et al., 2019), followed by processing with SAMtools (Li et al., 2009) and BEDTools (Quinlan & Hall, 2010). The sacCer3 (R64-2-1, GCA_000146045.1) genome build was used for alignment and transcriptome annotation was originally taken from the Saccharomyces Genome Database (Ng et al., 2020). 5PSeq reads contain UMIs, which were used to deduplicate reads using UMI-tools (Smith et al., 2017); QuantSeq reads do not. Counts to genomic regions of interest were calculated using FeatureCounts (Liao et al., 2014). For 5PSeq data, 5'P ends were also analysed using the fivepseq pipeline Nersisyan et al., 2020. Alternative Poly(A) site usage was tested using a Mann–Whitney U test (Mann & Whitney, 1947) on relative reads mapped to a 9 nucleotide window centred on the major Poly(A) site. p-values were calculated by comparing construct relative counts to wildtype relative counts and converted into p.adj values using the FDR (Benjamini & Hochberg, 1995b).

3.3.5. Determining 3'UTR decay motifs

We initially selected 69 3'UTR motifs to investigate from three separate studies of cis-regulatory elements suspected to regulate mRNA decay (Shalgi et al., 2005; Hogan et al., 2008; Cheng et al., 2017). To select a short list of motifs to test for context dependence, we determined the contribution of each motif to a linear model predicting half-life. Following (Cheng et al., 2017), we quantified the effect of motifs on transcript half-life using a linear model predicting half-life on the basis of codon usage, 3'UTR length, and 3'UTR motif

frequency.

$$log_2(\lambda_g^{1/2}) = \sum_c \beta_c p_{cg} + \sum_m \alpha_m n_{mg} + \gamma \omega_g + \varepsilon$$

where $\lambda_g^{1/2}$ is the half-life of gene g, β_c is the coefficient associated with codon c, p_{cg} is the proportion of gene g's coding sequence that corresponds to codon c, γ is the coefficient associated with 3'UTR length, ω_g is the 3'UTR length of gene g, α_m is the coefficient associated with motif m, n_{mg} is the number of occurrences of motif m in gene g's 3'UTR, and ε is the noise term. To choose 3'UTR lengths and to assess which sequence to use for the 3'UTR motif search, we used the median 3'UTR length estimates (precisely, the median length of clustered major transcript isoforms) reported from the TIF-seq analysis in (Pelechano et al., 2013).

We removed motifs that did not significantly contribute to half-life by using a greedy model selection algorithm that minimises the Akaike information criterion (AIC).

$$AIC = 2k - 2ln(\hat{L})$$

where k is the number of parameters in the model and $\hat{L}$ is the maximum value of the likelihood function (Akaike, 1998). We implement this motif comparison using the R function step (Ripley & Venables, 2002; R Core Team, 2020), to iteratively add the motif which reduces the AIC of the model the most until the penalty for adding news terms overcomes the benefit of including a new motif. This procedure was run on each decay data set independently. p-values were calculated use t-tests and converted into p.adj values using the FDR (Benjamini & Hochberg, 1995b).

The variance explained by the codon usage, 3'UTR length, and motif presence features are estimated in three ways for the linear model trained on the (Chan et al., 2018) data set. The motif presence feature is the total contribution to the linear model when the counts of the all of the shortlisted motifs are included together. The codon usage feature is the total contribution to the linear model when the counts of all codons are included together. The first way to estimate variance explained was with each feature individually as separate regression models. Then, features were added in descending order of their individual explained variance to create three models: codon, codon + motif and codon + motif + 3'UTR length. The variance explained by the joint models are denoted the cumulative variances explained. Finally, starting with the full model with all three features the drop in variance explained when one of the features is removed was reported.

We selected the specific versions of the HWNCATTWY and TGTAHMNTA motifs by running two additional linear models predicting half-life that inferred separate coefficients for each version of its consensus sequence. Coefficients were reported for the

significant motif versions (Supplementary Table B7, B8). We chose instances with similar effect size, statistical significance, and number of occurrences in native transcripts. We chose TTTCATTTC for HWNCATTWY and for TGTAHMNTA chose TGTACAATA over TGTATATTA specifically to avoid the 5nt stretch also found in ATATTC.

3.3.6. Design of modified 3'UTRs for testing the effects of mutated motifs

RPS3 was chosen as the first 3'UTR for inserting motifs into as it was the only terminator in the characterized library that did not contain any of the 69 original motifs of interest. The tRPS3 3'UTR-terminator was modified to incorporate three 9 nt insertion sites for motifs (M1, M2 and M3). The M1 was inserted 24 nt downstream of the stop codon, M2 was inserted 15 nt downstream of M1 and the final insert M3 was inserted 4 nt downstream of M2 (Figure 5.3A). These positions were selected based on key design criteria, including; minimal perturbations of RNA secondary structure as predicted by RNAfold (Lorenz et al., 2011), position of motifs in native 3'UTRs and position of other CREs important for transcriptional control (Supplementary Figure B3, Supplementary Table B9). A control tRPS3 3'UTR mod_NNN was designed to incorporate random bases in each insertion site. Further modified 3'UTR-terminator designs were designed to incorporate individual motifs of interest previously identified, within the insertion sites described (Figure 5.3A).

We chose an alternative 3'UTR for screening the effects of inserting motifs of interest by searching for characteristics similar to RPS3. To this end, median-length 3'UTRs were extracted from the (Pelechano et al., 2013) dataset filtered for the following criteria; 1) does not contain any of the original 69 motifs of interest, 2) < 300 nt in length, 3) from a highly expressed gene, 4) synthesisable as a gBlock by our manufacturer (IDT). The 3'UTR from TSA1 met these criteria.

Similar to the modified tRPS3 designs, in tTSA1 we designed three 9 nt motif insertion sites: M1 21 nt downstream of the stop codon, M2 20 nt downstream of M1, and M3 24 bp downstream of M2 (Figure 5.3A). The tTSA1 mod_NNN construct contained random bases in the M1, M2 and M3 sites, with the motif insertions in other modified constructs as for tRPS3.

To design deletion constructs we selected a native 3'UTR that contained the motifs of interest. Again, median-length 3'UTRs were extracted from the Pelechano et al. (2013) dataset filtered for the following criteria; 1) contains at least 3 shortlisted motifs of interest, 2) a highly expressed gene, 3) synthesisable. The PIR1 terminator chosen for motif deletion contains one copy each of the ATATTC and TGTAHMNT motifs, and 3 copies of the HWNCATTWY motifs (Figure 5.4A), although did not contain the putative stability motif GTATACCTA.

The mutation of motifs for their removal from the PIR1 3'UTR was carried out so that: 1) at least 50% of the motif sequence (specifically the motif consensus sequence) was mutated to a base that does not correspond to the consensus sequence, 2) GC content was minimally altered, 3) Mutations that resulted in a limited change in the predicted secondary structure and minimum free energy (MFE) according to RNAfold (Lorenz et al., 2011), see Supplementary Table B9).

The native and modified candidate 3'UTRs were screened for the presence of Esp3I and BsaI sites within the sequence. For incorporation into the YeastFab system, the sequence 'agcgtgCGTCTCgTAGC' was added to the 5'-end of the 3'UTR and the sequence 'CCTCcGAGACGcagcac' was added to the 3'-end of the 3'UTR. To check if the sequences were synthesisable, 100 nt downstream of the native 3'UTR was added to the candidate construct and the sequence was checked at the IDT gBlock entry form (eu.idtdna.com/site/order/gblockentry).

3.3.7. Determining motif effect on abundance

A linear model predicting construct ΔCq's using the presence or absence of the four selected motifs was trained on each promoter-terminator pairing separately. The model included a term to account for interactions between the TGTAHMNTA and HWNCATTWY motifs The linear model also included a term for batch effects, between the 2 experimental batches of 3 biological replicates for each set of constructs, because this improved the quality of model fit. The model was:

$$\Delta Cq = \sum_{m=1}^{4} \alpha_m n_m + \beta e_{rep} + \gamma n_{int} + \varepsilon$$

where n_m is the copy number of motif m in the construct, e_{rep} is which experimental batch the construct was part of and n_{int} is the interaction term with value 1 if both TGTAHMNTA and HWNCATTWY motifs are present. p-values were calculated use t-tests and converted into p.adj values using the FDR (Benjamini & Hochberg, 1995b).

3.3.8. Predicting changes in transcript abundance from changes in half-life

A simple kinetic model of the production and decay of transcripts was used:

$$\varnothing \xrightarrow{k} m$$

$$m \xrightarrow{\delta} \varnothing$$

The steady state solution for the average number of transcripts, $\langle m \rangle$, is

$$\langle m \rangle = \frac{k}{\delta}$$

where k is the rate of transcription, which can include multiple active states, and δ is the rate of decay for the transcript (Sánchez & Kondev, 2008).

Now consider a control transcript m_0, and a similar transcript with an altered terminator m_a. Assuming the alterations to the terminator of the host gene have a minimal impact on the transcription rate, the above equation says that the ratio of predicted abundance $\langle m_a \rangle$ to the control transcript abundance, $\langle m_0 \rangle$, is the same as the ratio of their half-lives:

$$\frac{\langle m_a \rangle}{\langle m_0 \rangle} = \frac{\delta_0}{\delta_a} = \frac{\lambda_a^{1/2}}{\lambda_0^{1/2}}$$

This gives a linear effect on the log-scale abundance

$$\log\langle m_a \rangle = \Delta \log \lambda^{1/2} + \log\langle m_0 \rangle$$

and because PCR quantification cycle Cq is proportional to $\log 2(m)$, this directly leads to a linear effect on Cq.

All analyses made extensive use of the tidyverse and ggplot2 (Wickham, 2016; Wickham et al., 2019).

Chapter 4

tidyqpcr: Quantitative PCR Analysis in the tidyverse

4.1. Introduction

Quantitative PCR is the most common technique for the quantification of DNA and RNA. The specificity and sensitivity of the assay have led it to be considered the gold standard for nucleotide detection and quantification in medicine, legislation and academia (Kubista et al., 2006). However, qPCR experiments, especially assays requiring a reverse transcriptase step, are susceptible to several reliability issues if poorly designed (Bustin, 2002). Nevertheless, the widespread use of qPCR across distinct disciplines has enabled the creation of countless protocols, equipment and analysis methods without consistent standards (Bustin et al., 2021). The limitations and varying quality of qPCR experiments have led to a reproducibility crisis with significant consequences for academia (Garson et al., 2009) and public health (Bustin, 2013).

qPCR assays are susceptible to multiple biases that can cause highly variable or unrepeatable results (Bustin, 2002). Lab specific protocols for the extraction, pipetting and storage can introduce variation, with samples having $R^2 = 0.4$ between extraction methods (Bustin & Nolan, 2017; Dagnall et al., 2017). The position of a sample on a qPCR plate can also have a significant contribution to measured expression with one study showing 10% of the variation between replicates was due to well position (Eisenberg et al., 2015). Positional effects can be introduced as edge wells may be more susceptible to evaporation or thermal gradients may be uneven across the plate. Finally, qPCR experiments that include a reverse transcriptase step can determine Cq values that differ by up to 91-fold according to the choice of reverse transcriptase and sensitivity to original RNA concentration (Ståhlberg et al., 2004). As different reverse transcriptases interact with RNA secondary structures in different ways, the effect is not consistent across targets (Williams et al., 1992; Brooks et al., 1995).

In an effort to improve the reproducibility of qPCR results guidelines were created for publishing qPCR data called the Minimum Information for the Publication of Quantitative Real-Time PCR Experiments (MIQE) guidelines (Bustin et al., 2009). They provide a thorough checklist of every detail that needs to be reported in order to enable another researcher to accurately repeat an experiment. Over time they developed their own file format, the Real-Time PCR Data Markup Language or RDML, to standardise the way scientists describe their experiments which users can upload to an open source database (Lefever et al., 2009; Ruijter et al., 2015). The guidelines have been around for over a decade, but they are still not widely implemented with as little as 4% of qPCR articles citing them (Nour et al., 2020).

In order to improve the reproducibility of qPCR analyses, we propose the development of a novel analysis package that uses detailed documentation and open software practices to teach and facilitate transparent and quality-controlled qPCR analysis. In addition, the dependence on proprietary software for the calculation of threshold Cq values and quality control graphs remains a stumbling block for reproducible and trustworthy analysis. We believe a novel package developed in the programming language R, a language regularly used by biologists, could increase the quality of qPCR assays from initial experimental design to the final publication of results. The consistent structure of qPCR data also provides an opportunity to apply scalable data analysis practices. Cq values are the variables of interest for most qPCR experiments and they are inherently separated into values from independent wells. Therefore, developing functions that can take advantage of this regular, independent structure to Cq values can lead to a comprehensive and efficient analysis of data sets of varying sizes.

Data analysis practices have been developed to enable efficient and scalable analysis by modularising analysis pipelines and standardising the structure of data sets. Google's MapReduce and Hadley Wickham's split-apply-combine focus on separating the analysis into independent functions and applying each function to subsets of the full data set before combining the results to produce a final summary (Dean & Ghemawat, 2004; Wickham, 2011). Modularising the analysis follows the DRY philosophy as a task, such as calculating the mean and standard deviation, can be encapsulated in one function but applied to different subsets of the data. Designing analysis programs in a modular way also simplifies the task of parallelising the process, which leads to a scalable method as large data sets can be subsetted into manageable chunks according to the computation resources available at run-time (Chua et al., 2004).

To supplement the application of modular analysis practices, there is a broader idea of structuring data in a tidy format to ensure no information is lost when it is split into subsets (Wickham, 2014). Data is considered tidy if it follows a strict row and column

structure. Each column must hold all values of a variable and each row holds a separate observation. Unfortunately, data in the wild is often organised with combinations of observations and variables being spread across columns and rows. However, ensuring each row contains all the information from an observation simplifies the grouping and splitting of a data set. If all functions in an analysis pipeline accept and output data in a tidy form then the entire analysis can be applied to different groupings of data leading to a more comprehensive analysis.

The split-apply-combine paradigm developed by Hadley Wickham has been incorporated into the R programming language through the tidyverse. The tidyverse suite of data analysis packages contains functions covering tasks from graph plotting to conducting statistical tests which all follow a tidy architecture (Wickham, 2014). The implementation of the split-apply-combine paradigm within the tidyverse is facilitated by the `group_by` function. Passing a variable held in the column of a tidy data set to the `group_by` function splits the data into groups of observations with the same value in that column. All functions within the tidyverse are then applied to each group separately and their results are combined. In addition, the functions inside the tidyverse are templates for user-friendly software development. Their source code is accessible and comprehensible as they follow a strict coding style and contain extensive documentation. The functions are also intuitive to learn as they follow a verb-object naming convention that states what they do and what they act on.

Here, we outline the development of tidyqpcr, an R package built within the tidyverse which implements the MIQE guidelines to facilitate scalable, transparent and quality-controlled qPCR analysis. We begin by outlining the continued need for better design and analysis of qPCR data through a review of current software. tidyqpcr is then introduced as an R package that helps users create publication-ready figures of normalised qPCR results. tidyqpcr packages the principles outlined in the MIQE guidelines for the easy design, analysis and reporting of reproducible and accurate qPCR results. Well-documented functions and intuitive structure help users conduct reproducible research without depending on checklists. The chapter ends by discussing improvements to the package motivated by a series of user tests and code reviews as well as future extensions to its functionality.

4.2. Results

4.2.1. qPCR analysis software review

We review the current software landscape for analysing qPCR data and outline the continued need for scalable, user-friendly, and MIQE-compliant analysis software. In the last

two years, the use of qPCR in the global response to the COVID-19 pandemic has driven the development of new packages to support the reliable detection of COVID-19. However, the last published review of qPCR analysis packages was Pabinger et al., 2014. The review covered 27 different open-source packages with the R package qpcR highlighted as the tool with the most comprehensive functionality. It also described the varying quality of documentation, lack of compliance with the MIQE guidelines, inconsistent input and output file formats, and the use of CLIs over GUIs across the reviewed packages. A review of qPCR analysis software released since 2014 is described below in order to A) determine the need for another qPCR analysis package and B) discern any generalisable changes in qPCR analysis software since the last review.

A list of qPCR analysis software was gathered through searches on GitHub, bioconda, bioconductor, CRAN, and Google Scholar. All of the software packages reviewed are freely available for use and are open source. The majority of the packages were released after the previous published review. However, HTqPCR, qpcR, ReadqPCR, and NormqPCR are included for completeness as they are dependencies for several of the newer packages. The packages are grouped according to their primary usage: Web Apps require a server and typically provide a website for users to conduct analyses, R and Python packages primarily need to be downloaded and ran locally, and Misc requires other proprietary software. A table summarising the main functionality of all reviewed software can be found in Appendix A.

4.2.1.1. Software descriptions

Web Apps

QuantGenius A PHP web-app published in Feb 2017 for the quantification of target abundance using standard curves. Users manually copy Cq values for each target into the GUI and can export results as a comma separated file or excel file. The app automatically highlights samples that are outliers, are outside the limit of detection or have poor efficiency, but does not check melt/amplification curves. It does not contain any functions for conducting statistical analyses or for producing graphs. QuantGenius has not been updated since publication (Špela Baebler et al., 2017).

ELIMU-MDx A PHP web-app published in Oct 2019 for the storage and analysis of clinical qPCR data. The app extracts Cq values from the input RDML file and stores results as an RDML file. The PHP backend is able to deduce relative and absolute abundance as well as detect samples that are outliers, are outside the limit of detection or have poor efficiency. Users need to set up their own apache or nginx server to run analyses and

store the database. It does not contain any functions for conducting statistical analyses or for producing graphs. ELIMU-MDx was last updated in Dec 2020 (Krähenbühl et al., 2019).

PIPE-T An extension to the Galaxy web-based bioinformatics platform published in Oct 2019 for the relative quantification of qPCR data. It accepts Cq values for each sample/replicate/condition as separate tab separated files and outputs tab separated files. The extension facilitates a variety of Cq normalisation methods, mainly provided through the R package HTqPCR. QC can be conducted by flagging samples with Cq levels outside user defined thresholds, but no melt/amplification curves are available. There are also functions to test for significant differential expression in two condition experiments and to impute missing data. PIPE-T has not been updated since publication (Zanardi et al., 2019).

SATqPCR A standalone web-app published in Aug 2019 for the relative quantification of qPCR data. It accepts up to two tab separated text files as input: one contains a table of Cq values and primer efficiencies with columns representing different genes and rows representing samples and the other optional file relates samples to different factors for t-test or anova statistical tests. It outputs summary statistics and normalised Cq values in text files and as bar charts in PNG format. The software cannot calculate primer efficiencies but, if efficiencies are provided by the user, it can use primer efficiencies in the relative quantification calculation. It does not have any functionality to plot melt/amplification curves, detect outliers or interpolate missing data. The app does contain an algorithm to automatically detect the most stable genes and use them as normalising genes. The app is an update to a previous R package called RqPCRAnalysis, but has not been updated itself since publication. (Rancurel et al., 2019).

Python packages

Auto-qPCR A standalone web-app with Python back-end published in Oct 2021 for the relative and absolute quantification of qPCR data. It accepts a comma separated file or text input file with specific column names, such as well, sample name, target name, and Cq value. It outputs text files and bar charts in PNG format with normalised ΔCq, $\Delta\Delta$Cq or absolute copy number results. Users can download the python code to run the app locally or use the online server. The function for calculating relative Cq values does not include primer efficiency. The software does not process melt or amplification curves but uses a standard deviation cutoff for outlier identification. It also can conduct a 1 or 2-way anova to test for significance. Auto-qPCR has not been updated since publication (Maussion et al., 2021).

qpcr A python package released in Aug 2021 for the relative quantification of

qPCR data. It accepts comma separated files and excel files in a variety of different formats to import different combinations of experiments, targets and samples. It outputs text files and bar charts in JPEG format with normalised ΔΔCq values. The software does not process melt or amplification curves but uses a standard deviation cutoff for outlier identification. It does not contain any functions for conducting statistical tests but can calculate primer efficiencies and use them in the ΔΔCq calculations. qpcr was last updated in Feb 2022 (Kleinschmidt, 2022).

R Packages

Chainy An R Shiny web-based app published in May 2017 for the relative quantification of qPCR data. It accepts inputs in multiple forms including RDML files and several proprietary qPCR analysis software output files. It outputs a zip file of summary statistics and normalised Cq values in comma separated files and as bar charts in PNG format. The software can calculate Cq values and efficiencies directly from amplification curves using the qpcR package or accepts pre-determined values. It can determine stable normalising genes using the NormqPCR package and flags outlying samples that do not fit the sigmoidal amplification curve. The app can also determine significant fold changes between samples using a permutation test. Chainy was last updated in Aug 2020 (Mallona et al., 2017).

shinyCurves An R Shiny web-based app published in Oct 2021 for detecting viral infections from diagnostic qPCR assays. The app accepts excel spreadsheet and RDML file inputs from BioRad's proprietary analysis software CFX Maestro Software, Roche's LightCycler® Software, Agilent's Aria software and Applied Biosystems® qPCR analysis software. It outputs the results as comma separated files. The plate designs are either 96 or 384 wells and users can flag control wells if they follow specific formats. It extracts Cq values from excel spreadsheets and determines if samples are Positive, Negative or Undetermined for viral load depending on user defined thresholds. It can quantify abundance using a standard curve if the input files contain serial dilutions. Users can conduct QC by viewing melt and amplification curve plots created by the R package qpcR and define a standard deviation cutoff for outlier identification. It does not contain any functions for conducting statistical analyses. shinyCurves has not been updated since release (Olaechea-Lázaro et al., 2021).

LEMming An R script published in Sept 2015 for the relative quantification of qPCR data. It proposes a linear error model for qPCR experiments which it uses to normalise Cq values without the use of normalising genes. This novel normalising method can confound the treatment effects with some systematic errors. Therefore, if normalising genes have been verified, the standard ΔCq method is recommended. It does not provide

any methods to read in qPCR data and creates an R S4 class object as output. It does not conduct any standard QC checks such as plotting amplification curves or checking for outliers. It does include how to conduct several different differential expression tests. LEMming has not been updated since release (Feuer et al., 2015).

pcr An R package published in May 2018 for the relative quantification of transcript abundance. It does not provide any methods to read in qPCR data but expected the input to be structured with each row a different sample and each column a target gene. The package creates an R data.table of summary statistics and ggplot2 figures. If 100% primer efficiency is assumed, it can calculate ΔCq. Otherwise, it requires serial dilutions to create standard curves and deduce relative abundance. The package also includes functions to conduct t-tests, Wilcoxon signed-rank tests and ANOVA. However, it always normalises to one normalising gene. It does not conduct any standard QC checks such as plotting amplification curves or checking for outliers. However, if the assay includes serial dilutions then amplification efficiency can be checked before analysis. pcr was last updated in April 2020 (Ahmed & Kim, 2018).

HTqPCR An R Bioconductor package published in Dec 2009 for the relative quantification of qPCR data. It contains several functions to read in several proprietary qPCR analysis software files. It outputs normalised Cq values and summary statistics as an S4 class object as well as several plots. The software does not plot melt or amplification curves as QC, but does allow users to define a standard deviation cutoff for outlier identification. There are also functions to determine batch effects, spatial effects and hierarchical interactions across samples and experiments. It can normalise genes using the standard ΔCq method or, in the case of unreliable normalising genes, it can normalise by quantile means and rank-invariant normalising factors. The package also contains functions to test differential expression with linear models, Mann-Whitney test or t-tests. HTqPCR core functionality has not been changed in 10 years, but it is maintained by the R Bioconductor community (Dvinge & Bertone, 2009).

ReadqPCR and NormqPCR A pair of R Bioconductor packages published in July 2012 for the relative quantification of qPCR data. ReadqPCR contains functions for reading in raw Cq value files from several proprietary qPCR analysis software files. They output normalised Cq values and summary statistics as S4 class objects as well as several plots. The software does not plot melt or amplification curves, but uses a user defined standard deviation cutoff for outlier identification. NormqPCR can select reliable normalising genes and impute missing values. It does not contain any methods for detecting statistically significant differential expression. ReadqPCR and NormqPCR were last updated in July 2018 (Perkins et al., 2012).

qpcR An R package released in 2008 for selecting the best sigmoidal model to fit to the amplification curve of each qPCR target for the accurate determination of Cq values and PCR efficiency. It does not provide any methods to read in qPCR data, but outputs an S3 object with summary statistics. The package contains several methods to determine the model with the best fit which is then used to determine threshold Cq values and efficiency. It can detect sample outliers, calculate relative and absolute abundances, and plot summary data. qpcR was last updated in June 2018 (Ritz & Spiess, 2008).

Misc

Spreadsheet A guide published in Dec 2021 for standardising the use of spreadsheet software to determine relative abundance. It does not describe how to calculate primer efficiencies but does use them in the ΔCq calculations. It outlines the use of t-tests to determine statistically significant differences. The guide does not process melt or amplification curves but suggests using a standard deviation cutoff for outlier identification. It does not suggest how to plot any summary statistics (Ng et al., 2021).

4.2.1.2. Summary

Novel qPCR analysis packages continue to be released despite packages being available across platforms for decades and the fundamental principles of qPCR remaining unchanged. Similar to the conclusions of the Pabinger *et al* review, the packages have a wide range of functionality, documentation and compliance with the MIQE guidelines. Two reoccurring issues with the reviewed software are the lack of quality control checks and the inconsistent approaches to removing data points that are considered outliers. Over the last 8 years, there has been an increase in GUI-based apps, but most sacrifice scalability and reproducibility to maximise ease of use. The R package qpcR remains the most comprehensive analysis package for qPCR data. However, the depth of its functionality is limited by its documentation. The package includes a description for every function and its arguments, but it does not provide enough information for deciding between alternative options or to justify the arguments it sets as default. The package also does not provide an example workflow to show how its functions can be combined to complete an analysis. In addition, the software architecture does not follow the tidy paradigm.

4.2.2. tidyqpcr: Quantitative PCR Analysis in the tidyverse

tidyqpcr addresses the need for a qPCR analysis package that balances functionality, documentation, and quality control to facilitate reproducible and best-practice-compliant analysis. It is intended to be flexible enough to analyse qPCR data from any nucleic acid source - DNA for qPCR or ChIP-qPCR, RNA for RT-qPCR - on any scale - 96, 384,

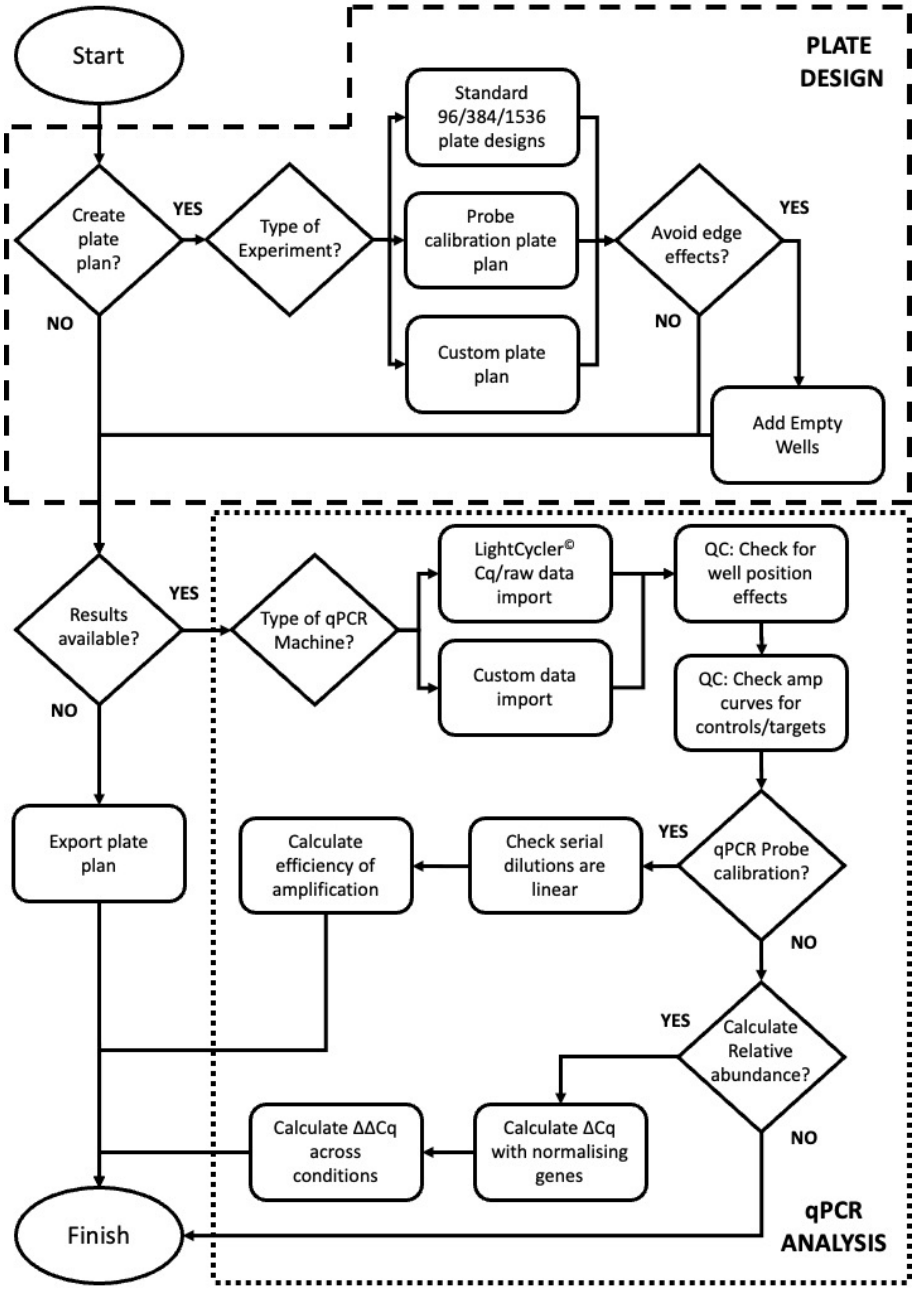


Figure 4.1: Block diagram showing the functionality of tidyqpcr. tidyqpcr has been designed to contribute to the design and analysis of qPCR experiments. tidyqpcr contains functions to aid in the design of balanced plates for manual or automatic loading across several common qPCR experiments and plate sizes. tidyqpcr facilitates reproducible analysis by separating each analysis step into a distinct function to create a human-readable analysis pipeline.

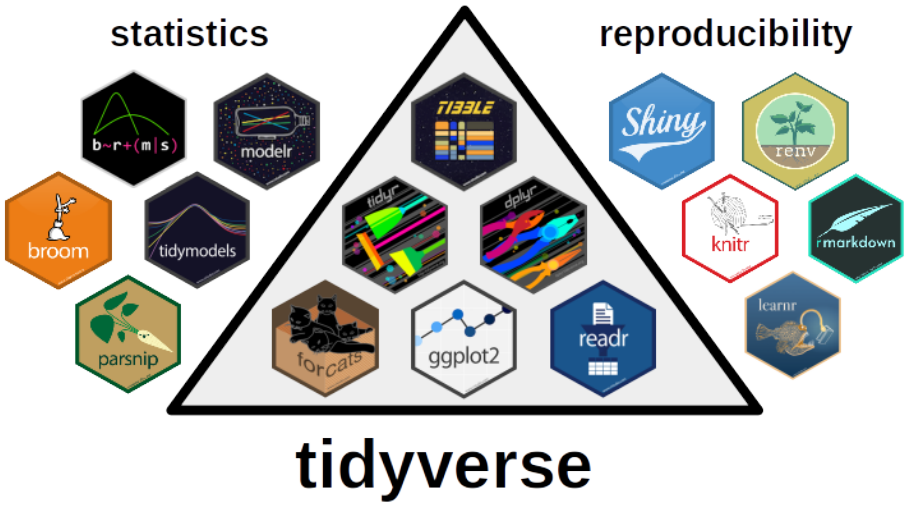


Figure 4.2: Developing tidyqpcr using the tidyverse packages grants access to a larger ecosystem of data analysis packages. The triangle contains the core tidyverse packages used to develop tidyqpcr. The wider ecosystem of packages all follow the same split-apply-combine paradigm and can be applied to any data analysed by tidyqpcr enabling statistical analysis or enhanced reproducibility.

1536+ well plates. Currently, tidyqpcr has functions that support relative quantification by the ΔCq method and calculating primer efficiencies, but not yet absolute quantification, Figure 4.1. A key component of tidyqpcr is its comprehensive documentation that teaches users how to use tidyqpcr and explains tidyqpcr's design decisions. These openly accessible teaching materials help to improve an entire qPCR experiment, from plate design to publication ready figures. The package follows the FAIR principles - Findable, Accessible, Interoperable, and Reusable - to ensure every stage of the analysis is transparent and verifiable. tidyqpcr is available to use now and can be downloaded from our GitHub page, github.com/ropensci/tidyqpcr/.

4.2.2.1. tidyqpcr design principles

Flexible and scalable analysis Within the R programming language, the tidyverse suite of packages has pioneered the use of tidy analysis. Mimicking the tidy structure in the creation of tidyqpcr not only opens the way to flexible analysis enabled by simply following the tidy data paradigm but it also directly allows access to a plethora of open-access and scalable data analysis tools already created in the tidyverse, Figure 4.2. Once users familiarise themselves with the tidy paradigm, they can conduct advanced downstream analysis such as linear regression analysis, complex visualisation and statistical summaries.

Experimental design In tidyqpcr, we help experimentalists decide how to set up their experiment by providing several plate plan helper functions built around block

designs. This enables samples to be spread across the plate and minimises well position biases but still contains regular patterns for loading with multi-channel pipettes, Figure 4.3. We also describe in detail different plate design strategies that users can explore depending on their pipettes and plates. Users can exclude loading samples into edge wells with the provided helper functions. We are also exploring introducing the automatic generation of loading recipes for common liquid handlers so users with access to the appropriate equipment can ensure the loader and plate plan match identically. Grouping biological/technical replicates so they are placed in the neighbouring wells can lead to systematic biases in the results. In an ideal situation, different samples and their replicates should be allocated entirely random well positions. However, if the sample loading is manual, then having inconsistent plans across plates will complicate the loading process and increase the likelihood of a mistake. Ultimately, having an incorrect map of samples in wells is significantly more detrimental to any analysis than systematic bias. tidyqpcr provides a balance between easy loading and good experimental design principles.

MIQE-compliant results tidyqpcr follows the MIQE-guidelines for analysis by allowing multiple normalising genes by default in the delta-Cq calculation. Helper functions are also provided for the design of serial dilutions plates for primer calibration and are available together with functions to calculate linearity, R^2 and plots to display behaviour across multiple primers. Importing the fluorescence across all cycles is also available so that quality control graphs for the melt and amplification curves across all wells can be seen, Figure 4.4. There are also default functions to plot Cq values across the plate to see biases, such as edge well bias. The vignettes also outline a reproducible analysis pipeline to standardise the analysis so reviewers can check it. All vignettes also describe the use of technical replicates, biological replicates and wells that contain RNA samples without reverse transcriptase.

4.2.2.2. Functionality

Overview tidyqpcr provides the functionality to aid with the implementation of qPCR assays from design to analysis. The design of complex plate plans is facilitated with the use of general plate formatting and labelling functions such as `label_plate_rowcol` and `create_blank_plate`. Meanwhile, helper functions that create commonly used 96 and 384 well plates are provided: `create_colkey_4diln_2ctrl_in_24` and `create_rowkey_4_in_16`. Once the plate has been designed, users can import the completed qPCR assay data from proprietary qPCR software. There are default functions for imported Roche LightCycler® data `read_lightcycler_1colour_raw` but users can create functions to import data from other machines as long as the end

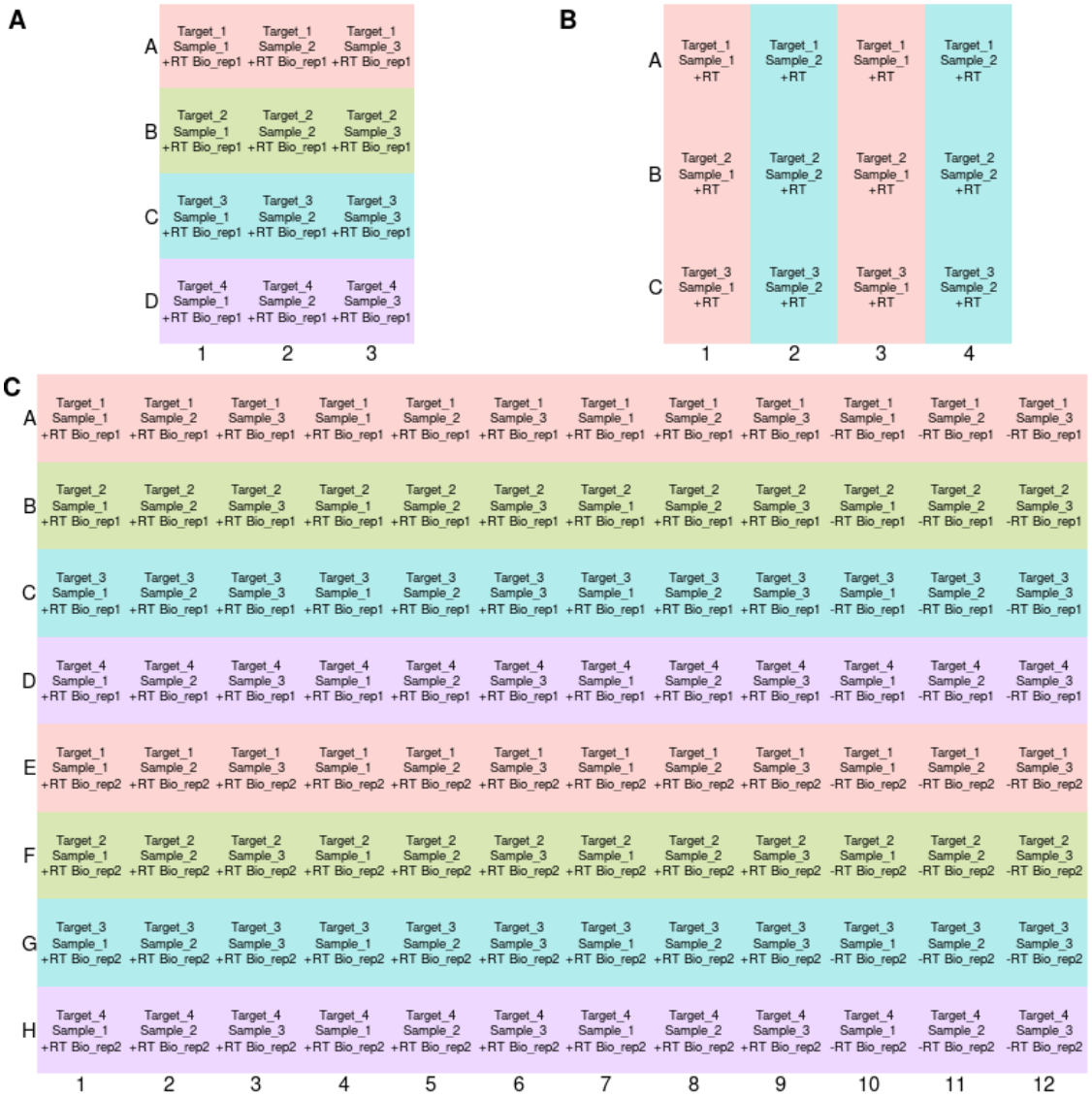


Figure 4.3: tidyqpcr facilitates flexible, modular plate design. **(A)** Single repeatable block containing important well information: Sample_id, target_id, RT and bio_rep number. **(B)** Alternative plate design with alternate sample replicates. Useful for pipetting with multichannel pipettes. **(C)** Full 96 well plate design based on the repeatable block of panel **A**. Shows a MIQE-compliant plate with 3 technical replicates, 2 biological replicates and -RT controls.

