Machine Learning Methods for MicroRNA Target Prediction

Ryan Phelan

School of Biological Sciences
University of East Anglia

A thesis submitted for the degree of
Doctor of Philosophy
September 2023

Acknowledgements

I would like to thank my supervisor Simon Moxon for providing me with this opportunity and guiding me throughout my MSc and PhD. I would also like to thank my family and friends for years of encouragement and support, in particular my parents, Jamie and Leigh, and my brother, Josh.

Funding

This work was supported by the UKRI Biotechnology and Biological Sciences Research Council Norwich Research Park Biosciences Doctoral Training Partnership [Grant number BB/M011216/1].

The research presented in this paper was carried out on the High Performance Computing Cluster supported by the Research and Specialist Computing Support service at the University of East Anglia.

Abstract

MicroRNAs are small non-coding RNA molecules that form a post-transcriptional layer of gene regulation. microRNA binds with messenger RNA in order to repress translation and accelerate its degradation, ultimately downregulating the expression of genes. The mechanics of these bindings in animals are complex and entrenched in a myriad of contextual factors which influence the specificity and efficacy of potential interactions.

This thesis describes the development of miRsight, a novel target prediction tool utilising advanced machine learning techniques. miRsight is trained using 44 target recognition features compiled through testing on published microRNA-transfected RNA sequencing data, an experimental procedure in which microRNA molecules are introduced into a sample to quantify their impact on gene expression. In addition to the tool itself, a database of pre-computed predictions is hosted at `https://mirsight.info`, which also provides search, filter, and export functionality for user convenience.

The results of this study indicate that miRsight is able to more effectively predict and rank microRNA targets compared to popular target prediction tools. This is validated by examining the downregulation of gene expression from predicted targets using microRNA transfection. In the 12 samples reserved for testing, miRsight is shown to more consistently identify true targets in the top 100, 300 and 500 of predictions by rank compared to TargetScan, MirTarget and DIANA-microT. Additionally, miRsight is capable of producing several thousand total predictions for each microRNA while maintaining this high rate of prediction accuracy.

Contents

Abstract **3**

1 Introduction **19**

2 Background **21**

2.1 Nucleic Acids 21

2.2 Genetic Regulatory Network 22

2.2.1 Gene Expression 23

2.2.1.1 mRNA Transcription 24

2.2.1.2 mRNA Processing 25

2.2.2 Post-transcriptional Regulation 26

2.2.2.1 miRNA Biogenesis 26

2.2.2.2 miRNA-mediated Regulation 27

2.3 Genetic Sequencing 28

2.3.1 Sequence Reconstruction 29

2.3.2 RNA-seq 30

2.3.2.1 miRNA Transfection 30

2.3.3 SHAPE-seq 31

2.4 Mechanics of miRNA:mRNA Binding 32

2.4.1 Seed Binding 33

2.4.2 Supplementary Binding 35

2.4.2.1 Compensatory Binding 36

2.4.3 Target Site Accessibility 36

2.4.4 Binding Gaps 37

2.4.5 Wobble Pairs 38

2.4.6 Evolutionary Conservation 39

2.4.7 Target Site Abundance 40
2.5 Computational Prediction of miRNA Targets 41
2.5.1 Machine Learning 42
2.5.1.1 Sampling and Splitting 43
2.5.1.2 Decision Tree 44
2.5.1.3 Random Forest 45
2.5.1.4 Support Vector Machine 46
2.5.1.5 Deep Neural Network 47
2.5.2 Prediction Tools 49
2.5.2.1 TargetScan 51
2.5.2.2 miRanda-mirSVR 52
2.5.2.3 MirTarget 52
2.5.2.4 DIANA-microT 53
2.5.3 Seed Definition 53
2.5.4 Common Target Recognition Features 55

3 Rule-based prediction of miRNA:mRNA Targets 56
3.1 Summary 56
3.2 Methods 59
3.2.1 Transfection Dataset 59
3.2.2 Read Trimming 60
3.2.3 Sequence Alignment 62
3.2.4 Gene Annotation 63
3.2.5 Seed Target Detection 64
3.2.6 Differential Expression Analysis 64
3.2.6.1 Filtering 65
3.2.6.2 Evaluating Transfection Quality 70
3.2.7 RNA Folding 71
3.2.7.1 Predictive Structural Accessibility 72
3.2.7.2 Predictive Base Pairing 73
3.2.7.3 Target Window Extraction 73
3.2.7.4 Supplementary Site Definition 74
3.2.8 Conservation Scoring 76
3.3 Results 77
3.3.1 Isolated Features 77

3.3.1.1 Seed Type . . . 77
3.3.1.2 Multiple Seed Target Sites . . . 78
3.3.1.3 Binding Stability . . . 80
3.3.1.4 Target Site Accessibility . . . 82
3.3.1.5 Supplementary Binding . . . 84
3.3.1.6 Evolutionary Conservation . . . 86
3.3.2 Rule-based Prediction . . . 89
3.4 Discussion . . . 94

4 Site Accessibility Measures 97
4.1 Summary . . . 97
4.2 Methods . . . 97
4.2.1 Secondary Structure Stability . . . 97
4.2.2 Local RNA Pairing Probability . . . 98
4.2.3 RNA Structure Analysis . . . 100
4.3 Results . . . 102
4.3.1 Local AU Content . . . 103
4.3.2 RNAplfold . . . 105
4.3.3 SHAPE-seq . . . 108
4.4 Discussion . . . 111

5 miRNA:mRNA Target Prediction using Machine Learning 114
5.1 Summary . . . 114
5.2 Methods . . . 114
5.2.1 Tool Architecture . . . 114
5.2.1.1 Bioinformatics Pipeline . . . 118
5.2.2 Functionality Enhancements . . . 118
5.2.2.1 Runtime Settings . . . 118
5.2.2.2 Output Caching . . . 118
5.2.2.3 Dynamic Experiment Loading . . . 119
5.2.2.4 Local Annotation Parsing . . . 119
5.2.2.5 Genome-wide Conservation Caching . . . 120
5.2.3 Dataset Construction . . . 123
5.2.3.1 Additional Datasets . . . 123
5.2.3.2 Data Quality Evaluation . . . 124

5.2.3.3 Train and Test Split 126
5.2.4 Data Preparation 130
5.2.4.1 Collinearity Reduction 130
5.2.4.2 Common Features 134
5.2.4.3 Full Feature List 136
5.2.4.4 Categorical Data Encoding 139
5.2.4.5 Incomplete Data Imputation 140
5.2.4.6 Scale Transformation 143
5.2.4.7 Outlier Removal 145
5.2.4.8 Normalisation 145
5.2.5 Machine Learning 146
5.2.5.1 Class Label Assignment 146
5.2.5.2 Evaluative Metrics 149
5.3 Results 151
5.3.1 Hyperparameter Tuning 151
5.3.1.1 Decision Tree 151
5.3.1.2 Random Forest 152
5.3.1.3 Support Vector Machine 153
5.3.1.4 Deep Neural Network 154
5.3.2 Classifier Performance 156
5.3.3 Benchmarking 157
5.3.4 Feature Importance 161
5.4 Discussion 163

6 Software Development 168

6.1 Summary 168
6.2 Methods 169
6.2.1 Command Line Application 169
6.2.2 Web Application 170
6.2.2.1 Database Layout 170
6.2.2.2 Back-end Architecture 171
6.2.2.3 Front-end Design and Development 172
6.2.2.4 Deployment 173
6.3 Results 174
6.3.1 Testing 174

6.3.2 User Interface 175
6.3.3 Responsive Design 178
6.4 Discussion 179

7 Conclusion and Future Work **181**
7.1 Summary 181
7.2 Future Work 183
7.3 Conclusion 185

A Additional Datasets **212**

B Individual Benchmark Results **219**

C Unused Datasets **232**

List of Tables

2.1 Summary of popular target prediction tools 1/2 49
2.2 Summary of popular target prediction tools 2/2 50
2.3 Seed type usage of several popular modern tools 54
2.4 Common features of several popular modern tools 55

3.1 An overview of 01-liu-HeLa 59
3.2 Annotations from biomaRt used to support target prediction 63
3.3 Mann-Whitney p-values for each transfection experiment 71

4.1 Example RNAplfold output 99
4.2 An overview of SH-sun-HS 100

5.1 Matrix size as a trade-off against transcript loss 122
5.2 Mann-Whitney p-values for each new transfection experiment 125
5.3 Transfected miRNAs with fully unique seeds 127
5.4 Transfected same-family miRNAs with the AGCAGC 6mer 128
5.5 Transfected same-family miRNAs with the AAUACU 6mer 128
5.6 Transfected same-family miRNAs with the AGUGCA 6mer 128
5.7 Transfected same-family miRNAs with the GGAAUG 6mer 128
5.8 Transfected same-family miRNAs with the GAGGUA 6mer 128
5.9 Dataset splits after accounting for same-family miRNAs 129
5.10 Two-way collinear features 132
5.11 Three-way collinear features 132
5.12 Clustered collinear features 133
5.13 Seed type usage in comparison to other tools 135
5.14 Common features in comparison to other tools 135
5.15 Features relating to seed definitions 136

5.16 Features relating to binding stability 136
5.17 Features relating to supplementary sites 137
5.18 Features relating to conservation 137
5.19 Features relating to site accessibility 138
5.20 Features relating to MTS 139
5.21 Features relating to target positioning 139
5.22 Features relating to the entire miRNA and complementary mRNA bases 139
5.23 Prediction tool data summary 158