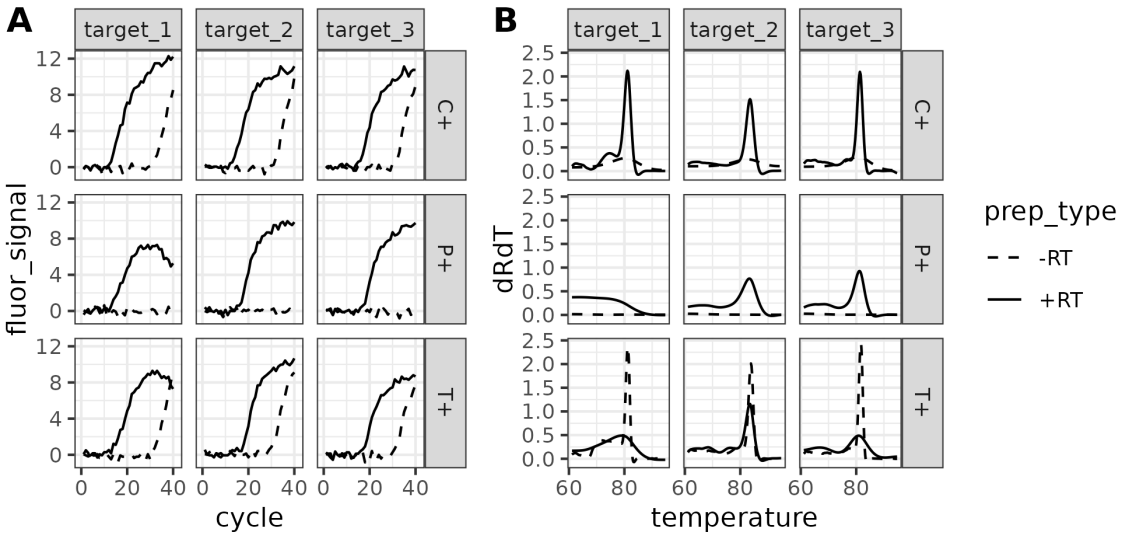


Figure 4.4: Extensive vignettes teach users MIQE-compliant analysis. **(A)** Amplification curve plots verifying that curves of the control samples without reverse transcriptase (dotted lines) do not overlap with the samples of interest. **(B)** Melt curves showing double-stranded DNA fragments become single-stranded above the threshold temperature for samples of interest.

data frame is in a tidy format. The import functions can be used to import threshold Cq values calculated from the qPCR machine or the Cq values across cycles for the entire time course. There are functions to conduct quality control on the imported data, as required by the MIQE guidelines. `calculate_drdt_plate` calculates the derivative of the melt curve enabling the user to confirm amplification occurs only at one temperature. Plate effects such as the reduction of efficiency at edge wells can be inspected using `display_plate_value`. Before the experiment of interest can be conducted, the amplification efficiency of the primers must be tested in order to ensure the assumptions of the qPCR threshold Cq comparisons are valid. Users can calculate primer efficiency with `calculate_efficiency_bytargetid` which enables calculations across targets given appropriate dilution assay data. Finally, on the completion of the quality control steps, user can calculate ΔCq and ΔΔCq values from any combination of samples and targets using `calculate_deltacq_bysampleid` and `calculate_deltadeltacq_bytargetid`. The mechanics around calculating ΔCq across replicates and plates is enabled by the `group_by` function in the tidyverse applied on the key variables `sample_id` and `target_id`. We chose the name `target_id` to hold information about the target of the qPCR primers and `sample_id` to hold the condition/strain/biorep information from each sample. Both words are necessary in the plate data frame in order to use tidyqpcr functions. The decision to name them `sample_id` and `target_id` was a balance between being specific enough to avoid ambiguity, but general enough to enable a variety of qPCR assays to be incorporated.

Use Case

An example use case is now described to show the power of tidyqpcr to analyse a complex 96-well qPCR assay. The data set is an assay inspecting the change in expression of 16 genes associated with the yeast stress response, as provided by Dr Edward Wallace. Yeast samples are exposed to heat shock in the presence of transcriptional inhibitors. The two transcriptional inhibitors are Phenanthroline and Thiolutin. Therefore, there are six conditions: no inhibitor present with and without heat shock, Phenanthroline with and without heat shock, and Thiolutin with and without heat shock. There are three technical replicates, two experimental replicates and each sample has a control which has not had any reverse transcriptase added. The example code will design the plate for this experiment, read in the results of the Roche LightCycler® qPCR machine and calculate normalised Cq values for all target-condition combinations.

The first stage of conducting a qPCR experiment with tidyqpcr involves designing a plate with the `label_plate_rowcol` function, Listing 4.1. It requires three data frame arguments: a blank plate data frame holding the shape and number of wells to be used, a rowkey data frame holding row-wise experimental metadata and a colkey data frame holding column-wise experimental metadata. The blank plate data frame can be any custom shape or size and follow any labelling system as long as each column and row is uniquely identifiable. tidyqpcr does provide boilerplate 96, 384 and 1024 well plates. The tidyqpcr example vignettes encourage users to hold `target_id` data in the rowkey data frame and `sample_id` data in the colkey data frame. This leads to entire rows containing the same primer and technical replicates and controls being grouped together. Creating the rowkey in the example is straightforward as there is the same number of rows as `target_ids` so the mapping is one-to-one. In other cases, if the number of `target_ids` is a factor of the number of rows then the `target_ids` are replicated until all rows are filled. This pattern is an easy way of introducing biological replicates onto a plate. Designing the colkey can be more complicated as different combinations of conditions, replicates and controls need to be included. Similar to the rowkey, if the number of unique samples is a factor of the number of columns then they can be repeated in blocks to represent the technical replicates and -rt control. Although it is not strictly necessary to use `label_plate_rowcol` to create a plate, we designed the function to encourage the users to design the plate in a logical row-wise and column-wise manner. This leads to an intuitive and reproducible method to load the plate which minimises mistakes and increases efficiency.

After the qPCR experiment has been conducted, the next step is to read in the results, Listing 4.2. The function `read_lightcycler_1colour_cq` is the default function in tidyqpcr for reading in the calculated threshold Cq values held in the ex-

```
# list target_ids of primer sets
target_id_levels <- c("HOR7",
    "HSP12", "HSP26", "HSP78",
    "HSP104", "RTC3", "SSA4",
    "PGK1", "ALG9", "HHT2",
    "HTB2", "RPS3", "RPS13",
    "RPS15", "RPS30A", "RPL39")

# Set up experimental samples
heat_levels <- c("-", "+")
heat_values <- factor(
    rep(heat_levels, each = 3),
    levels = heat_levels)
drug_levels <- c("C", "P", "T")
drug_values <- factor(
    rep(drug_levels, times = 2),
    levels = drug_levels)
condition_levels <- paste0(
    drug_levels,
    rep(heat_levels, each = 3))
condition_values <- factor(
    condition_levels,
levels = condition_levels)

# create plate plan
rowkey <- tibble(
    well_row = LETTERS[1:16],
    target_id = factor(
     target_id_levels,
     levels = target_id_levels))

colkey <- create_colkey_6_in_24(
    heat = heat_values,
    drug = drug_values,
    condition = condition_values)

plateplan <- label_plate_rowcol(
    create_blank_plate(
       well_row = LETTERS[1:16],
       well_col = 1:24),
    rowkey, colkey)
```

Listing 4.1: Example tidyqpcr code for designing a 96-well qPCR plate for an assay with 16 target stress response genes across 6 conditions.

cel file format used by the Roche LightCycler® software. The complementary function `read_lightcycler_1colour_raw` enables the user to load the Cq values across the entire time course for plotting quality control figures. Users using qPCR machines other than a Roche LightCycler® currently need to create their own function for reading in Cq data. The plate plans defined above can then quickly match the Cq values with the sample metadata. It is vital that the row and column labelling used by the qPCR machine is repeated correctly in the plate design data frame. As can be seen in the example code, the scalability of tidyverse functions enables tidyqpcr to easily incorporate multiple experimental replicates without significant changes in the pipeline.

Finally, to complete this example analysis the function `calculate_deltacq_bysampleid` will normalise all the Cq values from the targets of interest to the normalising genes, Listing 4.3. Following the MIQE guidelines, this function can accept multiple `target_ids` as normalising genes and calculate a mean or median value to subtract from all targets of interest. Again, using the flexibility of the tidyverse the mean Cq across any combination of samples, replicates and experiments can be calculated. This is possible because tidyqpcr consistently follows the tidy paradigm across all function outputs. The comparison of expression across all conditions and targets is plotted using ggplot2, Figure 4.5.

Function definitions and Documentation

tidyqpcr functions are designed following the tidyverse guidelines for compati-

```
file_path_cq_plate1 <-
   system.file("extdata",
     "Edward_qPCR_TxnInhibitors_
        HS_2018-06-15_
        plate1_Cq.txt.gz",
     package = "tidyqpcr")

plate1 <- file_path_cq_plate1 %>%
read_lightcycler_1colour_cq() %>%
   left_join(plateplan,
      by = "well") %>%
   mutate(biol_rep = "1",
      plate = "1")

file_path_cq_plate2 <-
   system.file("extdata",
   "Edward_qPCR_TxnInhibitors_
      HS_2018-06-15_
      plate2_Cq.txt.gz",
   package = "tidyqpcr")

plate2 <- file_path_cq_plate2 %>%
read_lightcycler_1colour_cq() %>%
  left_join(plateplan,
     by = "well") %>%
  mutate(biol_rep = "2",
     plate = "2")

# combine data from both plates
    into a single data frame
plates <- bind_rows(plate1,
   plate2) %>%
  unite(sample_id, condition,
     biol_rep, sep = "",
     remove = FALSE)
```

Listing 4.2: Example tidyqpcr code for reading threshold Cq values from LightCycler® qPCR machines and combining them with the designed plate plan.

```
platesnorm <- plates %>%
  filter(prep_type == "+RT") %>%
  calculate_deltacq_bysampleid(
     ref_target_ids = "PGK1")

platesmed <- platesnorm %>%
  group_by(sample_id, condition,
    biol_rep, heat, drug,
    target_id) %>%
  summarize(
    delta_cq = median(delta_cq,
       na.rm = TRUE),
    rel_abund = median(rel_abund,
       na.rm = TRUE))


ggplot(data = platesmed) +
  geom_point(aes(x = target_id,
     y = rel_abund,
     shape = biol_rep,
     colour = drug),
    position = position_jitter(
       width = 0.2,
       height = 0)) +
  facet_wrap(~heat, ncol = 3) +
  scale_y_log10("mRNA relative
    detection",
     labels = scales::
    label_number()) +
  theme(axis.text.x =
     element_text(angle = 90,
        vjust = 0.5))
```

Listing 4.3: Example tidyqpcr code for calculating ΔCq across multiple plates and plotting summary results across target genes.

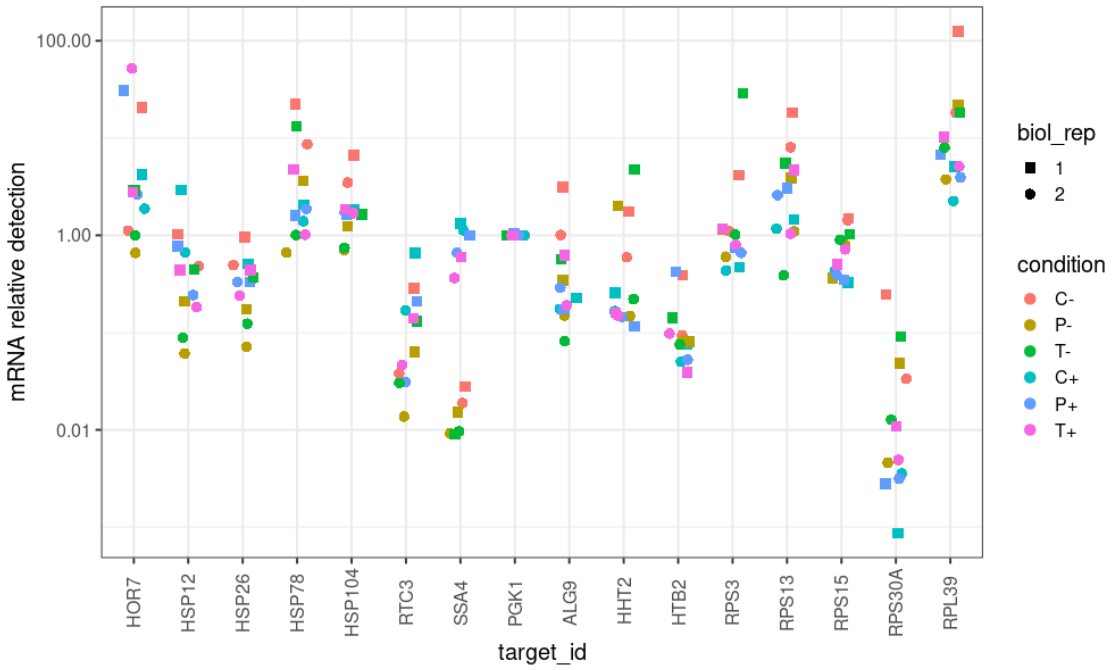


Figure 4.5: tidyqpcr can be used to quickly analyse multi-plate, multi-target, and multi-sample qPCR assays.

ble functions, Listing 4.4. The verb-object naming convention is followed throughout to help ensure each function is clearly named according to its purpose. The first argument of any tidyqpcr function is the primary data frame to be acted on. This allows the pipe operators commonly used in tidyverse code to continue their primary function. The input data frame is expected to be in the long tidy format and the outputs of any tidyqpcr function are also a data frame in a long tidy format. The definition of the `calculate_deltacq_bysampleid` function has been copied below as an example of a typical function code. The function groups Cq values by `sample_id` and subtracts the normalising `target_id` values from all Cq values in each group. Therefore, Cq, `sample_id`, and `target_id` are vital variables and are checked to be in the supplied data frame before the function attempts to calculate ΔCq. Once the presence of the required variables is asserted, the function calculates ΔCq and adds it as a new variable to the data frame. The function is entirely scalable as the internal `group_by` function can handle any number of `sample_id` and the ΔCq's can be calculated by any number of normalising `target_id`s.

Preceding the function definition is several commented lines documenting the details and use cases of the function, Listing 4.5. First, a brief description of the function, its input arguments and expected output are provided. Then, its dependencies on other functions both inside tidyqpcr and in other R packages are listed. Finally, short examples showing the use of the function are outlined. This preamble is converted into markdown

```
calculate_deltacq_bysampleid <-
 function(cq_df,
          ref_target_ids,
          norm_function = median)
{

 assertthat::assert_that(
  assertthat::has_name(
   cq_df,
   c("target_id",
     "sample_id",
     "cq")))

 cq_df %>%
  dplyr::group_by(
   .data$sample_id) %>%
   dplyr::do(
    calculate_normvalue(
     .data,
     ref_ids = ref_target_ids,
     value_name = "cq",
     id_name = "target_id",
     norm_function =
      norm_function)) %>%
dplyr::rename(
 ref_cq =
  .data$value_to_norm_by) %>%
    dplyr::ungroup() %>%
    dplyr::mutate(
     delta_cq =
      .data$cq - .data$ref_cq,
     rel_abund =
      2^ -.data$delta_cq)}
```

Listing 4.4: Function definition for the calculate ΔCq method within tidyqpcr. This example showcases the use of the `group_by` function provided by the core tidyverse package dplyr to split the Cq values by `sample_id` and apply the `calculate_normvalue` on each group separately.

formatted help documentation by the R package roxygen2 (Wickham et al., 2021). This documentation is accessible using the base help command once tidyqpcr has been downloaded and as a standalone documentation website hosted by rOpenSci.

```
#' Calculate delta cq to normalize
#, quantification cycle (log2-fold)
#' data within sample_id.
#'
#' This function implements
#' relative quantification by the
#' delta Cq method. For each
#' sample, the Cq values of all
#' targets (e.g. genes, probes,
#' primer sets) are compared to
#' one or more reference target
#' ids specified in
#' `ref_target_ids`.
#'
#' @param cq_df a data frame
#'  containing columns `sample_id`,
#'  value_name (default `cq`) and
#'  tid_name (default `target_id`).
#'  Crucially, sample_id should be
#'  the same for different technical
#'  replicates measuring identical
#'  reactions in different wells of
#'  the plate, but differ for
#'  different biological and
#'  experimental replicates. See
#'  tidyqpcr vignettes for examples.
#' @param ref_target_ids names of
#'  targets to normalize by, i.e.
#'  reference genes, hydrolysis
#'  probes, or primer sets. This can
#'  be one reference target id,
#'  a selection of multiple target
#'  ids, or even all measured
#'  target ids. In the case of all
#'  of them, the delta Cq value
#'  would be calculated relative to
#'   the median (or other
#'  `norm_function`) of all measured
#'  targets.
#' @param norm_function Function to
#'  use to calculate the value to
#'  normalize by on given scale.
#'  Default is median, alternatively
#'  could use mean.
#'
#' @return data frame like cq_df
#'  with three additional columns:
#'   ref_cq,  cq value for reference
#'            target ids;
#'   delta_cq,  normalized value;
#'   rel_abund, normalized ratio.
#' @export
#' @importFrom tidyr %>%
#' @importFrom stats median
#' @importFrom rlang .data
#' @examples
#' # create simple cq dataset
#' # with two samples, two
#' # targets  and 3 reps
#'
#' cq_tibble <- tibble(
#'  sample_id = rep(
#'   c("S_1", "S_1", "S_1",
#'     "S_2", "S_2", "S_2"),
#'     2),
#'  target_id = rep(
#'   c("T_1",
#'     "T_norm"),
#'   each = 6),
#'  tech_rep = rep(1:3, 4),
#'  well_row = rep(
#'   c("A", "B"),
#'   each = 6),
#'  well_col = rep(1:6, 2),
#'  well = paste0(well_row,
#'                well_col),
#'  cq = c(10, 10, 10, 12,
#'         12, 11,  9,  9,
#'          9,  9,  9,  9))
#'
#' # calculate deltacq using
#' # reference target_id
#' # called 'T_norm'
#'
#' # use case 1:
#' # median reference
#' # target_id value
#'
#' cq_tibble %>%
#'  calculate_deltacq
#'       _bysampleid(
#'    ref_target_ids = "T_norm")
#'
#' # use case 2:
#' # mean reference target_id
#' # value
#'
#' cq_tibble %>%
#'  calculate_deltacq
#'           _bysampleid(
#'   ref_target_ids = "T_norm",
#'   norm_function = mean)
```

Listing 4.5: Function documentation for the calculate ΔCq method within tidyqpcr as structured by the roxygen2 R package.

```
test_that("Unit test for the calculate_deltacq function",
{
   simulated_48_well_plate_plan <-
    create_blank_plate_96well() %>%
      dplyr::filter(well_row %in% c("A", "B",
                                    "C", "D")) %>%
      dplyr::mutate(
         target_id = rep(c("Target_1", "Target_2",
                           "Target_3", "Target_4"),
                         each = 12),
         sample_id = rep(rep(c("Sample_1", "Sample_2",
                               "Sample_3"),
                             each = 4),
                         times = 4),
         tech_rep = rep(c(1, 2, 3, 1),
                        times = 12),
         prep_type = rep(c("+RT", "+RT",
                           "+RT", "-RT"),
                         times = 12))

   calculated_48_well_plate_with_deltacq <-
      calculate_deltacq_bysampleid(
         simulated_48_well_plate_with_cq %>%
            dplyr::filter(prep_type == "+RT"),
         ref_target_ids = "Target_3") %>%
            dplyr::arrange(well_row, well_col)

    expect_equal(calculated_48_well_plate_with_deltacq,
                 simulated_48_well_plate_with_deltacq)})
```

Listing 4.6: Function test for the calculate ΔCq method within tidyqpcr as structure by the testthat R package.

Tests

tidyqpcr follows software development best practices by incorporating unit tests for all vital functions within the package, Listing 4.6. 95% of all functions within tidyqpcr are covered by a test. The development of tidyqpcr uses the continuous integration available in GitHub as a GitHub Action runs each unit test to check for bugs with every commit to the repository. The tests consist of small use cases where the simplest expected outcome from each function is compared to the actual output. Functions with multiple possible behaviours according to optional arguments have multiple tests to ensure functions perform as expected.

4.2.3. Reviewing and improving tidyqpcr

4.2.3.1. Themes derived from semi-structured interviews

We conducted a series of semi-structured interviews to explore current practices in qPCR experimental design and analysis. These interviews were conducted over Zoom with the video recorded by Zoom's proprietary software over a period between 45 and 90 minutes. The interview explored whether users were aware of the MIQE guidelines and if they cur-

rently executed any QC measures. We also wanted to know about the typical experiments users conducted; i.e. qPCR machine, reagents, and plate plans. Finally, we wanted to explore what software users currently used to analyse their results and if they were interested in learning R based analysis. The questions guiding the interview are available in Appendix A. The six interviewees covered several academic roles from senior postdoctoral research assistants to undergraduates. The interviewees had varied experiences in programming-based analysis and in conducting qPCR assays. The transcripts from the interviews are available at doi:10.5281/zenodo.7101606. Once the interview was completed the interviewers were introduced to tidyqpcr and asked to complete a user test.

Several key themes relating to the design and analysis of qPCR experiments appeared across the interviews, Figure 4.6. In terms of the design of qPCR assays, all interviewees reported doing RNA not DNA quantification. The inclusion of three technical replicates was typical although some users had a process of removing outliers. It also was not common practice to check amplification curves or confirm linear efficiency. Overall, few were aware of the MIQE guidelines and few recall published data giving QC results, analysis code or detailed protocols. In addition, no one reported trying to recreate any other published data set and a regular theme of not trusting conclusions based on qPCR results alone was common. In terms of the analysis of qPCR data Excel remains a common piece of software for the analysis and design of plates. Users almost universally depend on proprietary qPCR analysis software to determine Cq values. Few were aware of the concept of 'tidy' data outside of users already using R packages based on the tidyverse. Although most users are confident they could re-analysis their own results no one reported that their analysis was openly available for reviewers to access.

4.2.3.2. rOpenSci review and JOSS submission

In order to ensure tidyqpcr followed best software development practices and to verify the reliability of its functions, we submitted the package for an rOpenSci code review. rOpenSci offers transparent, constructive and open reviews of R packages that lower barriers to working with local and remote scientific data sources. A successful rOpenSci review can then be submitted to the Journal of Open Source Software (JOSS), enabling the software development work to be officially acknowledged with a citation. The submission to JOSS was published in June 2022, doi:10.21105/joss.04507.

The rOpenSci review highlighted several issues with tidyqpcr's implementation, including: insufficient compliance with FAIR software practices, failing CRAN software repository checks, and the inclusion of redundant functionality. rOpenSci requires all submitted software to be Findable, Accessible, Interoperable and Reusable (FAIR) and

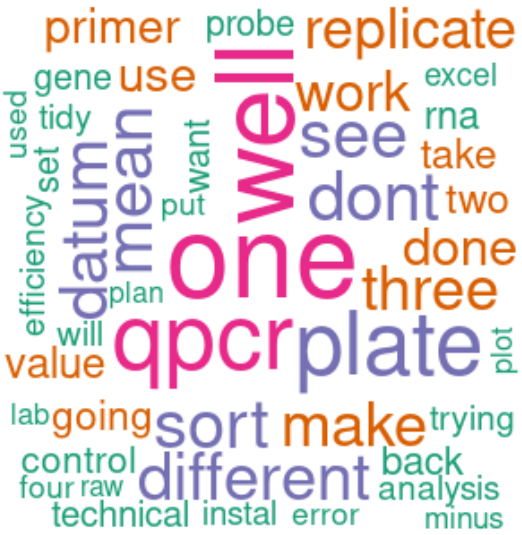


Figure 4.6: A text cloud showing the keywords repeatedly used across the semi-structured interviews. The greater the frequency of a word the larger it appears in the figure.

follow the practices of the CodeMeta Project (https://codemeta.github.io/). rOpenSci required us to add an codemeta.json file which standardises how the metadata associated with the software is held, such as maintainer details or software dependencies. Standardising the format of the metadata ensures a) all the data required to reuse a piece of software is available, and b) search functions can be developed to find software based on different metadata tags. Acceptance into rOpenSci suite of peer-reviewed packages also includes submission to the R package repository CRAN. However, the example data sets and extensive vignettes included in the tidyqpcr package led to it being over 5Mb, which is too large to be hosted on CRAN. We managed to reduce the package size by converting the example data set files to compress zip files. It was also highlighted that some functions originally packaged within tidyqpcr to aid with producing interpretable graph labels appeared out of place. We removed these plot helper functions as the scales package also provided the functionality (Wickham & Seidel, 2022).

4.2.3.3. User Feedback

Following the Agile software development practice, feedback from users was used to influence development priorities at regular intervals. Feedback was acquired from multiple sources: user testing following the user interviews outlined above, an independent

code review by a bioinformatics research assistant and the rOpenSci code review. The task-based user test section of the interview focused on three main themes of tidyqpcr: block-based plate planning, tidyverse-based API and conducting reproducible analysis. An independent code review following the Google code review guidelines was conducted by a colleague who had not previously contributed to the development of tidyqpcr. An overview of major improvements to tidyqpcr's functionality, usability, and documentation in response to user feedback is available in Table 4.7.

4.2.3.4. Future functionality

tidyqpcr is a fully functional package for the analysis of qPCR data using SYBR Green assays from Roche LightCycler® qPCR machines. However, there remain several planned improvements to enable tidyqpcr to easily analyse data from other qPCR assays and major additions to complete its aim to analyse qPCR data in an entirely open way according to the MIQE guidelines. First, to extend the import functions to read formats from other qPCR machines we intend to incorporate the plater R package (Hughes, 2016). This package follows the same tidy data principles as tidyqpcr, but is built to read data formats from a variety of plate-based experimental assays. Next, tidyqpcr has only been tested to work on SYBR Green qPCR assays. Fluorescent-quenched probe-based qPCR assays allow multiplexing so each well can measure multiple targets. Acquiring suitable fluorescent-quenched probe data and adding documentation on how tidyqpcr can be used to analyse such assays would also lead to additional functionality.

The previous improvements enhance the use cases of tidyqpcr, but tidyqpcr needs additional functionality to fulfil its aims to be MIQE-compliant and entirely open source. The major additions are: to include primer efficiency into ΔCq calculations, to provide methods to determine appropriate normalising genes, to import and export qPCR metadata in RDML form, and to calculate threshold Cq values itself. geNorm is an established method for incorporating primer efficiency into Cq values Meanwhile, the R package NormqPCR already contains the functionality to determine appropriate normalising genes to determine ΔCq values from a group of candidate genes. Integrating the geNorm method and NormqPCR functionality within the `calculate_normvalue` is a priority. As the default file type for MIQE-compliant qPCR analysis, creating the functionality to update RDML files using the RDML R package would increase the ease of conducting MIQE-compliant qPCR assays (Roediger et al., 2017). Finally, tidyqpcr needs to remove its reliance on the threshold Cq value calculations conducted by proprietary software. There are multiple methods to determine threshold Cq values with most being available within the qpcR R package described above. Rewriting the comprehensive set of functions in

Issue	Solution
Functionality	
tidyqpcr contains helper functions to create 96 and 384 well plates but 1536-plate wells are not supported.	Created a helper function to automatically create 1536-plate as well as a function to produce a "pick list" based on the plate to facilitate the use of robotic sample loaders.
Quality control should include a method for checking for positional effects on a qPCR plate.	Created the display_plate_value function to visualise threshold Cq values across the plate following the user-defined plate plan.
Usability	
Determining general but intuitive names for function arguments.	Depending on the assay used the measurement variable could be called Primer Set (for SYBR dye-style) or a fluorescent-quenched probe. Rather than committing to a specific assay, we decided on the more general term target_id.
The benefit of using tidyqpcr over the other available packages remains unclear.	The GitHub Repo README file now contains a summary table comparing functionality and MIQE compliance across tidyqpcr and its popular alternative packages.
Documentation	
Current package vignettes overwhelm new users as they introduce the basic concepts of tidyqpcr on multi-condition, multi-target data sets.	The interviewee provided a simpler 96-well plate data set for us to use as an example. We created a simpler vignette introducing the basic concepts of tidyqpcr using this data set for users to understand before moving on to the larger example.

Table 4.7: The tidyqpcr development cycle included regular opportunities for users to suggest improvements.

qpcR to follow a tidy format usable by tidyqpcr should complete the open source requirement from tidyqpcr's statement of intent.

4.3. Conclusion

qPCR remains one of the most widely used microbiology assays with uses across medicine, law and biology. However, awareness of what is required for someone else to repeat an experiment remains a key obstacle in ensuring reliable, reproducible results. Solutions have been widely published, such as the MIQE guidelines, but few publications appear to follow them. In addition, surveying the current qPCR analysis software landscape it is clear that there remains a gap in accessible, reproducible and scalable analysis packages. A combination of low levels of awareness and high levels of required effort impedes the publication of protocols and analyses in an open way. The consequences of the lack of reproducibility continue to have a detrimental effect on the quality and reliability of conclusions based on qPCR data. In summary, there remains a demand for methods that support and teach the best practices in qPCR assay design and analysis.

Here, we have described the R package tidyqpcr which aims to facilitate the analysis of qPCR data in an open, reproducible and reliable way. Created using best practices in open source software development, the tidyverse suite of data analysis packages and the MIQE guidelines we believe this tool can help experimentalists improve the quality of their analysis and the confidence in their results. We utilised multiple documentation formats to enable users to access the required level of detail for their expertise. Step-by-step workflows are provided in detail in the vignettes with specific reference to the MIQE guidelines and experimental plans that include controls, replicates, and block designs. More experienced programmers can take advantage of the complete function documentation to cherry-pick the steps they require and combine them with the plethora of tidyverse packages to create bespoke pipelines of their own. We believe that the extensive work put into ensure the package is open and accessible will manifest in to users contributing their own improvements and functions to the package. The initial response from users outside of our lab has been positive suggesting this package can improve the quality and clarity of results across the wider research community.

Chapter 5

Limitations of Composability of Cis-Regulatory Elements in Messenger RNA

5.1. Introduction

5.1.1. Discovering sequence determinants of gene expression

Since the beginning of molecular biology, biologists struggled to explain the size of the genomes of eukaryotes (Mirsky & Ris, 1951), It was quickly understood that only a fraction directly encoded protein sequences (Thomas, 1971). Around the same time, scientists were discovering that RNA and protein synthesis within cells can be regulated by hormones (Ui & Mueller, 1963). In addition, investigations into carbon regulation in bacteria discovered the promoter as a sequence adjacent to a coding sequence which regulated expression through the initiation of transcription (Jacob et al., 1964). All of this work led to the theory that a significant portion of the non-coding genome was dedicated to the regulation of expression. The theory included the classification of sequences that either affect the expression of its own gene; for example, promoters, or signal changes in the expression of other genes; for example, hormones (Britten & Davidson, 1969).

Today, these two types of regulatory sequences are called cis-regulatory elements (CRE) and trans-regulatory elements. Massively parallel approaches now characterise smaller CREs by exploring a large library of promoter elements with a single terminator (Sharon et al., 2012), or conversely a library of terminators with a single promoter (Shalem et al., 2015). Cis-regulatory elements include: sequences imbuing mRNA secondary structures, RNA binding protein motifs, and enhancer regions (Li et al., 2015). The possible effects of regulatory elements have also been expanded to include contributions to localisation, degradation, and translation. Meanwhile, trans-regulatory elements include: transcription factors (Spitz & Furlong, 2012), splicing regulators (Will

& Lührmann, 2011), and RNA-binding proteins (Bleichert & Baserga, 2010). However, the line between trans- and cis- regulatory elements is becoming increasingly blurred (Savarese & Grosschedl, 2006).

Computational methods continued to be developed for the discovery and quantification of cis-regulatory elements, such as sequence motifs recognised by regulatory proteins. Prominent methods for mapping CRE sequence-function relationships include predicting functional data with short sequence features, often using linear models (Shalgi et al., 2005; Cheng et al., 2017; Vijayabaskar et al., 2019) or, directly comparing the sequences of genes with similar characteristics to determine the presence of short consensus sequence motifs (Hogan et al., 2008; Kretz et al., 2013) using motif discovery software (Elemento et al., 2007; Bailey et al., 2015). Both of these approaches make the implicit assumption that the contribution of a short CRE is independent of context so that the effect of combining motifs is composed of a linear sum (on the appropriate scale) of the individual CRE contributions. The approximation that short CREs act independently helps to find elements that have clear contributions and to simplify a vast search space that would be made exponentially larger by accounting for CRE interactions. However, framing the search for CREs around independent contributions from short motifs overlooks multi-part motifs, interactions between motifs, and motifs that may be active only in specific contexts.

5.1.2. From regulatory elements to modular cloning

The discovery of regulatory elements with predictable contributions to gene expression opened the door to the idea of creating synthetic genes. Technology developed over the later half of the 20th century has turned the creation of synthetic genes into an industry. First, the discovery of restriction enzymes allowed the precise cutting of DNA using designed recognition sites. The original type I and type II restriction enzymes, which cut within the recognition site, (Meselson & Yuan, 1968; Arber & Linn, 1969; Kelly & Smith, 1970; Smith & Welcox, 1970) was followed by the type IIS restriction enzyme, which cut a few base pairs upstream of the recognition site (Pingoud et al., 2014). Type IIS enzymes keep the recognition site intact after cleavage, enabling complex multi-step processes. Next, the understanding of the recombination mechanisms enabled the pasting of multiple different segments of DNA together (Jackson et al., 1972; Cohen et al., 1973). Then, ten years after the discovery of promoters scientists transformed E.coli cells with a synthetic, replicating plasmid which expressed resistance to the antibiotic tetracycline (Cohen et al., 1973). The tools for separating and combining segments of DNA into replicating plasmids laid the groundwork for integratable cassettes of interchangeable genetic parts and the origins of modular cloning. Up to then, work was mostly completed in

prokaryotes, but work to efficiently transform yeast (Beggs, 1978) and later mammalian cells (Capecchi, 1989) enabled access to more complex synthesis machinery and more complex proteins. The creation of polymerase chain reaction dramatically increased the efficiency of transforming organisms and accessing new genes as regions of genomes could be selected and amplified by designing primers (Saiki et al., 1988). The invention and improvement of these tools culminated in the creation of synthetic circuits and regulatory elements at the turn of century (Elowitz & Leibier, 2000; Gardner et al., 2000).

The last 20 years of pioneering synthetic biology research as been defined by the standardisation of libraries of regulatory parts. BioBricks is the de facto library of choice that pioneers the "pick and mix" paradigm (Knight, 2003). Shifting the focus from fundamental cell biology to metabolic engineering, users no longer need to understand specific genomes to find suitable regulatory parts, but instead focus on the end product. Bioengineering is quickly becoming a staple of industry with the production of high-value compounds with high yields and reduced waste (Krivoruchko & Nielsen, 2015). Genetically modified organisms have now been created to detect oil spills, arsenic in water and even the presence of specific viruses (Willardson et al., 1998; Tancharoen et al., 2019; Wan et al., 2019).

Intrinsic to the success of synthetic biology is the development of modular processes to assemble genetic parts (Curran et al., 2013; Andreou & Nakayama, 2018; Garcia-Ruiz et al., 2018; Lee et al., n.d.). Several groups have constructed standardised libraries comprising promoter (including 5'UTR) and terminator (including 3'UTR) parts, which can be combined to achieve desired expression of synthesized proteins. The characterisation of all possible promoter-terminator pairings is impractical, so standardised libraries generally characterise parts independently, for example testing many promoters with the same coding sequence and terminator (Lee et al., n.d.). Because these experiments do not measure interactions between different parts, they rely on the untested assumption that those parts have more or less identical effects across different contexts. The majority of interactions between parts are negligible, but unpredictable behaviour at crucial stages of genetic pathways continues to hamper synthetic biology projects (Kittleson et al., 2012).

5.1.3. Current limitations in predicting gene expression

The unpredictability of the expression of synthetic constructs highlights the current limitations in our understanding of gene expression. Currently, the most common solution to this problem is to take advantage of the scale of high throughput assays now available. Automation of construct transformation, growth and characterisation is enabling 1000s of variants to be compared at the same time point. Massively parallel report assays can

now test thousands of constructs with high throughput sequencing and flow cytometry and those that behave as expected can be selected (Kosuri et al., 2013; Inoue & Ahituv, 2015). However, this remains a costly and high expertise endeavour that highlights how much there remains to be understood about gene regulation.

The continued development of statistical models to accurately predict gene expression from DNA sequences offers another route to better predict behaviour without needing to conduct additional experiments (Li et al., 2015). However, many CRE finding algorithms tend to rely on models of independent effects, that is, composability. More complex models that can account for non-linear interactions typically hide their decision-making. Interpreting the decisions made from "black box" machine learning algorithms (Guidotti et al., 2018) could be significant in understanding the biological mechanisms through which gene expression is regulated. Understanding gene regulation pathways requires a more balanced approach that enables interactions but in an interpretable way.

Here, we explore the limitations of composability beyond regulatory regions with a focus on interactions between regulatory sequence motifs. We first show the changing regulatory behaviour of terminator sequences paired with different promoters and coding sequences. We measure the fluorescence of 120 chimera constructs and highlight the irregular but significant changes in gene expression between contexts. Then, we investigate whether the changes in regulatory behaviour are reflected in the cis-regulatory elements within these regions. We conduct an analysis of contributions to half-life from suspected cis-regulatory elements using published data sets and create a shortlist of motifs of interest. Then, we show that these motifs on their own have different contributions to gene expression depending on context. Finally, we confirm motif contributions to gene expression using RNA-Seq experiments and explore possible positional effects on motif behaviour.

5.2. Results

5.2.1. Terminator effects on gene expression depend on cis-regulatory context

To investigate the context dependence of terminator regions, we created a library of 120 constructs (Figure 5.1A), containing all combinations of 6 promoters (including 5'UTR), 2 coding sequences, and 10 terminators (including 3'UTR). We selected promoters and terminators from native yeast genes spanning a variety of different expression patterns and functions (Table 5.1). To choose specific sequence lengths for the terminators of our constructs, we referred to published measurements of median 3'UTR length (Pelechano et al., 2013) because sequences that are necessary and sufficient for efficient transcriptional termination are found upstream of the termination site (Guo & Sherman, 1996). For

Gene Name	Systematic Name	Median 3'UTR Length	Construct Terminator Length	Usage	Function
PGK1	YCR012W	158	189	S	Glycolysis
RPS3	YNL178W	86	200	M&S	Ribosomal
RPS13	YDR064W	92	200	S	Ribosomal
PAB1	YER165W	150	200	S	RNA Binding
HSP26	YBR072W	164	200	S	Heat Shock
CLN2	YPL256C	203	200	S	Cell Cyclin
SRO9	YCL037C	543	545	S	RNA Binding
TOS6	YNL300W	256	256	S	Cell Wall
SUN4	YNL066W	198	198	S	Cell Wall
PMA1	YGL008C	421	421	S	Trans-membrane ATPase
TSA1	YML028W	112	219	M	Redox Homes-tasis
PIR1	YKL164C	235	358	M	Cell Wall

Table 5.1: Summary of the terminator library. The common gene name from which the terminator is extracted is included alongside its systematic name. The median 3'UTR reported by (Pelechano et al., 2013) is a median over the lengths of each distinct isoform they detect. The usage of each motif is signified by S, for promoter-terminator swaps, or M, for motif insertion or deletion. A short summary of the protein function is included.

terminators with measured lengths under 200nt, we used the standardised parts length of 200nt from the YeastFab library. The 2 coding sequences (CDS) expressed mCherry or mTurquoise2 fluorescent proteins, which are bright fluorophores with only 30% amino acid identity (Shaner et al., 2004; Goedhart et al., 2012).

Measuring fluorescence with a plate reader showed that, as expected, promoter choice dominated overall protein output. We observed up to 100-fold changes in fluorescence between the 4 highest expressing promoters (Supplementary Figure B1A) and the 2 lowest expressing promoters (Supplementary Figure B1B; mCherry log2 fold change = 7.33, p.value = 0.000). Expression from the stress-induced pHSP26 was notably more variable than from pPGK1 and pRPS's, across biological replicates, when combined with both coding sequences and a variety of terminators. We also confirmed that most differences in protein outputs are accounted for by changes in mRNA abundance by checking a subset of mCherry constructs using RT-qPCR ($R = 0.888$, Figure 5.1D).

Terminators also affect protein output with 5-fold changes in fluorescence seen within the same promoter-CDS sets (pPGK1-mTurq-tPMA1 log2 fold change = -2.73, p.value =1.19×10^{-41}), relative to the tPGK1 terminator of each group (Figure 5.1B, Supplementary Tables B1 and B2). We focus on constructs with high-expression promoters due to the poor signal-to-noise ratio at low expression levels. The interaction of coding sequence and terminator is seen most clearly for tPAB1. tPAB1 is consistently the most highly expressed terminator in mTurquoise2 constructs but is more variable in

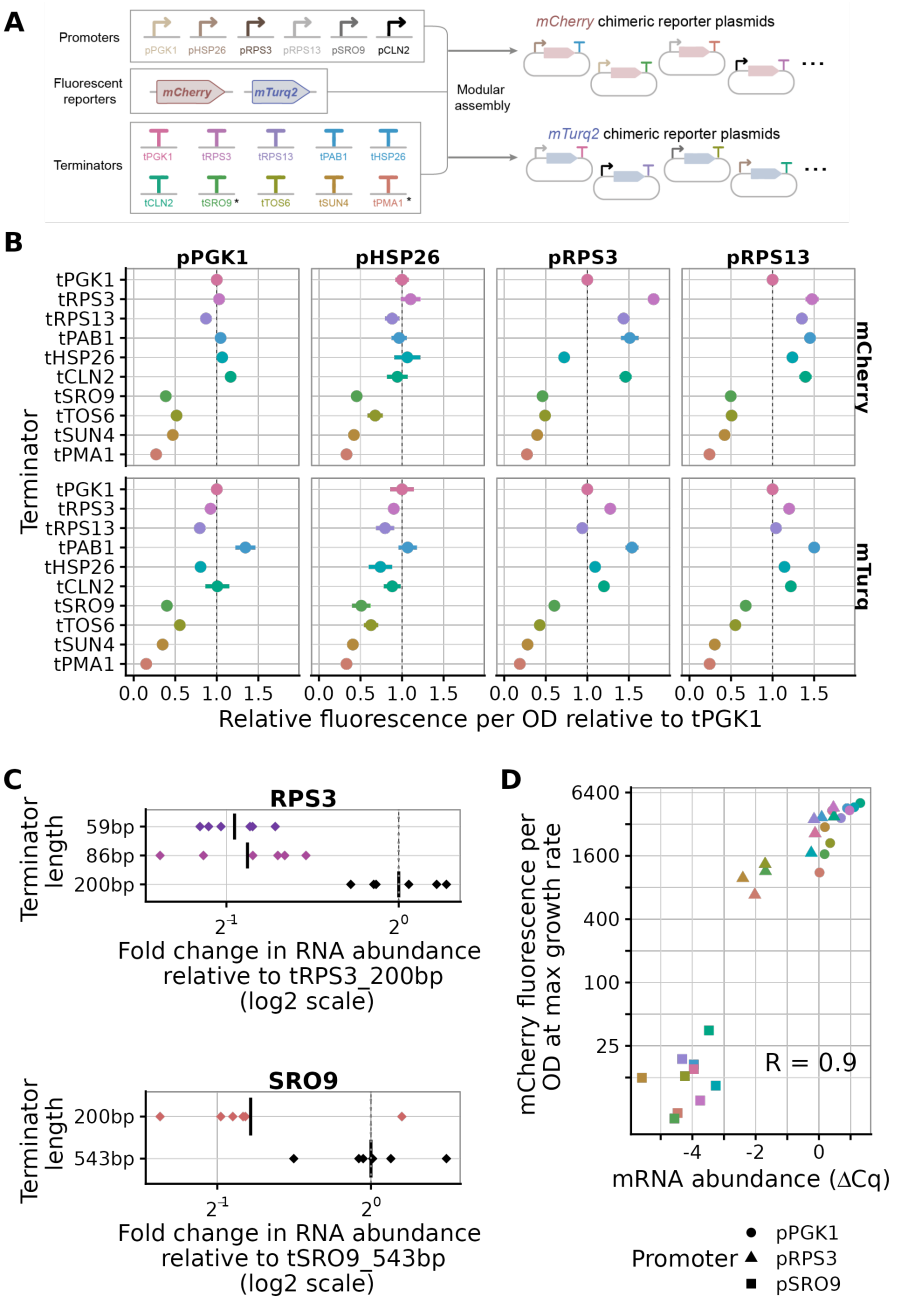


Figure 5.1: Terminator contributions to gene expression are promoter, coding sequence and length dependent. (**A**) Design of chimeric reporter constructs with all combinations of 6 promoters, 2 fluorescent proteins, and 10 terminators, on a centromeric plasmid. Terminators highlighted with an asterisk have a terminator length longer than the library standard of 200bp because they have a median 3'UTR length greater than 200bp according to (Pelechano et al., 2013). Panel created by Jamie Auxillos. (**B**) Relative protein abundance from each terminator, normalized to a reference terminator tPGK1 for each matched promoter. The plot shows statistical summaries (mean and standard error) of at least 6 replicates for high-expression promoter data shown in Figure 1A. (**C**) RT-qPCR mRNA results targeting the mCherry ORF for pRPS3-mCherry-tRPS3 and pSRO9-mCherry-tSRO9 constructs with different terminator lengths. (**D**) mCherry fluorescence correlates with RT-qPCR mRNA abundance for 3 promoters paired with all 10 terminators. Note that both axes use a log_2 scale.

mCherry constructs. Meanwhile, tPGK1 highlights the interactions of promoter and terminator. tPGK1 is one of the most highly expressed terminators when paired with pPGK1 and pHSP26, but is up to 40% lower in expression when paired with pRPS3 or pRPS13. Overall, our results show that the contributions of terminators (including 3'UTRs) to gene expression depend on other parts within the gene.

We further investigated the effects of terminators on mRNA levels by comparing constructs with extended vs truncated terminators. It is known that disrupting the transcription termination signal lowers expression (Guo & Sherman, 1996; Shalem et al., 2015). However, standardised parts libraries often assume a fixed terminator length for all genes, which is likely to omit the termination signal for genes with longer terminators. In the case of the YeastFab parts library (Guo et al., 2015) the fixed terminator length is 200bp. We compared a gene with a median 3'UTR length less than 200bp, RPS3 to a gene with a median 3'UTR length greater than 200bp, SRO9. The median length of the native RPS3 3'UTR is 86nt (Pelechano et al., 2013); truncating the terminator to 86bp or 59bp reduces transcript protein output by almost 2-fold (86bp Fold Change = 0.544, p-value = 2.6×10^{-4}; 59bp Fold Change = 0.517, p-value = 7.8×10^{-6}; Figure 5.1C). The median length of the native SRO9 3'UTR is 543nt (Pelechano et al., 2013); extending the terminator length to 543bp increases transcript protein output by almost 2-fold (Fold Change = 0.581, p-value = 0.01; Figure 5.1C). This validates the ability of our assay to detect known regulatory signals affecting transcription termination while highlighting the importance of using well-informed annotations to construct parts libraries for synthetic biology. Note that we used the longer 543bp SRO9 terminator, and a similarly extended 421bp PMA1 terminator, for the main set of constructs (Figure 5.1).

5.2.2. Candidate cis-regulatory elements contribute to transcript decay rates

Next, we investigated how the regulatory effects of CREs contained within terminator regions depend on their context. First, 69 suitable CREs to test for context dependence were found through a literature search. All were suspected sequence motifs for mRNA binding proteins, several directly associated with proteins involved in mRNA degradation (Shalgi et al., 2005; Hogan et al., 2008; Cheng et al., 2017). Any motifs that were found in fewer than 6 gene 3'UTRs, as annotated by (Pelechano et al., 2013), were removed.

We quantified the regulatory effects of the remaining 38 candidate motifs by applying a linear model predicting half-life to 2 recent transcriptome-wide analyses of mRNA decay that used metabolic labelling (Sun et al., 2013; Chan et al., 2018). These datasets are loosely correlated in their half-life measurements across 4188 genes reported in both datasets, $R = 0.63$ (Figure 5.2A). However, (Chan et al., 2018) estimated substantially smaller half-lives. (Chan et al., 2018) also had greater coverage of genes in the yeast

genome, 5529 vs 4304, and used multiple time points to determine half-lives. Following (Cheng et al., 2017), we constructed a linear model to predict the half-life of a transcript using the counts of motifs in its 3'UTR, the length of the 3'UTR, and the relative codon usage in the coding sequence of each transcript (see Material and Methods). The linear model performed similarly on both datasets by explaining 44% and 41% of the variability in half-lives for the (Chan et al., 2018) and (Sun et al., 2013) datasets respectively (Figure 5.2C). This predictive power is comparable to the squared correlation between the datasets ($R^2 = 0.40$). Motifs that did not significantly contribute to the model were automatically filtered out using a greedy algorithm maximising the Akaike information criterion (AIC) during both training stages. Approximately 1.7% of the variance is explained by 7 significant motifs, with 42.0% explained by codon usage (Supplementary Table B3), consistent with previous analyses (Presnyak et al., 2015; Cheng et al., 2017). The top 7 most significant motifs from the (Chan et al., 2018) data showed similar regulatory behaviour when tested on their own in the (Sun et al., 2013) data, except for TGTAAATA which was stabilising in one dataset and destabilising in another, as we later discuss (Figure 5.2B).

We selected 4 motifs for exploring context dependence: TGTAHMNTA, GTATACCTA, HWNCATTWY, and ATATTC (Table 5.2). TGTAHMNTA and GTATACCTA were chosen as they had the largest coefficients amongst significant decay and stability motifs, respectively. HWNCATTWY was chosen due to its statistically significant effect in both datasets and, as it co-occurs with TGTAHMNTA in 68 native 3'UTRs, because it could be used for testing motif interactions. The final selected motif was ATATTC, as it is a statistically significant decay motif in both datasets, and it has been previously shown to lower mRNA abundance when inserted in reporter constructs (Cheng et al., 2017). Functionally, TGTAHMNTA is the binding motif for Puf4p, and HWNCATTWY is associated with Khd1p/Hek2p-bound transcripts (Hogan et al., 2008). However, it is not known how ATATTC and GTATACCTA affect mRNA decay.

5.2.3. Quantification of differential expression due to motif insertion or mutagenesis in multiple 3'UTRs

To quantify the effects and composability of selected motifs in different contexts, we designed a further set of reporter constructs (Figure 5.3A). We first chose the ribosomal protein terminator tRPS3, as it was the only terminator in our initial library that did not contain any of the selected motifs. We selected thioredoxin peroxidase terminator tTSA1 as the second host terminator because it also lacks selected motifs and has a similar length to tRPS3. In each host terminator, we chose 3 motif insertion sites, selecting for: minimum impact on transcript secondary structure, avoiding known transcription termination elements, and matching the positions of motifs in native genes. Having 3

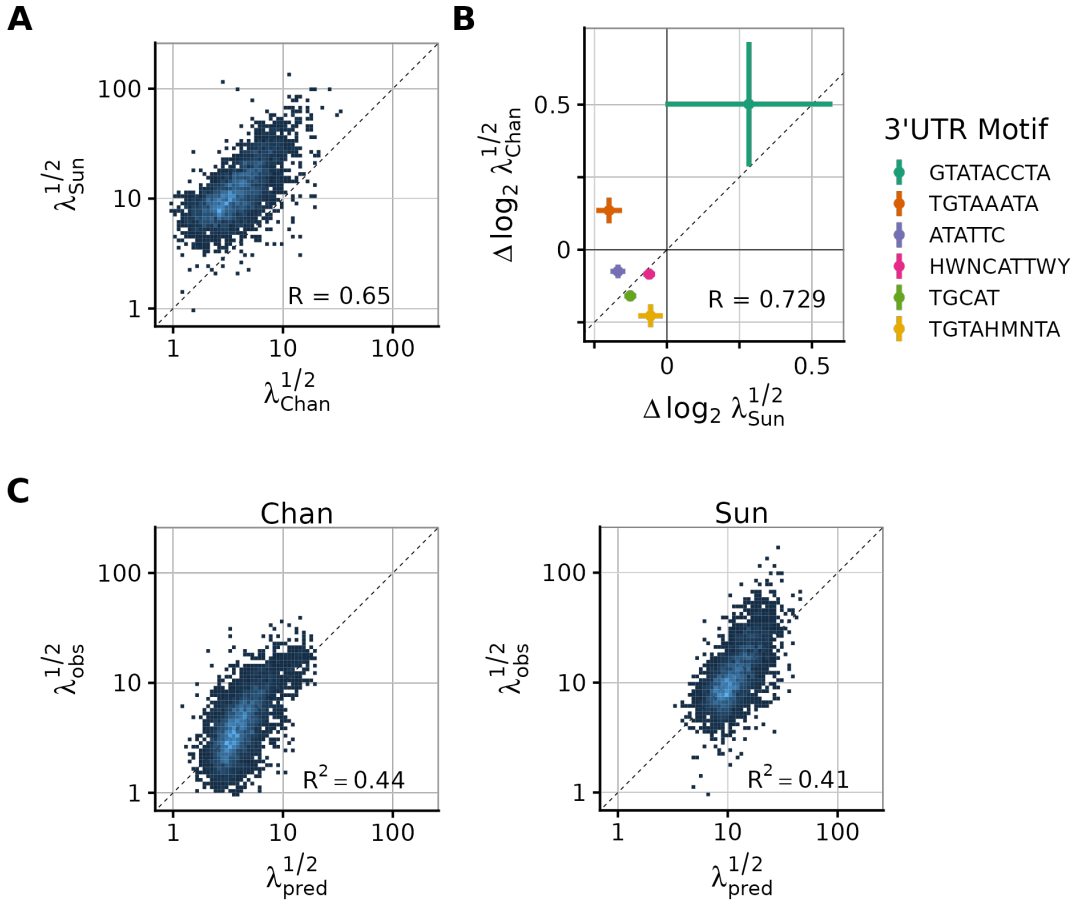


Figure 5.2: A linear model of transcript half-life quantifies the effect of candidate terminator motifs on half-life. (**A**) Correlation between the 2 transcript half-lives (λ), in minutes, reported in the (Chan et al., 2018) and (Sun et al., 2013) datasets. (**B**) Predicted contributions to log2 half-life for chosen motifs in the (Chan et al., 2018) and (Sun et al., 2013) datasets. The motifs were chosen using a linear model with the length of the 3'UTR and the relative codon usage of each gene as the base covariates and a greedy algorithm selecting the motif counts that increased the AIC. (**C**) Predicted vs actual transcript half-lives calculated by a linear model of codon and motif usage trained on the (Chan et al., 2018) and (Sun et al., 2013) datasets.

Consensus Seq	Inserted Motif	Deleted Motif	Source	Chan Coef	Chan p.value	Sun Coef	Sun p.value	Notes
GTATACCTA	GTATACCTA	GTATACCTA	Shalgi	0.500	1.9e-02	0.280	0.3300	Unknown
ATATTC	ATATTC	ATATTC	Cheng	-0.075	1.4e-03	-0.170	0.0000	Decay motif
HWNCATTWY	TTTCATTTC	CTTCATTTC ATACATTAT AATCAT-TAT	Hogan	-0.084	4.9e-06	-0.061	0.0026	Khd1/Hek2 associated motif
TGTAHMNTA	TGTACAATA	TGTACATTA	Hogan	-0.230	0.0e+00	-0.056	0.1800	Puf4p binding motif

Table 5.2: Summary of shortlisted motif characteristics. The first 3 columns hold the consensus sequence for each motif and the exact versions deleted from or inserted into the host terminators. Then, we report the paper from which the motif was selected from; (Hogan et al., 2008), (Cheng et al., 2017) or (Shalgi et al., 2005). Next, the coefficient given by the linear model predicting either the (Chan et al., 2018) or the (Sun et al., 2013) half-life datasets is included for each motif. Finally, the table includes notes on motif functions.

insertion sites enabled us to quantify combinations of motifs, including duplicates of weaker motifs to increase the likelihood of detecting a clear effect on gene expression. We chose TGTACAATA and TTTCATTTC sequences as explicit versions of the TGTAHMNTA and HWNCATTWY consensus motifs respectively, and checked that these explicit versions have similar predicted effects on half-life transcriptome-wide (Supplementary Tables B7, B8). Altogether, 7 variant terminators were designed for these 2 host terminators: the wildtype terminator, a control to test the insertion sites with randomly generated sequences, 4 testing the effects of inserting each motif individually and a final variant to test interactions between the TGTAHMNTA and HWNCATTWY motifs. We created a construct library by pairing each terminator with three different promoters; its native promoter pairing (pRPS3 or pTSA1), the high-expression promoter pPGK1, and the low-expression promoter pSRO9.

Motifs predicted to contribute to half-life when inserted into an RPS3 terminator are seen to have the predicted effect on mRNA abundance in the tRPS3 reporter constructs (Figure 5.3B). We measured mRNA abundance by RT-qPCR across 6 biological replicates, each quantified in 3 technical replicates and normalised by the ΔCq method against values from 3 reference mRNAs (see methods). Insertion of 2 copies of ATATTC (mod_NAA) generally lowers the mRNA abundance, as much as 4-fold when paired with the pRPS3 promoter. Insertion of either TGTAHMNTA (mod_NTN), or 2 copies of HWNCATTWY (mod_HNH), tends to decrease mRNA abundance, and their combined insertion (mod_HTH) tends to decrease mRNA abundance even further. The putative stability motif GTATACCTA (mod_NGG) does not consistently or strongly affect mRNA abundance. However, a comparison of the WT and control (mod_NNN) terminators does

show that the creation of the insertion sites alone affects mRNA levels (Supplementary Table B10).

Inserting the same motifs into our second host terminator gives qualitatively similar results (Figure 5.3C). Decay motifs generally lead to decay, although ATATTC (mod_NAA) has a weaker effect in tTSA1 than in tRPS3, and TGTAHMNTA (mod_NTN) has a stronger effect in tTSA1 than tRPS3. The putative stability motif GTATACCTA (mod_NGG) again has little effect.

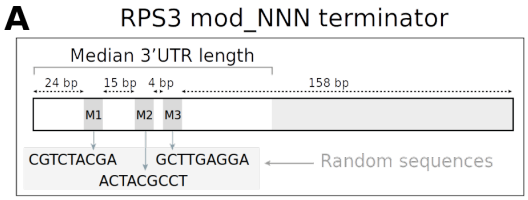
We next quantified the effects of removing decay motifs from a native yeast terminator. We selected the cell wall protein PIR1 as our host terminator as it is only 258 bp (Pelechano et al., 2013) and a *de-novo*-synthesizable terminator that contains the ATATTC, TGTAHMNTA, and HWNCATTWY motifs. We designed 8 terminators in which the motif occurrences in tPIR1 were replaced by scrambled sequences (Figure 5.4A). We found that the removal of almost any decay motif from tPIR1 results in an increase in mRNA levels (Figure 5.4B; Supplementary Table B11).

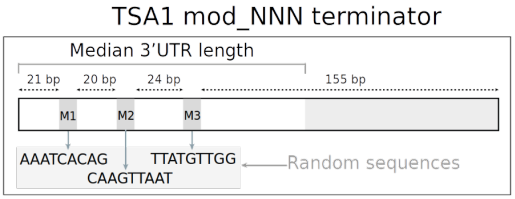
We confirmed that motif-dependent changes in mRNA abundance are reflected in protein abundance by measuring the fluorescence from a subset of reporter constructs with native promoter-terminator pairings (Supplementary Figure B2). The high correlation ($R = 0.96, 0.68, 0.86$ for tRPS3, tTSA1 and tPIR1 constructs respectively) demonstrates that these combinations of decay motifs that change mRNA abundance also change the protein output, as expected.

Comparison of mRNA abundance across all constructs (Figure 5.3B, 5.3C, 5.4B) shows motif contributions change in magnitude but not direction depending on the context of the rest of the construct. The insertion of almost any decay motif into tTSA1 or tRPS3 results in a decrease in mRNA abundance, and the removal of these from tPIR1 results in an increase in mRNA abundance. However, the quantitative effects vary depending both on the immediate motif context in the host terminator and the more distant context given by the promoter.

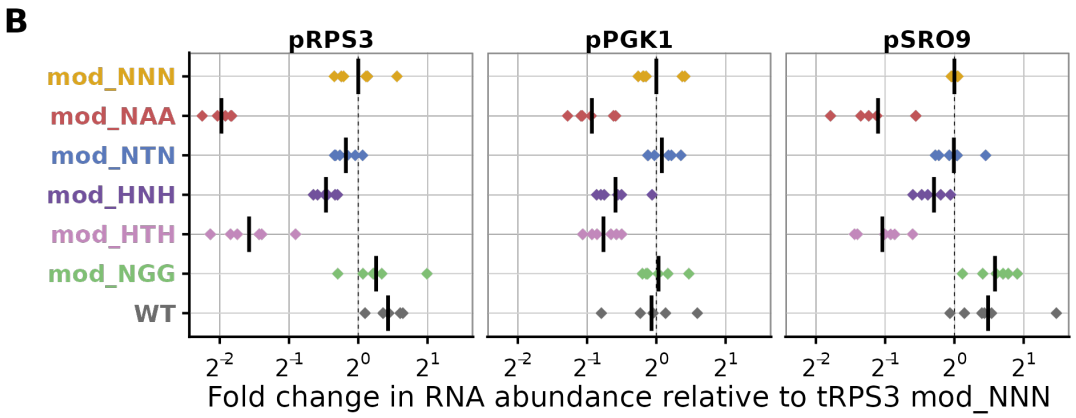
5.2.4. Motif effects on gene expression depend both on terminator context and promoter pairing

We compared the effects of cis-regulatory motifs on mRNA abundance to predicted effects inferred from the transcriptome-wide measurements of half-life. First, we trained a linear model using the RT-qPCR results to estimate the change in log2 mRNA abundance (i.e. ΔCq) due to the presence of a motif in each promoter and terminator combination. Using a simple model of transcript production and decay, we can argue that changes in mRNA abundance are directly proportional to changes in mRNA half-life (see methods).





Terminator	Motif 1	Motif 2	Motif 3
mod_NNN	Random	Random	Random
mod_NAA	Random	ATATTC	ATATTC
mod_NTN	Random	TGTAHMNTA	Random
mod_HNH	HWNCATTWY	Random	HWNCATTWY
mod_HTH	HWNCATTWY	TGTAHMNTA	HWNCATTWY
mod_NGG	Random	GTATACCTA	GTATACCTA
WT	NA	NA	NA



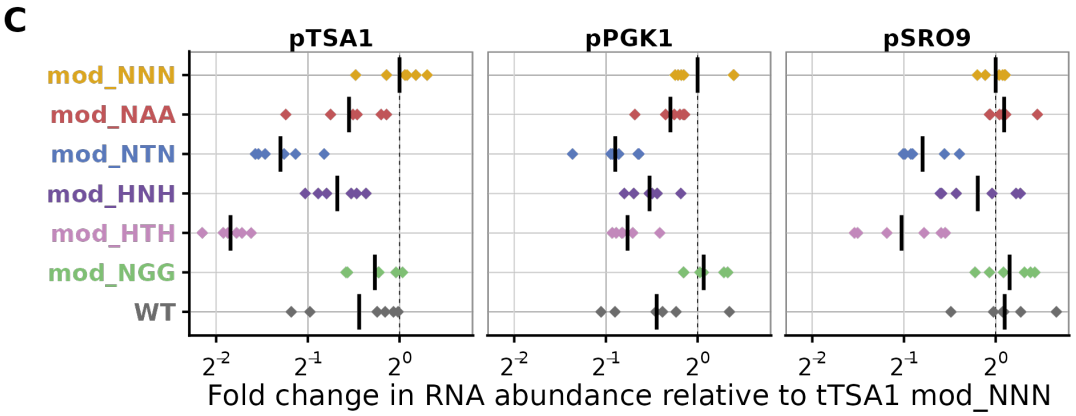
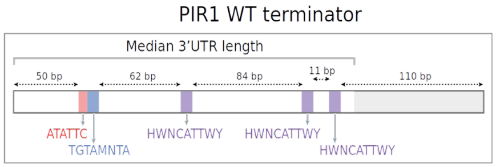


Figure 5.3: Motifs inserted into RPS3 and TSA1 host terminators change transcript abundance in RT-qPCR measurements. (**A**) Design of motif insertion sites in native RPS3 and TSA1 terminators, highlighting random insertion used as a negative control. Panel created by Jamie Auxillos. (**B**) Fold changes in transcript abundance for tRPS3 constructs paired with 3 promoters: pRPS3, pPGK1 and pSRO9. (**C**) Fold changes in transcript abundance for tTSA1 constructs paired with three promoters: pTSA1, pPGK1 and pSRO9. Each diamond represents a biological replicate, averaged over 3 technical replicates. The vertical line represents the mean of all 6 biological replicates. Fold changes are relative to the abundance of the mod_NNN construct, i.e. $2^{\Delta\Delta Cq}$ (see methods).

A



B

Terminator	Motif 1	Motif 2	Motif 3	Motif 4	Motif 5
WT	ATATTC	TGTAHMNTA	HWNCATTWY	HWNCATTWY	HWNCATTWY
mod_NTHHH	Random	TGTAHMNTA	HWNCATTWY	HWNCATTWY	HWNCATTWY
mod_ANHHH	ATATTC	Random	HWNCATTWY	HWNCATTWY	HWNCATTWY
mod_ATNHH	ATATTC	TGTAHMNTA	Random	HWNCATTWY	HWNCATTWY
mod_ATHNH	ATATTC	TGTAHMNTA	HWNCATTWY	Random	HWNCATTWY
mod_ATNNN	ATATTC	TGTAHMNTA	Random	Random	Random
mod_ANNNN	ATATTC	Random	Random	Random	Random
mod_NTNNN	Random	TGTAHMNTA	Random	Random	Random

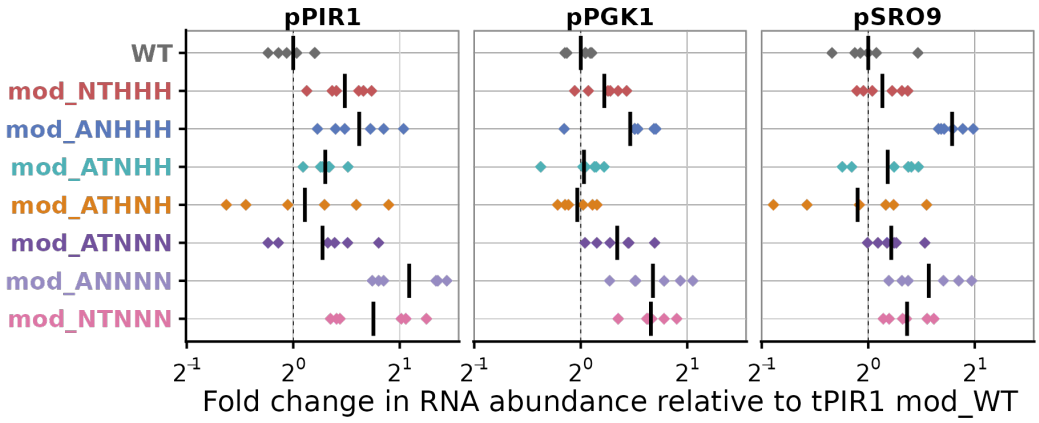


Figure 5.4: Motifs removed from PIR1 host terminators change transcript abundance in RT-qPCR measurements. (**A**) Design of PIR1 constructs with combinations of motifs replaced by random nucleotide sequences. Panel created by Jamie Auxillos. (**B**) Fold changes in transcript abundance for tPIR1 constructs paired with 3 promoters: pPIR1, pPGK1 and pSRO9. Each diamond represents a biological replicate, averaged over 3 technical replicates. The vertical line represents the mean of all 6 biological replicates. Fold changes are relative to the abundance of the WT construct, i.e. $2^{\Delta\Delta Cq}$ (see methods).

Therefore, we compared the estimated change in log2 mRNA abundance to that predicted due to changes in log2 half-life for each motif, estimated from our transcriptome-wide analysis of the (Chan et al., 2018) dataset. The effect of motifs in reporter constructs is correlated with the predictive model, but the strength of the correlation depends on context (Figure 5.5A). Constructs with inserted motifs (i.e. tRPS3 and tTSA1 constructs) had a lower correlation with predicted effects than constructs with deleted motifs (i.e. tPIR1 constructs). Interestingly, motif effects on mRNA abundance appear to be greater than that predicted from their effect on half-life when their host terminator is paired with its native promoter.

We next directly compared the estimated coefficients for the effect of each motif on mRNA abundance across promoter-terminator pairing (Figure 5.5B). The effect of a motif depends on terminator context. For example, ATATTCA reduces mRNA abundance substantially more when inserted in tRPS3 than in tTSA1 (pRPS3-tRPS3 log2 Fold Change = -0.99, p-value = 1.5×10^{-13}; pTSA1-tTSA1 log2 Fold Change = -0.26, p-value = 7.3×10^{-3}). Meanwhile, TGTAHMNTA significantly reduces mRNA abundance when inserted in tTSA1, but not tRPS3, whichever promoter is chosen (pRPS3-tRPS3 log2 fold Change = -0.18, p-value = 0.37; pTSA1-tTSA1 log2 fold Change = -1.30, p-value = 1.0×10^{-7}). Promoter choice also influences the magnitude of the contribution of a motif to mRNA levels. For the ATATTC, TGTAHMNTA and HWNCATTWY motifs, the greatest reduction in mRNA abundance occurred when native promoter-terminator pairings are measured. This is true for all 3 decay motifs across all 3 host terminators, except for HWNCATTWY in pRPS3-tRPS3 constructs (Supplementary Table B12).

Regulatory interactions between different motifs also change depending on host terminator and promoter context. We included an interaction term that quantifies how the effect of including both HWNCATTWY and TGTAHMNTA together differs from the sum of the effects of including these motifs individually. The combination of TGTAHMNTA and HWNCATTWY in tRPS3 has no significant effect beyond a simple sum of their individual effects when paired with pPGK1 (p-value = 0.39). However, when tRPS3 is paired with pRPS3 or pSRO9, the combination has a greater effect than expected from the sum of the individual effect of each motif (pRPS3 log2 fold Change = -0.47, p-value = 0.0015; pSRO9 log2 fold Change = -0.37, p-value = 0.02). The combination of TGTAHMNTA and HWNCATTWY in tTSA1 has no additional effect (pTSA1 p-value = 0.67, pSRO9 p-value = 0.91), except when paired with pPGK1, where it has a lesser effect than expected (log2 fold Change = 0.33, p-value = 0.02). Finally, the combination of TGTAHMNTA and HWNCATTWY in tPIR1 has no significant additional effect (Supplementary Table B12).

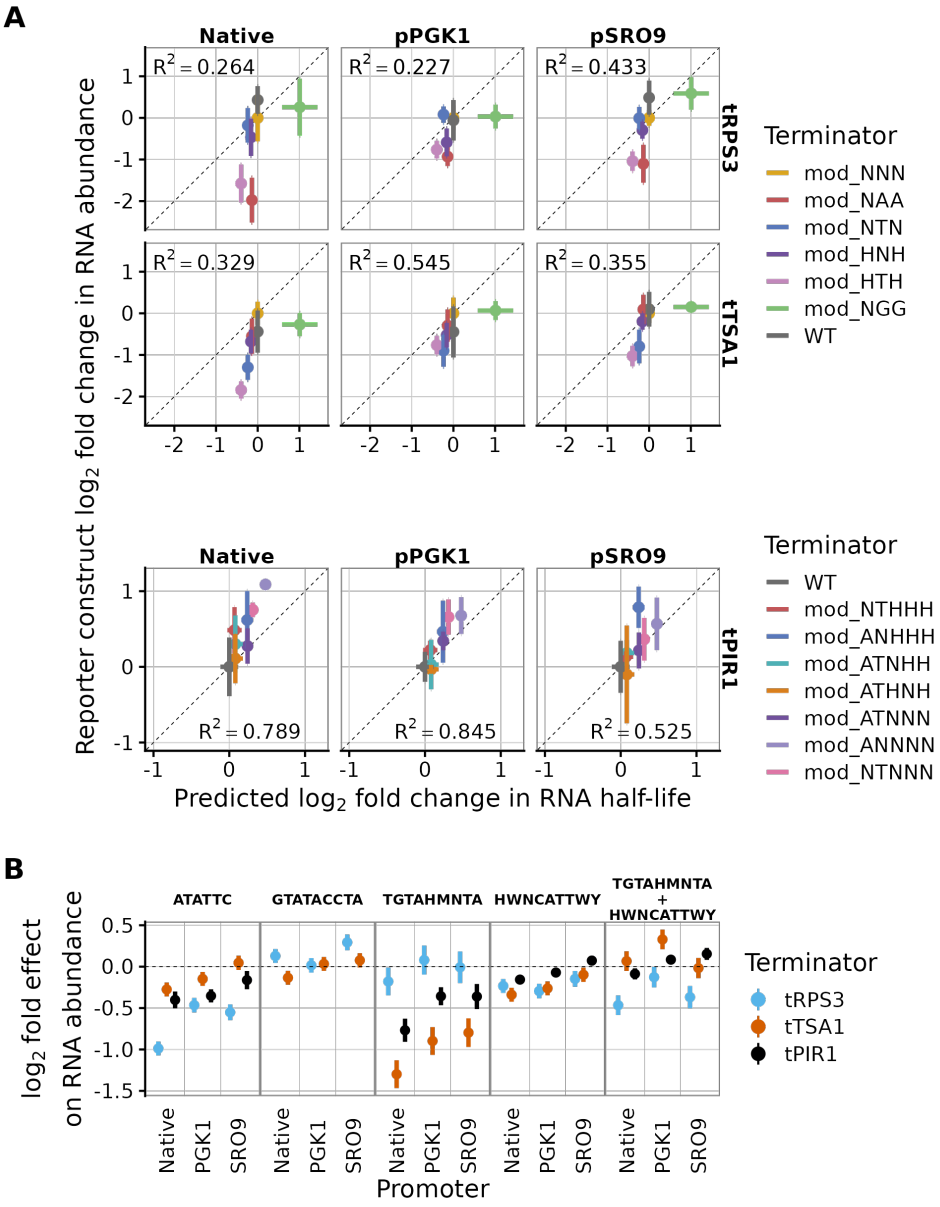


Figure 5.5: Promoter and terminator context alter the regulatory effect of motifs. (**A**) Predicted change in transcript abundance inferred from transcriptome-wide motif contributions to half-life, compared to RT-qPCR measurements of reporter transcript abundance. The y-axis shows statistical summaries (mean and standard error) of 6 replicates for data shown in figures 4 and 5. The fold change is relative to the mod_NNN construct in each promoter-terminator pairing for the tRPS3 and tTSA1 sets, and relative to the WT constructs for tPIR1 sets. Native promoter panels show the promoter paired with the terminator from the same set, e.g. pRPS3 with tRPS3. (**B**) Motif contributions to fold changes in mRNA abundance for reporter constructs with different promoter and terminator contexts. This is calculated by a linear model with a coefficient for the effect of each motif in each set, applied to ΔCq against 3 reference genes. The last column shows the interaction term between TGTAHMNTA and HWNCATTWY.

5.2.5. Inserting motifs into terminators shifts poly(A) site usage downstream

We next mapped poly(A) site usage in a subset of reporter constructs, for 2 reasons. First, changes in poly(A) site usage might mean that motifs placed in our reporters were unintentionally absent from the mature mRNA. Second, we wanted to know if motif effects on mRNA abundance might be due to changes in the poly(A) site usage. We chose three constructs with large effect sizes in the qPCR results: mod_NAA, mod_HTH and mod_NTN, together with WT and mod_NNN controls, within three promoter-terminator contexts: pRPS3-tRPS3, pPGK1-tRPS3 and pTSA1-tTSA1. For these constructs, we performed poly(A)-site proximal paired-end sequencing of 3' mRNA-Seq libraries following Lexogen's oligo(dT) priming QuantSeq protocol (Moll et al., 2014 and see methods). Read 1 allows precise inference of poly(A) site position while read 2 generally overlaps the CDS and allows distinguishing terminators in native loci from reporter constructs. We mapped the reads to genome sequences extended by the relevant reporter plasmid sequence.

We detected 1000s of reads on each reporter construct, which is enough to quantify expression confidently as well as to assign poly(A) sites. We checked that counts of all other RNAs are highly correlated between samples, giving us confidence that changes in construct detection are meaningful (Supplementary Figure B5). Transcript abundance, relative to mod_NNN, correlates strongly with qPCR results for most constructs (Figure 5.6A). However, some constructs were detected as more abundant by RNA-seq for reasons that are unclear.

We display the poly(A) sites as the cumulative fraction of poly(A)-site reads mapped at each location downstream of construct stop codons, out of all reads mapped to the terminator (Figure 5.6B). We confirmed that in other genes the poly(A) site locations were highly reproducible across samples, giving us confidence that changes in reporter poly(A) sites are meaningful (Supplementary Figure B6). Poly(A) sites are in the same relative positions in native loci and the constructs with wild-type terminator (Supplementary Figure B7). Then, we compared 3'end positions of reads between modified and wild-type reporter constructs to determine the poly(A) site usage. In both wild-type and mod_NNN tTSA1 constructs, the relative usage of major poly(A) sites does not significantly change (Site 1 p-value = 0.068 and Site 2 p-value = 1). Surprisingly, tRPS3 mod_NNN constructs appear to be using a novel upstream poly(A) site, that appears in about 50 percent of reads and is located upstream of the 3rd motif insertion site (Site 1 p-value = 0.016; Figure 5.6B).

Next, we compared changes in poly(A) site usage between constructs with different inserted motifs. We highlight the 2 major poly(A) sites for the tRPS3 constructs and 2 for the tTSA1 constructs (black vertical lines on the mod_NNN constructs in Figure

5.6B) and track the cumulative fraction of reads upstream of each major site. However, for tRPS3 constructs, there is a distinct shift to downstream poly(A) sites in constructs with verified decay motifs inserted, rather than random sequences. There are also smaller differences in poly(A) site usage between constructs with different motifs (Supplementary Figure 5.6C, Supplementary Table B13).

We then asked whether modifications in the 3'UTR region impact 5'-3' degradation following mRNA decapping, using the 5PSeq method targeted to the 3'-end regions of mRNA (Pelechano et al., 2016). 5PSeq can detect changing ribosome dynamics through 5'-3' co-translational degradation, however, the novel modification to 5Pseq here uses an anchored oligo(dT) reverse primer so detects only the poly(A)-site proximal region of the mRNA instead of the entire coding sequence. The 5Pseq counts per gene are highly reproducible between samples (Supplementary Figure B5B), and the abundances of reporter mRNAs from different constructs correlate well between 5PSeq and QuantSeq data (Supplementary Figure B5C). Our 5PSeq data finds no detectable changes in 5'-phosphorylated intermediates between wild-type and modified reporter constructs, and thus does not indicate detectable changes in ribosome dynamics near the 3' end of transcripts (Supplementary Figure B7, B9). It does confirm the behaviour of the inserted motifs correlates with RT-qPCR results and that an upstream alternative polyadenylation site is introduced in the tRPS3 constructs (Supplementary Figure B4). Moreover, the poly(A) site distribution for each construct matches that obtained from QuantSeq data, suggesting that 3'UTR isoforms are not differentially degraded using this pathway regardless of the motifs inserted (Supplementary Figure B8, Supplementary Table B14). Finally, the abundances of different reporter mRNAs from different constructs correlate well between 5PSeq and QuantSeq data (Supplementary Figure B5C).

Overall, poly(A) site mapping showed that most reporter mRNAs retained the expected poly(A) site and motifs, except for a new alternative poly(A) site in tRPS3 mod_NNN constructs. This highlights the potential for unexpected consequences from composing cis-regulatory elements, even when introducing "random" insertions of no known function.

5.3. Conclusions

This work explored the limitations of composability in cis-regulatory elements. The effects of interacting promoter and terminator regions have been well documented in the synthetic biology literature (Ito et al., 2013; Dhillon et al., 2020). However, the standard narrative describes this degree of unpredictability as a nuisance obstructing the creation of reliable genetic circuits and high-value products (Kittleson et al., 2012). The focus on creating reliable components with predictable contributions overlooks the evidence for

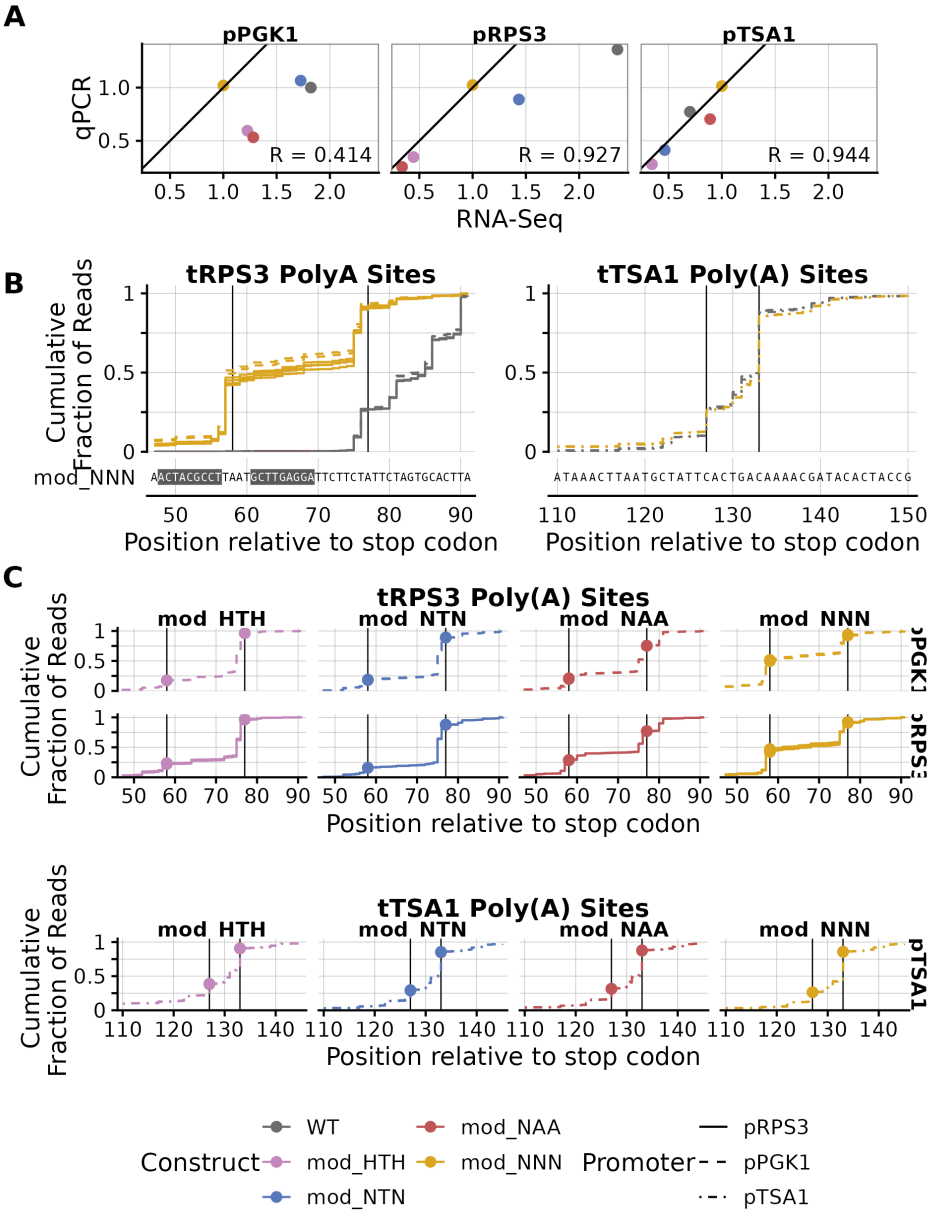


Figure 5.6: Inserting motifs into RPS3 and TSA3 terminators changes 3'UTR length. A subset of constructs were chosen to investigate changes in poly(A) site usage: WT, mod_NNN, mod_NTN, mod_NAA, and mod_HTH; within three promoter-terminator contexts: pRPS3-tRPS3, pPGK1-tRPS3, and pTSA1-tTSA1. (**A**) Comparison of construct transcript abundance as independently measured by RT-qPCR and RNA-Seq assays. Transcript abundance was normalised to the median abundance of plasmid URA3, genomic PGK1, and RPS3 or TSA1 transcripts for each construct. Fold change is relative to the mod_NNN construct in each promoter-terminator context. The black diagonal line represents the expected values if RNAseq and RT-qPCR results correlated perfectly. (**B**) Cumulative count of reads mapped downstream of WT (grey) and mod_NNN (golden) construct stop codons as fraction of total reads mapped to the constructs terminator. The x-axis shows the terminator sequence of the mod_NNN construct with inserted random motifs highlighted with a black background. WT reads have been shifted downstream to align with the mod_NNN sequence by accounting for motif insertion sites. Major poly(A) sites have been highlighted by a black vertical line. (**Left**) shows the poly(A) sites for the pRPS3-tRPS3 and pPGK1-tRPS3 promoter-terminator constructs. (**Right**) shows the same for the pTSA1-tTSA1 constructs. (**C**) Similar to Figure **B** but with each motif insertion construct plotted separately. Columns designate cumulative plots from different terminator constructs and rows designate cumulative plots from different promoter contexts.

a far more complex picture of regulatory mechanisms that is biologically interesting in itself.