6.1 Prediction table data type declaration 171

A.1 Additional datasets used to support ML 212

C.1 Dataset EX-guo-U20S summary 232

List of Figures

2.1 Carbon atoms in a furanose molecule 21
2.2 DNA canonical base pairing 22
2.3 Central dogma of molecular biology 23
2.4 Creation of pre-mRNA by RNAP II 24
2.5 RNA splicing 25
2.6 Structure of mature mRNA 25
2.7 Trafficking miRNA from Drosha to Dicer 26
2.8 Canonical pathway of miRNA biogenesis 27
2.9 Binding methods of miRNA to degrade or suppress mRNA 27
2.10 Nucleobases in a sequenced read 28
2.11 Sequence assembly process 29
2.12 Differential expression analysis 30
2.13 Cumulative expression fold change plot 31
2.14 SHAPE-seq reactivity values 32
2.15 Model of miRNA:mRNA interaction 32
2.16 Potential seed identifiers 33
2.17 Core seed type summary 34
2.18 Comparison of relative seed site efficacy 34
2.19 Key bases in supplementary binding 35
2.20 Compensatory binding for a mismatched seed 36
2.21 Secondary structure of an mRNA 36
2.22 Binding abnormalities 37
2.23 Comparison of perfect and imperfect seed binding 38
2.24 Wobble pairing within the seed 38
2.25 Multiple aligned species conservation tracks 39
2.26 Conservation levels for different seed variations 40

2.27 Synergistic effect of CDS targets on 3′ UTR targets 41
2.28 Model fitting types 43
2.29 k-fold cross-validation 44
2.30 Visualisation of the DT algorithm 45
2.31 SVM kernel transformation to create linear separation 46
2.32 Densely connected DNN layers 47
2.33 Mathematical breakdown of an artificial neuron 48
2.34 Rule flowchart of miRanda 2.0 52

3.1 Rule-based prediction pipeline 58
3.2 Unfiltered expression fold change variance between samples in 01-liu-HeLa 60
3.3 FastQC read quality before and after trimming 61
3.4 Overview of kallisto pseudoalignment 62
3.5 Comparison of different TPM filter thresholds on miR-124-3p 67
3.6 Comparison of different TPM filter thresholds across aggregated transfections 69
3.7 Expression fold change variance between samples in 01-liu-HeLa 70
3.8 Overview of accessibility computation using a three-window approach . 72
3.9 Fold prediction using RNAcofold with a 6mer constraint 73
3.10 Comparison of supplementary definitions across aggregated transfections 75
3.11 Comparison of aggregate seed type efficacy 77
3.12 Comparison of aggregate MTS efficacy 78
3.13 Comparison of MTS efficacy on four individual transfections 79
3.14 Comparison of aggregate binding stability efficacy 80
3.15 MFE distribution by seed type 81
3.16 Comparison of aggregate target site accessibility efficacy 82
3.17 Comparison of target site accessibility efficacy on four individual transfections 83
3.18 Comparison of aggregate supplementary binding efficacy 84
3.19 Comparison of aggregate supplementary binding efficacy on 7mers and 8mers 85
3.20 Comparison of supplementary binding efficacy on miR-9-3p 86
3.21 Comparison of aggregate seed conservation efficacy 87
3.22 Comparison of aggregate supplementary conservation efficacy 88
3.23 Comparison of aggregate rule-based prediction thresholds 90

3.24 Rule set 1 expressed as a decision tree 91
3.25 Rule set 2 expressed as a decision tree 92
3.26 Rule set 3 expressed as a decision tree 93

4.1 Visualisation of weights used in weighted AU content 98
4.2 Correlation between SHAPE-seq reactivity in HeLa and other cell lines 101
4.3 Correlation between non-zero SHAPE-seq reactivity in HeLa and other cell lines 102
4.4 Comparison of flanking AU content efficacy 103
4.5 Comparison of weighted and unweighted flanking AU content 104
4.6 Comparison of dynamic and fixed seed AU content 105
4.7 Comparison of aggregate RNAplfold efficacy 106
4.8 Comparison of aggregate RNAplfold supplementary efficacy 107
4.9 Comparison of aggregate SHAPE-seq efficacy from HeLa 108
4.10 Comparison of aggregate SHAPE-seq supplementary efficacy from HeLa 109
4.11 Comparison of aggregate SHAPE-seq efficacy from five cell lines 110
4.12 Comparison of aggregate SHAPE-seq supplementary efficacy from five cell lines 111

5.1 Setup module activity diagram 115
5.2 Feature extraction module activity diagram 116
5.3 Machine learning module activity diagram 117
5.4 Bioinformatics pipeline 118
5.5 Cached output directory structure 119
5.6 Histogram of 3′ UTR lengths 121
5.7 Filtered expression fold change variance between all datasets 124
5.8 Grouped vs random data splits 126
5.9 Training set feature correlation heatmap 131
5.10 Training set feature correlation heatmap after reducing collinearity . . . 134
5.11 Training set missing data visualisations 140
5.12 Multiple imputation steps in MICE 141
5.13 Comparison of original and MICE-imputed phylo100 5′ conservation scores 142
5.14 Comparison of original and MICE-imputed SHAPE-seq reactivity values 142
5.15 Comparison of aggregate SHAPE-seq efficacy before and after imputation 143
5.16 Feature scale prior to transformation 144

5.17 Feature scale after transformation 144
5.18 Feature scale after normalisation 146
5.19 Distribution of training set log_2 fold change 147
5.20 Training set class balance at different log_2 fold change thresholds 148
5.21 ROC chart 150
5.22 Layout of a confusion matrix 151
5.23 Hyperparameter tuning results for DT 152
5.24 Hyperparameter tuning results for RF 153
5.25 Hyperparameter tuning results for SVM 154
5.26 Hyperparameter tuning results for DNN 155
5.27 Tuned DNN layers 155
5.28 CD diagram for AUC and F_1 rankings on the test set 156
5.29 ROC chart collage for the test set 157
5.30 Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT 159
5.31 Heatmap of miRsight predictions against TargetScan, MirTarget and DIANA-microT 160
5.32 Feature coefficients in the trained RF 162
5.33 Feature coefficients in the trained SVM 162

6.1 miRsight activity diagram 169
6.2 Unit testing for miRsight 174
6.3 miRsight website search bar 175
6.4 miRsight website results table 176
6.5 miRsight website results filtering 176
6.6 miRsight website printing filtered results 177
6.7 miRsight website downloading filtered results 177
6.8 miRsight website scaled to tablet dimensions 178
6.9 miRsight website scaled to mobile dimensions 178

B.1 Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-125a-5p 220
B.2 Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-642a-5p 221

B.3 Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-6133 222
B.4 Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-214-3p 223
B.5 Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-137-3p 224
B.6 Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-181a-5p 225
B.7 Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-30a-3p 226
B.8 Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-30a-5p 227
B.9 Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-27b-3p 228
B.10 Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-130a-3p 229
B.11 Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-130b-3p 230
B.12 Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-204-5p 231

Abbreviations

A	adenine
AGO	argonaute
ANN	artificial neural network
API	application programming interface
AUC	area under the curve
bagging	bootstrap aggregating
bp	base pair
C	cytosine
CART	classification and regression trees
CD	critical difference
CDS	coding sequence
CLI	command-line interface
CNN	convolutional neural network
CPSF	cleavage and polyadenylation specificity factor
CRE	cis-regulatory element
CSS	Cascading Style Sheets
DNA	deoxyribonucleic acid
DNN	deep neural network
DT	decision tree
FN	false negative
FP	false positive
FPR	false positive rate
G	guanine
GTF	gene transfer format
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
MANE	Matched Annotation from the NCBI and EMBL-EBI

MFE	minimum free energy
MICE	multivariate imputation by chained equations
mids	multiply imputed dataset
mipo	multiple imputed pooled outcomes
mira	multiply imputed repeated analysis
miRNA	microRNA
ML	machine learning
MRE	miRNA-recognition element
mRNA	messenger RNA
MTS	multiple seed target sites
MW	Mann-Whitney test
N	negative
NCBI	National Center for Biotechnology Information
NGS	next-generation sequencing
nt	nucleotide
P	positive
PAS	polyadenylation signal
poly(A) tail	polyadenine tail
pre-mRNA	precursor mRNA
pri-miRNA	primary miRNA
RF	random forest
RFE	recursive feature elimination
RNA	ribonucleic acid
RNA-seq	RNA sequencing
RNAP	RNA polymerase
ROC	receiver operating characteristic
RPM	reads per million
SBS	sequencing by synthesis
SHAPE-seq	Selective 2′-hydroxyl acylation analyzed by primer extension sequencing
sncRNA	small non-coding RNA
SPA	single-page application
SVM	support vector machine
SVR	support vector regressor
T	thymine

TF transcription factors

TLS Transport Layer Security

TN true negative

TP true positive

TPM transcripts per million

TPR true positive rate

U uracil

UTR untranslated region

Chapter 1

Introduction

The expression of genes is a key mechanism in driving cell development, requiring precise regulation in order to coordinate the effective functioning of an organism's biological processes. Gene expression is largely centred around the metabolism of messenger RNA (mRNA), a molecule responsible for transferring genetic information for use in protein synthesis. Regulating the levels of gene expression is a complex and layered process; in addition to the transcription factors (TFs) that control the rate that genes are transcribed into mRNA, post-transcriptional suppressive mechanisms, such as microRNA (miRNA) binding, ensure its continuous regulation (Bartel, 2004).

mRNA suppression by miRNA in plants and animals occurs following complementary base pairing between the two molecules. In animals, the miRNA:mRNA interaction is complicated as a result of limited sequence complementarity (Zhang et al., 2007). Furthermore, there are beyond two thousand unique miRNAs in *Homo sapiens* (Alles et al., 2019), and mRNAs not limited to interactions with any single miRNA (Guo et al., 2014b). As a result, the efficacy and specificity of miRNA targeting is dependent on numerous interconnected contextual factors, the appreciation of which allows predictive models to determine candidate targets.