We first built on current literature to show the variability in contribution to gene expression from terminators from known regulatory genes in response to different coding regions and promoters. As expected, our protein fluorescence results showed up to 100-fold changes by promoter choice alone and then up to 5-fold by terminator choice. Then, we highlighted significant changes in several terminator contributions to gene expression across contexts. We highlighted the quantitative limitations of the assumption of composability with 1.5-fold change in the relative effect of terminators depending on coding sequence and promoter choice.

We extended our understanding of this unexplained behaviour by testing the regulatory behaviour of cis-regulatory elements within the terminator, namely short sequence motifs within the 3'UTR. We showed that the analysis of published data sets can enhance experimental design by building on previous work by Cheng et al., 2017 and shortlisted several prospective 3'UTR motifs using a linear model predicting half-life. Inserting or removing these motifs from three native host terminators, we showed that three of the four motifs performed as expected on their effect on transcript levels as measured by RT-qPCR. Furthermore, the magnitude of their contributions changed for all motifs across promoter and host terminator. TGTAHMNTA had no measurable effect when inserted into tRPS3 but had the expected effect in tTSA1, ATATTC had the expected effect in tRPS3 but little effect in tTSA1, and HWNCATTWY can either decrease or increase mRNA levels when removed from tPIR1, depending on the promoter. Also, the two tPIR1 constructs with different mutated HWNCATTWY motifs had different expression levels suggesting HWNCATTWY has a position-dependent effect. However, the exact sequences of the two HWNCATTWY motif instances did also differ by 4 nucleotides. Interestingly, when two motifs were inserted/removed together their combined contribution also changed across contexts.

RNA-Seq results confirmed our conclusions on the effect of motif insertions on gene expression. pRPS3-tRPS3 and pTSA1-tTSA1 constructs show similar relative abundances in the RT-qPCR results as the RNA-Seq results. However, pPGK1-tRPS3 results were skewed due to the unexpectedly low abundance of mod_NNN constructs in the RNA-Seq results, which all other constructs were normalised to. We investigated if positional effects could be contributing to the differing behaviour. In TSA1 the poly(A) site usage was unchanged between WT and the insertion constructs. However, in RPS3 a new alternative poly(A) site had been unintentionally introduced in between insertion sites 2 and 3 for all constructs. Although we tried to avoid altering elements that affect poly(A) sites in native terminators, the creation of a novel poly(A) site is likely due to

the inserted motifs extending the distance between the efficiency elements and the native poly(A) sites. Another possible explanation for the novel poly(A) site is that motifs 2 and 3 were inserted into a conserved element of tRPS3. As TGTAHMNTA and ATATTC were both inserted in site 2 (and ATATTC is also inserted into site 3) there is a chance that their behaviour is affected in RPS3. Interestingly, despite all tTSA1 constructs having 2 ATATTC motifs they had less effect on transcript level than in tRPS3, which had just one copy in nearly 50% of transcripts. ATATTC could need to be close to a Poly(A) site, so the novel poly(A) site in tRPS3 is actually beneficial, but ATATTC is not near a poly(A) site in tPIR1 and still had a greater effect than in tTSA1. Meanwhile, TGTAHMNTA could be disrupted by the proximity to the novel poly(A) site. Although, it is interesting that only in tRPS3 is the combined effect of HWNCATTWY and TGTAHMNTA synergistic. Further constructs with motifs inserted in different positions across a terminator are required to confirm the positional effects.

Our results support long-standing mechanistic observations that CRE contributions depend on the presence of other CREs in the host gene. Promoter and terminator sequences are known to share transcription factors common to initiation and termination by forming gene loops (O'Sullivan et al., 2004). Promoter sequences have also been shown to have contributions to mRNA degradation by interacting with trans-regulatory elements that have binding motifs on mature mRNA (Bregman et al., 2011). Within mature mRNA, CREs associated with stability and degradation can require proximity to the Poly(A) tail in order to interact with deadenylation factors that initiate degradation (Duttagupta et al., 2005; Geisberg et al., 2014; Cheng et al., 2017). Several RNA binding protein motifs located in the 3'UTR have also been shown to depend on the presence of specific secondary structures, such as hairpin loops, in order to be effective (Olivier et al., 2005; Aviv et al., 2006; Geisberg et al., 2020).

We believe that the study of interactions between cis-regulatory elements is an understudied research area. It promises to improve the predictability of contributions to gene expression required to enhance the design of synthetic pathways and is also a fruitful region for discovering novel mechanisms through which cell regulate their expression. We have shown that changes in contributions according to context can affect cis-regulatory elements such as motifs as well as regions like terminators. We have also shown that changes in contributions due to co-occurring motifs can be measured using linear interaction terms trained on RT-qPCR data. Our work designing suitable insertion sites into host terminators to detect these changes also offers a framework to inspect context effects on cis-regulatory elements at scale. Finally, this work also showcases the usefulness of our R package tidyqpcr as our analysis of complex, multi-experimental RT-qPCR data is entirely open, reproducible and quality controlled, github.com/DimmestP/chimera_project_

manuscript.

Chapter 6

DiffFracSeq: A Bayesian Model for the Detection of Differential Fractionation of Sequencing Data

6.1. Introduction

Localisation of RNA populations to specific sub-cellular compartments is used to regulate gene expression across the tree of life (Das et al., 2021). A ubiquitous example is the distinct populations of RNA found to be localised to the nucleus or the cytoplasm. Most mRNA transcripts are localised in the cytoplasm in order to be translated (Köhler & Hurt, 2007). Many non-coding RNA (ncRNA) transcripts, such as those that facilitate splicing, are localised to the nucleus (Will & Lührmann, 2011). In plant and animal cells, microRNA (miRNA), long non-coding RNA (lncRNA) and small interfering RNA (siRNA) contribute to gene expression regulation through splicing, degradation or translation pathways specific either to the nucleus or cytoplasm (Hombach & Kretz, 2016). Localisation also aids in the transportation of secretory proteins as their mRNA transcripts are co-translationally translocated to the endoplasmic reticulum (Jan et al., 2014).

Localisation can enable rapid changes in gene expression in response to stress or facilitate precise changes at highly sensitive stages of cell cycles. In response to stimuli, such as heat stress, surplus mRNA transcripts can be collected into stress granules where translation may be suppressed (Anderson & Kedersha, 2009). Previous studies have shown that between 10%-15% of mRNA transcripts are localised to granules when exposed to stress (Khong et al., 2017; Treeck et al., 2018). The mRNA transcripts enriched in stress granules are characterised by poor translatability and long coding region/3' UTR length (Khong et al., 2017). Granules of ribonucleoproteins, such as processing bodies (P-bodies), also facilitate the tight regulatory control of translation without degrading mRNA (Buchan & Buchan, 2014). In *C. elegans*, oogenesis includes the release of nuclear-bound

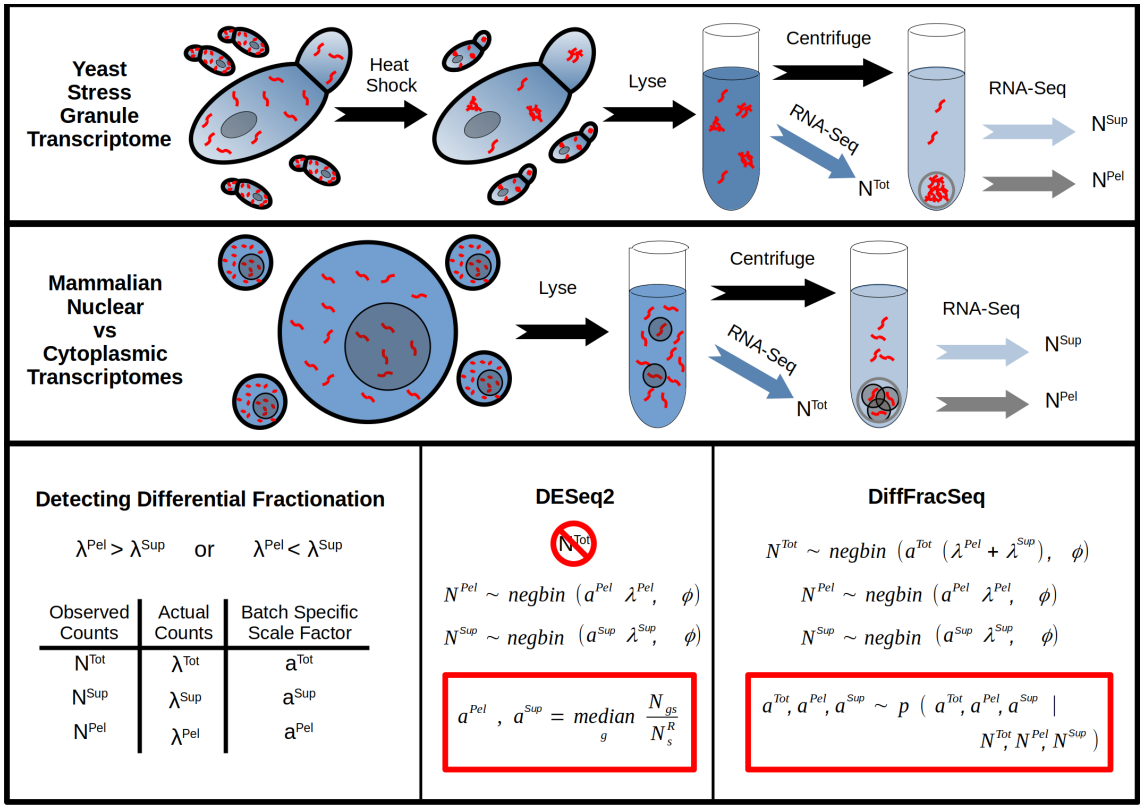


Figure 6.1: Summary of two different fractionation experiments and comparison of detecting differential fractionation with DESeq2 vs DiffFracSeq. The top panel summarises a fractionation experiment by Iserman *et al* that investigates the transcriptome of stress granules formed by yeast cells in response to heat stress (Iserman et al., 2020). The middle panel summarises a fractionation experiment by the ENCODE consortium to compare the nuclear and cytoplasmic transcriptomes in lymphoblastoid cells (Dunham et al., 2012). Both panels highlight the stages at which RNA-Seq samples are taken: one before fractionation, N^{Tot}, and two after fractionation, N^{Sup} and N^{Pel}. The bottom panel outlines the task of detecting differential fractionation using noisy RNA-Seq transcript counts. The RNA-Seq count models behind DESeq2 and DiffFracSeq are also summarised with key differences highlighted in red.

.

P-bodies into the cytoplasm during prophase I (Voronina et al., 2011). Meanwhile, mutations that affect the formation of ribonucleoprotein granules are associated with several human diseases (Mackenzie et al., 2017).

Several RNA-Seq assays have been developed to investigate RNA localisation by isolating and comparing RNA populations in different sub-cellular compartments. Fractions of mRNA from organelles of different densities can be separated by centrifuging cell lysate. Centrifuging using a sucrose gradient or by repeatedly centrifuging the supernatant under increasing speeds can separate a sample into multiple fractions (Dunham et al., 2012; Hu et al., 2017; Iserman et al., 2020). Experiments can also include an immunopurification step after centrifugation to further purify samples with compartments of interest (Khong et al., 2017). Alternatively, fractions of freely floating vs protein-bound mRNA can be separated by orthogonal organic phase separation (Queiroz et al., 2019).

Despite the development of multiple fractionation-based RNA-Seq assays, there are no statistical methods designed to detect differential fractionation. Previous studies to determine changes in transcript abundance across different fractions have had to use statistical software developed to determine differential expression across conditions. DESeq (Anders & Huber, 2010), edgeR (Robinson & Oshlack, 2010) and Cuffdiff (Trapnell et al., 2010) have been used in attempts to determine differential fractionation in yeast and human data sets (Hubstenberger et al., 2017; Khong et al., 2017; Treeck et al., 2018; Matheny et al., 2019). Unfortunately, these methods address sequencing bias by assuming only a subset of genes will experience biologically significant differences across conditions. Under this assumption, the change in the average gene is not biological, but must be down to sequencing bias; predominately the batch-specific library size. Quantile normalisation techniques, such as the median of medians used by DESeq, attempt to find an average gene to normalise to across the conditions and reduce the sequencing bias (Anders & Huber, 2010). However, the assumption breaks down in cases where the majority of genes are expected to have different expressions across RNA-Seq samples. For example, in assays that extract RNA from different fractions, changes in abundance across the entire transcriptome may be expected. These methods also estimate sequencing bias *a priori* and any uncertainty in their values is not included in the tests for differential expression.

This work presents DiffFracSeq, a Bayesian statistical model specifically designed to detect differential fractionation. The chapter outlines an alternative way of normalising RNA-Seq data sets from different fractions using the additional information that can be gathered when measuring subsets of a complete sample. The normalising method uses the transcript counts of samples taken before fractionation to enable reliable inference of RNA-Seq batch-specific scale factors within the Bayesian model, rather than relying on

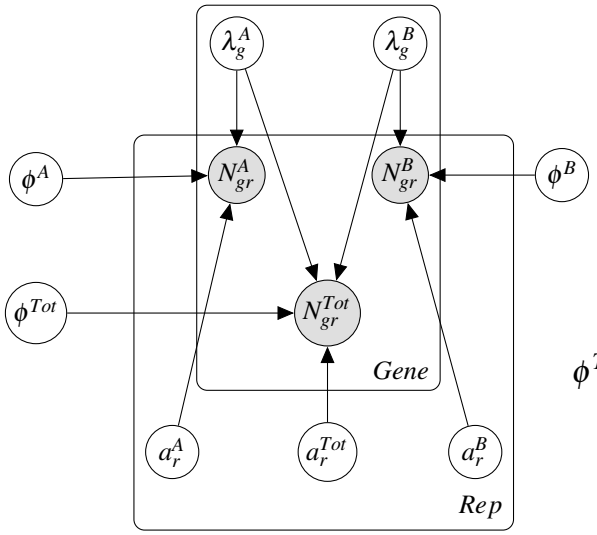
a priori estimations. First, the model behind DiffFracSeq is introduced and its ability to model noisy RNA-Seq transcript counts is tested using a simulated data set. The simulated data set is also used to show the limitations of using quantile normalisation methods in experiments with global changes in the transcriptome by using DESeq2 to detect differential fractionation (Love et al., 2014). Then, DESeq2 and DiffFracSeq are applied to two experimental data sets, outlined in Figure 6.1, and their ability to detect differential fractionation is compared. DiffFracSeq is an open-source R software package that will enable even more sensitive comparisons of transcript localisation across conditions and cell types.

6.2. Results

6.2.1. Bayesian hierarchical model

The DiffFracSeq model reliably detects differential fractionation by: A) using RNA-Seq counts taken from a sample before fractionation as a quasi-replicate to provide additional information with which to normalise the counts taken from the fractionation samples, and B) including the determination of batch-specific scale factors within the Bayesian framework rather than depending on *a priori* estimations. The normalisation of the counts from different fractions can be aided by the counts from a pre-fractionation sample if the sum of transcripts counts from sub-fractions is assumed to equal the counts from the total body; i.e. $N^{Tot} = N^A + N^B$. Summing the counts across all fractions accounts for the expected global changes in their transcriptomes, but any difference between this sum and a sample from the total body can be assumed to come from sequencing bias rather than biological effect. Therefore, the sample taken before fractionation can be considered a quasi-replicate that enables information to be shared across fractions to account for the sequence bias.

The Bayesian model determines values for noiseless transcript counts λ, overdispersion parameters ϕ, and batch-specific library scale factors a. The noise of transcript counts from RNA-Seq data is modelled by a negative binomial with mean λ and overdispersion parameter ϕ, as RNA-Seq transcript counts are positive integers and overdispersion is often present (Cameron & Trivedi, 1998; Robinson & Smyth, 2007). The DiffFracSeq model contains three negative binomial distributions: one for samples from the total body and one for each of the two sub-fractions Figure 6.2. The three negative binomial distributions each have an overdispersion parameter: ϕ^{Tot}, ϕ^A and ϕ^B, that is shared across genes and conditions. Separate mean parameters for transcript counts are learnt for the sub-fraction negative binomials: λ^A and λ^B, and their sum is used as the mean for the total negative binomial. The λ parameters are determined in log space to help the model fit the broad range of transcript count levels expected across an entire genome. In



$$N_{gr}^{Tot} \sim negbin(a_r^{Tot}(e^{\lambda_g^A} + e^{\lambda_g^B}), \phi^{Tot})$$
$$N_{gr}^{B} \sim negbin(e^{a_r^B + \lambda_g^B}, \phi^B)$$
$$N_{gr}^{A} \sim negbin(e^{a_r^A + \lambda_g^A}, \phi^A)$$

$$\lambda_g^A, \lambda_g^B \sim normal(0,1)$$
$$\phi^{Tot}, \phi^A, \phi^B \sim normal(100,10)$$
$$a_r^{Tot} \sim normal(10, 0.1)$$
$$a_r^A, a_r^B \sim normal(0, 0.1)$$

Figure 6.2: Plate diagram summarising the basic Bayesian hierarchical model behind DiffFracSeq. Shaded circles represent observed variables, in this case the unnormalised RNA-Seq transcript counts, and white circles represent latent variables learnt by the model. Circles placed within a square plate have separate random variables for each possible value associated with that plate, i.e. the a^A circle is within the Rep plate and Con plate as there is a a^A for each replicate and condition.

addition to the mean transcript count parameter, each negative binomial has a total reads scale factor term: a^{Tot}, a^A, a^B, that is unique to the condition and replicate used. The R package RStan is used to sample from the posterior distribution, the core stan code is available in Appendix C.

6.2.2. Overview of the simulated test data set

The validity of DiffFracSeq as a model of RNA-Seq fractionation data sets was first tested using a simulated data set. The simulated data set consisted of 300 genes with total transcript counts, X^{Tot}, sampled from a lognormal distribution to simulate the range of gene expressions in an RNA-Seq data set. The fractionation of total transcripts into fractions A and B were simulated by sampling the ratio of transcripts, γ, between fractions A and B from a beta distribution. The parameters of the beta distribution were set to simulate three regimes, Figure 6.6A. The first regime, $beta(2,2)$, randomly allocates genes such that 50% of genes have transcripts that are biased to be in fraction A and 50% of genes have transcripts that are biased to be in fraction B. The second regime, $beta(4,2)$ introduces a marginal bias towards fraction B in the global transcriptome with $\approx$ 70% of genes having transcripts that are biased to be in fraction B. The final regime, $beta(4,1)$ represents the largest differential fractionation effect as only a specific subset of genes, $\approx$ 15%, have transcripts that are biased to be in fraction A, and $\approx$ 85% are in fraction B.

The noiseless total counts and fraction ratios were then converted into noisy RNA-

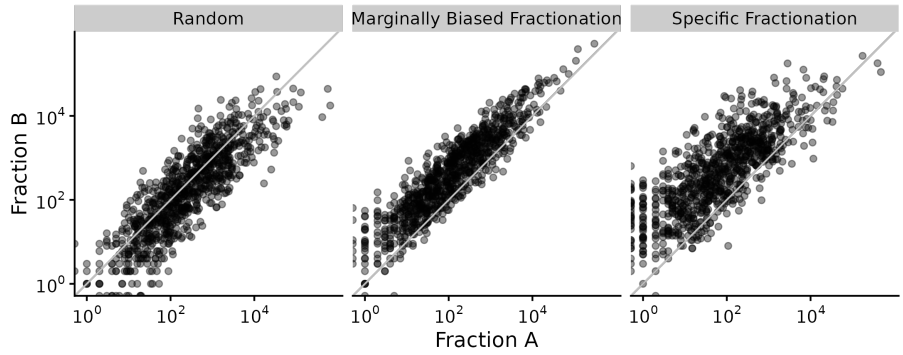


Figure 6.3: Overview of the simulated data set used in this study. Simulated mRNA transcript counts between two fractions in three regimes: all genes are randomly allocated a fraction, most genes have a marginal bias towards fraction B, and the majority of transcripts are in fraction B but transcripts from a specific subset of genes are found in fraction A.

Seq counts to test the DiffFracSeq model. First, noiseless counts from the two fractions were created by multiplying the noiseless total counts by the simulated ratios, $\gamma_{gc}X_g^{Tot}$ and $(1-\gamma_{gc})X_g^{Tot}$. The batch-specific scale factor parameter, α, is introduced to represent the varying total reads expected from every replicate, condition and fraction sequencing run. Noise typically associated with RNA-Seq data sets was introduced to the ideal gene-wise transcript counts by sampling from a negative binomial with mean equal to the ideal value times by the scale factor creating the training data, N^A, N^B and N^{Tot}. An appropriate overdispersion parameter value of 100 was estimated from experimental count data. Three data points are sampled for each gene in each of the two fractions and the three regimes to create three replicates, Figure 6.3.

$$X_g^{Tot} \sim lognormal(\mu, \sigma^2)$$

$$N_{grc}^A \sim negbin(\gamma_{gc}\alpha_{rc}^A X_g^{Tot}, \phi)$$

$$N_{grc}^B \sim negbin((1-\gamma_{gc})\alpha_{rc}^B X_g^{Tot}, \phi)$$

$$N_{grc}^{Tot} \sim negbin(\alpha_{rc}^{Tot} X_g^{Tot}, \phi)$$

$$\alpha \sim uniform(min = 0.5, max = 3)$$

$$\gamma_{gc} \sim beta(a_c, b_c)$$

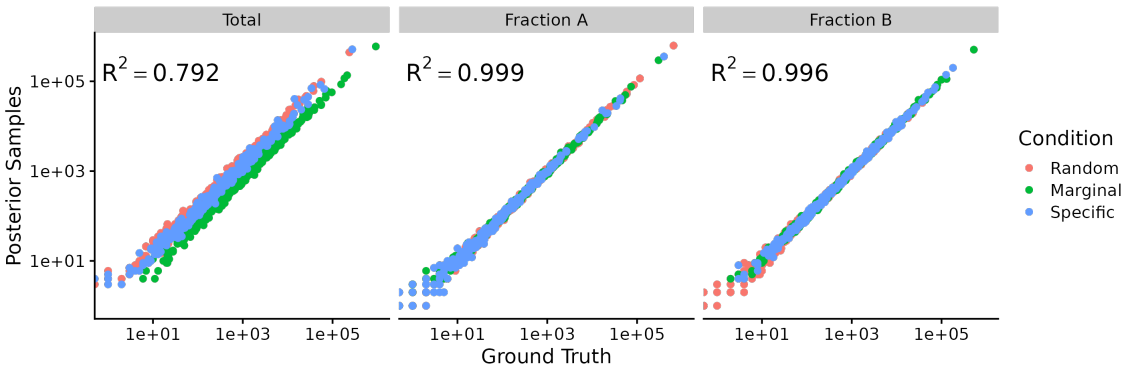


Figure 6.4: Comparison of noisy transcript counts sampled from DiffFracSeq's posterior distribution to the simulated ground truth. Noisy transcript counts for all genes in the total sample and both fractions as sampled from DiffFracSeq vs from the simulated ground truth.

6.2.3. Posterior checks using the simulated test data set

Posterior checks confirm the reliability of the DiffFracSeq model as it correctly recreates the simulated test data set. The three replicates of noisy N^{Tot}, N^{A} and N^{B} counts from each regime of the simulated data set are used to train the DiffFracSeq model. 1000 samples of all parameters are taken from the posterior distribution after a 1000 iteration burn-in. The median value of the 1000 samples is used as a summary statistic for all parameters. N^{Tot}, N^{A} and N^{B} counts from the simulated data set are compared to the posterior samples taken from the DiffFracSeq model, Figure 6.4. The counts sampled from the DiffFracSeq model show the model explains the majority of the variation in the counts from the ground truth with R^2 values of 0.999 and 0.996 for counts from the two fractions. Counts from the total sample vary more from the ground truth, $R^2 = 0.792$, likely due to the distribution being explored in linear space, to enable the addition of fractions A and B, rather than log space.

6.2.4. Detecting differential fractionation using DESeq2

The detection of differential fractionation with DiffFracSeq is compared to the results of a widely used R package for detecting differential expression using *a priori* normalisation, DESeq2. To run DESeq2, the counts for a given condition were combined as one matrix with every column holding a different fraction and replicate. The design matrix passed to DESeq2 consisted of whether the counts of a gene came from fraction A or fraction B only, `design_matrix = ~fraction`. This method was repeated separately for

```
column_data_simulated <- data.frame(fraction = factor(rep(c("pel",
                                                            "sup"),
                                                          3)),
                                    condition = factor(rep("Random",
                                                           6)))

rownames(column_data_simulated) = colnames(fractionation_count_matrix_simulated)

DESeq2_data_set_simulated_random <- DESeqDataSetFromMatrix(
    countData = fractionation_count_matrix_simulated[,c("random_1_pel", "random_1_sup",
                                                        "random_2_pel", "random_2_sup",
                                                        "random_3_pel", "random_3_sup")
    ],
    colData = column_data_simulated[1:6,],
    design = ~fraction)

DESeq2_data_set_simulated_random <- DESeq(DESeq2_data_set_simulated_random)


DESeq2_result_simulated_random <- results(DESeq2_data_set_simulated_random)
```

Listing 6.1: Example R code for using DESeq2 to detect differential fractionation in the random regime of the simulated data set.

each of the three conditions, Listing 6.1. As well as the determination of normalisation factors, DESeq2 also estimated shrinkage and calculated significance differences using a Wald test following the standard workflow outlined in its documentation.

6.2.5. Detecting differential fractionation with DiffFracSeq and DESeq2 with the simulated ground truth

The simulated ground truth in gene-wise differential fractionation reveals DiffFracSeq's ability to allow changes in the global transcriptome. However, DESeq2's normalisation method confounds batch-specific effects with global changes in expression. The $\log_2$ ratio of transcript counts between the two fractions as calculated by DESeq2 and DiffFracSeq were compared to the $\log_2$ ratio of noiseless counts in the simulated data set, $\gamma/(1-\gamma)$, Figure 6.5A. The coefficient of variation for the predicted $\log_2$ ratio and ground truth is greater than 0.95 for both methods in all regimes. Genes considered to be differentially fractionated in faction B over fraction A are highlighted in blue. For the DiffFracSeq model, a gene is considered to be significantly localised to fraction B if 97.5% of the λ^B samples are greater than all of the λ^A samples for that gene in that regime. This method provides a suitable summary statistic to compare to DESeq2's frequentist p-value, although DESeq2's results also have an FDR-based multiple-testing correction which is not applied to the DiffFracSeq model p-values.

The disparity between the models is revealed as the difference in the global transcriptome between the two fractions increases across the three regimes in the simulated data sets. The $\log_2$ ratios from DiffFracSeq consistently match the ground truth across all conditions and magnitudes. However, the $\log_2$ ratios from DESeq2 shift below the ground

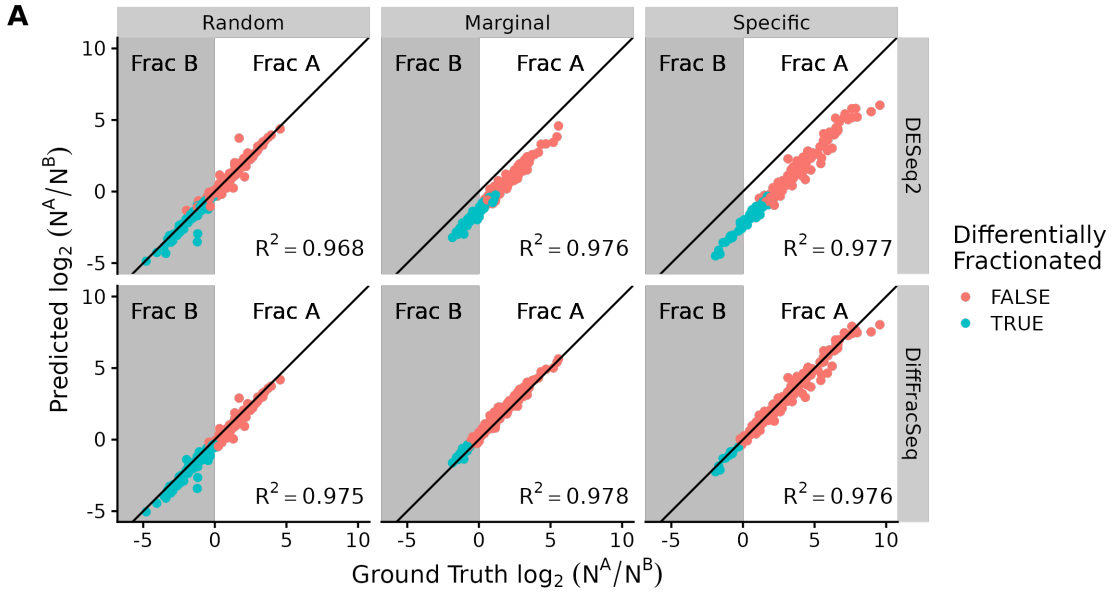
truth for all genes as the fraction transcriptomes change from a balanced 50%-50% random regime to an asymmetric 85%-15% specific regime. The shift in $\log_2$ ratios across all genes is explained by a divergence in DESeq2's normalisation scale factors in the regimes with asymmetric transcriptomes, Figure 6.5B. In the balanced random regime, DESeq2's normalisation scale factors all follow the same linear relationship with the ground truth scale factor. For the asymmetric marginal and specific regimes DESeq2's normalisation scale factors for fraction A diverge from the scale factors from fraction B. Therefore, the absorption of global changes in transcriptome by DESeq2's scale factors limits its ability to detect fractionation.

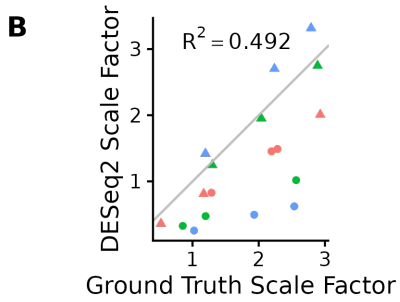
The shift in DESeq2's $\log_2$ ratios is reflected in the detection of significant differential fractionation across all transcript counts. Across all genes, DESeq2 has a larger false discovery rate (FDR) in the marginal and specific regimes than DiffFracSeq. In both cases, over 50% of genes detected to be differentially fractionated by DESeq2 are false positives, but all genes detected by DiffFracSeq are true positives. Although DESeq2 does have a better FDR in the random regime, 0 vs 0.01, its true positive rate (TPR) is less than DiffFracSeq, 0.78 vs 0.93, Figure 6.5C. This behaviour is replicated over the 60 least abundant genes and the 60 genes with the smallest change between the conditions, Figure 6.5D. Overall, genes determined to be differentially fractionated by DiffFracSeq are more likely to be true positives than those determined by DESeq2.

6.2.6. Overview of the experimental test data sets

DiffFracSeq is shown to handle the scale of real experimental data sets by analysing the results of a $\approx 16,000$ gene fractionation experiment. The experimental data set compares the nuclear and cytoplasmic transcriptomes of human cells. The data set is from the Encyclopedia of DNA Elements (ENCODE) consortium (Dunham et al., 2012). The data set consists of total, nuclear and cytoplasmic poly(A) RNA transcripts from a human GM12878 lymphoblastoid cell line. The fractions were separated using centrifugation and include two biological replicates which have a high correlation, Figure 6.6A. The reads were already aligned to the hg38 human reference genome and counted by the RSEM software (Li & Dewey, 2011) following the standard ENCODE analysis pipeline (Luo et al., 2020). It was downloaded from the ENCODE portal with the following identifiers: ENCSR000COR, ENCSR000COQ, ENCSR000CPO.

Finally, DiffFracSeq's ability to detect changes in the fractionation of the transcripts of a gene across conditions is tested using a multi-temperature yeast data set (Iserman et al., 2020). This experimental data set is from a paper investigating the transcriptomes of heat-induced stress granules at 30°C, 40°C and 42°C in *Saccharomyces cerevisiae*. Stress granules were isolated in pellets by 18,000g centrifugation before their

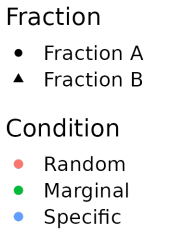




C

All Genes

Condition	Method	TPR	FDR
Random	DESeq2	0.782	0
Random	DiffFracSeq	0.932	0.0144
Marginal	DESeq2	0.979	0.598
Marginal	DiffFracSeq	0.458	0
Specific	DESeq2	1	0.849
Specific	DiffFracSeq	0.688	0



D

Condition	Method	Least Abundant TPR	Least Abundant FDR	Smallest Change TPR	Smallest Change FDR
Random	DESeq2	0.529	0	0.312	0
Random	DiffFracSeq	0.882	0.0323	0.688	0.0833
Marginal	DESeq2	0.9	0.4	0.952	0.643
Marginal	DiffFracSeq	0.3	0	0	0
Specific	DESeq2	1	0.812	1	0.817
Specific	DiffFracSeq	0	0	0.545	0

Figure 6.5: Comparison of DESeq2 and DiffFracSeq performance a simulated data set. A Predicted vs ground truth $\log_2$ ratio of transcripts in fraction A to fraction B. The first row presents the results determined from DESeq2 and the second row presents the results from DiffFracSeq. The results from the three regimes in the simulated data set are shown across the columns. **B** Comparison of DESeq2 values of RNA-Seq run specific total read scale factors to the ground truth. **C** True positive rates (TPR) and false discovery rates (FDR) of the two methods across three regimes for all genes. **D** Similar to **C** but across the 60 least abundance genes and the 60 genes with the smallest change between fractions.

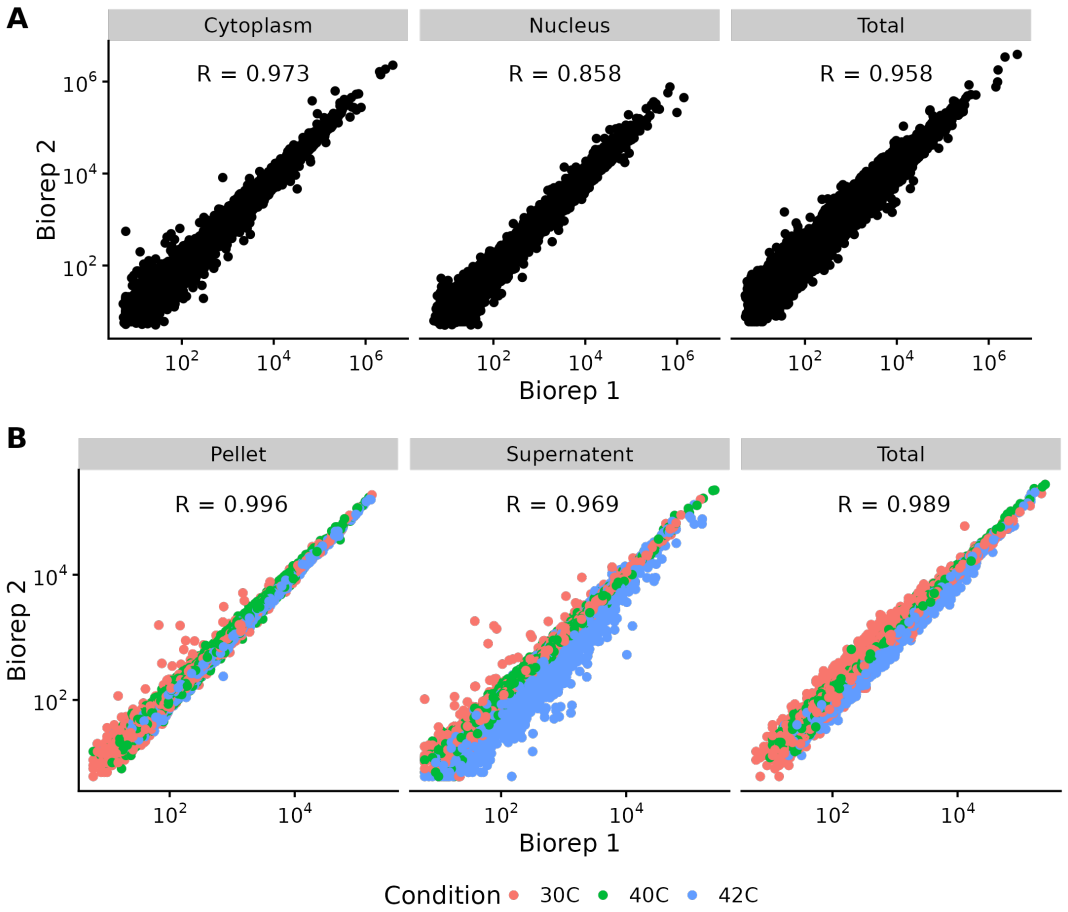


Figure 6.6: Overview of the two experimental data sets used in this study. **(A)** Correlation between two biological replicates of nuclear vs cytoplasmic mRNA transcript counts in a human lymphoblastoid cell line from the ENCODE project (Dunham et al., 2012). **(B)** Correlation between two biological replicates of mRNA transcript counts in heat-shock-induced stress granules vs freely floating in *Saccharomyces* cerevisiae cells. The dataset includes three temperature conditions: optimal 30C, mild 40C heat-shock, and extreme 42C heat-shock (Iserman et al., 2020).

transcriptomes were extracted and sequenced. Samples from the total transcriptome were taken before centrifugation and samples of the unbound transcriptome were taken using the supernatant post-centrifugation. Two highly correlated biological replicates are available on GEO with accession GSE131176, Figure 6.6B. The reads were already aligned to the S288C reference genome (release R64-2-1) and counted using the STAR aligner (Dobin et al., 2013). The *Saccharomyces* data set also includes a deletion strain in each of the three temperatures, but it is not used here.

6.2.7. Quantifying fractionation in the experimental data sets

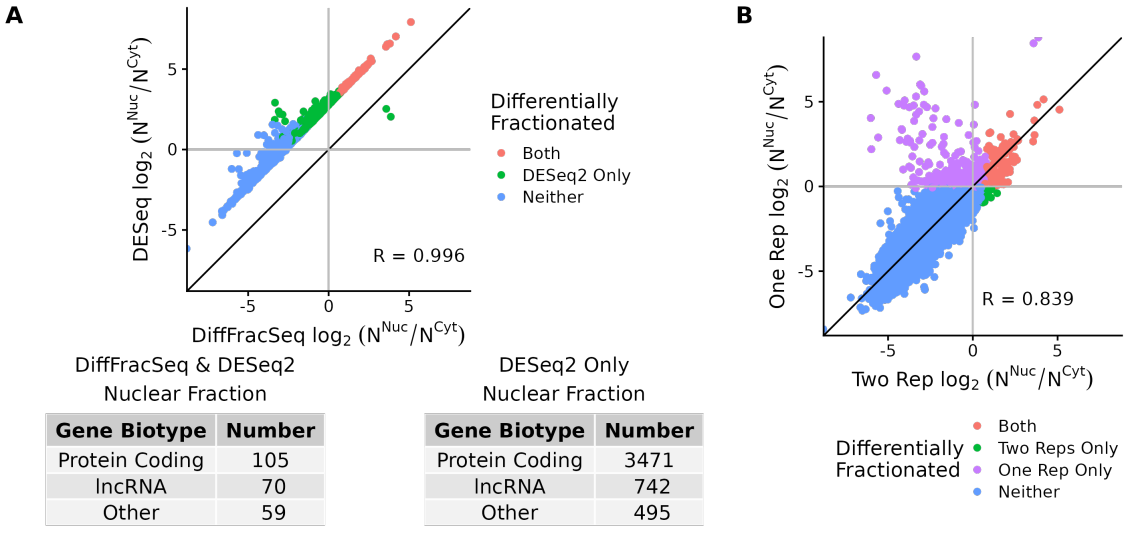
In contrast to DESeq2, DiffFracSeq trained on the ENCODE data set suggests the nuclear transcriptome is more selective and that it predominately consists of ncRNA. Similar to the results from the simulated data set, the correlation in $\log_2$ fraction ratios between the

two methods is high, but DiffFracSeq is able to suggest a more asymmetric distribution in total transcript counts between the two fractions, 6.7A. DiffFracSeq determines that over 90% of poly(A) tailed RNA transcripts are significantly localised to the cytoplasm, compared to 30% for DESeq2. Over half of the 234 genes that DiffFracSeq detects as nuclear localised are associated with ncRNA, according to the PANTHER database (Mi et al., 2013). DESeq2 detects almost 20 times more genes as fractionated to the nucleus, 70% of which are known to be mRNA.

The use of the total transcriptome sample as a quasi-replicate allows DiffFracSeq to detect differential fractionation without any experimental replicates. The results from training DiffFracSeq using one of the replicates in the ENCODE data set were compared to the results when using both. The results between using one and two replicates are correlated, $R = 0.84$ with 185 determined to be differentially expressed using either data set, 6.7B. However, 49 genes are only detected to be significantly localised to the nucleus when using both replicates and 367 additional genes are detected to be significantly localised when using just one replicate. A similar analysis is not available when using DESeq2 as it requires at least two replicates for each RNA-Seq sample.

DiffFracSeq can detect global changes in the stress granule transcriptome as temperatures increase. DESeq2 and DiffFracSeq were trained on the Iserman *et al* data set on the yeast stress granule transcriptome at 30°C, 40°C and 42°C. DiffFracSeq and DESeq2 have similar correlations in $\log_2$ ratio across the temperatures. However, there is a global shift in ratios between the two methods at 30°C that reduces as the temperature increase to 40°C and 42°C, Figure 6.7C. DiffFracSeq detects an increasing number of genes that are differentially fractionated to the stress granule across temperatures with 144, 956, and 2,174 genes selected at 30°C, 40°C, and 42°C respectively. DESeq2 detects a relatively consistent number of genes as differentially fractionated to the stress granule across temperatures with 1,952, 2,196, and 1,862 genes selected at 30°C, 40°C, and 42°C respectively.

The 40°C stress granule transcriptome as determined by DiffFracSeq lacks transcripts from genes that are crucial for fundamental cellular processes. A gene ontology analysis was conducted using PANTHER on genes detected to be differentially fractionated in the pellet at 40°C by DiffFracSeq or by DESeq2. Genes associated with primary metabolic processes, including those associated with processing organic substances and nitrogen compounds, are significantly underrepresented in the DiffFracSeq subset. The DESeq2 subset of genes included the same number of genes associated with primary metabolic processes as would be expected if the same number of genes were randomly sampled from the yeast genome. Instead, the DESeq2 subset was enriched with genes relating to localisation and transmembrane transport.

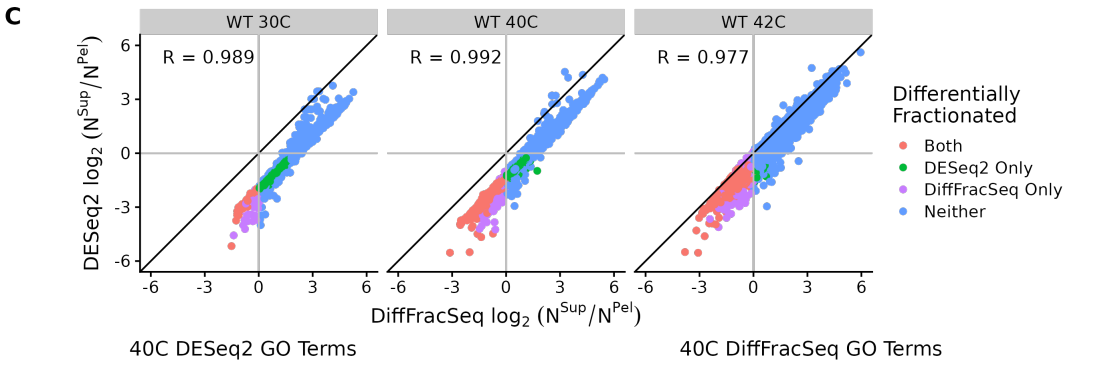


DiffFracSeq & DESeq2 Nuclear Fraction

Gene Biotype	Number
Protein Coding	105
lncRNA	70
Other	59

DESeq2 Only Nuclear Fraction

Gene Biotype	Number
Protein Coding	3471
lncRNA	742
Other	495



40C DESeq2 GO Terms

GO Term	Rep	FDR
transmembrane transport	+	1.25e-07
transposition	-	1.15e-04
localization	+	2.24e-04

40C DiffFracSeq GO Terms

GO Term	Rep	FDR
organic substance metabolic process	-	8.57e-08
primary metabolic process	-	3.57e-07
nitrogen compound metabolic process	-	4.16e-06

Figure 6.7: Comparison of DESeq2 and DiffFracSeq performance on two experimental data sets. **(A)** DESeq2 vs DiffFracSeq predicted $\log_2$ ratio of transcripts in the nucleus to the cytoplasm from the ENCODE data set. The colour denotes whether DiffFracSeq or DESeq2 consider the gene to be differentially fractionationed to the nucleus. The two tables show the associated biotype of genes considered differentially fractionated either by both methods or by DESeq2 only, as retrieved from PANTHERdb. **(B)** DiffFracSeq predicted $\log_2$ ratio of transcripts when trained with both or only one of the biological replicates in the ENCODE data set. **(C)** DESeq2 vs DiffFracSeq predicted $\log_2$ ratio of transcripts in the supernatant to the pellet from the Iserman *et al* data set. The wild type samples across three temperature conditions are shown across the columns. The colour denotes whether DiffFracSeq or DESeq2 considers the gene to be differentially fractionationed to the stress granule. The two tables show the top three terms in a gene ontology analysis conducted on genes that are considered differentially fractionated either by DESeq2 only or by DiffFracSeq only in the 40°C condition. The second column of each table denotes whether the GO term is overrepresented (+) or underrepresented (−) in the gene group.

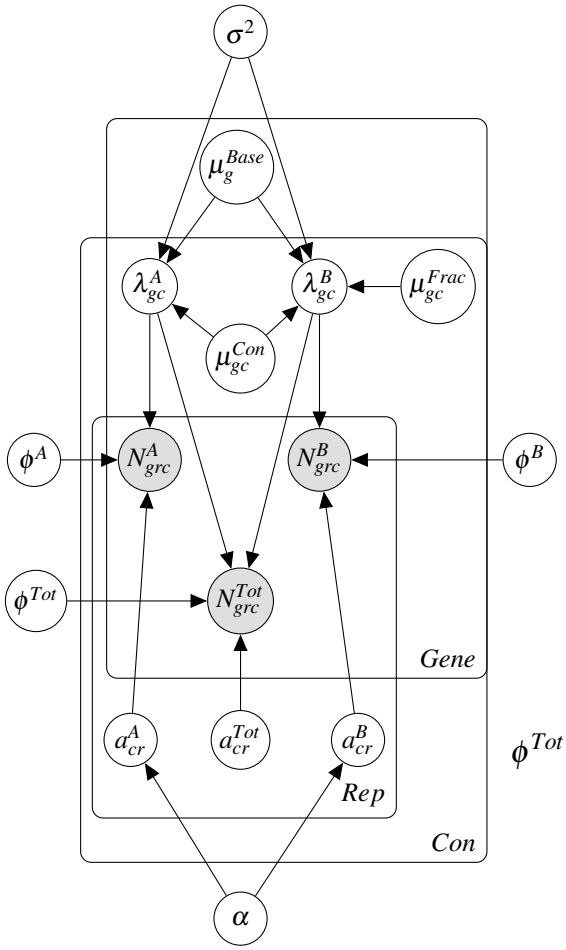
6.2.8. Extending DiffFracSeq to include a generalised linear model on transcript counts

DiffFracSeq's Bayesian model was extended to determine whether changes in transcript counts across conditions were due to changes in the overall RNA abundance in each fraction or changes in the ratio of RNA abundance between fractions. The original model enabled comparisons of transcript counts between fractions of the same sample by accounting for differences in the global transcriptomes of each fraction. However, the comparison of transcript counts from the same fraction across samples with different conditions, i.e. counts in stress granules across different temperatures, remains unsolved as the counts from each condition are normalised to different quasi-replicates, i.e. total transcript counts prior to centrifugation. This problem can be addressed by re-introducing the normalising assumption used by DESeq2 and applying it to total transcript counts across conditions. Assuming the majority of genes have constant transcripts counts in the total transcriptome across all conditions allows for the determination of batch-specific scale factors between samples and conditions. DiffFracSeq can then detect relative changes in the total expression of a subset of genes.

Decomposing changes in the transcript counts of a fraction across conditions was enabled by introducing a linear model of latent counts, λ^A, λ^B, with three terms: μ^{Base}, μ^{Con}, and μ^{Frac}. μ^{Base} is shared across all conditions and fractions and represents the base expression of a given gene. μ^{Con} is shared across fractions and represents the change in the overall expression of a given gene across conditions. μ^{Frac} is unique to fraction B for each condition and represents the change in fractionation of a given gene across conditions. μ^{Base} has a broad normal prior to enable the model to correctly determine the range of transcript abundances across a genome. The prior distributions for μ^{Con} and μ^{Frac} are normal distributions with zero mean to encourage the model to set parameters to zero if it believes there are no changes across conditions. The linear model with normal noise shares the same variance parameter, σ^2, across all conditions, fractions and genes. Finally, a hyperparameter, α, was added as the mean of the scale factor prior distribution to enable the model to find an appropriate average value for the scale factors given the high variability of this parameter across batches.

6.2.9. Detecting relative changes in fractionation and expression across conditions

In response to heat-shock, DiffFracSeq determines an increase in transcripts localised to stress granules and a global reduction in expression. Over 1/5 of genes have a significant increase in the fraction of transcripts found in stress granules over the change from 30°C to 40°C. Comparing 30°C to 42°C, even more genes are detected to have an increase in fractionation to stress granules, 2421 genes compared to 1407 genes at 40°C. As temper-



$$N_{grc}^{Tot} \sim negbin(a_{cr}^{Tot}(e^{\lambda_{gc}^A} + e^{\lambda_{gc}^B}), \phi^{Tot})$$
$$N_{grc}^{B} \sim negbin(e^{a_{cr}^B + \lambda_{gc}^B}, \phi^B)$$
$$N_{grc}^{A} \sim negbin(e^{a_{cr}^A + \lambda_{gc}^A}, \phi^A)$$

$$\lambda_{gc}^A \sim normal(\mu_g^{Base} + \mu_{gc}^{Con}, \sigma^2)$$
$$\lambda_{gc}^B \sim normal(\mu_g^{Base} + \mu_{gc}^{Con} + \mu_{gc}^{Frac}, \sigma^2)$$
$$\mu_g^{Base} \sim normal(7, 2)$$
$$\mu_{gc}^{Con} \sim normal(0, 1)$$
$$\mu_{gc}^{Frac} \sim normal(0, 1)$$
$$\sigma^2 \sim normal(2, 1)$$

$$\phi^{Tot}, \phi^A, \phi^B \sim normal(100, 10)$$
$$a_{cr}^A, a_{cr}^B \sim normal(\alpha, 0.1)$$
$$\alpha \sim normal(0, 0.5)$$

Figure 6.8: Plate diagram summarising the complete Bayesian hierarchical model behind DiffFracSeq. A generalised linear model of transcript counts across conditions was introduced by adding linear terms, μ_g^{Base}, μ_{gc}^{Con}, and μ_{gc}^{Frac}, to the means of the two latent count variables, λ_{gc}^A and λ_{gc}^B.

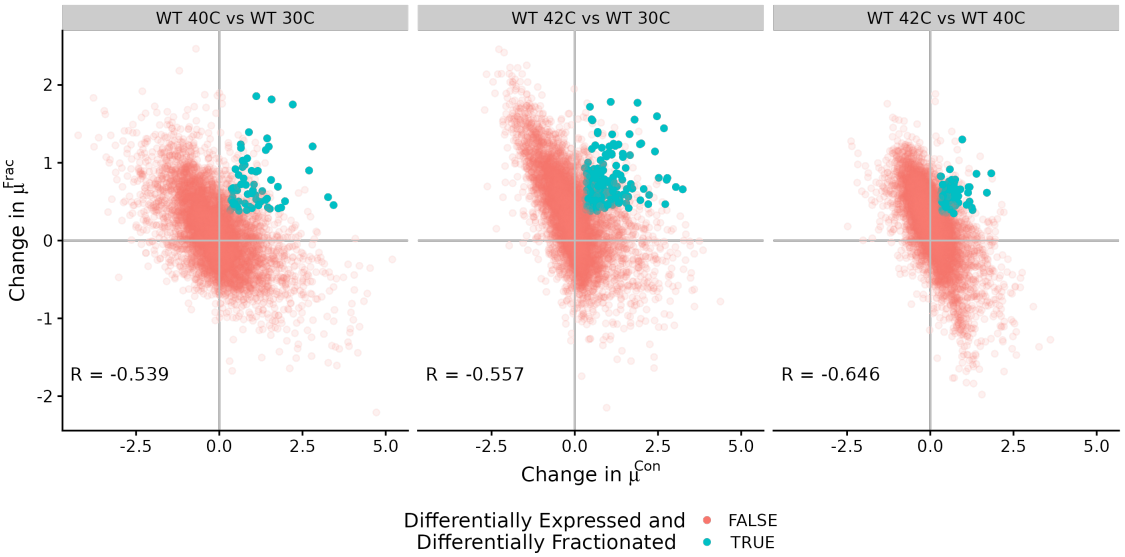


Figure 6.9: Detection of differential fractionation and differential expression across the Iserman data set. Changes in log_2 fold expression, μ^{Con}, vs changes in log_2 fold ratio between fractions, μ^{Frac}, between all pairs of the three conditions. Genes in the top left quadrant are overexpressed at the higher temperature and more localised to the stress granule. Genes in the bottom right quadrant are underexpressed in the higher temperature and are more localised to the cytoplasm. Genes that DiffFracSeq detects as differentially fractionated and differentially expressed across the pair of conditions are highlighted in blue.

ature increases from 30°C to 40°C DiffFracSeq detects 2373 genes in the yeast genome as significantly underexpressed, i.e. over 97.5% of $\mu^{Con}_{40°C}$ samples are significantly lower than all $\mu^{Con}_{30°C}$ samples for a particular gene. DiffFracSeq detects only 1945 genes as underexpressed for the transition from 30°C to 42°C. However, as DiffFracSeq is detecting nearly 1/3 of the yeast genome as differentially expressed across conditions the normalising assumption is likely invalid and conclusions about changes in total expression are unreliable.

DiffFracSeq uncovers general behaviours in genes in response to heat shock. Overall, there is a negative correlation between genes predicted to have an increase in overall expression and genes that change to be more concentrated in the supernatant, -0.539, -0.557, and -0.646. Four categories of behaviour in response to heat stress can be detected by DiffFracQuant: increase in total expression and increase in localisation to the stress granule, decrease in total expression and decrease in localisation to the stress granule, increase in expression and decrease in localisation, and decrease in expression and increase in localisation. For example, 191 genes at 42°C and 72 genes at 40°C are detected to increase in expression and fractionation to stress granules as temperature increases from 30°C 6.9.

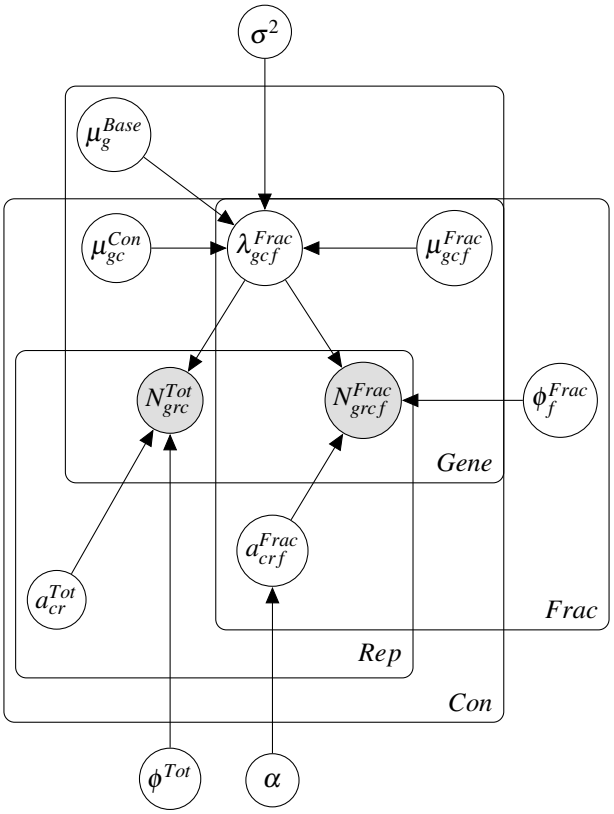


Figure 6.10: Plate diagram summarising an improved multi-fraction model for DiffFracSeq. The next iteration of the Bayesian hierarchical model behind DiffFracSeq will enable the normalisation and detection of differential fractionation in RNA-Seq experiments with more than two sub-fractions.

6.3. Conclusion

This chapter introduced DiffFracSeq as a novel Bayesian model for analysing RNA-Seq data and, to our knowledge, the only statistical model specifically designed to detect differential fractionation. The inclusion of pre-fractionation counts as a quasi-replicate to help normalise sub-fraction counts enables the Bayesian model to accurately determine batch-specific scale factors. The ability of the model to perform even with a single replicate data set and to extract changes in total transcript abundance as well as relative fractions from data sets with multiple conditions means it is a versatile tool in exploring fractionation data sets.

DiffFracSeq's has been shown to outperform DESeq2 in detecting differential fractionation using a simulated data set. DESeq2's inflated false positive rate is revealed when fractions contain global changes in their transcriptomes. Even in the 50%-50% random regime, when DESeq2's normalisation method successfully accounts for the batch-specific scale factors, DiffFracQuant has a better true positive rate when detecting significant differential fractionation than DESeq2. This behaviour is repeated across genes with

the lowest overall expression and the smallest effect sizes.

DiffFracSeq can determine asymmetric transcript abundances between fractions at scale with the human nuclear-cytoplasmic data set. Of the $\approx 16,000$ uniquely mapped genes, DiffFracSeq only determined 234 genes as having transcripts localised to the nucleus compared to 4708 genes detected by DESeq2. A recent study by Zaghlool *et al* also analysed the GM12878 cell line, together with three other human cell lines, and similarly determined 4500 transcripts that are localised to the nucleus using DESeq2, see (Zaghlool et al., 2021) Supplementary Figure 2. However, DiffFracSeq's result correlates with the original ENCODE analysis that placed the majority of protein coding transcripts in the cytoplasm without using DESeq2, (Djebali et al., 2012) Figure 3.

DiffFracSeq can determine relative changes in fractionation across conditions in a yeast stress granule data set. Gene ontology analysis of the heat stress granule transcriptome according to DiffFracSeq shows it lacks key transcripts associated with key metabolic processes. However, the same analysis using the DESeq2 transcriptome shows enrichment for transmembrane proteins in contradiction to other stress granule studies (Unworth et al., 2010; Khong et al., 2017). As temperature increases, DiffFracSeq detects more genes as differentially fractionated to the stress granule, but DESeq2 does not have a clear pattern. DiffFracSeq also detects a correlation between genes that are overexpressed under stressed conditions and increase the fraction of their transcripts in the cytoplasm. However, further exploration of changes in total expression is limited by the breaking of the underlying normalising assumption used to compare expression across conditions.

DiffFracSeq can only determine relative changes in RNA abundance in each fraction and cannot estimate overall changes in total RNA abundance per cell. The model can be enhanced to allow normalisation using external RNA spike-ins so that changes in total RNA abundance per cell can be measured across conditions. The model behind DiffFracSeq can be further expanded by modelling counts from more than two fractions, Figure 6.10. The linear sum of sub-fraction counts can theoretically be extended to include any number of sub-samples $N^{Tot} = \sum_i N^i$. Enabling more fractions to be normalised could unlock more use cases for the software as it could also be used to characterise libraries of constructs by comparing fractions from cell sorting assays. Entire libraries of synthetic constructs can be characterised by sorting pools of constructs by some desired characteristic, for example high protein fluorescence, which are subsequently sequenced (Sharon et al., 2012). Alternatively, the linear model predicting latent transcript counts could be improved by letting users define their own design matrices. The latent count linear model is fixed to determine separate coefficients for each condition, but users may be interested

in exploring interactions between conditions. For example, the Iserman *et al* data set also includes stress granule samples from a mutant strain across the three temperatures which could be directly compared to the wild type. This assumption enabled DiffFracSeq to detect differential expression and fractionation across conditions, but this functionality is not applicable in many cases.

The software containing the two-fraction model is available to download from GitHub, github.com/DimmestP/ DiffFracSeq, and has been shown to be successful at investigating localisation across organisms and subcellular fractions. DiffFracSeq has the functionality to improve the quality of experiments across biology from the uncovering of novel regulatory mechanisms in fundamental cell biology to the characterisation of constructs libraries in synthetic biology.

Chapter 7

Discussion

Technology developed over the last 40 years has enabled investigations into the regulatory mechanisms that cells use to control their gene expression at unprecedented resolutions. Meanwhile, recent trends in molecular biology continue to emphasise the use of high-throughput experiments and the application of novel machine learning analysis methods. Massively parallel reporter assays using multiplexed RNA-Seq and flow cytometry techniques are allowing researchers to test the expression levels of hundreds of thousands of constructs (Sharon et al., 2012; Shalem et al., 2015; Klein et al., 2020). Google Deep Minds' AlphaFold changed expectations of machine learning applied to complex biological problems with the accuracy and scale of its protein folding predictions (Jumper et al., 2021). In transcriptomics, deep-learning techniques continue to be successfully applied to RNA degradation prediction with increasing accuracy (Wayment-Steele et al., 2022; He et al., 2023).

Investigations with larger data sets and more extensive analysis methods are discovering subtle mechanisms that are easily confounded by experimental noise. As the volume and detail of data sets continue to increase the likelihood of mistaking experimental noise for biological phenomena also increases. Inadequate analysis of noisy data sets is contributing to a reproducibility crisis across biology. This thesis shows how quality research software can support rigorous statistical methods and improve the design and conclusion of experiments.

In chapter 4, the R package tidyqpcr was introduced as an open-source R package for the analysis of qPCR data. The widespread usage of qPCR across biology and medicine was described as well as the lack of understanding of best practices in the design and analysis of qPCR experiments. The current software landscape was reviewed showing there are some open-source packages with extensive functionality, but their steep learning curves lead users to search for alternatives. tidyqpcr was shown to combine best practices in qPCR experimental design as outlined by the MIQE-guidelines together with the latest

developments in data analysis provided by the R tidyverse. The development of extensive documentation together with feedback from user interviews and an rOpenSci code review has ensured that tidyqpcr is accessible to prospective users with varying coding experiences. tidyqpcr is a freely available tool for analysing qPCR data that is downloadable from GitHub, github.com/ropensci/tidyqpcr/, with an associated publication in doi:10.21105/joss.04507 JOSS.

Chapter 5 described the use of tidyqpcr together with the rigorous analysis of published data sets and the integration of data from multiple experiments to detect subtle interactions between regulatory elements. We showed that the terminator sequence of mRNA transcripts can have different contributions to protein fluorescence when paired with different promoters and ORFs. Furthermore, we showed that context-dependent contributions to gene expression can also be detected for short cis-regulatory elements. We selected motifs within the 3'UTR sequence of terminators and showed that they also express different contributions to gene expression when inserted into different constructs terminators and paired with different promoters. The contributions of the chosen motifs were further supported by poly(A) anchored RNA-Seq and comparison of construct 3'UTR sequences.

The results from chapter 5 contribute to the growing evidence for a more complex picture of cis-regulatory elements with consequences for computational and synthetic biology. Motifs that are dependent on other sequences have previously been detected by high-resolution maps of protein-RNA interactions. These maps have discovered gapped, multi-partite motifs (Olivier et al., 2005) and motifs that must be repeated in the same transcript to be effective (Gu et al., 2004; Jackson et al., 2004). Therefore, computational methods to find motifs, such as the MEME Suite (Bailey et al., 2015), need to include motifs that: are varied in length, contain gaps between conserved sequences, do not act independently, and/or occur more than once per transcript (Frith et al., 2008). In synthetic biology, the unpredictability in the expression of combinations of otherwise well-characterised regulatory elements has led some experiments to depend on time-consuming directed evolution assays to overcome mis-matches in component expression levels (Yokobayashi et al., 2002). Developing methods that introduce pools of combinations of suitable CREs when characterising synthetic libraries can help design more reliable synthetic pathways (Kosuri et al., 2013).

In chapter 6, a Bayesian hierarchical model was introduced that can rigorously detect differential fractionation if combined with appropriate experimental design. We began by outlining the assumptions of normalising techniques that enable the detection of differential expression despite known biases in RNA-Seq. These assumptions are shown to break down when analysing data from fractionation-based RNA-Seq assays that in-

vestigate the localisation of RNA transcripts. DiffFracSeq was described as a Bayesian hierarchical model that normalises and detects differential fractionation without the assumptions of other techniques. The performance of DiffFracSeq is shown to outperform DESeq2 on three data sets when detecting changes in fractionation. DiffFracSeq is a freely available tool for normalising RNA-Seq data sets that are investigating RNA transcript localisation and is downloadable from GitHub github.com/DimmestP/DiffFracSeq

7.1. Future Work

tidyqpcr is a fully functional qPCR analysis package that has contributed to the research of several labs. However, the package is missing functionality that would extend its application and contribute to its overall aim of removing any dependence on proprietary software. Adding functions to read alternative qPCR data file formats, calculate Cq values directly from amplification curves and enable the analysis of qPCR assays other than SYBR Green remain priorities. There also remains work to be done on promoting its comprehensive documentation as a teaching resource both for conducting reproducible analysis and for implementing MIQE-compliant qPCR experimental design. The development and organisation of tidyqpcr workshops inspired by the widely successful Carpentries workshop for coding and data science will help grow tidyqpcr's user base.

The limitations of composability of cis-regulatory elements can be further explored through the creation of a larger construct library which can be characterised using high-throughput gene expression assays. Several questions remain about the composability of the four 3'UTR motifs explored in chapter 5. First, is the behaviour of these motifs observed in the three host terminators representative of their behaviour across the yeast genome? Second, are positional effects changing motif behaviour, particularly with respect to distance from the poly(A) tail? Finally, the design of more constructs that include multiple motifs together could uncover new interactions between motifs. The extended construct library could be characterised by high-throughput flow cytometry to determine protein fluorescence and multiplexed RNA-Seq to determine transcript abundance.

DiffFracSeq is currently able to analyse experiments with two fractions and two conditions. The first development of DiffFracSeq would be to allow normalisation using external RNA spike-ins so that total expression levels can be compared across conditions. DiffFracSeq could be further extended by enabling more complex design matrices. The inclusion of more than two fractions would enable it to be applied to more experimental assays. Furthermore, allowing design matrices that facilitate interactions between multiple conditions and between fractions will enhance the quality of the conclusions that can be deduced with DiffFracSeq. The implementation of the R package that contains the DiffFracSeq model needs further development. The function documentation needs to be

enhanced and a vignette describing a typical DiffFracSeq workflow remains to be written. Investigating the default priors and the method for posterior sampling could also lead to a reduced run time.

Appendix A

A.1. tidyqpcr User Questionnaire

A.1.1. Subject questionnaire

Estimated Time : 20 Minutes

Tell me about your experience with qPCR experiments.

Question notes:

- RNA or DNA qPCR ?
- primers only or probe sets?
- How many qPCR experiments in last year/two?
- How many planned in next 6 months?
- How difficult would it be to reanalyse/repeat your own qPCR experiment?

Describe how qPCR experiments are used/presented in published papers related to your research.

Question notes:

- Recount a time where you questioned results/conclusion from qPCR experiments
- Did you attempt to reanalyse/recreate their qPCR data?

Describe the design of your most recent qPCR experiment.

Question notes:

- Technical/biological/experimental replicates?
- Plate design?
- Software to design plate (excel?)
- Methodology for ordering samples
- Number of wells?
- Typical number of probes?

- Primer efficiency calculation?
- MIQE best practises qPCR guidelines?
- How did you load your plates - single-channel pipette, multichannel pipette, electronic or manual, automatic loading with what robot?
- What qPCR instrument did you use?
- How do you tell if your experiment worked - what do you do for quality control?

Describe the analysis pipeline of your most recent RT-qPCR experiment.

Question notes:

- GUI / Terminal / R based?
- Proprietary software?
- See, understand and repeat every step?
- Customisable, paper ready output graph?
- Whats the biggest frustration? (is there something you know you should be doing but don't)
- Would it be easy to redo an experiment (because something went wrong) using the same analysis?
- The features you require from qPCR software

What is your previous R programming / terminal experience?

Question notes:

- Previous courses?
- Previous obstacles?
- Familiar with the concept of tidy data?
- Interest in learning?

A.1.2. tidyqpcr worksheet

Estimated Time : 40 minutes

Follow installation instructions on github.com/ewallace/tidyqpcr.

Read through the vignette on plate designing

Create a example plate design for the following experiment:

- 8 by 12 well plate
- Three Biological Replicates
- Three Technical Replicates + “-RT” control
- One strain: “WT”
- Two conditions: + and - “menadione”
- Four probes: "PGK1","ALG9", " HHT2", "HTB2"

Read through the instructions on conducting qPCR analysis with tidyqpcr in the multifactor vignette

Load in the example plate plan using data (tidyqpcr_plateplan) and associated experimental data.

Normalise raw data and produce plot of differential expression under two stresses.

A.2. Software review table

	Platform	Normalisation	Method of Cq determination	Quantification method	Plots summary statistics
QuantGenius	Web	To ref genes	N/A	Absolute (Standard Curve)	No
ELIMU-MDx	Web	To ref genes	N/A	Absolute (Standard Curve) and Relative	No
shinyCurves	Web/R	N/A	N/A	Relative	No
PIPE-T	Web	Numerous methods (from HTqPCR)	N/A	Relative	No
SATqPCR	Web	To ref genes	N/A	Relative	Yes
Auto-qPCR	Web/Python	To ref genes	N/A	Absolute (Standard Curve) and Relative	Yes
Chainy	Web/R	To ref genes	Several methods to determine gradient of Amp curve	Relative	Yes
LEMming	R	Linear Error Mode	N/A	Relative	No
pcr	R	To ref genes	N/A	Relative (delta Cq or standard curve)	Yes
HTqPCR	R	Ref genes, quartile mean, rank-invariant normalising feature	N/A	Relative	Yes
ReadqPCR/ NormqPCR	R	To ref genes	N/A	Relative	No
qpcR	R	Numerous methods	Fitting multi parameter logistic curve	Relative and absolute	Yes
qpcr	python	To ref genes	N/A	Relative	Yes
Spreadsheet	Misc	To ref genes	N/A	Relative	No

	Scalability	Summary statistics	Outlier identification	Calculate primer calibration	Use primer efficiency
QuantGenius	Requires copy and pasting input	No	Yes	No	No but filters based on it)
ELIMU-MDx	If you can create RDML files	No	No	No	No
shinyCurves	If in correct excel format and sample NT/control positions	No	Manual	No	No
PIPE-T	If files in tab separated format	two condition tests	Yes	No	No
SATqPCR	If files in tab separated txt format	t-test or ANOVA	No	No	Yes
Auto-qPCR	If input files in right format	t-test or ANOVA	Yes	No	No
Chainy	Manual upload of files	permutation approach equivalent to the REST software	Yes	Yes	Yes
LEMming	If imported into R	t-test and Wilcoxon signed-rank test	No	No	No
pcr	If imported into R	t-test, ANOVA and signed-rank test	No	Yes	No
HTqPCR	Yes	t-test, Mann-Whitney Test and limma package linear models	Yes	No	No
ReadqPCR/ NormqPCR	Yes	No	No	No	No
qpcR	Yes	F-test for model selection	Yes	Yes	Yes
qpcr	If input files in right format	No	Yes	Yes	Yes
Spreadsheet	No	t-test	No	No	Yes

	QC	Reproducible	Copy number	Input
QuantGenius	No melt/amp curve. Highlights outliers, outside LOD and efficiency	No	Yes	Copy and paste each target and reference gene separately
ELIMU-MDx	No melt/amp curves. Checks LOD, efficiency and control Cq	If you can host it	No	RDML, excel
shinyCurves	Both melt/amp curve with qpcR	Yes (If you identify the same outliers)	Yes	csv, (custom) xlsx, xls
PIPE-T	No melt/amp curve. Highlights outliers, outside LOD and efficiency	Yes	No	tsv
SATqPCR	No	Yes	No	txt
Auto-qPCR	Filters out samples with SD cutoff. No melt/amp curve	Yes	Yes	csv, txt
Chainy	Plots amp curve and highlight outliers that dont fit sigmoidal	Yes	No	RDML, csv, raw qPCR machine excel, plate plan
LEMming	No	Yes	No	R data.frame
pcr	No	Yes	No	R data.frame
HTqPCR	No melt/amp curve. Functions to plot wells, conduct PCA, calc variation. Can flag High Cq value and high variable samples	Yes	No	raw qPCR machine excel
ReadqPCR/ NormqPCR	No melt/amp curve. Functions to pairwise results across replicates/plates. Can impute missing values. flag High Cq value and high variable samples	Yes	No	raw qPCR machine excel
qpcR	Yes	Yes	No	R data.frame
qpcr	Filters out samples with SD cutoff. No melt/amp curve	Yes	No	csv, (custom) excel
Spreadsheet	Filters out samples with SD cutoff. No melt/amp curve	No	No	(custom) excel

	Output	GUI	Last update	Release date	Number of wells	Normalising gene selection
Quant Genius	txt, xls	Yes	2017 Feb	2017 Feb	Unlimited	No
ELIMU-MDx	RDML, excel	Yes	2020 Dec	2019 Oct	Unlimited	No
shinyCurves	csv, png	Yes	2021 Oct	2021 Oct	96 or 364	No
PIPE-T	tsv, PNGs	Yes	2019 Nov	2019 Nov	Unlimited	No
SATqPCR	txt, png	Yes	2019 Aug	2019 Aug	Unlimited	stability parameter and coefficient of variation
Auto-qPCR	csv, png	Yes	2021 Oct	2021 Oct	Unlimited	No
Chainy	csv, png	Yes	2020 Aug	2017 May	Unlimited	Yes (geNorm method from NormqPCR)
LEMming	R data.frame	No	2015 Sept	2015 Sept	Unlimited	No
pcr	R data.frame and plots	No	2020 April	2018 May	Unlimited	No
HTqPCR	R S4 object and plots	No	N/A	2009 Dec	Unlimited	No
ReadqPCR/ NormqPCR	R S4 object and plots	No	2018 July	2012 Jul	Unlimited	Yes (geNorm or NormFinder)
qpcR	R S3 object and plots	No	2018 June	2008 July	Unlimited	Yes
qpcr	txt, jpg	No	2022 Feb	2021 Aug	Unlimited	No
Spreadsheet	excel	No	N/A	N/A	Unlimited	No

Appendix B

B.1. Chapter 5 Supplementary Tables

mCherry

pRPS3

Terminator	log2 fold change	p.adj
tPMA1	-1.8667482	0.0e+00
tSUN4	-1.3307405	0.0e+00
tSRO9	-1.1145008	0.0e+00
tTOS6	-1.0278710	0.0e+00
tHSP26	-0.4661509	4.9e-05
tRPS13	0.5234415	1.5e-05
tCLN2	0.5464283	1.1e-05
tPAB1	0.6018695	2.2e-06
tRPS3	0.8958268	0.0e+00

pHSP26

Terminator	log2 fold change	p.adj
tPMA1	-1.6289649	0.0e+00
tSUN4	-1.2574579	8.0e-07
tSRO9	-1.1746048	3.4e-06
tTOS6	-0.5805808	1.1e-01
tRPS13	-0.1965647	1.0e+00
tCLN2	-0.1252409	1.0e+00
tPAB1	-0.0616474	1.0e+00
tHSP26	0.0319040	1.0e+00
tRPS3	0.1164520	1.0e+00

pRPS13

Terminator	log2 fold change	p.adj
tPMA1	-2.0853624	0.0e+00
tSUN4	-1.2491007	0.0e+00
tSRO9	-1.0162570	0.0e+00
tTOS6	-0.9791352	0.0e+00
tHSP26	0.3061052	7.2e-03
tRPS13	0.4382561	4.1e-04
tCLN2	0.4762219	2.0e-04
tPAB1	0.5324084	4.7e-05
tRPS3	0.5549583	2.9e-05

pPGK1

Terminator	log2 fold change	p.adj
tPMA1	-1.8971163	0.00
tSRO9	-1.3729199	0.00
tSUN4	-1.0917919	0.00
tTOS6	-0.9565605	0.00
tRPS13	-0.2020877	0.14
tRPS3	0.0381100	1.00
tPAB1	0.0650108	1.00
tHSP26	0.0881576	1.00
tCLN2	0.2156118	0.12

Supplementary Table B1: Tables showing changing contributions to gene expression from terminators paired with different promoters and coding sequences. log_2 fold change is calculated with respect to the PGK1 terminator of each promoter-coding sequence set. p.adj values are FDR-adjusted t-test pvalues.

mTurq

pRPS3

Terminator	log2 fold change	p.adj
tPMA1	-2.4173566	0.000
tSUN4	-1.8362856	0.000
tTOS6	-1.2512970	0.000
tSRO9	-0.7268833	0.000
tRPS13	-0.0881267	0.330
tHSP26	0.1310147	0.300
tCLN2	0.2638312	0.015
tRPS3	0.3529039	0.001
tPAB1	0.6543571	0.000

pHSP26

Terminator	log2 fold change	p.adj
tPMA1	-1.5432817	8.4e-06
tSUN4	-1.2474233	3.0e-04
tSRO9	-1.0514050	2.7e-03
tTOS6	-0.6661552	1.2e-01
tHSP26	-0.5293229	3.0e-01
tRPS13	-0.3155979	1.0e+00
tCLN2	-0.1610090	1.0e+00
tRPS3	-0.0913532	1.0e+00
tPAB1	0.1215476	1.0e+00

pRPS13

Terminator	log2 fold change	p.adj
tPMA1	-2.0420263	0.0e+00
tSUN4	-1.7210314	0.0e+00
tTOS6	-0.8907257	0.0e+00
tSRO9	-0.5765566	3.0e-07
tRPS13	0.0611790	5.0e-01
tHSP26	0.1966032	6.8e-02
tRPS3	0.2642338	1.5e-02
tCLN2	0.2901645	9.0e-03
tPAB1	0.5886753	2.0e-07

pPGK1

Terminator	log2 fold change	p.adj
tPMA1	-2.7277132	0.0000
tSUN4	-1.5208626	0.0000
tSRO9	-1.3242251	0.0000
tTOS6	-0.8517308	0.0000
tRPS13	-0.3294545	0.0390
tHSP26	-0.3162353	0.0390
tRPS3	-0.1138039	0.7200
tCLN2	-0.0457990	0.7200
tPAB1	0.4671738	0.0019

Supplementary Table B2: Tables showing changing contributions to gene expression from terminators paired with different promoters and coding sequences. log_2 fold change is calculated with respect to the PGK1 terminator of each promoter-coding sequence set. p.adj values are FDR-adjusted t-test pvalues.

CRE	Individual	Drop	Cumulative
Codon Usage	42.0%	-40.5%	42.0%
Motifs	3.2%	-1.6%	43.7%
3'UTR Length	0.6%	0.0%	43.7%

Supplementary Table B3: Variance explained by each type of CRE in the half-life model applied to data from Chan et al (2018). Variance explained was estimated in 3 different ways. The individual variance explained by a linear model containing each type of CRE on its own. The drop in variance explained when one type of CRE is removed from the full model. The cumulative variance explained when each CRE is added to the linear model in sequence; starting with codon usage, then adding motifs and finally 3'UTR length.

mCherry construct sequences are available on the manuscript GitHub repo.
https://github.com/DimmestP/chimera_project_manuscript/tree/main/supplementary_data_chapter/data/mCherry_fluorescence_reporter_dna_sequences.csv

Supplementary Table B4: Table showing the DNA sequences for all mCherry reporter constructs.

mTurq construct sequences are available on the manuscript GitHub repo
https://github.com/DimmestP/chimera_project_manuscript/tree/main/supplementary_data_chapter/data/mTurq_fluorescence_reporter_dna_sequences.csv

Supplementary Table B5: Table showing the DNA sequences for all mTurq reporter constructs.

Primer name	Primer sequence	Purpose
mCh_F7	AGGACGGCGAGTT CATCTA	qPCR primer for mCherry ORF
mCh_R7	CCCATGGTCTTCTT CTGCATTA	qPCR primer for mCherry ORF
RPS3_F1	TCGCTGACGGTGT CTTCTACG	qPCR primer for RPS3 ORF
RPS3_R1	TCGGTCTTGGTTGGA GTGACA	qPCR primer for RPS3 ORF
RPS13_EF	CTAGAAATGCTCCAGC TTGGTTCAA	qPCR primer for RPS13 ORF
RPS13_ER	TCAAACCCTTTCTCG CGTACTTG	qPCR primer for RPS13 ORF
PGK1_F2	GCTGCTTTGCCAAC CATCAA	qPCR primer for PGK1 ORF
PGK1_R2	TCGTTTCTTTCACCG TTTGGTC	qPCR primer for PGK1 ORF
mTu_F2	TTGGGGTGTTCAATG TTTTGC	qPCR primer for mTurq2 ORF
mTu_R2	TGAACATAACCTTCT GGCATGG	qPCR primer for mTurq2 ORF

Supplementary Table B6: Primer sequences created for all qPCR experiments.