The application for miRNA:mRNA target prediction is multifaceted due to the macro importance of gene regulation in molecular biology and biomedical research, particularly in gene studies and biomarker discovery. miRNAs are embedded in a complex regulatory network that mediates the host response to pathogens (Drury et al., 2017). Dysfunctional miRNA gene regulation is also implicated at numerous stages in

the development of various diseases (Corbett, 2018), such as cancer, where aberrant miRNA abundance is known to disturb the expression of tumour-suppressive target genes (Ali Syeda et al., 2020). Consequently, controlled miRNA delivery has emerged as a potential therapeutic strategy by modulating specific immune responses (Lee et al., 2019), including cellular response to treatments such as radiotherapy (Podralska et al., 2020). Synthetic miRNAs have also shown promise in targeting virus structural proteins and enhancing immune responses when encoded into vaccine formulations (Leon-Icaza et al., 2019). To facilitate further research in these areas, computational tools capable of accurately predicting miRNA interaction are important in disentangling the complexity of miRNA-mediated regulation.

This thesis aims to build upon established research for miRNA:mRNA prediction through the use of data mining and machine learning (ML) techniques on a sophisticated feature set drawn from a wide array of published datasets. This goal is recognised by the following chapters:

- **Chapter 2** provides an examination of the broader biological systems relating to gene regulation and reviews the target recognition features from established research that will inform the basis of this study.

- **Chapter 3** observes the impact of core target recognition features from popular prediction tools and applies them in a basic rule-based model.

- **Chapter 4** investigates several alternative methods for measuring target site accessibility, a staple feature in many target prediction algorithms.

- **Chapter 5** describes the development of an ML model for target prediction, integrating discoveries from previous chapters as features.

- **Chapter 6** discusses the conversion and finalisation of tooling from previous chapters into the miRsight command line tool, in addition to the development of a web application platform to host its predicted targets.

- **Chapter 7** concludes this thesis with a summary and discussion of key findings and potential further research.

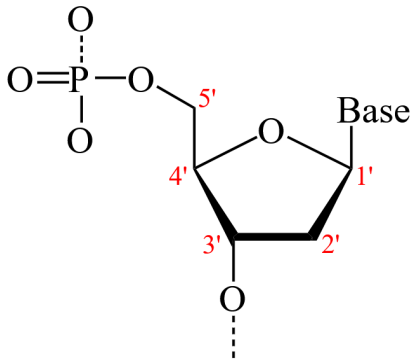
Chapter 2

Background

2.1 Nucleic Acids

Within the cells of an organism, copies of genetic information are stored inside nuclei in the form of deoxyribonucleic acid (DNA). Comprised of 23 pairs of chromosomes in *Homo sapiens*, each containing thousands of genes, the genome encodes quaternary sequences of adenine (A), cytosine (C), guanine (G) and thymine (T) nucleobase compounds.

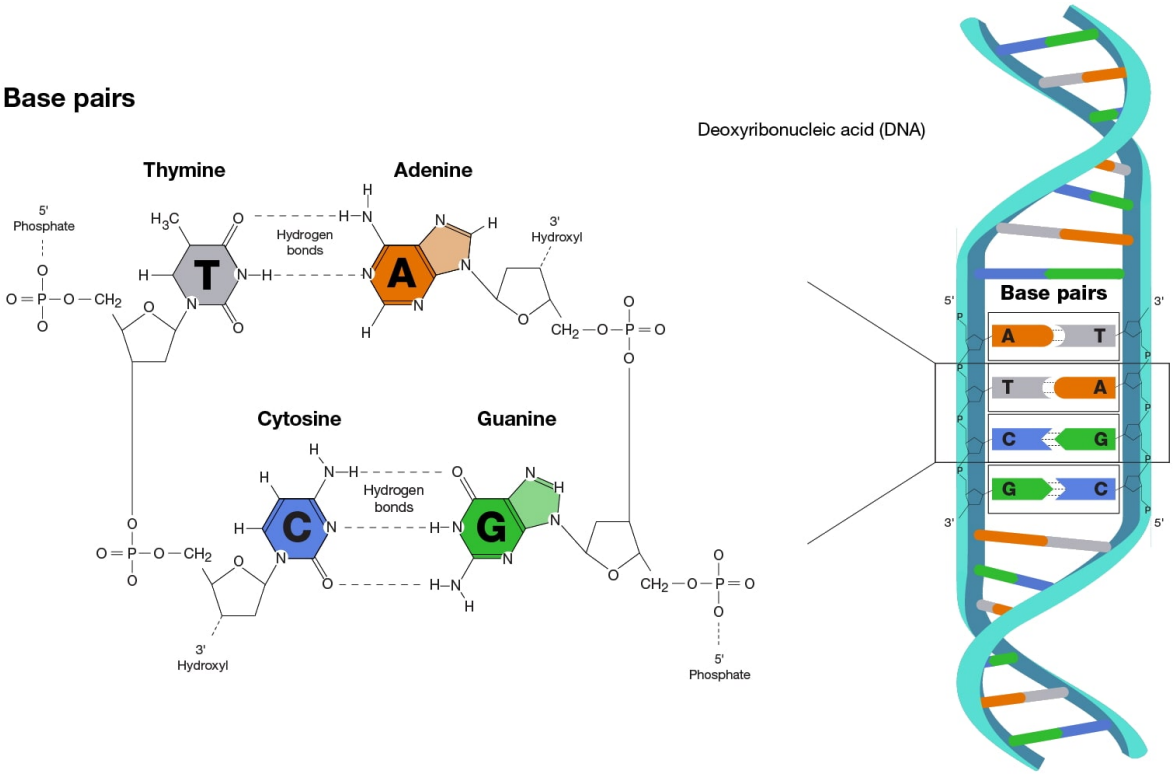
Nucleobases combine with deoxyribose (sugar) and a phosphate group to form nucleotides (nt), which in turn constitute polynucleotide chains by way of covalent bonding of alternating sugar and phosphate groups (Watson and Crick, 1953b). Polynucleotide chains form a single strand of DNA from $5' \rightarrow 3'$, in reference to the directionality of the 5th and 3rd carbon atoms present in the furanose (sugar ring) between chained molecules.



Source: Neurotiker, Public Domain

Figure 2.1: Carbon atoms in a furanose molecule. Chaining occurs at the $5'$ and $3'$ carbon atoms, where a chained molecule would be considered upstream and downstream respectively.

DNA is formed of two sugar-phosphate backbones bound together through canonical Watson-Crick base pairing, in which hydrogen bonding occurs between complementary A:T and C:G bases (Figure 2.2). The resulting double-stranded binding forms a double helix structure (Watson and Crick, 1953a).



Source: Darryl Leja, Public Domain

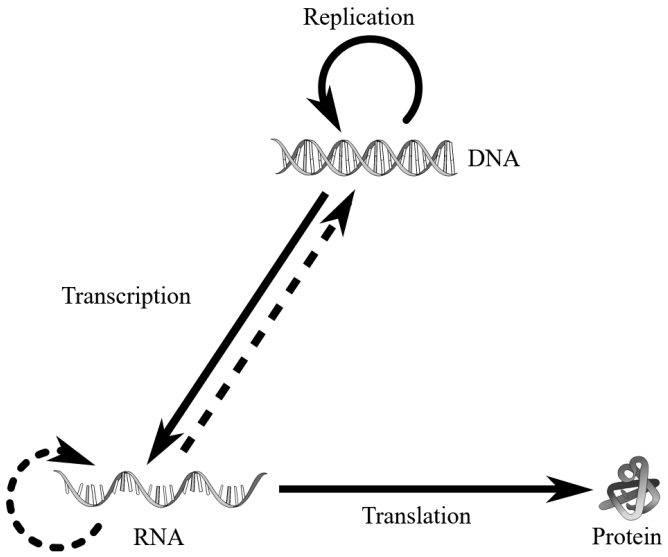
Figure 2.2: DNA canonical base pairing. Pairing occurs between A:T and C:G on opposing strands, themselves formed of chains between sugar and phosphate.

A derivative of DNA is ribonucleic acid (RNA), produced during transcription as a functional intermediary between DNA and protein. Like DNA, RNA encodes genetic information using a combination of four nucleobases, though uracil (U) is used in place of T. The chemical structure of RNA differs from DNA as it is single-stranded, significantly shorter and comprised of ribose as opposed to deoxyribose.

2.2 Genetic Regulatory Network

Cell function is varied despite the dissemination of identical DNA throughout an organism. This specialisation is driven by differences in which genes of the DNA are selected in a process known as gene expression. Key to this process is transcription, where genetic information is copied from DNA into a new RNA molecule, and translation, which synthesises proteins from the RNA template (Schwanhäusser et al., 2011).

The overarching system is outlined in the central dogma of molecular biology, which defines rules regarding the flow of genetic information between these three states (Figure 2.3). It highlights that information flows in a one-way direction from DNA → RNA → protein, although atypical flows can occur in abnormal scenarios (Crick, 1970). The process of gene expression can be quantified using a combination of experimental and computational methods (Section 2.3).



Source: Barillot et al. (2012)

Figure 2.3: Central dogma of molecular biology. The solid lines define the flow of genetic information between states, ruling that it may not flow back from protein to nucleic acid. The dotted lines indicate abnormal scenarios typically associated with viruses.

Proteins are molecules formed of one or more amino acid residues, linked together linearly in peptide or polypeptide chains (Ramachandran and Sasisekharan, 1968). They are responsible for a range of vital processes within an organism and may be further classified by their specialisation, for example chemical signalling (hormones), immunity (antibodies) and catalysis (enzymes). RNA was initially characterised by the role of mRNA in protein synthesis. However, the discovery of regulatory small non-coding RNAs (sncRNAs), such as miRNA, has shown that its role is substantially more diverse (Morris and Mattick, 2014).