Motif	Total Count	Coef	p.value
CTTCATTTC	14	-0.52197	0.00474
ATTCATTTC	22	-0.44858	0.00577
TAGCATTTT	19	-0.43109	0.00934
TTGCATTTT	46	-0.27615	0.01088
CTGCATTTT	15	-0.47916	0.01162
TTTCATTTC	42	-0.27049	0.01421
AAACATTTC	13	-0.46682	0.01904
CTGCATTAT	10	-0.53910	0.02027
TTTCATTTT	103	-0.14469	0.03671
CTGCATTTC	6	-0.63943	0.03775
TTACATTAC	18	-0.43018	0.03918
TTCCATTAT	15	0.37314	0.04106
ATGCATTTT	31	-0.26015	0.04115
CATCATTAT	16	-0.38091	0.04879
ATTCATTAT	39	-0.22684	0.04928

Supplementary Table B7: Selecting the HWNCATTWY motif, TTTCATTTC. Summary of the number of occurrences and contributions to a linear model predicting half-life for each possible version of the HWNCATTWY motif.

Motif	Total Count	Coef	p.value
TGTATATTA	83	-0.51920	0.00000
TGTATCATA	17	-0.67490	0.00004
TGTATAATA	72	-0.27914	0.00066
TGTACCATA	6	-0.98705	0.00115
TGTACACTA	16	-0.56750	0.00373
TGTACAATA	27	-0.37788	0.00551
TGTAACATA	19	-0.44503	0.00687
TGTATACTA	36	-0.30274	0.00863
TGTACATTA	26	-0.29426	0.02791

Supplementary Table B8: Selecting the UGUAHMNUA motif, TGTACAATA. Summary of the number of occurrences and contributions to a linear model predicting half-life for each possible version of the UGUAHMNUA motif.

Terminator	Construct	Minimum free energy (kcal/mol)
RPS3	WT	-6.10
RPS3	mod_NNN	-11.50
RPS3	mod_NTN	-13.00
RPS3	mod_NAA	-7.40
RPS3	mod_NGG	-10.80
RPS3	mod_HNH	-5.50
RPS3	mod_HTH	-7.40
TSA1	WT	-6.90
TSA1	mod_NNN	-10.14
TSA1	mod_NTN	-10.60
TSA1	mod_NAA	-8.90
TSA1	mod_NGG	-9.80
TSA1	mod_HNH	-6.10
TSA1	mod_HTH	-6.67
PIR1	WT	-28.00
PIR1	mod_ANHHH	-30.60
PIR1	mod_NTHHH	-27.70
PIR1	mod_ATNHH	-28.40
PIR1	mod_ATHNH	-27.00
PIR1	mod_ATNNN	-30.40
PIR1	mod_ANNNN	-32.60
PIR1	mod_NTNNN	-30.10

Supplementary Table B9: Table showing minimum free energies of 3'UTR constructs with inserted/deleted motifs.

mod_NAA

Promoter	Terminator	Fold change	p.adj
pRPS3	tRPS3	0.2541415	8.0e-07
pSRO9	tRPS3	0.4656789	1.0e-03
pPGK1	tRPS3	0.5245889	1.5e-03
pTSA1	tTSA1	0.6826257	3.0e-02
pPGK1	tTSA1	0.8142880	1.2e-01
pSRO9	tTSA1	1.0662162	4.2e-01

mod_NTN

Promoter	Terminator	Fold change	p.adj
pTSA1	tTSA1	0.4065174	3.7e-05
pPGK1	tTSA1	0.5367128	1.5e-03
pSRO9	tTSA1	0.5755669	4.3e-04
pRPS3	tRPS3	0.8833831	2.7e-01
pSRO9	tRPS3	0.9947510	9.5e-01
pPGK1	tRPS3	1.0568824	8.1e-01

mod_HNH

Promoter	Terminator	Fold change	p.adj
pTSA1	tTSA1	0.6247665	0.0029
pPGK1	tRPS3	0.6645988	0.0150
pPGK1	tTSA1	0.6954945	0.0150
pRPS3	tRPS3	0.7238204	0.0210
pSRO9	tRPS3	0.8151770	0.0240
pSRO9	tTSA1	0.8734047	0.4000

mod_HTH

Promoter	Terminator	Fold change	p.adj
pTSA1	tTSA1	0.2786772	6.0e-07
pRPS3	tRPS3	0.3352986	1.0e-04
pSRO9	tRPS3	0.4865148	2.9e-04
pSRO9	tTSA1	0.4909374	1.1e-03
pPGK1	tTSA1	0.5889067	1.5e-03
pPGK1	tRPS3	0.5891336	1.8e-03

mod_NGG

Promoter	Terminator	Fold change	p.adj
pTSA1	tTSA1	0.8299585	0.1100
pPGK1	tRPS3	1.0221923	0.8500
pPGK1	tTSA1	1.0470925	0.8100
pSRO9	tTSA1	1.1112799	0.3700
pRPS3	tRPS3	1.1957891	0.2700
pSRO9	tRPS3	1.5013874	0.0017

WT

Promoter	Terminator	Fold change	p.adj
pPGK1	tTSA1	0.7338775	0.130
pTSA1	tTSA1	0.7371346	0.110
pPGK1	tRPS3	0.9530783	0.830
pSRO9	tTSA1	1.0705360	0.620
pRPS3	tRPS3	1.3474065	0.033
pSRO9	tRPS3	1.4012037	0.093

Supplementary Table B10: Tables showing fold changes in transcript abundance of promoter-terminator constructs with different inserted motifs. Tables are sorted by fold change which is calculated with respect to the mod_NNN construct of that promoter-terminator pairing.

mod_NTHHH

Promoter	Terminator	Fold change	p.adj
pSRO9	tPIR1	1.096403	0.4200
pPGK1	tPIR1	1.165855	0.0500
pPIR1	tPIR1	1.399047	0.0038

mod_ANHHH

Promoter	Terminator	Fold change	p.adj
pPGK1	tPIR1	1.379786	0.01500
pPIR1	tPIR1	1.536283	0.00380
pSRO9	tPIR1	1.725748	0.00045

mod_ATNHH

Promoter	Terminator	Fold change	p.adj
pPGK1	tPIR1	1.021209	0.830
pSRO9	tPIR1	1.134849	0.400
pPIR1	tPIR1	1.231856	0.012

mod_ATHNH

Promoter	Terminator	Fold change	p.adj
pSRO9	tPIR1	0.9330330	0.73
pPGK1	tPIR1	0.9777246	0.81
pPIR1	tPIR1	1.0788127	0.67

mod_ATNNN

Promoter	Terminator	Fold change	p.adj
pSRO9	tPIR1	1.161598	0.230
pPIR1	tPIR1	1.209994	0.180
pPGK1	tPIR1	1.267952	0.018

mod_ANNNN

Promoter	Terminator	Fold change	p.adj
pSRO9	tPIR1	1.482810	0.0200
pPGK1	tPIR1	1.599674	0.0015
pPIR1	tPIR1	2.126692	0.0002

mod_NTNNN

Promoter	Terminator	Fold change	p.adj
pSRO9	tPIR1	1.287386	0.0460
pPGK1	tPIR1	1.578562	0.0004
pPIR1	tPIR1	1.685034	0.0038

Supplementary Table B11: Tables showing fold changes in transcript abundance of promoter-terminator constructs with different deleted motifs. Tables are sorted by fold change which is calculated with respect to the WT construct of that promoter-terminator pairing.

ATATTC

Promoter	Terminator	log2 fold change	p.adj
pRPS3	tRPS3	-0.9881481	0.0e+00
pSRO9	tRPS3	-0.5512963	2.4e-05
pPGK1	tRPS3	-0.4653704	5.2e-05
pPIR1	tPIR1	-0.4017119	1.3e-03
pPGK1	tPIR1	-0.3524806	3.5e-04
pTSA1	tTSA1	-0.2754167	7.3e-03
pSRO9	tPIR1	-0.1625775	2.1e-01
pPGK1	tTSA1	-0.1481944	1.6e-01
pSRO9	tTSA1	0.0462500	6.8e-01

TGTAHMNTA and HWNCATTWY

Promoter	Terminator	log2 fold change	p.adj
pRPS3	tRPS3	-0.4656481	0.0015
pSRO9	tRPS3	-0.3685185	0.0210
pPGK1	tRPS3	-0.1268519	0.3900
pPIR1	tPIR1	-0.0872287	0.2800
pSRO9	tTSA1	-0.0170833	0.9100
pTSA1	tTSA1	0.0669444	0.6700
pPGK1	tPIR1	0.0843282	0.1900
pSRO9	tPIR1	0.1539836	0.0760
pPGK1	tTSA1	0.3288889	0.0200

TGTAHMNTA

Promoter	Terminator	log2 fold change	p.adj
pTSA1	tTSA1	-1.2986111	1.0e-07
pPGK1	tTSA1	-0.8977778	5.2e-05
pSRO9	tTSA1	-0.7969444	3.3e-04
pPIR1	tPIR1	-0.7679716	2.4e-05
pSRO9	tPIR1	-0.3607687	4.4e-02
pPGK1	tPIR1	-0.3570995	7.0e-03
pRPS3	tRPS3	-0.1788889	3.7e-01
pSRO9	tRPS3	-0.0075926	9.7e-01
pPGK1	tRPS3	0.0798148	7.2e-01

HWNCATTWY

Promoter	Terminator	log2 fold change	p.adj
pTSA1	tTSA1	-0.3393056	0.0013
pPGK1	tRPS3	-0.2947222	0.0069
pPGK1	tTSA1	-0.2619444	0.0100
pRPS3	tRPS3	-0.2331481	0.0200
pPIR1	tPIR1	-0.1563889	0.0120
pSRO9	tRPS3	-0.1474074	0.2000
pSRO9	tTSA1	-0.0976389	0.3500
pPGK1	tPIR1	-0.0711111	0.1600
pSRO9	tPIR1	0.0729630	0.2800

GTATACCTA

Promoter	Terminator	log2 fold change	p.adj
pTSA1	tTSA1	-0.1344444	0.190
pPGK1	tRPS3	0.0158333	0.900
pPGK1	tTSA1	0.0331944	0.750
pSRO9	tTSA1	0.0761111	0.460
pRPS3	tRPS3	0.1289815	0.200
pSRO9	tRPS3	0.2931481	0.011

Supplementary Table B12: Tables showing fold changes in transcript abundance of motifs in different contexts. Tables are sorted by fold change.

tRPS3 Site 1

promoter	label	max_rel_counts	p.adj
pPGK1	WT	0.0013042	0.016
pPGK1	mod_NTN	0.1052831	0.016
pRPS3	WT	0.0013267	0.016
pRPS3	mod_NTN	0.0819971	0.035
pPGK1	mod_HTH	0.1007454	0.420
pRPS3	mod_HTH	0.1161378	0.530
pPGK1	mod_NAA	0.1248866	0.640
pRPS3	mod_NAA	0.1769533	0.970

tRPS3 Site 2

promoter	label	max_rel_counts	p.adj
pPGK1	mod_HTH	0.3172375	0.016
pPGK1	mod_NTN	0.3945115	0.016
pRPS3	mod_NTN	0.3940147	0.016
pPGK1	mod_NAA	0.2223265	0.420
pRPS3	mod_HTH	0.3281569	0.530
pPGK1	WT	0.1708060	0.640
pRPS3	WT	0.1697265	0.970
pRPS3	mod_NAA	0.2062429	0.970

tTSA1 Site 1

promoter	label	max_rel_counts	p.adj
pTSA1	mod_NTN	0.1389566	0.016
pTSA1	mod_HTH	0.1610745	0.027
pTSA1	mod_NAA	0.1503364	0.027
pTSA1	WT	0.1715484	0.068

tTSA1 Site 2

promoter	label	max_rel_counts	p.adj
pTSA1	mod_NAA	0.3671093	0.64
pTSA1	mod_HTH	0.3198939	0.70
pTSA1	WT	0.4000555	1.00
pTSA1	mod_NTN	0.3519398	1.00

Supplementary Table B13: Tables showing relative usage of the two major Poly(A) sites in tRPS3 and tTSA1 terminators across constructs. p values are calculated using a two-sided Wilcoxon signed rank exact test with respect to the mod_NNN construct.

tRPS3 Site 1

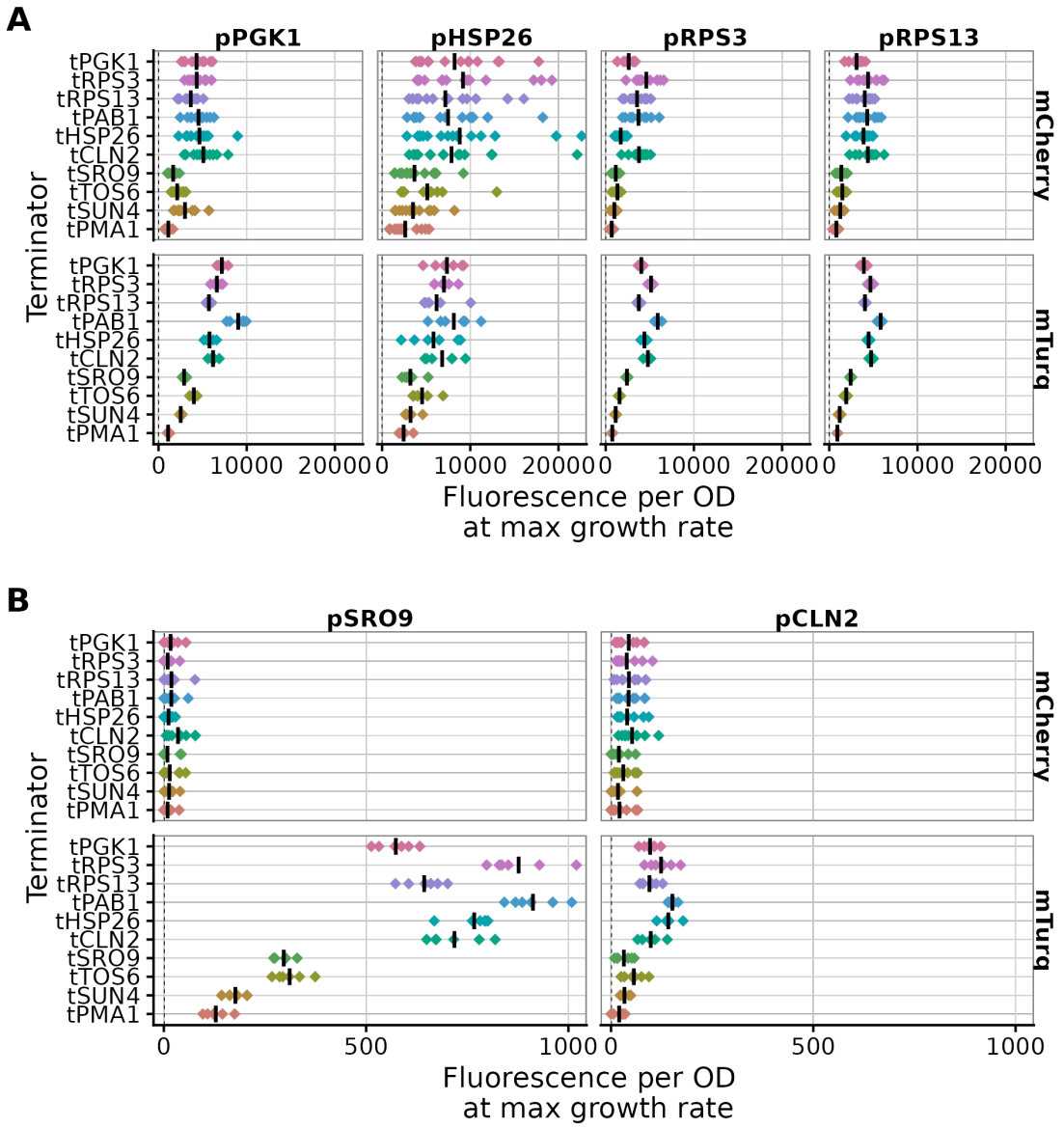
promoter	label	p.value	max_rel_counts
pRPS3	WT	0.20	0.0013634
pRPS3	mod_NNN	0.32	0.1016874
pRPS3	mod_HTH	1.00	0.0937931
pRPS3	mod_NTN	1.00	0.0651922
pRPS3	mod_NAA	1.00	0.1047647

tRPS3 Site 2

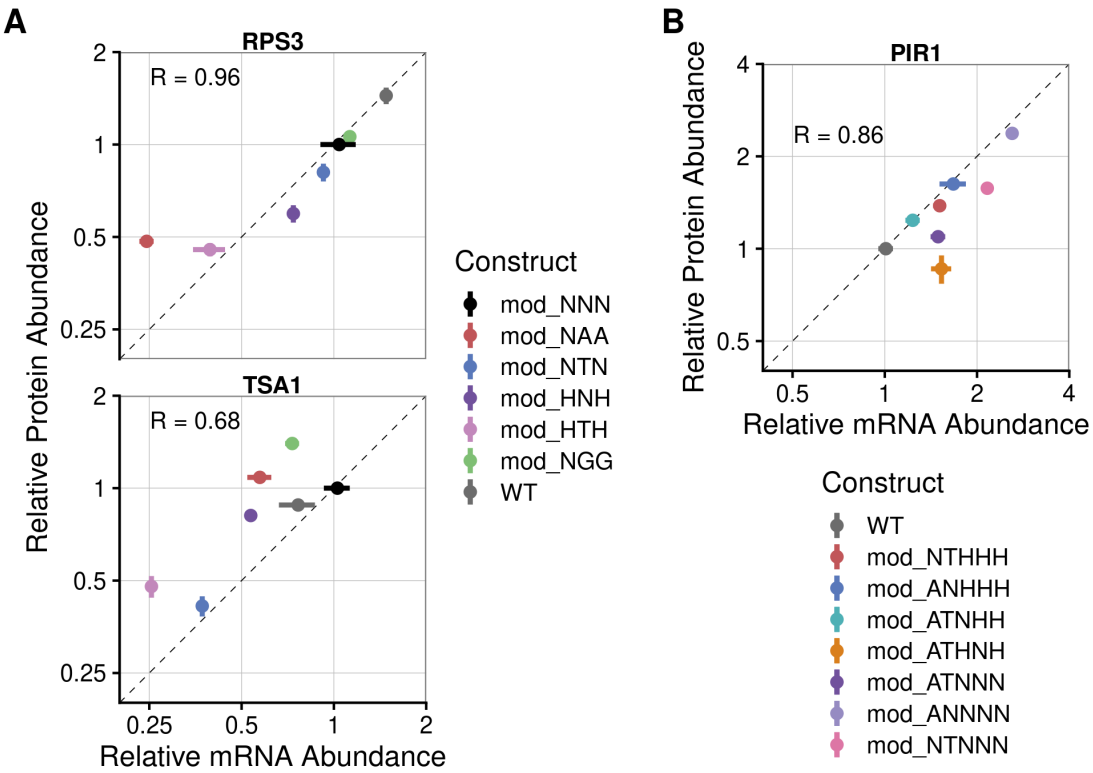
promoter	label	p.value	max_rel_counts
pRPS3	mod_NAA	0.16	0.2064649
pRPS3	mod_NNN	0.16	0.1988836
pRPS3	mod_HTH	0.28	0.3016092
pRPS3	WT	0.62	0.1012708
pRPS3	mod_NTN	0.92	0.3802624

Supplementary Table B14: Tables showing relative usage of the two major Poly(A) sites in tRPS3 across constructs as detected by 5PSeq. p.values are calculated by comparing the rel_counts of the same construct across the two RNA-seq methods using a two-sided Wilcoxon signed rank exact test.

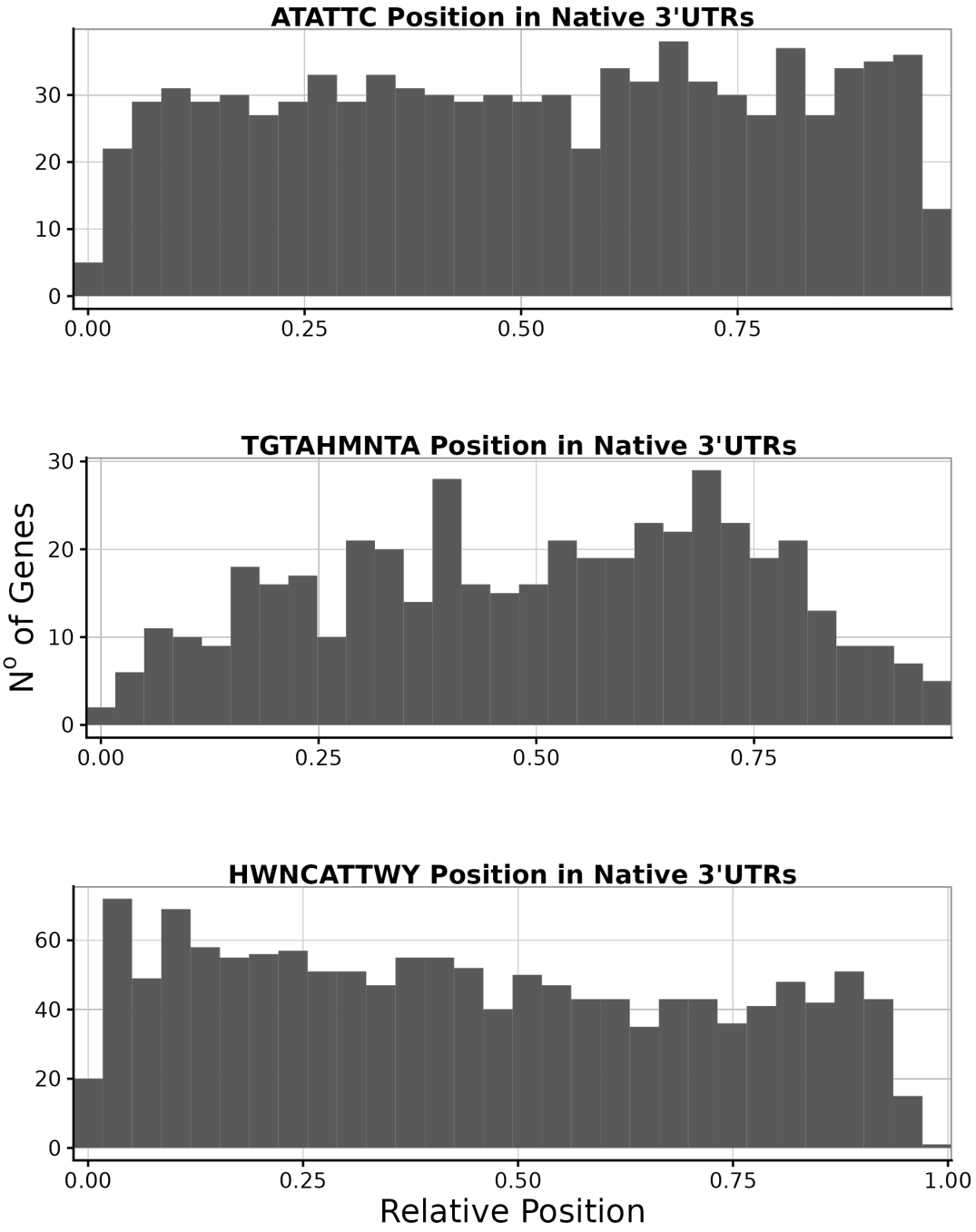
B.2. Chapter 5 Supplementary Figures



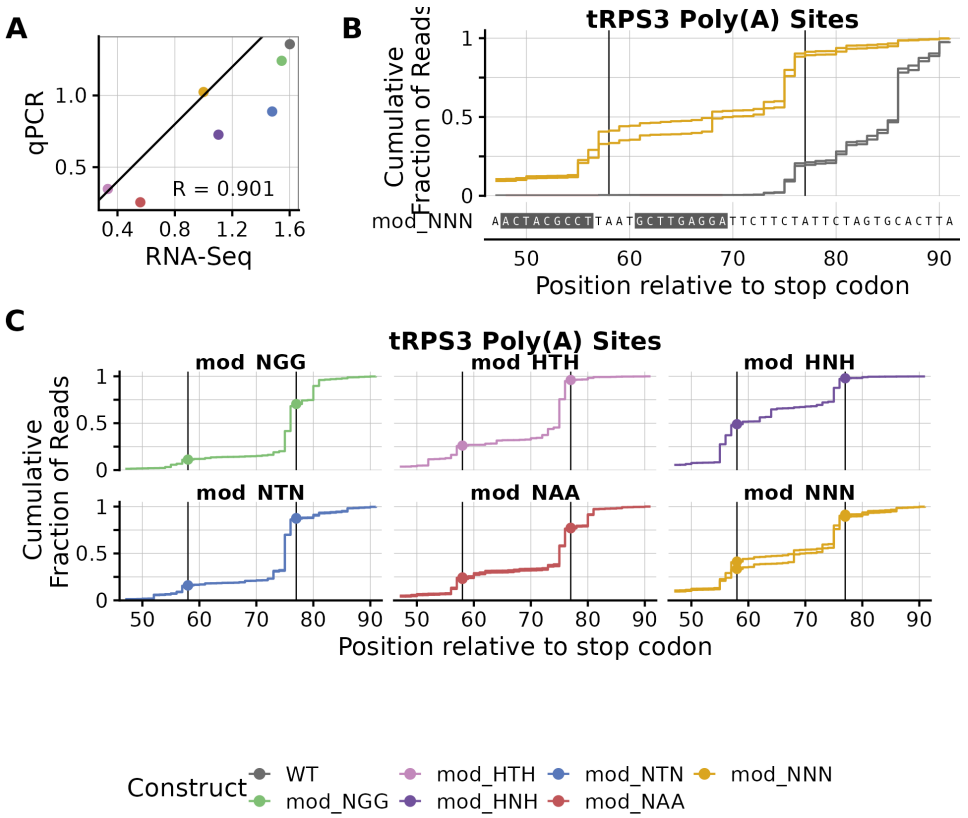
Supplementary Figure B1: Both terminator and promoter contribute to gene expression. (**A**) Protein abundance estimated by mCherry and mTurquoise2 (mTurq) fluorescence for 10 terminators paired with 4 high expressing promoters. Fluorescence and OD were measured in cultures grown in a plate reader and reported at the time of the maximum growth rate of each sample (see methods). Each diamond represents a biological replicate, averaged over 3 technical replicates. The vertical line is the mean of all 6 biological replicates. (**B**) Same as panel B, but for 2 low expressing promoters. Negative fluorescence values arising from instrument noise dominating measurements of constructs with negligible fluorescence are automatically set to 0.



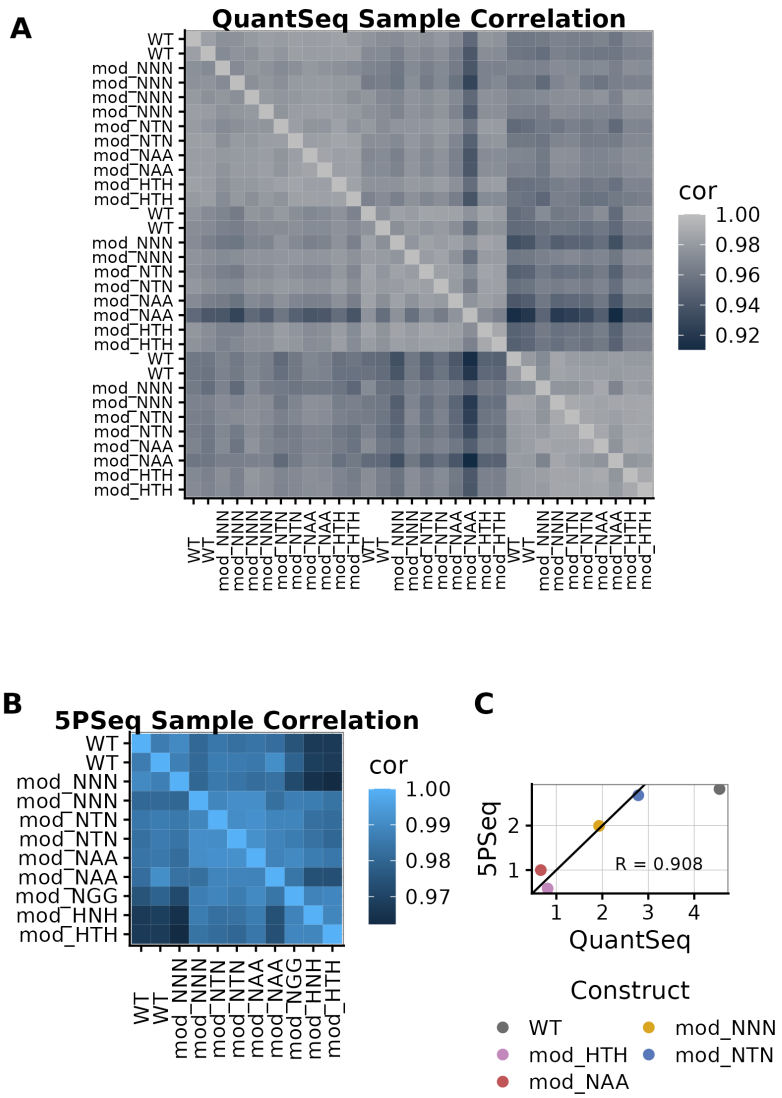
Supplementary Figure B2: Relative protein abundance correlates with relative mRNA abundance for reporter constructs with modified 3'UTRs. (**A**) Fold changes in RT-qPCR vs fold changes in mCherry fluorescence for all terminator constructs in the pRPS3-mCherry-tRPS3 and pTSA1-mCherry-tTSA1 pairings. Transcript abundance is relative to the mod0 construct of each promoter-terminator pairing. (**B**) Fold changes in RT-qPCR vs fold changes in mCherry fluorescence for all terminator constructs in the pPIR1-mCherry-tPIR1 pairing. Transcript abundance is relative to WT construct of the promoter-terminator pairing. The mean and standard error calculated over 6 biological replicates are plotted for each construct.



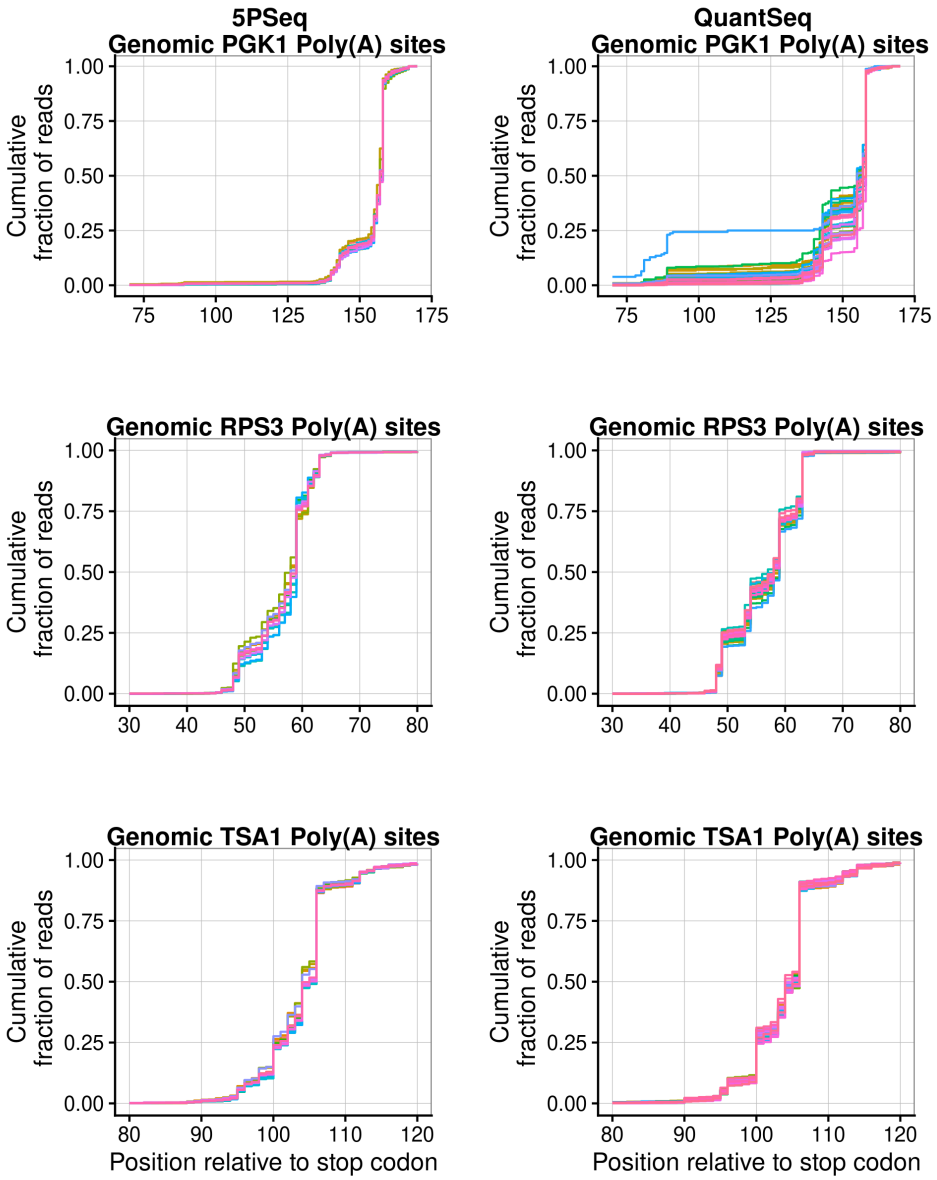
Supplementary Figure B3: Relative positions of ATATTC, TGTAHMNTA and HWNCATTWY motifs in native 3'UTRs. Histogram shows the counts of the motif occurrences in 3'UTRs relative to the total length of 3'UTR, where 0 would be starting exactly at the stop codon and 1 would be at the reported poly(A)-site. See (methods or ref to data repository) for details.



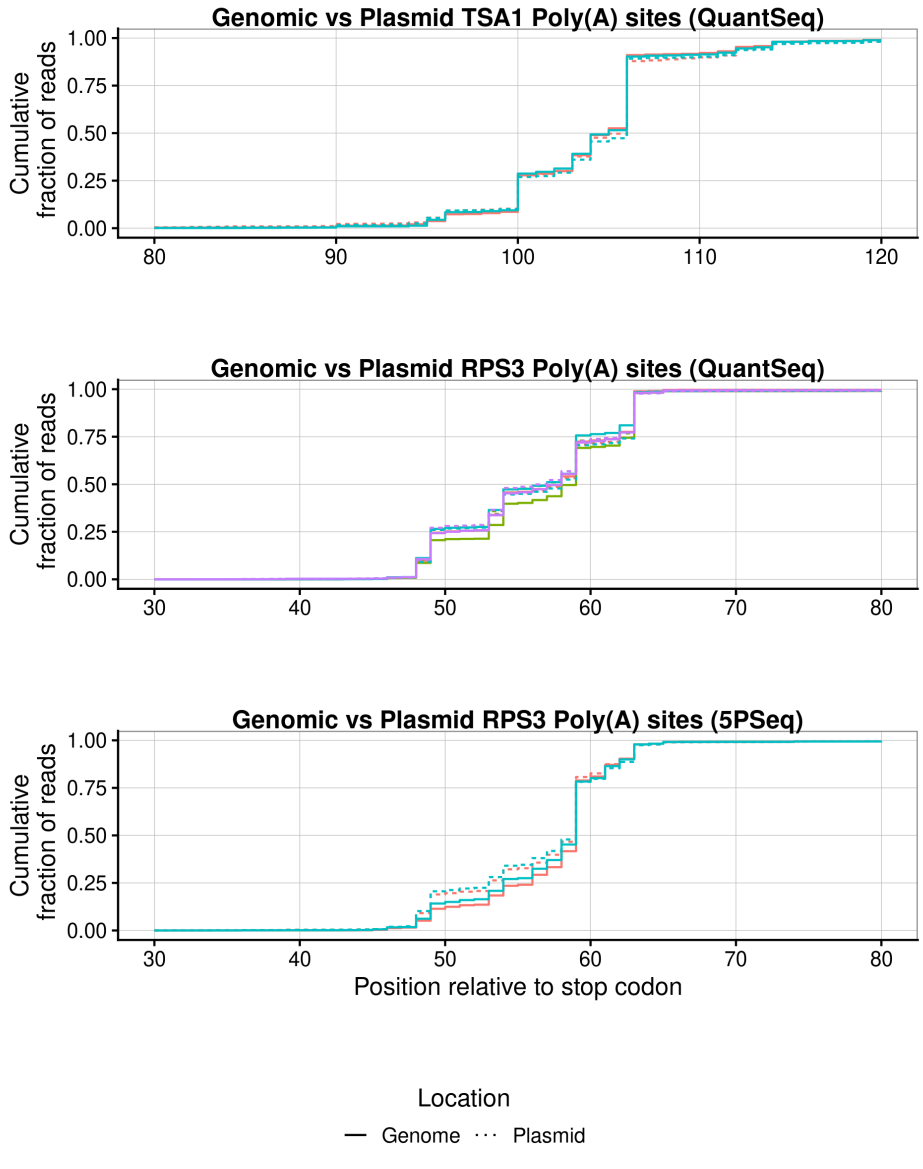
Supplementary Figure B4: Decapped constructs measured by 5Pseq match mature constructs measured by QuantSeq. pRPS3-tRPS3 constructs were chosen to investigate changes in poly(A) site usage across 5' decapped transcripts flagged for decay. (**A**) Comparison of construct transcript abundance as independently measured by qPCR and RNA-Seq assays. Transcript abundance was normalised to the median abundance of plasmid URA3, genomic PGK1 and RPS3 transcripts for each construct. Fold change is relative to the mod_NNN construct in each promoter-terminator context. The black diagonal line represents the expected values of RNAseq and qPCR results correlated perfectly. (**B**) Cumulative counts of reads mapped downstream of WT (grey) and mod_NNN (golden) construct stop codons as a fraction of the total reads mapped to the constructs terminator. WT reads have been shifted downstream to align with the mod_NNN sequence by accounting for motif insertion sites. Major poly(A) sites have been highlighted by a black vertical line. Constructs also used in the QuantSeq analysis have their QuantSeq cumulative graphs plotted in dotted lines. (**C**) Similar to Figure **B** but with each motif insertion construct plotted separately. Columns designate cumulative plots from different terminator constructs.



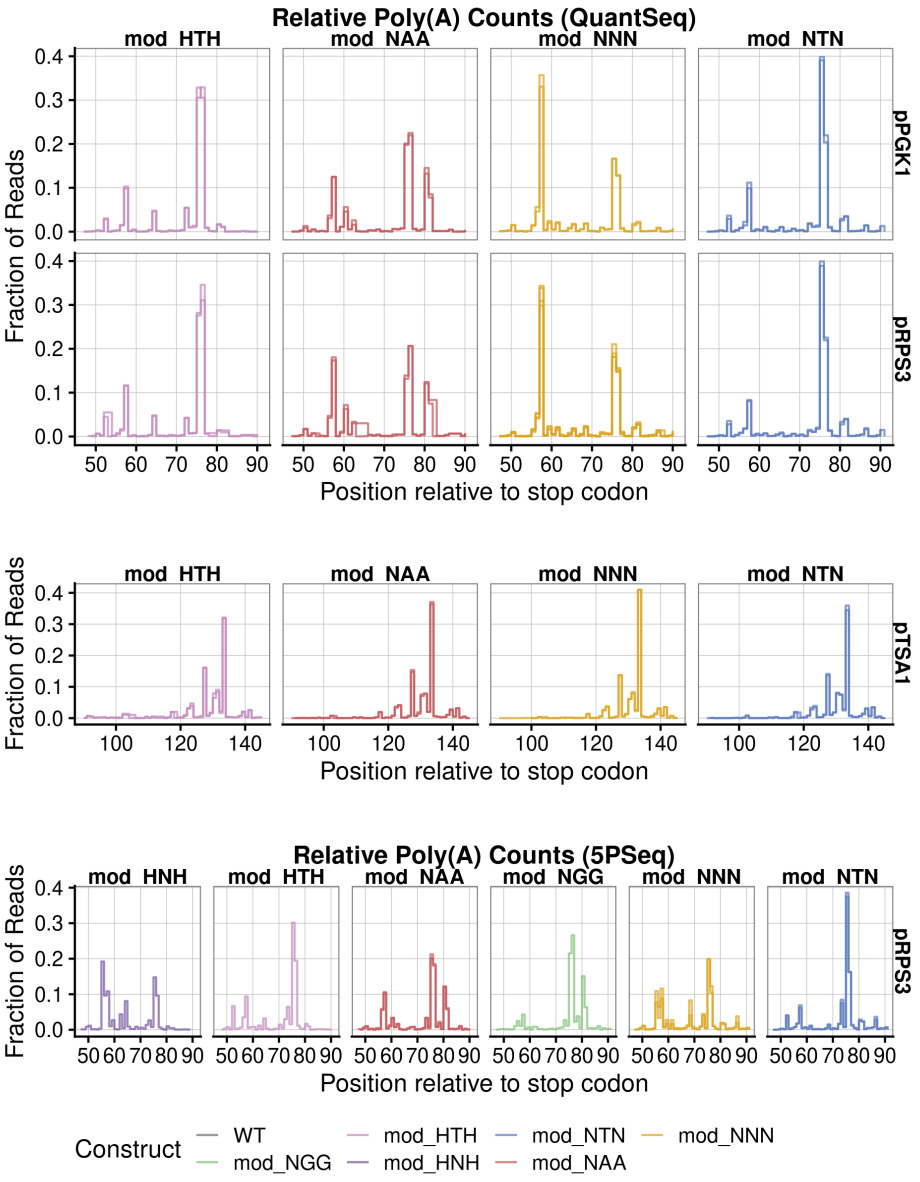
Supplementary Figure B5: High correlation in transcript counts between samples for both RNA-Seq assays. (**A**) Correlation heat map of DESeq2 normalised log2 pseudocounts across all genes for each sample pair in the QuantSeq assay. From top to bottom on the y-axis (left to right on the x-axis) the terminator contexts are pRPS3-tRPS3, pPGK1-tRPS3 and pTSA1-tTSA1. (**B**) Similar correlation heat map for all sample pairs in the 5PSeq assay. (**C**) Comparison of mean log2 abundance of construct mRNA transcripts as measured by 5PSeq and QuantSeq. Only 5 terminator constructs were measured by both methods and only in the pRPS3-tRPS3 context. Transcript abundance for each sample is normalised to the median of the genomic PGK1, TSA1 and RPS3, and the plasmid URA3 genes to match the normalisation used in the qPCR analysis.



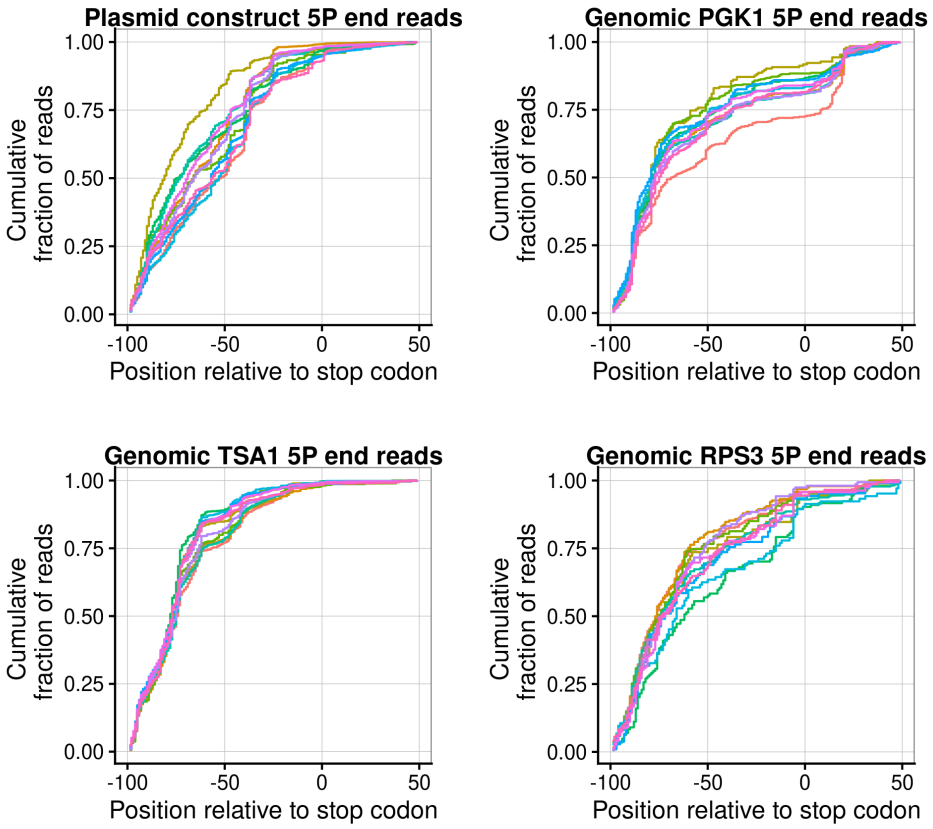
Supplementary Figure B6: Poly(A) site usage for genomic PGK1, TSA1 and RPS3 terminators remains the same across samples for each RNA-Seq assay. Cumulative counts of reads mapped downstream of native genomic gene stop codons as a fraction of the total reads mapped to the terminator. Each line plots the cumulative counts of a separate sequencing run (including two tech reps per sample) which is given a unique colour. Relative usage of poly(A) sites remains similar across samples and across RNA-seq assays. Higher variability in QuantSeq PGK1 poly(A) site usage is due to low overall transcript counts.



Supplementary Figure B7: Poly(A) site usage remains the same for genomic TSA1 and RPS3 terminators as for plasmid expressed WT constructs in QuantSeq and 5PSeq. Cumulative counts of reads mapped downstream of WT stop codons for native genomic terminators and plasmid construct terminators as a fraction of total reads mapped to the constructs terminator. Each line plots the cumulative counts of a separate sequencing run (including two tech reps per sample) which is given a unique colour. Expressing constructs from a low copy number plasmid does not seem to affect poly(A) site usage.



Supplementary Figure B8: Construct poly(A) site usage across 5PSeq and QuantSeq. Relative counts of reads mapped downstream of construct stop codons as a fraction of the total reads mapped to the constructs terminator. Peaks represent the position of major poly(A) sites. Replicates are plotted on top of each other where available.



Supplementary Figure B9: 5PSeq data finds no detectable changes in 5'-phosphorylated intermediates between reporter constructs. Relative counts of 5' end reads of 5'-phosphorylated intermediates as a fraction of the total 5' end reads mapped to the terminator. Results for plasmid constructs are plotted alongside selected genomic genes for comparison.

Appendix C

C.1. Chapter 6 Supplementary Code

DiffFracSeq Stan Code

```
data {
  // Number of RNAs
  int<lower=1> NRNA;

  // Number of replicates
  int<lower=1> NREP;

  // Number of conditions
  int<lower=1> NCON;

  // Note: These are all integers
  // columns t, s, p
  int<lower=0> tot_obs[NCON, NREP, NRNA];
  int<lower=0> sup_obs[NCON, NREP, NRNA];
  int<lower=0> pel_obs[NCON, NREP, NRNA];
}
parameters {
  // Normalising factors
  real scale_factor_mean;
  real<lower=0> tot_scale_factor[NCON, NREP];
  real sup_scale_factor[NCON, NREP];
  real pel_scale_factor[NCON, NREP];

  // latent counts
  vector[NRNA] base_count;
  vector[NRNA] total_count_condition[NCON];
  vector[NRNA] frac_count_condition[NCON];

  // latent count prior parameters
  real norm_alpha;
  real<lower=0> norm_beta;

  // dispersion parameter for counts
  real<lower=0> phi[3];
}

transformed parameters{
// latent counts
  vector[NRNA] sup_latent[NCON];
  vector[NRNA] pel_latent[NCON];

  for(con in 1:NCON){
    sup_latent[con] = base_count + total_count_condition[con];

    pel_latent[con] = sup_latent[con] + frac_count_condition[con];
  }
}
```

```
model{

norm_alpha ~ normal(7,2);
norm_beta ~ normal(2,1);

for(con in 1:NCON){

    scale_factor_mean ~ normal(0, 0.5);
    tot_scale_factor[con] ~ normal(10, 0.1);
    pel_scale_factor[con] ~ normal(scale_factor_mean, 0.1);
    sup_scale_factor[con] ~ normal(scale_factor_mean, 0.1);

    phi ~ normal(100, 10);

    // latent counts
    base_count ~ normal(norm_alpha, norm_beta);
    total_count_condition[con] ~ normal(0, 1);
    frac_count_condition[con] ~ normal(0, 1);


    for(rep in 1:NREP){
          // fractions
          sup_obs[con, rep] ~ neg_binomial_2_log(sup_scale_factor[con, rep] +
            sup_latent[con],
            phi[2]);

        pel_obs[con, rep] ~ neg_binomial_2_log(pel_scale_factor[con, rep] +
            pel_latent[con],
            phi[3]);

        // count distn negative binomial with specified means
        // total
        tot_obs[con, rep] ~ neg_binomial_2(tot_scale_factor[con, rep] *
            (exp(pel_latent[con]) +
            exp(sup_latent[con])),
            phi[1]);
      }
  }
}
```

Bibliography

Bonferroni, C. E. (1936). Teoria statistica delle classi e calcolo delle probabilità. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*.

Cox, R. T. (1946). Probability, frequency and reasonable expectation. *American Journal of Physics*, *14*(1), 1–13. https://doi.org/10.1119/1.1990764

Mann, H. B., & Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *https://doi.org/10.1214/aoms/1177730491*, *18*, 50–60. https://doi.org/10.1214/AOMS/1177730491

Mirsky, A. E., & Ris, H. (1951). The desoxyribonucleic acid content of animal cells and its evolutionary significance. *The Journal of General Physiology*, *34*, 451. https://doi.org/10.1085/JGP.34.4.451

Ui, H., & Mueller, G. C. (1963). The role of rna synthesis in early estrogen action. *Proceedings of the National Academy of Sciences of the United States of America*, *50*, 256. https://doi.org/10.1073/PNAS.50.2.256

Jacob, F, Ullman, A, & Monod, J. (1964). Le promoteur, élément génétique nécessaire à l'expression d'un opéron. *Comptes rendus hebdomadaires des seances de l'Academie des sciences*, *258*, 3125–3128. https://gallica.bnf.fr/ark:/12148/bpt6k4011c/f713.item

Meselson, M., & Yuan, R. (1968). Dna restriction enzyme from e. coli. *Nature 1968 217:5134*, *217*, 1110–1114. https://doi.org/10.1038/2171110a0

Arber, W, & Linn, S. (1969). Dna modification and restriction [PMID: 4897066]. *Annual Review of Biochemistry*, *38*(1), 467–500. https://doi.org/10.1146/annurev.bi.38.070169.002343

Britten, R. J., & Davidson, E. H. (1969). Gene regulation for higher cells: A theory. *Science*, *165*, 349–357. https://doi.org/10.1126/SCIENCE.165.3891.349

Crick, F. (1970). Central dogma of molecular biology. *Nature 1970 227:5258*, *227*, 561–563. https://doi.org/10.1038/227561a0

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*, 55.

Kelly, T. J., & Smith, H. O. (1970). A restriction enzyme from hemophilus influenzae: Ii. base sequence of the recognition site. *Journal of Molecular Biology*, *51*, 393–409. https://doi.org/10.1016/0022-2836(70)90150-6

Smith, H. O., & Welcox, K. W. (1970). A restriction enzyme from hemophilus influenzae: I. purification and general properties. *Journal of Molecular Biology*, *51*, 379–391. https://doi.org/10.1016/0022-2836(70)90149-X

Akaike, H. (1971). Determination of the number of factors by an extended maximum likelihood principle. *Research Memo.*

Thomas, C. A. (1971). The genetic organization of chromosomes. *Annual Review of Genetics*, *5*, 237–256. https://doi.org/10.1146/ANNUREV.GE.05.120171.001321

Jackson, D. A., Symonst, R. H., Berg, P., Lobban, D. P., & Kaiser, A. D. (1972). Biochemical method for inserting new genetic information into dna of simian virus 40: Circular sv40 dna molecules containing lambda phage genes and the galactose operon of escherichia coli. *Proceedings of the National Academy of Sciences*, *69*, 2904–2909. https://doi.org/10.1073/PNAS.69.10.2904

Cohen, S. N., Chang, A. C., Boyer, H. W., & Helling, R. B. (1973). Construction of biologically functional bacterial plasmids in vitro. *Proceedings of the National Academy of Sciences of the United States of America*, *70*, 3240. https://doi.org/10.1073/PNAS.70.11.3240

Sanger, F., Nicklen, S., & Coulson, A. R. (1977). Dna sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America*, *74*, 5463–5467. https://doi.org/10.1073/PNAS.74.12.5463

Beggs, J. D. (1978). Transformation of yeast by a replicating hybrid plasmid. *Nature 1978 275:5676*, *275*, 104–109. https://doi.org/10.1038/275104a0

Schwarz, G. (1978). Estimating the Dimension of a Model. *The Annals of Statistics*, *6*(2), 461 –464. https://doi.org/10.1214/aos/1176344136

Chang, T. (1983). Binding of cells to matrixes of distinct antibodies coated on solid surface. *Journal of Immunological Methods*, *65*, 217–223. https://doi.org/10.1016/0022-1759(83)90318-6

Saiki, R. K., Gelfand, D. H., Stoffel, S., Scharf, S. J., Higuchi, R., Horn, G. T., Mullis, K. B., & Erlich, H. A. (1988). Primer-directed enzymatic amplification of dna with a thermostable dna polymerase. *Science*, *239*, 487–491. https://doi.org/10.1126/SCIENCE.2448875

Capecchi, M. R. (1989). Altering the genome by homologous recombination. *Science*, *244*, 1288–1292. https://doi.org/10.1126/SCIENCE.2660260

Holland, P. M., Abramson, R. D., Watson, R., & Gelfand, D. H. (1991). Detection of specific polymerase chain reaction product by utilizing the 5 $\rightarrow$ 3 exonuclease activity of thermus aquaticus dna polymerase. *Proceedings of the National Academy of Sciences of the United States of America*, *88*, 7276–7280. https://doi.org/10.1073/PNAS.88.16.7276

Williams, W. V., Rosenbaum, H., & Weiner, D. B. (1992). Technical effect of rna concentration on cdna synthesis for dna amplification. *Genome Res*, 86–88. https://doi.org/10.1101/gr.2.1.86

Lee, R. C., Feinbaum, R. L., & Ambros, V. (1993). The C. elegans heterochronic gene lin-4 encodes small rnas with antisense complementarity to lin-14. *Cell*, *75*, 843–854. https://doi.org/10.1016/0092-8674(93)90529-y

Wightman, B., Ha, I., & Ruvkun, G. (1993). Posttranscriptional regulation of the heterochronic gene lin-14 by lin-4 mediates temporal pattern formation in C. elegans. *Cell*, *75*, 855–862. https://doi.org/10.1016/0092-8674(93)90530-4

Benjamini, Y., & Hochberg, Y. (1995a). Controlling the false discovery rate - a practical and powerful approach to multiple testing. *J. Royal Statist. Soc., Series B*, *57*, 289 –300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x

Benjamini, Y., & Hochberg, Y. (1995b). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society.*

Series B (Methodological), *57*(1), 289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x

Brooks, E. M., Sheflin, L. G., & Spaulding, S. W. (1995). Secondary structure in the 3' utr of egf and the choice of reverse transcriptases affect the detection of message diversity by rt-pcr. *Biotechniques*, *19*, 806–12, 814. https://europepmc.org/article/med/8588921

Schena, M., Shalon, D., Davis, R. W., & Brown, P. O. (1995). Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, *270*, 467–470. https://doi.org/10.1126/SCIENCE.270.5235.467

Guo, Z., & Sherman, F. (1996). 3'f-end-forming signals of yeast mRNA. *Trends in biochemical sciences*, *21*(12), 477–481. https://doi.org/10.1128/mcb.15.11.5983

McConnell, S. (1996). Rapid development : Taming wild software schedules.

Sil, A., & Herskowitz, I. (1996). Identification of an asymmetrically localized determinant, Ash1p, required for lineage-specific transcription of the yeast HO gene. *Cell*, *84*, 711–722. https://doi.org/10.1016/S0092-8674(00)81049-1

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, *58*(1), 267–288. Retrieved October 27, 2022, from http://www.jstor.org/stable/2346178

Akaike, H. (1998). Information theory and an extension of the maximum likelihood principle, 199–213. https://doi.org/10.1007/978-1-4612-1694-0_15

Cameron, A. C., & Trivedi, P. K. (1998). *Regression analysis of count data*. Cambridge University Press. https://faculty.econ.ucdavis.edu/faculty/cameron/racd2/

Willardson, B. M., Wilkins, J. F., Rand, T. A., Schupp, J. M., Hill, K. K., Keim, P., & Jackson, P. J. (1998). Development and testing of a bacterial biosensor for toluene-based environmental contaminants. *Applied and Environmental Microbiology*, *64*, 1006–1012. https://doi.org/10.1128/AEM.64.3.1006-1012.1998

Olsen, P. H., & Ambros, V. (1999). The lin-4 regulatory RNA controls developmental timing in Caenorhabditis elegans by blocking LIN-14 protein synthesis after the initiation of translation. *Developmental Biology*, *216*, 671–680. https://doi.org/10.1006/dbio.1999.9523

Thomas, D., & Hunt, A. (1999). *The pragmatic programmer*. https://pragprog.com/titles/tpp20/the-pragmatic-programmer-20th-anniversary-edition/

Elowitz, M. B., & Leibier, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature 2000 403:6767*, *403*, 335–338. https://doi.org/10.1038/35002125

Gardner, T. S., Cantor, C. R., & Collins, J. J. (2000). Construction of a genetic toggle switch in escherichia coli. *Nature 2000 403:6767*, *403*, 339–342. https://doi.org/10.1038/35002131

Gasch, A. P., Spellman, P. T., Kao, C. M., Carmel-Harel, O., Eisen, M. B., Storz, G., Botstein, D., & Brown, P. O. (2000). Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, *11*, 4241–4257. https://doi.org/10.1091/MBC.11.12.4241/ASSET/IMAGES/LARGE/MK1201389008.JPEG

Lander, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., Baldwin, J., Devon, K., Dewar, K., Doyle, M., Fitzhugh, W., Funke, R., Gage, D., Harris, K., Heaford, A., Howland, J., Kann, L., Lehoczky, J., Levine, R., McEwan, P., ... Morgan,

M. J. (2001). Initial sequencing and analysis of the human genome. *Nature 2001 409:6822*, *409*, 860–921. https://doi.org/10.1038/35057062

Bustin, S. A. (2002). Quantification of mrna using real-time reverse transcription pcr (rt-pcr): Trends and problems. *Journal of Molecular Endocrinology*, *29*, 23–39. https://doi.org/10.1677/JME.0.0290023

Forward, A., & Lethbridge, T. C. (2002). The relevance of software documentation, tools and technologies. *Proceedings of the 2002 ACM symposium on Document engineering - DocEng '02*, 26. https://doi.org/10.1145/585058.585065

Gietz, R. D., & Woods, R. A. (2002). Transformation of yeast by lithium acetate/single-stranded carrier DNA/polyethylene glycol method. *Methods in enzymology*, *350*, 87–96. https://doi.org/10.1016/S0076-6879(02)50957-5

Ripley, B., & Venables, B. (2002). *Modern applied statistics with s* (4th). Springer.

Walker, N. J. (2002). A technique whose time has come. *Science*, *296*. https://doi.org/10.1126/SCIENCE.296.5567.557

Yokobayashi, Y., Weiss, R., & Arnold, F. H. (2002). Directed evolution of a genetic circuit. *Proceedings of the National Academy of Sciences of the United States of America*, *99*, 16587–16591. https://doi.org/https://doi.org/10.1073/pnas.252535999

Knight, T. (2003). Idempotent vector design for standard assembly of biobricks. https://dspace.mit.edu/handle/1721.1/21168

Beck, K., & Andres, C. (2004). *Extreme programming explained: Embrace change (2nd edition)*. Addison-Wesley Professional.

Chua, C. L., Tang, F., Lim, Y. P., Ho, L. Y., & Krishnan, A. (2004). Implementing a bioinformatics workflow in a parallel and distributed environment. *Lecture Notes in Computer Science*, *3320*, 1–4. https://doi.org/10.1007/978-3-540-30501-9_1/COVER

Dean, J., & Ghemawat, S. (2004). Mapreduce: Simplified data processing on large clusters, 137–150.

Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., Hornik, K., Hothorn, T., Huber, W., Iacus, S., Irizarry, R., Leisch, F., Li, C., Maechler, M., Rossini, A. J., ... Zhang, J. (2004). Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology 2004 5:10*, *5*, 1–16. https://doi.org/10.1186/GB-2004-5-10-R80

Gu, W., Deng, Y., Zenklusen, D., & Singer, R. H. (2004). A new yeast PUF family protein, Puf6p, represses ASH1 mRNA translation and is required for its localization. *Genes and Development*, *18*, 1452–1465. https://doi.org/10.1101/gad.1189004

Jackson, J. S., Houshmandi, S. S., Leban, F. L., & Olivas, W. M. (2004). Recruitment of the Puf3 protein to its mRNA target for regulation of mRNA decay in yeast. *RNA*, *10*, 1625–1636. https://doi.org/10.1261/rna.7270204

O'Sullivan, J. M., Tan-Wong, S. M., Morillon, A., Lee, B., Coles, J., Mellor, J., & Proudfoot, N. J. (2004). Gene loops juxtapose promoters and terminators in yeast. *Nature Genetics*, *36*. https://doi.org/10.1038/ng1411

Shaner, N. C., Campbell, R. E., Steinbach, P. A., Giepmans, B. N., Palmer, A. E., & Tsien, R. Y. (2004). Improved monomeric red, orange and yellow fluorescent proteins derived from Discosoma sp. red fluorescent protein. *Nature biotechnology*, *22*(12), 1567–1572. https://doi.org/10.1038/nbt1037

Ståhlberg, A., Håkansson, J., Xian, X., Semb, H., & Kubista, M. (2004). Properties of the reverse transcription reaction in mrna quantification. *Clinical Chemistry*, *50*, 509–515. https://doi.org/10.1373/CLINCHEM.2003.026161

Duttagupta, R., Tian, B., Wilusz, C. J., Khounh, D. T., Soteropoulos, P., Ouyang, M., Dougherty, J. P., & Peltz, S. W. (2005). Global analysis of pub1p targets reveals a coordinate control of gene expression through modulation of binding and stability. *Molecular and Cellular Biology*, *25*. https://doi.org/10.1128/mcb.25.13.5499-5513.2005

Gui, X., Hwang, J. T., Qiu, J., Blades, N. J., & Churchill, G. A. (2005). Improved statistical tests for differential gene expression by shrinking variance components estimates. *Biostatistics*, *6*, 59–75. https://doi.org/10.1093/BIOSTATISTICS/KXH018

Ihmels, H., & Otto, D. (2005). Intercalation of organic dye molecules into double-stranded dna - general principles and recent developments. *Topics in Current Chemistry*, *258*, 161–204. https://doi.org/10.1007/B135804/COVER

Olivier, C., Poirier, G., Gendron, P., Boisgontier, A., Major, F., & Chartrand, P. (2005). Identification of a conserved RNA motif essential for She2p recognition and mRNA localization to the yeast bud. *Molecular and Cellular Biology*, *25*, 4752–4766. https://doi.org/10.1128/MCB.25.11.4752-4766.2005

Rasmussen, C. E., & Williams, C. K. I. (2005, November). *Regression*. The MIT Press. https://doi.org/10.7551/mitpress/3206.003.0005

Shalgi, R., Lapidot, M., Shamir, R., & Pilpel, Y. (2005). A catalog of stability-associated sequence elements in 3' UTRs of yeast mRNAs. *Genome biology*, *6*. https://doi.org/10.1186/gb-2005-6-10-r86

Smyth, G. K. (2005). Limma: Linear models for microarray data. *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*, 397–420. https://doi.org/10.1007/0-387-29362-0_23

Wall, M. E., Rechtsteiner, A., & Rocha, L. M. (2005). Singular value decomposition and principal component analysis. *A Practical Approach to Microarray Data Analysis*, 91–109. https://doi.org/10.1007/0-306-47815-3_5/COVER

Wong, M. L., & Medrano, J. F. (2005). Real-time pcr for mrna quantitation. *BioTechniques*, *39*, 75–85. https://doi.org/10.2144/05391RV01

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*, 301–320. https://doi.org/10.1111/J.1467-9868.2005.00503.X

Aviv, T., Lin, Z., Ben-Ari, G., Smibert, C. A., & Sicheri, F. (2006). Sequence-specific recognition of rna hairpins by the sam domain of vts1p. *Nature Structural and Molecular Biology*, *13*, 168–176. https://doi.org/10.1038/nsmb1053

Kane, D. W., Hohman, M. M., Cerami, E. G., McCormick, M. W., Kuhlmman, K. F., & Byrd, J. A. (2006). Agile methods in biomedical software development: A multi-site experience report. *BMC Bioinformatics*, *7*, 1–12. https://doi.org/10.1186/1471-2105-7-273

Kubista, M., Andrade, J. M., Bengtsson, M., Forootan, A., Jonák, J., Lind, K., Sindelka, R., Sjöback, R., Sjögreen, B., Strömbom, L., Ståhlberg, A., & Zoric, N. (2006). The real-time polymerase chain reaction [Real-time Polymerase Chain Reaction]. *Molecular Aspects of Medicine*, *27*(2), 95–125. https://doi.org/https://doi.org/10.1016/j.mam.2005.12.007

Savarese, F., & Grosschedl, R. (2006). Blurring cis and trans in gene regulation. *Cell*, *126*, 248–250. https://doi.org/10.1016/j.cell.2006.07.008

Elemento, O., Slonim, N., & Tavazoie, S. (2007). A universal framework for regulatory element discovery across all genomes and data types. *Molecular Cell*, *28*, 337–350. https://doi.org/10.1016/j.molcel.2007.09.027

Garneau, N. L., Wilusz, J., & Wilusz, C. J. (2007). The highways and byways of mrna decay. *Nature Reviews Molecular Cell Biology 2007 8:2*, *8*, 113–126. https://doi.org/10.1038/nrm2104

Köhler, A., & Hurt, E. (2007). Exporting rna from the nucleus to the cytoplasm. *Nature Reviews Molecular Cell Biology 2007 8:10*, *8*, 761–773. https://doi.org/10.1038/nrm2255

Paquin, N., Ménade, M., Poirier, G., Donato, D., Drouet, E., & Chartrand, P. (2007). Local activation of yeast ASH1 mRNA translation through phosphorylation of Khd1p by the casein kinase Yck1p. *Molecular Cell*, *26*, 795–809. https://doi.org/10.1016/j.molcel.2007.05.016

Robinson, M. D., & Smyth, G. K. (2007). Moderated statistical tests for assessing differences in tag abundance. *Bioinformatics*, *23*, 2881–2887. https://doi.org/10.1093/BIOINFORMATICS/BTM453

Rusk, N., & Kiermer, V. (2007). Primer: Sequencing—the next generation. *Nature Methods 2008 5:1*, *5*, 15–15. https://doi.org/10.1038/nmeth1155

Craig, D. W., Pearson, J. V., Szelinger, S., Sekar, A., Redman, M., Corneveaux, J. J., Pawlowski, T. L., Laub, T., Nunn, G., Stephan, D. A., Homer, N., & Huentelman, M. J. (2008). Identification of genetic variants using bar-coded multiplexed sequencing. *Nature Methods 2008 5:10*, *5*, 887–893. https://doi.org/10.1038/nmeth.1251

Deng, Y., Singer, R. H., & Gu, W. (2008). Translation of ASH1 mRNA is repressed by Puf6p-Fun12p/eIF5B interaction and released by CK2 phosphorylation. *Genes and Development*, *22*, 1037–1050. https://doi.org/10.1101/gad.1611308

Dohm, J. C., Lottaz, C., Borodina, T., & Himmelbauer, H. (2008). Substantial biases in ultra-short read data sets from high-throughput dna sequencing. *Nucleic Acids Research*, *36*, 105. https://doi.org/10.1093/NAR/GKN425

Feinerer, I., Hornik, K., & Meyer, D. (2008). Text mining infrastructure in r. *Journal of Statistical Software*, *25*(5), 1–54. https://www.jstatsoft.org/v25/i05/

Frith, M. C., Saunders, N. F. W., Kobe, B., & Bailey, T. L. (2008). Discovering sequence motifs with arbitrary insertions and deletions (G. Stormo, Ed.). *PLoS Computational Biology*, *4*, e1000071. https://doi.org/10.1371/journal.pcbi.1000071

Hogan, D. J., Riordan, D. P., Gerber, A. P., Herschlag, D., & Brown, P. O. (2008). Diverse RNA-binding proteins interact with functionally related sets of RNAs, suggesting an extensive regulatory system (S. R. Eddy, Ed.). *PLoS Biology*, *6*, e255. https://doi.org/10.1371/journal.pbio.0060255

Ritz, C., & Spiess, A. N. (2008). Qpcr: An r package for sigmoidal model selection in quantitative real-time polymerase chain reaction analysis. *Bioinformatics*, *24*, 1549–1551. https://doi.org/10.1093/BIOINFORMATICS/BTN227

Sánchez, Á., & Kondev, J. (2008). Transcriptional control of noise in gene expression. *Proceedings of the National Academy of Sciences*, *105*(13), 5081–5086. https://doi.org/10.1073/pnas.0707904105

Swillens, S., Dessars, B., & Housni, H. E. (2008). Revisiting the sigmoidal curve fitting applied to quantitative real-time pcr data. *Analytical Biochemistry*, *373*, 370–376. https://doi.org/10.1016/J.AB.2007.10.019

VanGuilder, H. D., Vrana, K. E., & Freeman, W. M. (2008). Twenty-five years of quantitative pcr for gene expression analysis. *BioTechniques*, *44*, 619–626. https://doi.org/10.2144/000112776

Anderson, P., & Kedersha, N. (2009). Rna granules: Post-transcriptional and epigenetic modulators of gene expression. *Nature Reviews Molecular Cell Biology 2009 10:6*, *10*, 430–436. https://doi.org/10.1038/nrm2694

Bustin, S. A., Benes, V., Garson, J. A., Hellemans, J., Huggett, J., Kubista, M., Mueller, R., Nolan, T., Pfaffl, M. W., Shipley, G. L., Vandesompele, J., & Wittwer, C. T. (2009). The miqe guidelines: Minimum information for publication of quantitative real-time pcr experiments. *Clinical chemistry*, *55*, 611–622. https://doi.org/10.1373/CLINCHEM.2008.112797

Dvinge, H., & Bertone, P. (2009). Htqpcr: High-throughput analysis and visualization of quantitative real-time pcr data in r. *Bioinformatics (Oxford, England)*, *25*, 3325–3326. https://doi.org/10.1093/BIOINFORMATICS/BTP578

Garson, J. A., Huggett, J. F., Bustin, S. A., Pfaffl, M. W., Benes, V., Vandesompele, J., & Shipley, G. L. (2009). Unreliable real-time pcr analysis of human endogenous retrovirus-w (herv-w) rna expression and dna copy number in multiple sclerosis. *https://home.liebertpub.com/aid*, *25*, 377–378. https://doi.org/10.1089/AID.2008.0270

Granneman, S., Kudla, G., Petfalski, E., & Tollervey, D. (2009). Identification of protein binding sites on u3 snorna and pre-rrna by uv cross-linking and high-throughput analysis of cdnas. *Proceedings of the National Academy of Sciences of the United States of America*, *106*, 9613–9618. https://doi.org/10.1073/PNAS.0901997106

Hannay, J. E., MacLeod, C., Singer, J., Langtangen, H. P., Pfahl, D., & Wilson, G. (2009). How do scientists develop and use scientific software? *Proceedings of the 2009 ICSE Workshop on Software Engineering for Computational Science and Engineering, SECSE 2009*, 1–8. https://doi.org/10.1109/SECSE.2009.5069155

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning. https://doi.org/10.1007/978-0-387-84858-7

Ioannidis, J. P., Allison, D. B., Ball, C. A., Coulibaly, I., Cui, X., Culhane, A. C., Falchi, M., Furlanello, C., Game, L., Jurman, G., Mangion, J., Mehta, T., Nitzberg, M., Page, G. P., Petretto, E., & Noort, V. V. (2009). Repeatability of published microarray gene expression analyses. *Nature Genetics 2009 41:2*, *41*, 149–155. https://doi.org/10.1038/ng.295

Jansen, J. M., Wanless, A. G., Seidel, C. W., & Weiss, E. L. (2009). Cbk1 regulation of the rna-binding protein Ssd1 integrates cell fate with translational control. *Current Biology*, *19*, 2114–2120. https://doi.org/10.1016/j.cub.2009.10.071

Lefever, S., Hellemans, J., Pattyn, F., Przybylski, D. R., Taylor, C., Geurts, R., Untergasser, A., & Vandesompele, J. (2009). Rdml: Structured language and reporting guidelines for real-time quantitative pcr data. *Nucleic acids research*, *37*, 2065–2069. https://doi.org/10.1093/NAR/GKP056

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., & Durbin, R. (2009). The sequence alignment/map format and samtools. *Bioinformatics*, *25*, 2078–2079. https://doi.org/10.1093/bioinformatics/btp352

Oshlack, A., & Wakefield, M. J. (2009). Transcript length bias in rna-seq data confounds systems biology. *Biology Direct*, *4*, 1–10. https://doi.org/10.1186/1745-6150-4-14

Anders, S., & Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biology*, *11*, 1–12. https://doi.org/10.1186/GB-2010-11-10-R106

Andrews, S. (2010). Fastqc: A quality control tool for high throughput sequence data. https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Auer, P. L., & Doerge, R. W. (2010). Statistical design and analysis of rna sequencing data. *Genetics*, *185*, 405–416. https://doi.org/10.1534/GENETICS.110.114983

Bleichert, F., & Baserga, S. J. (2010). Ribonucleoprotein multimers and their functions. *Critical reviews in biochemistry and molecular biology*, *45*, 331. https://doi.org/10.3109/10409238.2010.496772

Juskowiak, B. (2010). Nucleic acid-based fluorescent probes and their analytical potential. *Analytical and Bioanalytical Chemistry 2010 399:9*, *399*, 3157–3176. https://doi.org/10.1007/S00216-010-4304-5

Prause, C. R., Reiners, R., & Dencheva, S. (2010). Empirical study of tool support in highly distributed research projects. *Proceedings - 5th International Conference on Global Software Engineering, ICGSE 2010*, 23–32. https://doi.org/10.1109/ICGSE.2010.13

Quinlan, A. R., & Hall, I. M. (2010). Bedtools: A flexible suite of utilities for comparing genomic features. *Bioinformatics*, *26*, 841–842. https://doi.org/10.1093/bioinformatics/btq033

Robinson, M. D., & Oshlack, A. (2010). A scaling normalization method for differential expression analysis of rna-seq data. *Genome Biology*, *11*, 1–9. https://doi.org/10.1186/GB-2010-11-3-R25/

Trapnell, C., Williams, B. A., Pertea, G., Mortazavi, A., Kwan, G., Baren, M. J. V., Salzberg, S. L., Wold, B. J., & Pachter, L. (2010). Transcript assembly and quantification by rna-seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology 2010 28:5*, *28*, 511–515. https://doi.org/10.1038/nbt.1621

Unworth, H., Raguz, S., Edwards, H. J., Higgins, C. F., & Yagüe, E. (2010). Mrna escape from stress granule sequestration is dictated by localization to the endoplasmic reticulum. *The FASEB Journal*, *24*, 3370–3380. https://doi.org/10.1096/FJ.09-151142

Bregman, A., Avraham-Kelbert, M., Barkai, O., Duek, L., Guterman, A., & Choder, M. (2011). Promoter elements regulate cytoplasmic mrna decay. *Cell*, *147*. https://doi.org/10.1016/j.cell.2011.12.005

Kivioja, T., Vähärautio, A., Karlsson, K., Bonke, M., Enge, M., Linnarsson, S., & Taipale, J. (2011). Counting absolute numbers of molecules using unique molecular identifiers. *Nature Methods 2011 9:1*, *9*, 72–74. https://doi.org/10.1038/nmeth.1778

Kurischko, C., Kuravi, V. K., Herbert, C. J., & Luca, F. C. (2011). Nucleocytoplasmic shuttling of ssd1 defines the destiny of its bound mrnas. *Molecular Microbiology*, *81*, 831–849. https://doi.org/10.1111/J.1365-2958.2011.07731.X

Li, B., & Dewey, C. N. (2011). Rsem: Accurate transcript quantification from rna-seq data with or without a reference genome. *BMC Bioinformatics*, *12*, 1–16. https://doi.org/10.1186/1471-2105-12-323

Lorenz, R., Bernhart, S. H., zu Siederdissen, C. H., Tafer, H., Flamm, C., Stadler, P. F., & Hofacker, I. L. (2011). Viennarna package 2.0. *Algorithms for Molecular Biology*, *6*, 26.