2.2.1 Gene Expression

During transcription, different portions of genetic code are copied from DNA into RNA. Depending on whether the transcribed RNA is coding (synthesises protein) or non-coding (regulatory), the post-transcriptional pathway may include translation, in which the RNA molecule conveys genetic information to the ribosome in order to

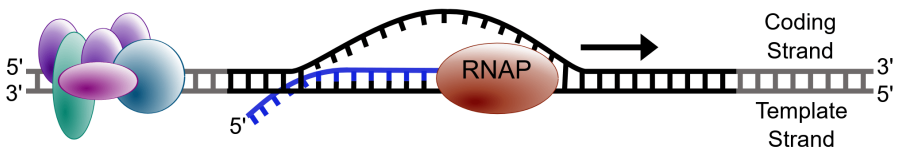
produce proteins (Yusupova et al., 2001).

2.2.1.1 mRNA Transcription

mRNAs are large RNA molecules defined by their role in conveying genetic information for a range of cellular processes. mRNA is first introduced in the form of precursor mRNA (pre-mRNA) by RNA polymerase (RNAP) II enzymes, recruited to the DNA at core promoter regions (DeHaseth et al., 1998). The likelihood of RNAP II binding is influenced by cis-regulatory elements (CREs), non-coding DNA sequences in the DNA that manage the interaction through the recruitment of TFs (Ong and Corces, 2011). TFs play an important role in regulating gene expression and are broadly categorised as activators and repressors; those that increase transcriptional activity and those that suppress it.

Enhancers are positive-influencing CREs which bind with activators to deliver accessory factors to the promoter (Calo and Wysocka, 2013). Antithetical to enhancers, silencers suppress transcription through binding with repressor proteins to physically block elongation and prevent splice recognition (Ogbourne and Antalis, 1998). Most CREs must be local to the promoter to function effectively; however, enhancers act relatively independent to sequence distance (Pennacchio et al., 2013). Instead, loaded TFs act to reduce an enhancer's relative physical proximity to promoters by taking advantage of the coiling of DNA in the chromatin (Kolovos et al., 2012).

If successful, RNAP II binds to the promoter and creates a transcription bubble by detaching the two DNA strands (Figure 2.4). RNAP II then moves upstream while adding complementary bases to the template strand in a process known as elongation (Pal et al., 2005). When the 5′ end of the pre-mRNA emerges from the RNAP II, it is capped with an altered G by a capping enzyme (Hirose and Manley, 2000). Elongation continues until RNAP II has transcribed the remainder of the transcript, including the polyadenylation signal (PAS) (Rodríguez-Molina et al., 2023).



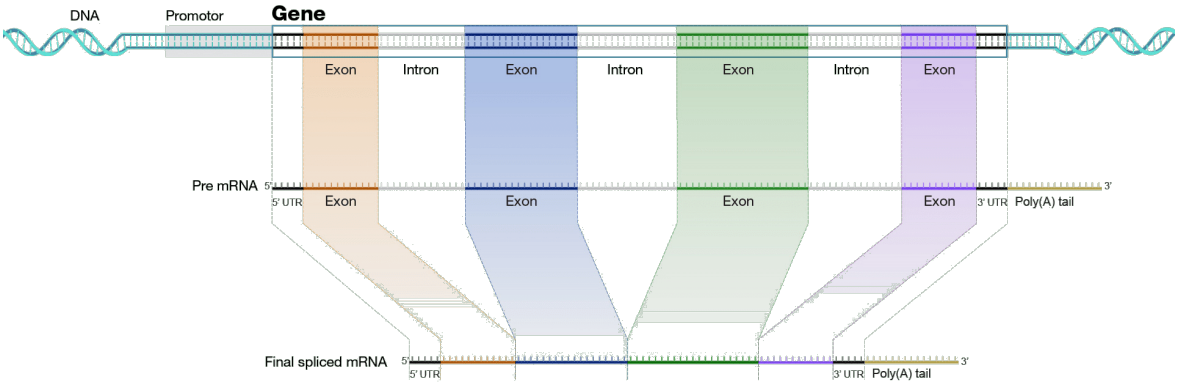
Source: Proanonicholas, CC BY-SA 4.0

Figure 2.4: Creation of pre-mRNA by RNAP II. RNAP II creates pre-mRNA within the transcription bubble by adding complementary bases to the template strand. A build-up of TFs can be seen toward the 3′ end of the template strand.

2.2.1.2 mRNA Processing

Protein-coding mRNAs undergo several stages of post-transcriptional modification in order to transform them into their mature state for translation. Transcription termination in mammals is coordinated by a multi-protein complex of at least sixteen polypeptides, notably encompassing the cleavage and polyadenylation specificity factor (CPSF) (Schönemann et al., 2014). The binding of the CPSF to the PAS region catalyses the cleavage and synthesis of a polyadenine tail (poly(A) tail) at the 3′ end of the pre-mRNA in protein-coding mRNA. The 5′ cap and poly(A) tail terminal modifications protect and stabilise the newly synthesised RNA and have a broad influence on gene expression, with the latter also playing a role in translation (Gao et al., 2000).

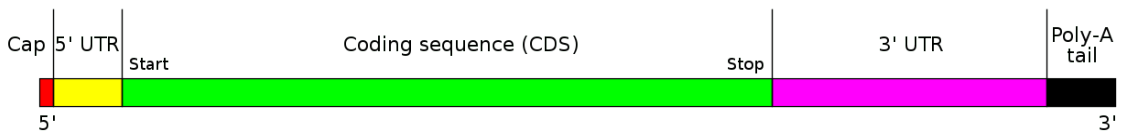
A two-stage splicing process then strips the pre-mRNA of regions that do not code for proteins (introns) (Figure 2.5). The removal of introns allows the coding regions (exons) to be spliced to form the mature mRNA's coding sequence (CDS), preparing it for ribosomal interaction and facilitating translation (Zeitlin et al., 1987).



Source: Eric Green, Public Domain

Figure 2.5: RNA splicing Pre-mRNA is formed from DNA in transcription. After exons are spliced following the removal of introns, the CDS is formed and the resultant single-stranded molecule is a mature mRNA.

Mature mRNA is composed of a subset of the original segments produced from transcription (Figure 2.6). In the work presented in this thesis, the 3′ untranslated region (UTR) and CDS are of particular importance.



Source: Daylite, Public Domain

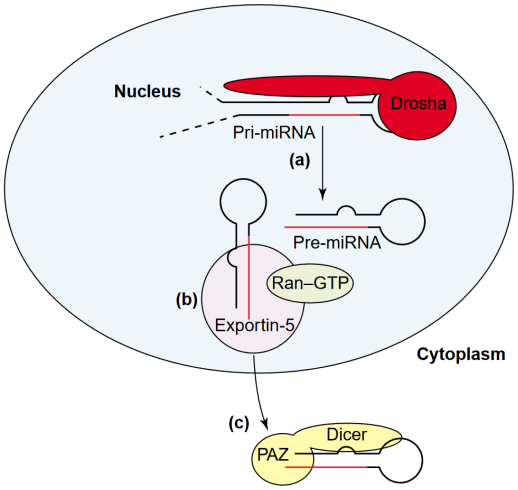
Figure 2.6: Structure of mature mRNA. Mature mRNA is composed of five distinct parts, formed through several layers of processing.

2.2.2 Post-transcriptional Regulation

Gene expression is managed by a variety of regulatory factors, most predominantly TFs and miRNAs (Hobert, 2008). Unlike TFs, which regulate during transcription, miRNAs form a post-transcriptional regulatory layer in both plants and animals (Krol et al., 2010). miRNAs are a type of sncRNA, meaning that they are comparatively small (18-25 nt) and do not code for proteins, instead binding with mRNA to disrupt the expression of target genes.

2.2.2.1 miRNA Biogenesis

miRNAs are first transcribed from DNA by RNAP II in the form of primary miRNA (pri-miRNA). In addition to a 5′ cap and poly(A) tail (Lee et al., 2004), pri-miRNA is notably structured with a hairpin loop (Ha and Kim, 2014). Drosha, a type of RNA III enzyme, cleaves the pri-miRNA into a smaller stem-loop of around 70nt, resulting in precursor-miRNA (pre-miRNA) (Lee et al., 2003). Pre-miRNA is then exported from the nucleus to the cytoplasm for further processing by the Dicer RNAP III enzyme (Murchison and Hannon, 2004), which binds to cleave the loop and ultimately produce an 18-25nt long duplex with a 3′ overhang (Lund and Dahlberg, 2006).

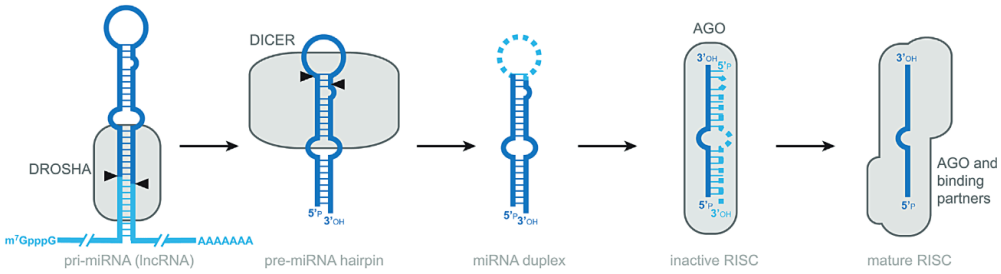


Source: Murchison and Hannon (2004)

Figure 2.7: Trafficking miRNA from Drosha to Dicer. After cleavage by Drosha (a), the pre-miRNA is transported out of the nucleus by an Exportin-5 protein, itself dependent on a Ras protein export factor (b). In the cytoplasm, miRNA is prepared for interaction with Dicer (c).

Argonaute (AGO) is a central protein component in RNA-induced silencing complexes (RISC), a type of heterogeneous molecular complex that targets genes for silencing (Pratt and MacRae, 2009). AGO is responsible for recruiting sncRNAs to the RISC to function as binding site hosts for potential targets (Cloonan, 2015) (Figure 2.8). A

strand of the miRNA duplex is selected by the AGO to function as the 'guide strand' in accordance with its 5′ stability characteristics (O'Brien et al., 2018), favouring low thermodynamic stability and the presence of a 5′ terminal U (Khvorova et al., 2003). Once loaded into AGO, a typical mature miRNA molecule consists of 22 nt, of which the first eight bases constitute the seed region. Two miRNAs are said to belong to the same family if they have identical seed sequences (Brancati and Großhans, 2018).

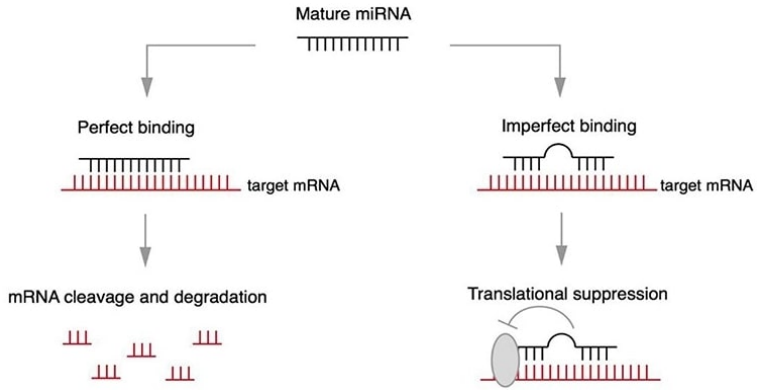


Source: Cloonan (2015)

Figure 2.8: Canonical pathway of miRNA biogenesis. Pri-miRNA is cleaved by Drosha and then Dicer to produce short duplexes. After loading into AGO, a strand is selected and prepared for binding with miRNA, forming a RISC.

2.2.2.2 miRNA-mediated Regulation

The miRNA-loaded RISC degrades or suppresses the transcription of mRNA molecules through the repression (reduction of translation rate), deadenlyation (removal of adenosine from the poly(A) tail) and decay of the mRNA, following a successful binding between the RNAs (Bazzini et al., 2012). Bindings may also have imperfect complementarity, leading to a different suppressive mechanism whereby the mRNA bases are blocked and rendered inaccessible to other molecules (Bartel, 2009). The mechanics of miRNA:mRNA binding is further discussed in Section 2.4.



Source: Teixeira et al. (2014)

Figure 2.9: Binding methods of miRNA to degrade or suppress mRNA. miRNA:mRNA binding may occur through two main scenarios. (Left) miRNA binds perfectly with mRNA, leading to thorough cleavage and degradation. (Right) miRNA binds imperfectly with mRNA, suppressing the mRNA target by preventing base access to other molecules.

2.3 Genetic Sequencing

Genetic sequencing refers to the process of determining the sequence order of bases within nucleic acids. A typical sequencing method applies biochemical techniques to obtain a number of shorter sequence 'reads' (Figure 2.10). These reads are then re-assembled to reconstruct the original sequence (Section 2.3.1).

Figure 2.10: Nucleobases in a sequenced read. A read encodes nucleobases using their representative letters in a continuous sequence, each accompanied by a per-base sequencing quality score.

Sequencing methods are divided into technological generations in accordance with their notable traits and underpinning techniques (Pettersson et al., 2009). Early endeavours suffered a number of drawbacks; they were slow, manual and often involved the use of radiation (Heather and Chain, 2016). The first generation Sanger technique (Sanger et al., 1977) evolved to remedy these issues, becoming an industry standard at the end of the twentieth century (Gharizadeh et al., 2006). Late first generation approaches are instead recognised by their relatively long read length, limited throughput and low rate of error (Hebert et al., 2018).

The next-generation sequencing (NGS) era, which comprises the second and third generations, began in the early 2000s following concurrent advancements in computing and sequencing chemistry (Slatko et al., 2018). The invention of sequencing by synthesis (SBS) techniques was key in facilitating NGS, originally finding application in the 2005 Solexa (Illumina) and 454 (Roche) platforms as reverse terminator sequencing (Bentley et al., 2008) and pyrosequencing (Margulies et al., 2005) respectively.

The SBS techniques of the second generation forsake read length in favour of higher throughput by short-sequencing reads in parallel (Reis-Filho, 2009), leading to a higher rate of error due to a deterioration in read quality (Patel and Jain, 2012). Sequence reassembly of these shorter reads is offset to advanced computational algorithms able to benefit from periodic technological developments, such as cloud computing, enabling a faster process that is more scalable to larger genomes compared to Sanger sequencing (Muir et al., 2016). In contrast, third generation sequencing utilises various techniques that can produce longer reads (Schadt et al., 2010) and includes platforms that prioritise portability and real-time sequencing (Jain et al., 2016). As a result, NGS

is broadly recognised for its reduction in time and sequencing costs, yet substantially increased output compared to the first generation (Kumar et al., 2019).

2.3.1 Sequence Reconstruction

In NGS, millions of reads are produced in parallel, requiring reassembly using computational tools and algorithms. When there is no reference genome available, the *de novo* assembly process uses linked data structures to infer the likely positions of reads from overlapping portions of code. Figure 2.11 demonstrates how reads with overlapping code 'contigs' are used to create 'scaffolds', bridges between contigs with known gaps (Miller et al., 2010). *De novo* assembly is further complicated by short NGS reads, as the overlap between sequences is not always sufficient (Li et al., 2010).

Source: Aaron Mayo, CC BY-SA 4.0

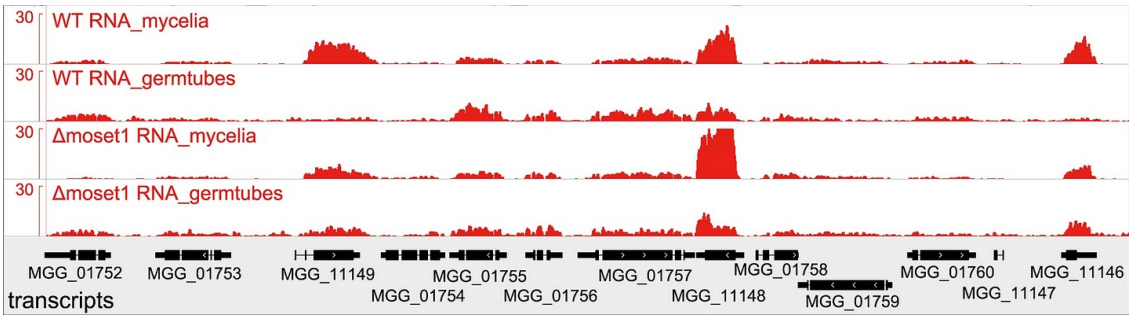
Figure 2.11: Sequence assembly process. In *de novo* assembly, overlaps between reads are grouped into contigs, which are then assembled into scaffolds to represent the overall sequence.

When a species' genome is published, such as with *Homo sapiens*, the reconstruction process is simplified. In contrast to *de novo* assembly, sequence alignment reconstructs the original sequence by mapping them against a reference genome. Sequence alignment tools explore alignment possibilities by attempting to maximise a scoring function while considering overlapping regions (Chowdhury and Garai, 2017). Per-base scoring is also applied to mitigate read sequencing errors that would otherwise complicate the process with erroneous bases (Engström et al., 2013).

2.3.2 RNA-seq

RNA sequencing (RNA-seq) is an NGS sequence alignment technique that quantifies gene expression by mapping RNA reads against a reference genome (Finotello and Di Camillo, 2015). The core principle of RNA-seq holds that the presence of RNA is indicative of an expressed gene; therefore, the greater the number of aligned reads, the higher the level of gene expression (Figure 2.12). RNA-seq is the de facto standard for estimating gene expression due to its superior precision compared to alternative techniques (Wang et al., 2009).

In a differential expression analysis, the levels of expression in a sample are compared against a control group. When the RNA level near a gene differs, the sample's tested condition can be said to affect the gene.



Source: Pham et al. (2015)

Figure 2.12: Differential expression analysis. Expression peaks (red) become more abundant around gene sequences (black), where a higher peak implies greater expression of the gene. The tracks show expression levels in two wild type samples (top) and corresponding mutant samples (bottom). 'MGG_11149' can be seen downregulated in the mutant mycelia sample compared to the wild type.

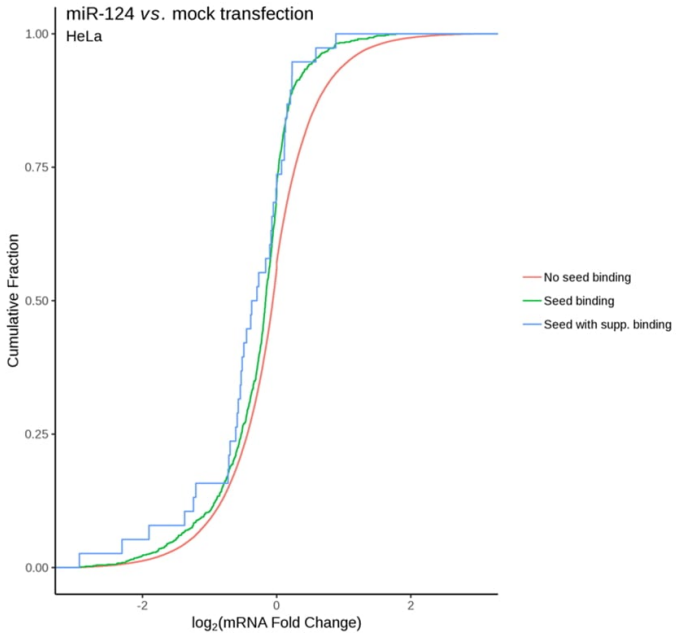
The difference in RNA abundance in the treated sample compared to the control sample is often represented as expression fold change, the ratio difference between two expression values. If the fold change of a gene is zero, it is unaffected by the treatment (Love et al., 2014). Expression fold change is typically transformed to a logarithmic scale, for example, log_2 fold change.