Martin, M. (2011). Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal*, *17*, 10. https://doi.org/10.14806/ej.17.1.200

Sletholt, M. T., Hannay, J., Pfahl, D., Benestad, H. C., & Langtangen, H. P. (2011). A literature review of agile practices and their effects in scientific software development. *Proceedings - International Conference on Software Engineering*, 1–9. https://doi.org/10.1145/1985782.1985784

Voronina, E., Seydoux, G., Sassone-Corsi, P., & Nagamori, I. (2011). Rna granules in germ cells. *Cold Spring Harbor Perspectives in Biology*, *3*, a002774. https://doi.org/10.1101/CSHPERSPECT.A002774

Wickham, H. (2011). The split-apply-combine strategy for data analysis. *Journal of Statistical Software*, *40*, 1–29. https://doi.org/10.18637/JSS.V040.I01

Will, C. L., & Lührmann, R. (2011). Spliceosome structure and function. *Cold Spring Harbor Perspectives in Biology*, *3*, 1–2. https://doi.org/10.1101/CSHPERSPECT.A003707

Djebali, S., Davis, C. A., Merkel, A., Dobin, A., Lassmann, T., Mortazavi, A., Tanzer, A., Lagarde, J., Lin, W., Schlesinger, F., Xue, C., Marinov, G. K., Khatun, J., Williams, B. A., Zaleski, C., Rozowsky, J., Röder, M., Kokocinski, F., Abdelhamid, R. F., … Gingeras, T. R. (2012). Landscape of transcription in human cells. *Nature 2012 489:7414*, *489*, 101–108. https://doi.org/10.1038/nature11233

Dunham, I., Kundaje, A., Aldred, S. F., Collins, P. J., Davis, C. A., Doyle, F., Epstein, C. B., Frietze, S., Harrow, J., Kaul, R., Khatun, J., Lajoie, B. R., Landt, S. G., Lee, B. K., Pauli, F., Rosenbloom, K. R., Sabo, P., Safi, A., Sanyal, A., … Lochovsky, L. (2012). An integrated encyclopedia of dna elements in the human genome. *Nature 2012 489:7414*, *489*, 57–74. https://doi.org/10.1038/nature11247

EMBL-EBI. (2012). Embl-ebi annual scientific report 2012 – embl documents. https://www.embl.org/documents/document/embl-ebi-annual-scientific-report-2012/

Goedhart, J., Von Stetten, D., Noirclerc-Savoye, M., Lelimousin, M., Joosen, L., Hink, M. A., Van Weeren, L., Gadella, T. W., & Royant, A. (2012). Structure-guided evolution of cyan fluorescent proteins towards a quantum yield of 93%. *Nature communications*, *3*(1), 1–9. https://doi.org/10.1038/ncomms1738

Kittleson, J. T., Wu, G. C., & Anderson, J. C. (2012). Successes and failures in modular genetic engineering. *Current Opinion in Chemical Biology*, *16*, 329–336. https://doi.org/10.1016/J.CBPA.2012.06.009

Langmead, B., & Salzberg, S. L. (2012). Fast gapped-read alignment with bowtie 2. *Nature Methods 2012 9:4*, *9*, 357–359. https://doi.org/10.1038/nmeth.1923

Pathak, S., & Saxena, P. (2012). Hybrid methodology involving scrum and waterfall model towards the software project development in academic knowledge centers. *International Journal of Evaluation and Research in Education (IJERE)*, *1*. https://doi.org/10.11591/IJERE.V1I1.456

Perkins, J. R., Dawes, J. M., McMahon, S. B., Bennett, D. L., Orengo, C., & Kohl, M. (2012). Readqpcr and normqpcr: R packages for the reading, quality checking and normalisation of rt-qpcr quantification cycle (cq) data. *BMC Genomics*, *13*, 1–8. https://doi.org/10.1186/1471-2164-13-296

Prlić, A., & Procter, J. B. (2012). Ten simple rules for the open development of scientific software. *PLOS Computational Biology*, *8*(12), 1–3. https://doi.org/10.1371/journal.pcbi.1002802

Sharon, E., Kalma, Y., Sharp, A., Raveh-Sadka, T., Levo, M., Zeevi, D., Keren, L., Yakhini, Z., Weinberger, A., & Segal, E. (2012). Inferring gene regulatory logic from high-throughput measurements of thousands of systematically designed promoters. *Nature biotechnology*, *30*(6), 521–530. https://doi.org/10.1038/nbt.2205

Spitz, F., & Furlong, E. E. (2012). Transcription factors: From enhancer binding to developmental control. *Nature Reviews Genetics 2012 13:9*, *13*, 613–626. https://doi.org/10.1038/nrg3207

Berchowitz, L. E., Gajadhar, A. S., van Werven, F. J., Rosa, A. A. D., Samoylova, M. L., Brar, G. A., Xu, Y., Xiao, C., Futcher, B., Weissman, J. S., White, F. M., & Amon, A. (2013). A developmentally regulated translational control pathway establishes the meiotic chromosome segregation pattern. *Genes and Development*, *27*, 2147–2163. https://doi.org/10.1101/gad.224253.113

Bustin, S. A. (2013). Why there is no link between measles virus and autism. *Recent Advances in Autism Spectrum Disorders - Volume I*. https://doi.org/10.5772/52844

Curran, K. A., Karim, A. S., Gupta, A., & Alper, H. S. (2013). Use of expression-enhancing terminators in saccharomyces cerevisiae to increase mRNA half-life and improve gene expression control for metabolic engineering applications. *Metabolic Engineering*, *19*, 88–97. https://doi.org/10.1016/j.ymben.2013.07.001

Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M., & Gingeras, T. R. (2013). Star: Ultrafast universal rna-seq aligner. *Bioinformatics*, *29*, 15–21. https://doi.org/10.1093/BIOINFORMATICS/BTS635

Ito, Y., Yamanishi, M., Ikeuchi, A., Imamura, C., Tokuhiro, K., Kitagawa, T., & Matsuyama, T. (2013). Characterization of five terminator regions that increase the protein yield of a transgene in saccharomyces cerevisiae. *Journal of Biotechnology*, *168*(4), 486–492. https://doi.org/10.1016/j.jbiotec.2013.09.024

Kosuri, S., Goodman, D. B., Cambray, G., Mutalik, V. K., Gao, Y., Arkin, A. P., Endy, D., & Church, G. M. (2013). Composability of regulatory sequences controlling transcription and translation in escherichia coli. *Proceedings of the National Academy of Sciences*, *110*(34), 14024–14029. http://www.jstor.org/stable/42713044

Kretz, M., Siprashvili, Z., Chu, C., Webster, D. E., Zehnder, A., Qu, K., Lee, C. S., Flockhart, R. J., Groff, A. F., Chow, J., Johnston, D., Kim, G. E., Spitale, R. C., Flynn, R. A., Zheng, G. X., Aiyer, S., Raj, A., Rinn, J. L., Chang, H. Y., & Khavari, P. A. (2013). Control of somatic tissue differentiation by the long non-coding RNA TINCR. *Nature*, *493*, 231–235. https://doi.org/10.1038/nature11661

Mi, H., Muruganujan, A., & Thomas, P. D. (2013). Panther in 2013: Modeling the evolution of gene function, and other gene attributes, in the context of phylogenetic trees. *Nucleic Acids Research*, *41*, D377–D386. https://doi.org/10.1093/NAR/GKS1118

Pelechano, V., Wei, W., & Steinmetz, L. M. (2013). Extensive transcriptional heterogeneity revealed by isoform profiling. *Nature*, *497*, 127–131. https://doi.org/10.1038/nature12121

Sun, M., Schwalb, B., Pirkl, N., Maier, K. C., Schenk, A., Failmezger, H., Tresch, A., & Cramer, P. (2013). Global analysis of eukaryotic mRNA degradation reveals Xrn1-dependent buffering of transcript levels. *Molecular Cell*, *52*, 52–62. https://doi.org/10.1016/j.molcel.2013.09.010

Buchan, J. R., & Buchan, R. (2014). Mrnp granules. *http://dx.doi.org/10.4161/15476286.2014.972208*, *11*, 1019–1030. https://doi.org/10.4161/15476286.2014.972208

da Veiga Leprevost, F., Barbosa, V. C., Francisco, E. L., Perez-Riverol, Y., & Carvalho, P. C. (2014). On best practices in the development of bioinformatics software. *Frontiers in Genetics*, *5*, 199. https://doi.org/10.3389/FGENE.2014.00199

Geisberg, J. V., Moqtaderi, Z., Fan, X., Ozsolak, F., & Struhl, K. (2014). Global analysis of mrna isoform half-lives reveals stabilizing and destabilizing elements in yeast. *Cell*, *156*. https://doi.org/10.1016/j.cell.2013.12.026

Gelman, A., Carlin, J. B. B., Stern, H. S. S., & Rubin, D. B. B. (2014). Bayesian data analysis, third edition (texts in statistical science). *Book*, 675. https://doi.org/10.1007/s13398-014-0173-7.2

Goble, C. (2014). Better software, better research. *IEEE Internet Computing*, *18*, 4–8. https://doi.org/10.1109/MIC.2014.88

Jan, C. H., Williams, C. C., & Weissman, J. S. (2014). Principles of er cotranslational translocation revealed by proximity-specific ribosome profiling. *Science (New York, N.Y.)*, *346*, 1257521–1257521. https://doi.org/10.1126/SCIENCE.1257521

Khabsa, M., & Giles, C. L. (2014). The number of scholarly documents on the public web. *PLOS ONE*, *9*, e93949. https://doi.org/10.1371/JOURNAL.PONE.0093949

Liao, Y., Smyth, G. K., & Shi, W. (2014). Featurecounts: An efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics*, *30*, 923–930. https://doi.org/10.1093/BIOINFORMATICS/BTT656

Lichten, C. A., White, R., Clark, I. B., & Swain, P. S. (2014). Unmixing of fluorescence spectra to resolve quantitative time-series measurements of gene expression in plate readers. *BMC biotechnology*, *14*(1), 1–11. https://doi.org/10.1186/1472-6750-14-11

Love, M. I., Huber, W., & Anders, S. (2014). Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome Biology*, *15*, 1–21. https://doi.org/10.1186/S13059-014-0550-8/FIGURES/9

Moll, P., Ante, M., Seitz, A., & Reda, T. (2014). Quantseq 3' mrna sequencing for rna quantification. *Nature Methods 2014 11:12*, *11*, i–iii. https://doi.org/10.1038/nmeth.f.376

Niednery, A., Edelmanny, F. T., & Niessing, D. (2014). Of social molecules: The interactive assembly of ASH1 mRNA-transport complexes in yeast. *RNA Biology*, *11*, 998–1009. https://doi.org/10.4161/rna.29946

Pabinger, S., Rödiger, S., Kriegner, A., Vierlinger, K., & Weinhäusel, A. (2014). A survey of tools for the analysis of quantitative pcr (qpcr) data. *Biomolecular Detection and Quantification*, *1*, 23–33. https://doi.org/10.1016/J.BDQ.2014.08.002

Pingoud, A., Wilson, G. G., & Wende, W. (2014). Type ii restriction endonucleases—a historical perspective and more. *Nucleic Acids Research*, *42*, 7489–7527. https://doi.org/10.1093/NAR/GKU447

Sultan, M., Amstislavskiy, V., Risch, T., Schuette, M., Dökel, S., Ralser, M., Balzereit, D., Lehrach, H., & Yaspo, M. L. (2014). Influence of rna extraction methods and library selection schemes on rna-seq data. *BMC Genomics*, *15*, 1–13. https://doi.org/10.1186/1471-2164-15-675

Tajadini, M., Panjehpour, M., & Javanmard, S. H. (2014). Comparison of sybr green and taqman methods in quantitative real-time polymerase chain reaction analysis of four adenosine receptor subtypes. *Advanced Biomedical Research*, *3*, 85. https://doi.org/10.4103/2277-9175.127998

Wickham, H. (2014). Tidy data. *Journal of Statistical Software*, *59*, 1–23. https://doi.org/10.18637/JSS.V059.I10

Alberts, B., Johnson, A., Lewis, J., Morgan, D., Raff, M., Roberts, K., & Walter, P. (2015). Molecular biology of the cell. *Molecular Biology of the Cell*, 369–438. https://doi.org/10.1201/9781315735368

Bailey, T. L., Johnson, J., Grant, C. E., & Noble, W. S. (2015). The MEME suite. *Nucleic Acids Research*, *43*, W39–W49. https://doi.org/10.1093/nar/gkv416

de Klerk, E., & 't Hoen, P. A. (2015). Alternative mrna transcription, processing, and translation: Insights from rna sequencing. *Trends in Genetics*, *31*, 128–139. https://doi.org/10.1016/J.TIG.2015.01.001

Eisenberg, D. T., Kuzawa, C. W., & Hayes, M. G. (2015). Improving qpcr telomere length assays: Controlling for well position effects increases statistical power. *American Journal of Human Biology*, *27*, 570–575. https://doi.org/10.1002/AJHB.22690

Feuer, R., Vlaic, S., Arlt, J., Sawodny, O., Dahmen, U., Zanger, U. M., Thomas, M., & Kaderali, L. (2015). Lemming: A linear error model to normalize parallel quantitative real-time pcr (qpcr) data as an alternative to reference gene based methods. *PLoS ONE*, *10*. https://doi.org/10.1371/JOURNAL.PONE.0135852

Guo, Y., Dong, J., Zhou, T., Auxillos, J., Li, T., Zhang, W., Wang, L., Shen, Y., Luo, Y., Zheng, Y., et al. (2015). Yeastfab: The design and construction of standard biological parts for metabolic engineering in saccharomyces cerevisiae. *Nucleic acids research*, *43*(13), e88–e88. https://doi.org/10.1093/nar/gkv464

Inoue, F., & Ahituv, N. (2015). Decoding enhancers using massively parallel reporter assays. *Genomics*, *106*, 159–164. https://doi.org/10.1016/J.YGENO.2015.06.005

Krivoruchko, A., & Nielsen, J. (2015). Production of natural products through metabolic engineering of saccharomyces cerevisiae. *Current Opinion in Biotechnology*, *35*, 7–15. https://doi.org/10.1016/J.COPBIO.2014.12.004

Lawlor, B., & Walsh, P. (2015). Engineering bioinformatics: Building reliability, performance and productivity into bioinformatics software. *Bioengineered*, *6*, 193–203. https://doi.org/10.1080/21655979.2015.1050162

Li, Y., yu Chen, C., Kaye, A. M., & Wasserman, W. W. (2015). The identification of cis-regulatory elements: A review from a machine learning perspective. *Biosystems*, *138*, 6–17. https://doi.org/10.1016/J.BIOSYSTEMS.2015.10.002

Liang, M., Li, Z., Chen, T., & Zeng, J. (2015). Integrative data analysis of multi-platform cancer data with a multimodal deep learning approach. *IEEE/ACM Transactions*

on Computational Biology and Bioinformatics, *12*, 928–937. https://doi.org/10.1109/TCBB.2014.2377729

Presnyak, V., Alhusaini, N., Chen, Y. H., Martin, S., Morris, N., Kline, N., Olson, S., Weinberg, D., Baker, K. E., Graveley, B. R., & Coller, J. (2015). Codon optimality is a major determinant of mRNA stability. *Cell*, *160*, 1111–1124. https://doi.org/10.1016/j.cell.2015.02.029

Prins, P., Ligt, J. D., Tarasov, A., Jansen, R. C., Cuppen, E., & Bourne, P. E. (2015). Toward effective software solutions for big biology. *Nature Biotechnology 2015 33:7*, *33*, 686–687. https://doi.org/10.1038/nbt.3240

Risca, V. I., & Greenleaf, W. J. (2015). Beyond the linear genome: Paired-end sequencing as a biophysical tool. *Trends in Cell Biology*, *25*, 716–719. https://doi.org/10.1016/J.TCB.2015.08.004

Ritchie, M. E., Phipson, B., Wu, D., Hu, Y., Law, C. W., Shi, W., & Smyth, G. K. (2015). Limma powers differential expression analyses for rna-sequencing and microarray studies. *Nucleic Acids Research*, *43*, e47–e47. https://doi.org/10.1093/NAR/GKV007

Ruijter, J. M., Lefever, S., Anckaert, J., Hellemans, J., Pfaffl, M. W., Benes, V., Bustin, S. A., Vandesompele, J., & Untergasser, A. (2015). Rdml-ninja and rdmldb for standardized exchange of qpcr data. *BMC bioinformatics*, *16*. https://doi.org/10.1186/S12859-015-0637-6

Shalem, O., Sharon, E., Lubliner, S., Regev, I., Lotan-Pompan, M., Yakhini, Z., & Segal, E. (2015). Systematic dissection of the sequence determinants of gene 3' end mediated expression control. *PLOS Genetics*, *11*(4), 1–21. https://doi.org/10.1371/journal.pgen.1005147

Tommaso, P. D., Palumbo, E., Chatzou, M., Prieto, P., Heuer, M. L., & Notredame, C. (2015). The impact of docker containers on the performance of genomic pipelines. *PeerJ*, *2015*, e1273. https://doi.org/10.7717/PEERJ.1273

Trafimow, D., & Marks, M. (2015). Editorial. *Basic and Applied Social Psychology*, *37*, 1–2. https://doi.org/10.1080/01973533.2015.1012991

Wu, C., & Ma, S. (2015). A selective review of robust variable selection with applications in bioinformatics. *Briefings in Bioinformatics*, *16*, 873–883. https://doi.org/10.1093/BIB/BBU046

Baker, M. (2016). 1,500 scientists lift the lid on reproducibility. *Nature*, *533*, 452–454. https://doi.org/10.1038/533452a

Conesa, A., Madrigal, P., Tarazona, S., Gomez-Cabrero, D., Cervera, A., McPherson, A., Szcześniak, M. W., Gaffney, D. J., Elo, L. L., Zhang, X., & Mortazavi, A. (2016). A survey of best practices for rna-seq data analysis. *Genome Biology 2016 17:1*, *17*, 1–19. https://doi.org/10.1186/S13059-016-0881-8

Ewels, P., Magnusson, M., Lundin, S., & Käller, M. (2016). Multiqc: Summarize analysis results for multiple tools and samples in a single report. *Bioinformatics*, *32*, 3047–3048. https://doi.org/10.1093/bioinformatics/btw354

Freese, N. H., Norris, D. C., & Loraine, A. E. (2016). Integrated genome browser: Visual analytics platform for genomics. *Bioinformatics*, *32*, 2089–2095. https://doi.org/10.1093/BIOINFORMATICS/BTW069

Hombach, S., & Kretz, M. (2016). Non-coding rnas: Classification, biology and functioning. *Advances in Experimental Medicine and Biology*, *937*, 3–17. https://doi.org/10.1007/978-3-319-42059-2_1

Hughes, S. M. (2016). Plater: Read, tidy, and display data from microtiter plates. *The Journal of Open Source Software*, *1*(7). https://doi.org/10.21105/joss.00106

Jain, M., Olsen, H. E., Paten, B., & Akeson, M. (2016). The oxford nanopore minion: Delivery of nanopore sequencing to the genomics community. *Genome Biology 2016 17:1*, *17*, 1–11. https://doi.org/10.1186/S13059-016-1103-0

McKiernan, E. C., Bourne, P. E., Brown, C. T., Buck, S., Kenall, A., Lin, J., McDougall, D., Nosek, B. A., Ram, K., Soderberg, C. K., Spies, J. R., Thaney, K., Updegrove, A., Woo, K. H., & Yarkoni, T. (2016). How open science helps researchers succeed. *eLife*, *5*. https://doi.org/10.7554/ELIFE.16800

Pelechano, V., Wei, W., & Steinmetz, L. M. (2016). Genome-wide quantification of 5'-phosphorylated mrna degradation intermediates for analysis of ribosome dynamics. *Nature Protocols 2016 11:2*, *11*, 359–376. https://doi.org/10.1038/nprot.2016.026

Swain, P. S., Stevenson, K., Leary, A., Montano-Gutierrez, L. F., Clark, I. B., Vogel, J., & Pilizota, T. (2016). Inferring time derivatives including cell growth rates using gaussian processes. *Nature Communications*, *7*, 1–8. https://doi.org/10.1038/ncomms13766

Wickham, H. (2016). *Ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York.

Bustin, S., & Nolan, T. (2017). Talking the talk, but not walking the walk: Rt-qpcr as a paradigm for the lack of reproducibility in molecular research. *European Journal of Clinical Investigation*, *47*, 756–774. https://doi.org/10.1111/ECI.12801

Cheng, J., Maier, K. C., Žiga Avsec, Petra, R. U., & Gagneur, J. (2017). Cis-regulatory elements explain most of the mRNA stability variation across genes in yeast. *RNA*, *23*, 1648–1659. https://doi.org/10.1261/rna.062224.117

Dagnall, C. L., Hicks, B., Teshome, K., Hutchinson, A. A., Gadalla, S. M., Khincha, P. P., Yeager, M., & Savage, S. A. (2017). Effect of pre-analytic variables on the reproducibility of qpcr relative telomere length measurement. *PLOS ONE*, *12*, e0184098. https://doi.org/10.1371/JOURNAL.PONE.0184098

Geiger, R. S. (2017). Summary analysis of the 2017 github open source survey. https://doi.org/10.17605/OSF.IO/ENRQ5

Hu, P., Fabyanic, E., Kwon, D. Y., Tang, S., Zhou, Z., & Wu, H. (2017). Dissecting cell-type composition and activity-dependent transcriptional state in mammalian brains by massively parallel single-nucleus rna-seq. *Molecular Cell*, *68*, 1006–1015.e7. https://doi.org/10.1016/J.MOLCEL.2017.11.017

Hubstenberger, A., Courel, M., Bénard, M., Souquere, S., Ernoult-Lange, M., Chouaib, R., Yi, Z., Morlot, J. B., Munier, A., Fradet, M., Daunesse, M., Bertrand, E., Pierron, G., Mozziconacci, J., Kress, M., & Weil, D. (2017). P-body purification reveals the condensation of repressed mrna regulons. *Molecular Cell*, *68*, 144–157.e5. https://doi.org/10.1016/J.MOLCEL.2017.09.003

Khong, A., Matheny, T., Jain, S., Mitchell, S. F., Wheeler, J. R., & Parker, R. (2017). The stress granule transcriptome reveals principles of mrna accumulation in stress

granules. *Molecular Cell*, *68*, 808–820.e5. https://doi.org/10.1016/J.MOLCEL.2017.10.015

Mackenzie, I. R., Nicholson, A. M., Sarkar, M., Messing, J., Purice, M. D., Pottier, C., Annu, K., Baker, M., Perkerson, R. B., Kurti, A., Matchett, B. J., Mittag, T., Temirov, J., Hsiung, G. Y. R., Krieger, C., Murray, M. E., Kato, M., Fryer, J. D., Petrucelli, L., … Rademakers, R. (2017). Tia1 mutations in amyotrophic lateral sclerosis and frontotemporal dementia promote phase separation and alter stress granule dynamics. *Neuron*, *95*, 808–816.e9. https://doi.org/10.1016/J.NEURON.2017.07.025

Mallona, I., Díez-Villanueva, A., Martín, B., & Peinado, M. A. (2017). Chainy: An universal tool for standardized relative quantification in real-time pcr. *Bioinformatics*, *33*, 1411–1413. https://doi.org/10.1093/BIOINFORMATICS/BTW839

Roediger, S., Burdukiewicz, M., Spiess, A.-N., & Blagodatskikh, K. (2017). Enabling reproducible real-time quantitative pcr research: The rdml package. *Bioinformatics*. https://doi.org/10.1093/bioinformatics/btx528

Smith, T., Heger, A., & Sudbery, I. (2017). Umi-tools: Modeling sequencing errors in unique molecular identifiers to improve quantification accuracy. *Genome Research*, *27*, 491–499. https://doi.org/10.1101/gr.209601.116

Taschuk, M., & Wilson, G. (2017). Ten simple rules for making research software more robust. *PLOS Computational Biology*, *13*, e1005412. https://doi.org/10.1371/JOURNAL.PCBI.1005412

Tommaso, P. D., Chatzou, M., Floden, E. W., Barja, P. P., Palumbo, E., & Notredame, C. (2017). Nextflow enables reproducible computational workflows. *Nature Biotechnology 2017 35:4*, *35*, 316–319. https://doi.org/10.1038/nbt.3820

Wilson, G., Bryan, J., Cranston, K., Kitzes, J., Nederbragt, L., & Teal, T. K. (2017). Good enough practices in scientific computing. *PLOS Computational Biology*, *13*, e1005510. https://doi.org/10.1371/JOURNAL.PCBI.1005510

Xie, R., Wen, J., Quitadamo, A., Cheng, J., & Shi, X. (2017). A deep auto-encoder model for gene expression prediction. *BMC Genomics*, *18*, 39–49. https://doi.org/10.1186/S12864-017-4226-0

Zheng, G. X., Terry, J. M., Belgrader, P., Ryvkin, P., Bent, Z. W., Wilson, R., Ziraldo, S. B., Wheeler, T. D., McDermott, G. P., Zhu, J., Gregory, M. T., Shuga, J., Montesclaros, L., Underwood, J. G., Masquelier, D. A., Nishimura, S. Y., Schnall-Levin, M., Wyatt, P. W., Hindson, C. M., … Bielas, J. H. (2017). Massively parallel digital transcriptional profiling of single cells. *Nature Communications 2017 8:1*, *8*, 1–12. https://doi.org/10.1038/ncomms14049

Špela Baebler, Svalina, M., Petek, M., Stare, K., Rotter, A., Pompe-Novak, M., & Gruden, K. (2017). Quantgenius: Implementation of a decision support system for qpcr-based gene quantification. *BMC Bioinformatics*, *18*, 1–11. https://doi.org/10.1186/S12859-017-1688-7/FIGURES/5

Ahmed, M., & Kim, D. R. (2018). Pcr: An r package for quality assessment, analysis and testing of qpcr data. *PeerJ*, *6*. https://doi.org/10.7717/PEERJ.4473

Andreou, A. I., & Nakayama, N. (2018). Mobius assembly: A versatile golden-gate framework towards universal DNA assembly. *PLOS ONE*, *13*(1), 1–18. https://doi.org/10.1371/journal.pone.0189892

Cereci, I., & Karakaya, Z. (2018). Need for a software development methodology for research-based software projects. *UBMK 2018 - 3rd International Conference on Computer Science and Engineering*, 648–651. https://doi.org/10.1109/UBMK.2018.8566613

Chan, L. Y., Mugler, C. F., Heinrich, S., Vallotton, P., & Weis, K. (2018). Non-invasive measurement of mRNA decay reveals translation initiation as the major determinant of mRNA stability. *eLife*, *7*. https://doi.org/10.7554/eLife.32536

Ching, T., Himmelstein, D. S., Beaulieu-Jones, B. K., Kalinin, A. A., Do, B. T., Way, G. P., Ferrero, E., Agapow, P. M., Zietz, M., Hoffman, M. M., Xie, W., Rosen, G. L., Lengerich, B. J., Israeli, J., Lanchantin, J., Woloszynek, S., Carpenter, A. E., Shrikumar, A., Xu, J., … Greene, C. S. (2018). Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, *15*. https://doi.org/10.1098/RSIF.2017.0387

Corbett, A. H. (2018). Post-transcriptional regulation of gene expression and human disease. *Current Opinion in Cell Biology*, *52*, 96–104. https://doi.org/10.1016/J.CEB.2018.02.011

Evans, C., Hardin, J., & Stoebel, D. M. (2018). Selecting between-sample rna-seq normalization methods from the perspective of their assumptions. *Briefings in Bioinformatics*, *19*, 776–792. https://doi.org/10.1093/BIB/BBX008

Fellows, I. (2018). *Wordcloud: Word clouds* [R package version 2.6]. https://CRAN.R-project.org/package=wordcloud

Garcia-Ruiz, E., Auxillos, J., Li, T., Dai, J., & Cai, Y. (2018). Yeastfab: High-throughput genetic parts construction, measurement, and pathway engineering in yeast. *Methods in enzymology*, *608*, 277–306. https://doi.org/10.1016/bs.mie.2018.05.003

Geiger, R. S., Varoquaux, N., Mazel-Cabasse, C., & Holdgraf, C. (2018). The types, roles, and practices of documentation in data analytics open source software libraries: A collaborative ethnography of documentation work. *Computer Supported Cooperative Work: CSCW: An International Journal*, *27*, 767–802. https://doi.org/10.1007/S10606-018-9333-1/

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM Computing Surveys (CSUR)*, *51*. https://doi.org/10.1145/3236009

Laros, J. F. (2018). Demultiplex. https://github.com/jfjlaros/demultiplex

Sadath, L., Karim, K., & Gill, S. (2018). Extreme programming implementation in academia for software engineering sustainability. *2018 Advances in Science and Engineering Technology International Conferences, ASET 2018*, 1–6. https://doi.org/10.1109/ICASET.2018.8376925

Soneson, C., & Robinson, M. D. (2018). Bias, robustness and scalability in single-cell differential expression analysis. *Nature Methods 2018 15:4*, *15*, 255–261. https://doi.org/10.1038/nmeth.4612

Treeck, B. V., Protter, D. S., Matheny, T., Khong, A., Link, C. D., & Parker, R. (2018). Rna self-assembly contributes to stress granule formation and defining the stress granule transcriptome. *Proceedings of the National Academy of Sciences of the United States of America*, *115*, 2734–2739. https://doi.org/10.1073/PNAS.1800038115

Witte, A. K., Sickha, R., Mester, P., Fister, S., Schoder, D., & Rossmanith, P. (2018). Essential role of polymerases for assay performance – impact of polymerase replacement in a well-established assay. *Biomolecular Detection and Quantification*, *16*, 12. https://doi.org/10.1016/J.BDQ.2018.10.002

Aghajani, E., Nagy, C., Vega-Marquez, O. L., Linares-Vasquez, M., Moreno, L., Bavota, G., & Lanza, M. (2019). Software documentation issues unveiled. *Proceedings - International Conference on Software Engineering*, *2019-May*, 1199–1210. https://doi.org/10.1109/ICSE.2019.00122

Attwood, T. K., Blackford, S., Brazas, M. D., Davies, A., & Schneider, M. V. (2019). A global perspective on evolving bioinformatics and data science training needs. *Briefings in Bioinformatics*, *20*, 398–404. https://doi.org/10.1093/BIB/BBX100

Balázs, Z., Tombácz, D., Csabai, Z., Moldován, N., Snyder, M., & Boldogkoi, Z. (2019). Template-switching artifacts resemble alternative polyadenylation. *BMC Genomics*, *20*, 1–10. https://doi.org/10.1186/S12864-019-6199-7

Cramer, P. (2019). Organization and regulation of gene transcription. *Nature 2019 573:7772*, *573*, 45–54. https://doi.org/10.1038/s41586-019-1517-4

Gomez-Diaz, T., & Recio, T. (2019). On the evaluation of research software: The cdur procedure. *F1000Research 2019 8:1353*, *8*, 1353. https://doi.org/10.12688/f1000research.19994.2

Kim, D., Paggi, J. M., Park, C., Bennett, C., & Salzberg, S. L. (2019). Graph-based genome alignment and genotyping with hisat2 and hisat-genotype. *Nature Biotechnology*, *37*, 907–915. https://doi.org/10.1038/s41587-019-0201-4

Krähenbühl, S., Studer, F., Guirou, E., Deal, A., Mächler, P., Hosch, S., Mpina, M., Mswata, S., Daubenberger, C., & Schindler, T. (2019). Elimu-mdx: A web-based, open-source platform for storage, management and analysis of diagnostic qpcr data. *BioTechniques*, *68*, 22–27. https://doi.org/10.2144/BTN-2019-0064

Matheny, T., Rao, B. S., & Parker, R. (2019). Transcriptome-wide comparison of stress granules and p-bodies reveals that translation plays a major role in rna partitioning. *Molecular and Cellular Biology*, *39*. https://doi.org/10.1128/MCB.00313-19

Queiroz, R. M., Smith, T., Villanueva, E., Marti-Solano, M., Monti, M., Pizzinga, M., Mirea, D. M., Ramakrishna, M., Harvey, R. F., Dezi, V., Thomas, G. H., Willis, A. E., & Lilley, K. S. (2019). Comprehensive identification of rna–protein interactions in any organism using orthogonal organic phase separation (oops). *Nature Biotechnology 2019 37:2*, *37*, 169–178. https://doi.org/10.1038/s41587-018-0001-2

Rancurel, C., van Tran, T., Elie, C., & Hilliou, F. (2019). Satqpcr: Website for statistical analysis of real-time quantitative pcr data. *Molecular and Cellular Probes*, *46*, 101418. https://doi.org/10.1016/J.MCP.2019.07.001

Stark, R., Grzelak, M., & Hadfield, J. (2019). Rna sequencing: The teenage years. *Nature Reviews Genetics 2019 20:11*, *20*, 631–656. https://doi.org/10.1038/s41576-019-0150-2

Tancharoen, C., Sukjee, W., Thepparit, C., Jaimipuk, T., Auewarakul, P., Thitithanyanont, A., & Sangma, C. (2019). Electrochemical biosensor based on surface imprinting for zika virus detection in serum. *ACS Sensors*, *4*, 69–75. https://doi.org/10.1021/ACSSENSORS.8B00885

Tang, B., Pan, Z., Yin, K., & Khateeb, A. (2019). Recent advances of deep learning in bioinformatics and computational biology. *Frontiers in Genetics*, *10*, 214. https://doi.org/10.3389/FGENE.2019.00214

Vijayabaskar, M. S., Goode, D. K., Obier, N., Lichtinger, M., Emmett, A. M., Abidin, F. N., Shar, N., Hannah, R., Assi, S. A., Lie-A-Ling, M., Gottgens, B., Lacaud, G., Kouskoff, V., Bonifer, C., & Westhead, D. R. (2019). Identification of gene specific cis-regulatory elements during differentiation of mouse embryonic stem cells: An integrative approach using high-throughput datasets. *PLoS Computational Biology*, *15*, e1007337. https://doi.org/10.1371/journal.pcbi.1007337

Wan, X., Volpetti, F., Petrova, E., French, C., Maerkl, S. J., & Wang, B. (2019). Cascaded amplifying circuits enable ultrasensitive cellular sensors for toxic metals. *Nature Chemical Biology 2019 15:5*, *15*, 540–548. https://doi.org/10.1038/s41589-019-0244-3

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., … Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://tidyverse.tidyverse.org/articles/paper.html

Zanardi, N., Morini, M., Tangaro, M. A., Zambelli, F., Bosco, M. C., Varesio, L., Eva, A., & Cangelosi, D. (2019). Pipe-t: A new galaxy tool for the analysis of rt-qpcr expression data. *Scientific Reports 2019 9:1*, *9*, 1–12. https://doi.org/10.1038/s41598-019-53155-9

Adams, G. (2020). A beginner's guide to rt-pcr, qpcr and rt-qpcr. *The Biochemist*, *42*, 48–53. https://doi.org/10.1042/BIO20200034

Dhillon, N., Shelansky, R., Townshend, B., Jain, M., Boeger, H., Endy, D., & Kamakaka, R. (2020). Permutational analysis of saccharomyces cerevisiae regulatory elements. *Synthetic biology (Oxford, England)*, *5*. https://doi.org/10.1093/synbio/ysaa007

Ewels, P. A., Peltzer, A., Fillinger, S., Patel, H., Alneberg, J., Wilm, A., Garcia, M. U., Tommaso, P. D., & Nahnsen, S. (2020). The nf-core framework for community-curated bioinformatics pipelines. *Nature Biotechnology 2020 38:3*, *38*, 276–278. https://doi.org/10.1038/s41587-020-0439-x

Geisberg, J. V., Moqtaderi, Z., & Struhl, K. (2020). The transcriptional elongation rate regulates alternative polyadenylation in yeast. *eLife*, *9*, 1–55. https://doi.org/10.7554/ELIFE.59810

Iserman, C., Altamirano, C. D., Jegers, C., Friedrich, U., Zarin, T., Fritsch, A. W., Mittasch, M., Domingues, A., Hersemann, L., Jahnel, M., Richter, D., Guenther, U. P., Hentze, M. W., Moses, A. M., Hyman, A. A., Kramer, G., Kreysing, M., Franzmann, T. M., & Alberti, S. (2020). Condensation of ded1p promotes a translational switch from housekeeping to stress protein production. *Cell*, *181*, 818–831.e19. https://doi.org/10.1016/J.CELL.2020.04.009

Klein, J. C., Agarwal, V., Inoue, F., Keith, A., Martin, B., Kircher, M., Ahituv, N., & Shendure, J. (2020). A systematic evaluation of the design and context dependencies of massively parallel reporter assays. *Nature Methods*, *17*. https://doi.org/10.1038/s41592-020-0965-y

Luo, Y., Hitz, B. C., Gabdank, I., Hilton, J. A., Kagda, M. S., Lam, B., Myers, Z., Sud, P., Jou, J., Lin, K., Baymuradov, U. K., Graham, K., Litton, C., Miyasato, S. R., Strattan, J. S., Jolanki, O., Lee, J. W., Tanaka, F. Y., Adenekan, P., … Cherry, J. M. (2020). New developments on the encyclopedia of dna elements (encode) data portal. *Nucleic Acids Research*, *48*, D882–D889. https://doi.org/10.1093/NAR/GKZ1062

Nersisyan, L., Ropat, M., & Pelechano, V. (2020). Improved computational analysis of ribosome dynamics from 5'p degradome data using fivepseq. *NAR Genomics and Bioinformatics*, *2*. https://doi.org/10.1093/nargab/lqaa099

Ng, P. C., Wong, E. D., MacPherson, K. A., Aleksander, S., Argasinska, J., Dunn, B., Nash, R. S., Skrzypek, M. S., Gondwe, F., Jha, S., Karra, K., Weng, S., Miyasato, S., Simison, M., Engel, S. R., & Cherry, J. M. (2020). Transcriptome visualization and data availability at the saccharomyces genome database. *Nucleic Acids Research*, *48*, D743–D748. https://doi.org/10.1093/nar/gkz892

Nour, A. M. A., Nemer, G., & Khalil, A. (2020). The miqe guidelines' tenth anniversary: The good and bad students. *Gene Reports*, *19*, 100630. https://doi.org/10.1016/J.GENREP.2020.100630

R Core Team. (2020). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria.

Rios, N., Mendes, L., Cerdeiral, C., Magalhães, A. P. F., Perez, B., Correal, D., Astudillo, H., Seaman, C., Izurieta, C., Santos, G., & Spínola, R. O. (2020). Hearing the voice of software practitioners on causes, effects, and practices to deal with documentation debt. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12045 LNCS*, 55–70. https://doi.org/10.1007/978-3-030-44429-7_4/FIGURES/1

Rudis, B., & Embrey, B. (2020). *Pluralize: Pluralize and 'singularize' any (english) word* [R package version 0.2.0]. https://CRAN.R-project.org/package=pluralize

Schwaber, K., & Sutherland, J. (2020). The scrum guide. https://www.scrum.org/resources/scrum-guide

Wilkinson, M. E., Charenton, C., & Nagai, K. (2020). Rna splicing by the spliceosome. *https://doi.org/10.1146/annurev-biochem-091719-064225*, *89*, 359–388. https://doi.org/10.1146/ANNUREV-BIOCHEM-091719-064225

Zhao, S., Ye, Z., & Stanton, R. (2020). Misuse of rpkm or tpm normalization when comparing across samples and sequencing protocols. *RNA*, *26*, 903–909. https://doi.org/10.1261/RNA.074922.120

Bayne, R. A., Jayachandran, U., Kasprowicz, A., Bresson, S., Tollervey, D., Wallace, E. W. J., & Cook, A. G. (2021). Yeast Ssd1 is a non-enzymatic member of the RNase II family with an alternative RNA recognition site. *Nucleic Acids Research*, *50*(5), 2923–2937. https://doi.org/10.1093/nar/gkab615

Bustin, S., Mueller, R., Shipley, G., & Nolan, T. (2021). Covid-19 and diagnostic testing for sars-cov-2 by rt-qpcr—facts and fallacies. *International Journal of Molecular Sciences 2021, Vol. 22, Page 2459*, *22*, 2459. https://doi.org/10.3390/IJMS22052459

Das, S., Vera, M., Gandin, V., Singer, R. H., & Tutucci, E. (2021). Intracellular mrna transport and localized translation. *Nature Reviews Molecular Cell Biology 2021 22:7*, *22*, 483–504. https://doi.org/10.1038/s41580-021-00356-8

EMBL-EBI. (2021). Embl-ebi highlights 2021. https://www.embl.org/documents/document/embl-ebi-highlights-2021/

Greener, J. G., Kandathil, S. M., Moffat, L., & Jones, D. T. (2021). A guide to machine learning for biologists. *Nature Reviews Molecular Cell Biology 2021 23:1*, *23*, 40–55. https://doi.org/10.1038/s41580-021-00407-0

Hu, T., Chitnis, N., Monos, D., & Dinh, A. (2021). Next-generation sequencing technologies: An overview. *Human Immunology*, *82*, 801–811. https://doi.org/10.1016/J.HUMIMM.2021.02.012

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., ... Hassabis, D. (2021). Highly accurate protein structure prediction with alphafold. *Nature 2021 596:7873*, *596*, 583–589. https://doi.org/10.1038/s41586-021-03819-2

Maussion, G., Thomas, R. A., Demirova, I., Gu, G., Cai, E., Chen, C. X., Abdian, N., Strauss, T. J., Kelaï, S., Nauleau-Javaudin, A., Beitel, L. K., Ramoz, N., Gorwood, P., & Durcan, T. M. (2021). Auto-qpcr; a python-based web app for automated and reproducible analysis of qpcr data. *Scientific Reports 2021 11:1*, *11*, 1–14. https://doi.org/10.1038/s41598-021-99727-6

Ng, H. F., Ngeow, Y. F., Rahman, A., Correspondence, M., & Ngeow, Y. F. (2021). A simple spreadsheet-based method for relative quantification using quantitative real-time pcr. *Biochemistry and Molecular Biology Education*. https://doi.org/10.1002/BMB.21596

Olaechea-Lázaro, S., García-Santisteban, I., Pineda, J. R., Badiola, I., Alonso, S., Bilbao, J. R., & Fernandez-Jimenez, N. (2021). Shinycurves, a shiny web application to analyse multisource qpcr amplification data: A covid-19 case study. *BMC Bioinformatics*, *22*, 1–6. https://doi.org/10.1186/S12859-021-04392-1

Wickham, H., Danenberg, P., Csárdi, G., & Eugster, M. (2021). *Roxygen2: In-line documentation for r* [R package version 7.1.2]. https://CRAN.R-project.org/package=roxygen2

Zaghlool, A., Niazi, A., Åsa K. Björklund, Westholm, J. O., Ameur, A., & Feuk, L. (2021). Characterization of the nuclear and cytosolic transcriptomes in human brain tissue reveals new insights into the subcellular distribution of rna transcripts. *Scientific Reports 2021 11:1*, *11*, 1–12. https://doi.org/10.1038/s41598-021-83541-1

Zhang, Y., & Pelechano, V. (2021). High-throughput 5'p sequencing enables the study of degradation-associated ribosome stalls. *Cell Reports Methods*, *1*, 100001. https://doi.org/10.1016/j.crmeth.2021.100001

Barker, M., Hong, N. P. C., Katz, D. S., Lamprecht, A.-L., Martinez-Ortiz, C., Psomopoulos, F., Harrow, J., Castro, L. J., Gruenpeter, M., Martinez, P. A., & Honeyman, T. (2022). Introducing the fair principles for research software. *Scientific Data 2022 9:1*, *9*, 1–6. https://doi.org/10.1038/s41597-022-01710-x

Dreier, M., Meola, M., Berthoud, H., Shani, N., Wechsler, D., & Junier, P. (2022). High-throughput qpcr and 16s rrna gene amplicon sequencing as complementary methods for the investigation of the cheese microbiota. *BMC microbiology*, *22*. https://doi.org/10.1186/S12866-022-02451-Y

Hermann, S., & Fehr, J. (2022). Documenting research software in engineering science. *Scientific Reports 2022 12:1*, *12*, 1–11. https://doi.org/10.1038/s41598-022-10376-9

Hong, N. P. C., Katz, D. S., Barker, M., Lamprecht, A.-L., Martinez, C., Psomopoulos, F. E., Harrow, J., Castro, L. J., Gruenpeter, M., Martinez, P. A., Honeyman, T., Struck, A., Lee, A., Loewe, A., van Werkhoven, B., Garijo, D., Plomp, E., Genova, F., Shanahan, H., … WG, F. (2022). Fair principles for research software (fair4rs principles). https://doi.org/10.15497/RDA00065

Jovic, D., Liang, X., Zeng, H., Lin, L., Xu, F., & Luo, Y. (2022). Single-cell rna sequencing technologies and applications: A brief overview. *Clinical and Translational Medicine*, *12*. https://doi.org/10.1002/CTM2.694

Kleinschmidt, N. (2022). *Qpcr - a python module for easy and versatile qpcr data analysis for small-scale datasets and high-throughput* (Version 3.1.5). https://github.com/NoahHenrikKleinschmidt/qpcr.git

Schindler, D., Bensmann, F., Dietze, S., & Krüger, F. (2022). The role of software in science: A knowledge graph-based analysis of software mentions in pubmed central. *PeerJ Computer Science*, *8*, e835. https://doi.org/10.7717/PEERJ-CS.835

Strasser, C., Hertweck, K., Greenberg, J., Taraborelli, D., & Vu, E. (2022). Ten simple rules for funding scientific open source software. *PLOS Computational Biology*, *18*, e1010627. https://doi.org/10.1371/JOURNAL.PCBI.1010627

The Turing Way Community. (2022). The turing way: A handbook for reproducible, ethical and collaborative research. https://doi.org/10.5281/ZENODO.6909298

Wayment-Steele, H. K., Kladwang, W., Watkins, A. M., Kim, D. S., Tunguz, B., Reade, W., Demkin, M., Romano, J., Wellington-Oguri, R., Nicol, J. J., Gao, J., Onodera, K., Fujikawa, K., Mao, H., Vandewiele, G., Tinti, M., Steenwinckel, B., Ito, T., Noumi, T., … Das, R. (2022). Deep learning models for predicting rna degradation via dual crowdsourcing. *Nature Machine Intelligence*, *4*. https://doi.org/10.1038/s42256-022-00571-8

Wickham, H., & Seidel, D. (2022). *Scales: Scale functions for visualization* [R package version 1.2.0]. https://CRAN.R-project.org/package=scales

Costa-Silva, J., Domingues, D. S., Menotti, D., Hungria, M., & Lopes, F. M. (2023). Temporal progress of gene expression analysis with rna-seq data: A review on the relationship between computational methods. *Computational and Structural Biotechnology Journal*, *21*, 86. https://doi.org/10.1016/J.CSBJ.2022.11.051

He, S., Gao, B., Sabnis, R., & Sun, Q. (2023). Rnadegformer: Accurate prediction of mrna degradation at nucleotide resolution with deep learning. *Briefings in bioinformatics*, *24*. https://doi.org/10.1093/bib/bbac581

Lee, M. E., DeLoache, W. C., Cervantes, B., & Dueber, J. E. (n.d.). https://doi.org/10.1021/sb500366v