2.3.2.1 miRNA Transfection

Transfection is a lab procedure for delivering foreign nucleic acids into a sample to observe its impact on protein expression against a control (non-transfected) sample. miRNA transfection refers specifically to the introduction of miRNAs to examine the impact of their interaction with endogenous mRNAs. A differential expression analysis utilising miRNA-transfected RNA-seq data is therefore capable of quantifying

the suppressive impact of transfected miRNAs on gene expression profiles. Where a downregulatory effect is observed, an mRNA can be considered a 'true target' of the miRNA. An alternative to miRNA transfection is knockout, in which an miRNA is instead removed from a sample to observe an opposite effect.

The log_2 fold change of each gene can be represented using a cumulative plot to visualise expression differentials following an miRNA transfection. In Figure 2.13, the lines show the log_2 fold change of genes with and without seed target sites. Where a leftward shift can be observed compared to the 'No seed binding' control line, a downregulation in expression of the genes in the associated category has occurred. This line separation can be quantified using a one-sided two-sample Mann-Whitney (MW) test, as the resulting p-value offers a statistical assessment of whether one group's distribution deviates significantly in a smaller (leftward) direction.



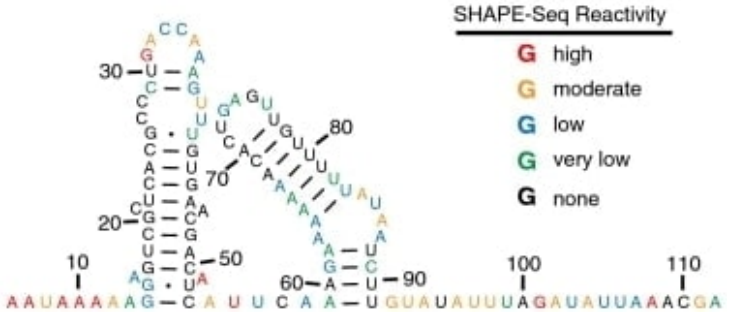
Source: prior work from the lab

Figure 2.13: Cumulative expression fold change plot. The fold change of miR-124 transfected HeLa cells is compared against a control sample. The orange line plots genes which do not contain a seed target site for miR-124. As genes without a seed target should not become downregulated as a result of the introduction of miR-124, this functions as a control line. A leftward shift can be observed in the green line compared to the orange, indicating the expression of genes containing a seed target for miR-124 are downregulated. The blue line shifts further still, meaning supplementary binding conferred a stronger downregulatory impact than seed binding alone. Meanwhile, the jagged shape of this line highlights that there are fewer data points in this category compared to the others.

2.3.3 SHAPE-seq

Selective 2′-hydroxyl acylation analysed by primer extension sequencing (SHAPE-seq) (Lucks et al., 2011) is an NGS technique for measuring RNA structure. Folded

RNA is treated with a reagent to block reactions from reverse transcriptase, leading to a series of truncated products that allow the original structure to be reconstructed (Fang and Fullwood, 2016). When sequenced, a SHAPE reactivity value is produced at each nt, where high values are generally indicative of weaker structure (Lucks et al., 2011).

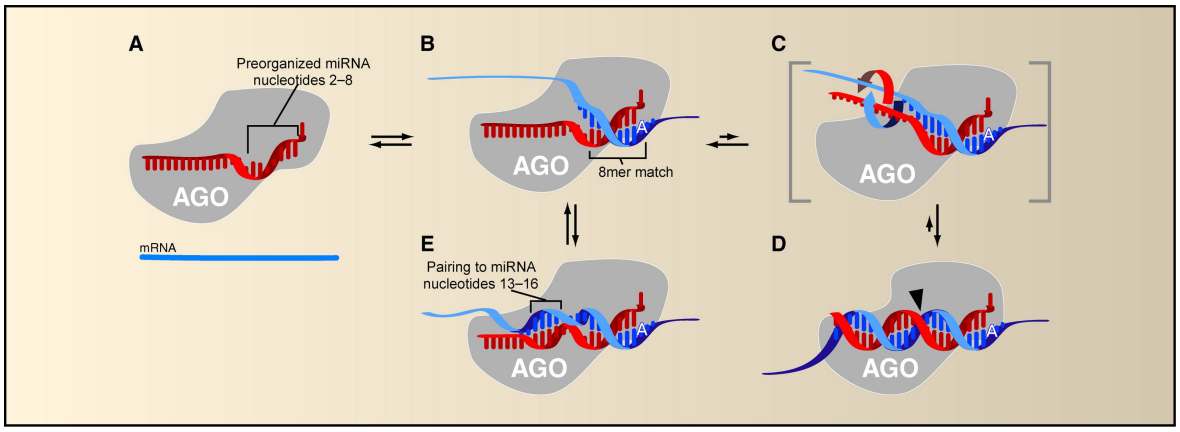


Source: Lucks et al. (2011)

Figure 2.14: SHAPE-seq reactivity values. At each base of an RNA molecule (numbered in black), SHAPE-seq outputs a reactivity value (colour coded). A high reactivity values indicates low levels of internal structure and therefore an openness to external binding (Section 2.4.3).

2.4 Mechanics of miRNA:mRNA Binding

Binding between miRNA and mRNA in animals typically occurs along the 3′ UTR of the mRNA at accessible seed target sites (Lewis et al., 2003). Figure 2.15 shows how the miRNA is bound by AGO to ensure effective binding at bases 2-8 relative to the 5′ end, with potential supplementary pairing at positions 13-16. It also demonstrates how certain bases, notably 1 and 9-12, are twisted from incoming mRNA to prevent their binding in most cases (Bartel, 2009).



Source: Bartel (2009)

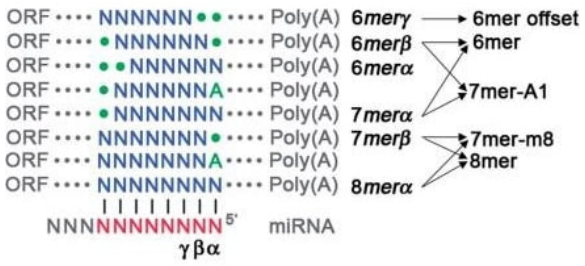
Figure 2.15: Model of miRNA:mRNA interaction. (A) miRNA (red) is loaded by AGO (grey) to prepare for binding with mRNA (blue). Base 1 is twisted away, while 2-8 are positioned to be accessible. (B) A match of bases 1-8. The mRNA's A is recognised by an external factor, such as the AGO. (C) Seed pairing causes AGO to loosen around the 3′ region of the miRNA to support further binding. (D) AGO locks the duplex and positions the active site (black arrow) to cleave the mRNA. (E) Supplementary binding of bases 13-16 (Section 2.4.2). The structure of bases 1-9 is unaffected.

Thermodynamic stability is a major determinant of the efficacy (down-regulatory impact of a binding on mRNA expression) of a binding. Binding stability is affected by many factors, including the number of paired bases (Section 2.4.1), presence of gaps (Section 2.4.4) and wobbles (Section 2.4.5). Minimum free energy (MFE) is a common measure of this stability (Ui-Tei et al., 2008).

2.4.1 Seed Binding

Seed sites are short and conserved sequences which function as fixed anchor points on the miRNA for binding (Friedman et al., 2009). The seed region is defined as positions 1-8 at the 5′ end of the miRNA (Sethupathy et al., 2006). To differentiate which specific combination of bases are utilised, a unique '$k$mer' identifier is designated, where k refers to the number of bases involved in binding.

Although precise $k$mer identifiers can differ by publication, the convention follows that seed sites are built relative to a '6mer' (six nt) match. There are theoretically three possible locations for a 6mer within the seed region (Figure 2.16): positions 1-6 (6merα), 2-7 (6merβ) and 3-8 (6merγ). However, due to the relatively low efficacy of bindings built about 6merα and 6merγ, 6merβ is considered the primary 6mer site, with 6merγ often referred to as the '6mer offset'. This means that a 6mer is strictly defined as positions 2-7, despite these other possibilities (Ellwanger et al., 2011).

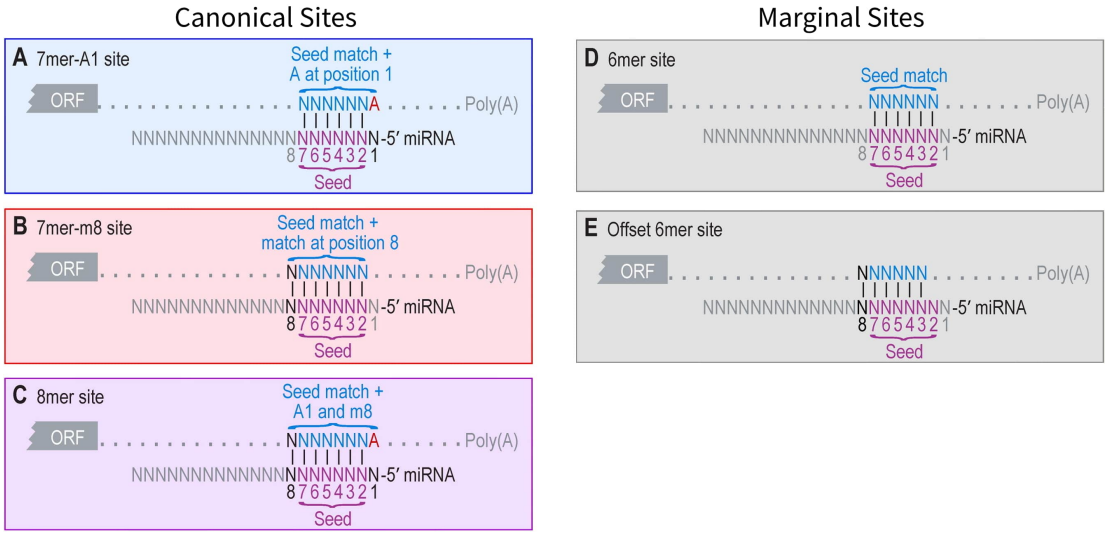


Source: Ellwanger et al. (2011)

Figure 2.16: Potential seed identifiers. Seed bindings are generally recognised relative to a six-base match. The 6mer, 7mer-a1, 7mer-m8 and 8mer definitions are the most common in literature.

Beyond the 6mer, it is possible for bases 1 and 8 to provide additional stability to a binding (Nielsen et al., 2007). These '7mers' are identical to 6mers, except an additional base is brought to relevance by one of two scenarios: a match at position 8 (7mer-m8) or an A present at position 1 (7mer-a1). It is important to note that for 7mer-a1, base 1 does not need to be matched, as the structure of miRNA:mRNA binding makes pairing here unlikely (Section 2.4). Finally, an '8mer' simply refers to a 6mer with a

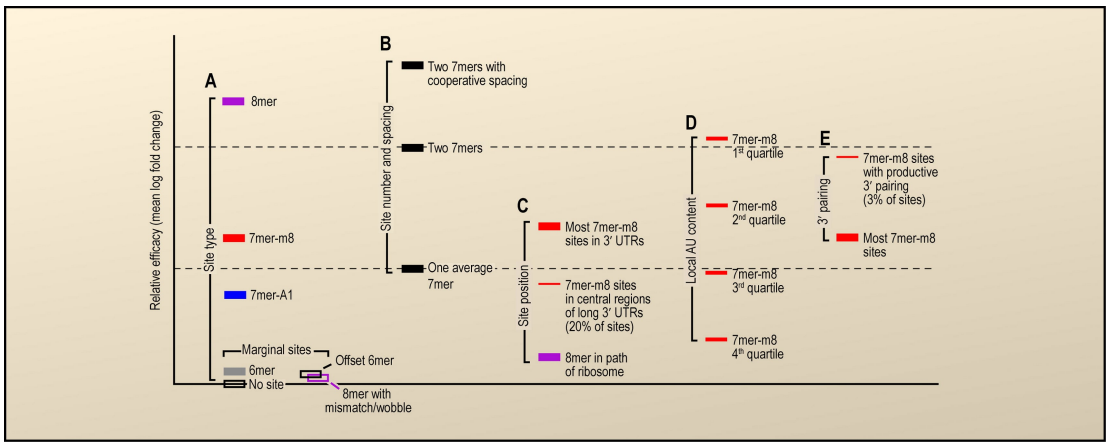
match at position 8 and an A present at position 1, or a combination of the 7mer rules. In cases where there is a match at position 1, these 6merα derived sites are sometimes distinctly referred to as '7mer-m1' and '8mer-m1'.



Source: Bartel (2009)

Figure 2.17: Core seed type summary. (A-C) The 7mer and 8mer sites are subsets of 6mer, specifically containing either an A at base 1 of the mRNA, a match at base 8, or both. (D-E) The two primary 6mer sites, in which six consecutive bases are paired beginning at either base 2 or 3.

Binding efficacy increases multiplicatively for each additional base pair (bp) at the seed site (Fang and Rajewsky, 2011). 7mer-m8 is also known to have a greater downregulatory effect than 7mer-a1, meaning the ranked efficacy of each seed site is: 8mer > 7mer-m8 > 7mer-a1 > 6mer (Baek et al., 2008). The performance of 6mers is superior, yet comparable, to instances of 6mer offset and no binding site (Figure 2.18). This ultimately leads to their exclusion from many target prediction algorithms (Section 2.5.3).



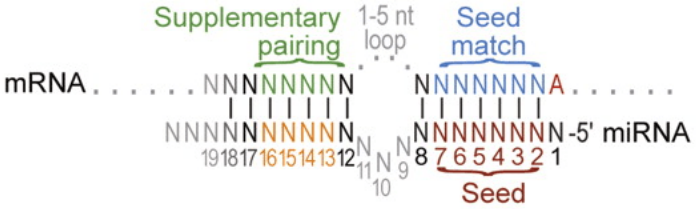
Source: Bartel (2009)

Figure 2.18: Comparison of relative seed site efficacy. Relative rankings from most (top) to least (bottom) effective: (A) Seed types. (B) 7mers based on the cumulative effect of multiple target sites (Section 2.4.7). (C) Seed positioning. (D) Local AU content (Section 2.4.3). (E) Additional supplementary pairing (Section 2.4.2).

In terms of frequency, a genome-wide mapping of RNA binding sites found that 6merα and 6mer offset occur most often, at approximately the same rate of 24% of all sites. This is followed by 6merβ, 7mer-m8, 7mer-a1 and finally 8mer, with 19%, 13%, 10% and 9% respectively (Ellwanger et al., 2011). Since alternate 6mer definitions are not always used, this would make the combined 7mer category the most populous.

2.4.2 Supplementary Binding

It is possible to have additional base pairing towards the 3′ end of the miRNA. These supplementary pairings offer increased efficacy to a binding by stabilising the regulatory interaction (Pasquinelli, 2012). In such instances, a seed binding is still required even with extensive 3′ pairing (Brennecke et al., 2005). The characteristics of supplementary binding are not as well understood as seed binding, though bases 12-17 are believed to be important, with 13-16 being of particular significance. In this region, contiguous pairings of 3-5 bases have a substantial impact on the binding (Grimson et al., 2007). After accounting for the lack of accessibility in bases 9-12, these bases are of closest proximity to the seed (Section 2.4).



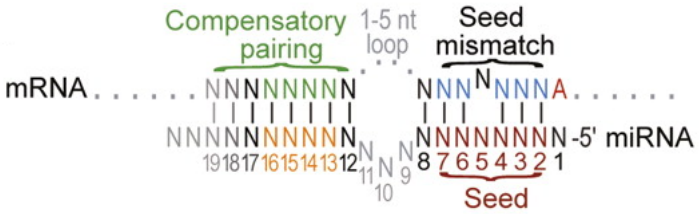
Source: Friedman et al. (2009)

Figure 2.19: Key bases in supplementary binding. Supplementary binding occurs toward the 3′ end of an miRNA. Bases 13-16 are thought to be the most important in supplementary binding, followed by 12 and 17.

Supplementary binding can also confer specificity (filters to the range of potential targets) to miRNAs of the same family (Broughton et al., 2016). This specificity is largely influenced by the degree and stability of supplementary binding, rather than the impact of specific base combinations (Brennecke et al., 2005). The relative frequency of supplementary binding is low across all types of seed, and significantly rarer than seed binding (Bartel, 2009).

2.4.2.1 Compensatory Binding

A subclass of supplementary binding is compensatory binding. In a perfect binding, there are no gaps (Section 2.4.4) or wobbles (Section 2.4.5), meaning the seed and its target are perfectly complementary. In cases of imperfect seed binding, it is possible for supplementary pairing to 'compensate' and improve binding stability (Bartel, 2009). Compensatory binding is rarer than supplementary binding, occurring at a rate of 1.5% of all bindings, compared to 5% for supplementary binding (McGeary et al., 2022).

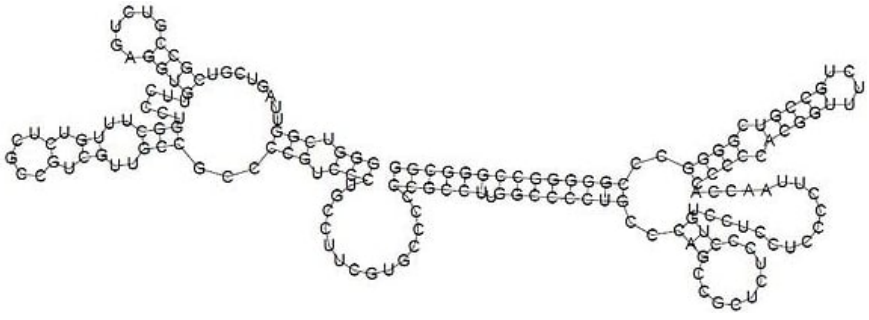


Source: Friedman et al. (2009)

Figure 2.20: Compensatory binding for a mismatched seed. Extensive binding in the 3′ region of the miRNA may partially compensate for seed instability.

2.4.3 Target Site Accessibility

The single strands of mRNA fold intramolecularly to form complex interconnected base pairing internally, known as secondary structure (Mahen et al., 2010). Secondary structure is a factor in mRNA stability, affecting translation and protein synthesis (Shabalina et al., 2006). Necessary to RNA function, unfavourable positioning can interfere with miRNA:mRNA interaction by blocking potential binding sites (Long et al., 2007).



Source: UCSC, Public Domain

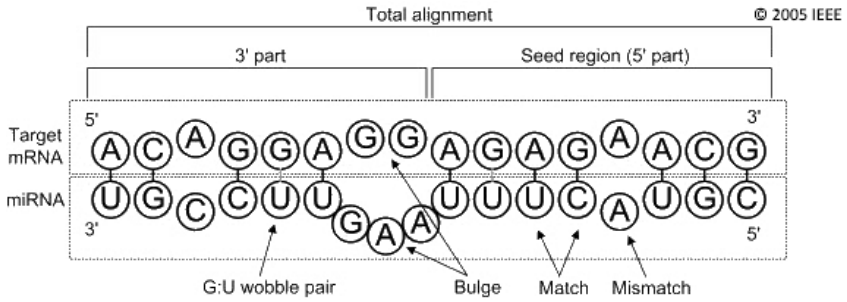
Figure 2.21: Secondary structure of an mRNA. An example of the predicted folding of an mRNA's 5′ UTR. Internal base pairing occurs frequently, forming motifs, such as loops.

Accessibility is defined as a measure of how open a base is to binding machinery as a result of secondary structure. Site inaccessibility leads to a reduction in binding efficacy similar to that of imperfect sequence complementarity in the seed (Kertesz et al.,

2007), making it important in understanding local site context. Local AU content has traditionally served as an indirect measure of accessibility (Grimson et al., 2007), as the presence of A and U flanking the seed are indicative of weaker secondary structure (Riffo-Campos et al., 2016). More recently, direct probing techniques, such as SHAPE-seq (Section 2.3.3), offer potentially more accurate accessibility measures.

2.4.4 Binding Gaps

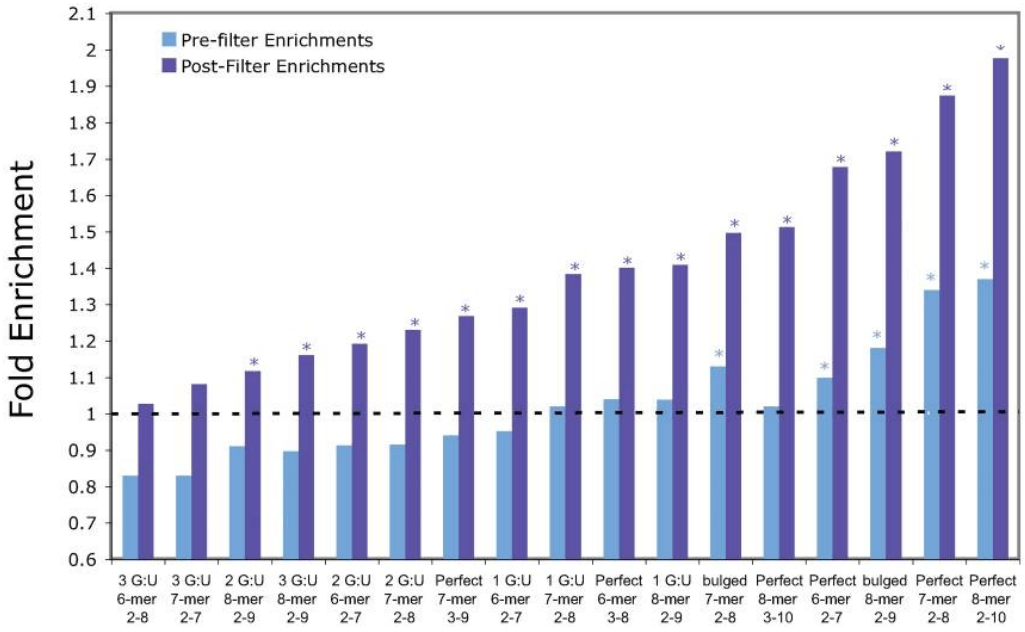
A gap, or bulge, is a type of structural abnormality where a base is mismatched on one or both sides of a binding to allow the rest of the sequence to pair. They generally occur as a result of secondary structure factors.



Source: Kim et al. (2005)

Figure 2.22: Binding abnormalities. An illustration of various binding structural abnormalities.

Gaps are particularly detrimental to miRNA efficacy when they occur in the alignment of the seed to the mRNA. Continuous sequences of at least 4-5 nt in this region are required for effective function, even with extensive compensatory binding (Brennecke et al., 2005). Nonetheless, sufficient compensation and favourable gap positioning can allow the overall binding to remain effective (Seok et al., 2016). Figure 2.23 shows how a bulged 7mer or 8mer may perform as well as a 6mer in these scenarios.



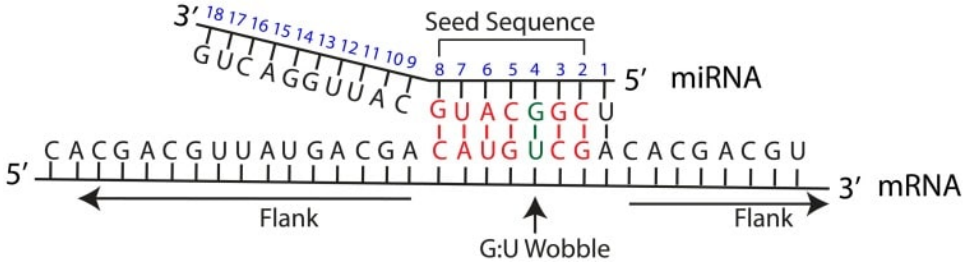
Source: Hammell et al. (2008)

Figure 2.23: Comparison of perfect and imperfect seed binding. A 7mer or 8mer containing a gap is as effective as a perfect 6mer, while a 7mer or 8mer with a single G:U wobble has similar efficacy to a 6mer offset binding. The inclusion of more than one wobble substantially reduces efficacy.

Gaps in the supplementary portion of a binding are tolerated until around five nt (Kiriakidou et al., 2004). There is no significant preference for symmetrical or asymmetrical gaps; however, traditional base pairing is still favoured over altered structural pairings (McGeary et al., 2022).

2.4.5 Wobble Pairs

Wobble pairs are those which occur outside the canonical Watson-Crick pairs, most notably G:U base pairings. Wobbles generally provide less binding stability than canonical pairings (Higgs, 2000), although this varies depending on their positioning and type (Didiano and Hobert, 2006). As with gaps, their presence in the seed is known to have a strong detrimental effect on miRNA efficacy (Doench and Sharp, 2004).



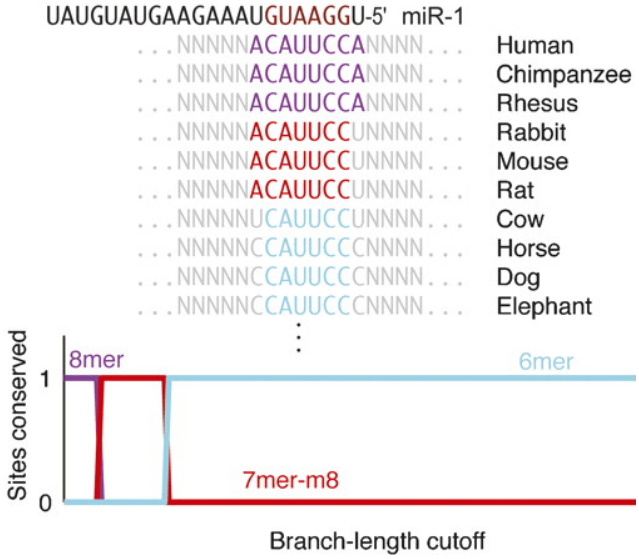
Source: Peterson et al. (2014)

Figure 2.24: Wobble pairing within the seed. The G:U pair in the seed is a wobble pair. While it may provide stability to the seed binding, it is inferior to a canonical base pairing.

A 7mer or 8mer with a single G:U wobble has similar performance to a 6mer offset binding (Xu et al., 2014), although more than one wobble substantially reduces efficacy (Figure 2.23).

2.4.6 Evolutionary Conservation

Sequences of genetic code may be retained across species irrespective of changes brought about by evolution. Such a resistance to mutation can therefore be interpreted as an indicator of biological importance. Conservation is often quantified by an alignment of multiple species sequence tracks (Figure 2.25). Where a set of bases do not commonly mismatch, they can be said to be conserved.



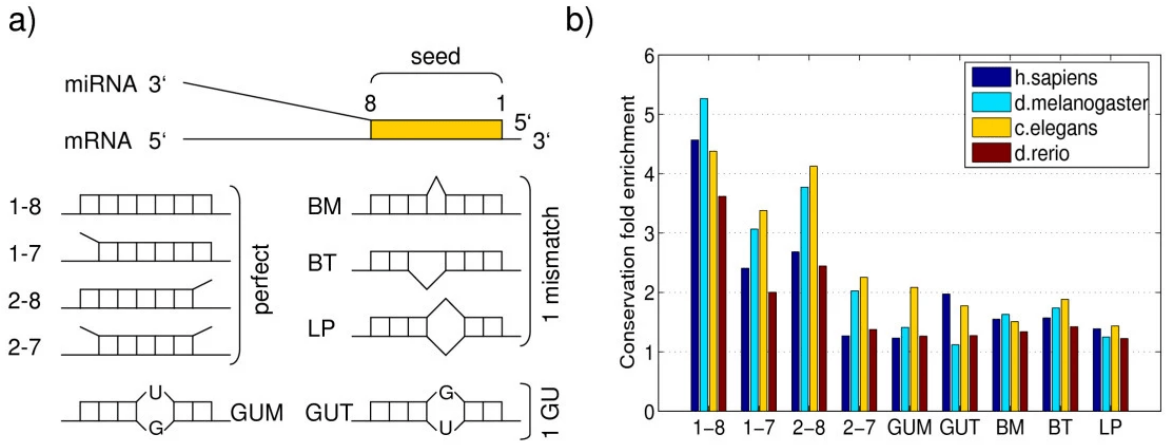
Source: Friedman et al. (2009)

Figure 2.25: Multiple aligned species conservation tracks. (Top) Ten species tracks aligned to compare sequence conservation. (Bottom) Conservation scores for the sequence tracks. The 6mer is more conserved than the 7mer-m8, itself more conserved than the 8mer.

In the context of miRNA:mRNA binding, seeds sites are well conserved in animals (Gaidatzis et al., 2007), across both mammalian and vertebrate species (Yang et al., 2011). A high level of conservation in a weaker seed type may make it more effective than a non-conserved, yet stronger seed type; a highly conserved 6mer will be more effective compared to a non-conserved 7mer (Lewis et al., 2005). However, it should be noted that low conservation does not necessarily mean a sequence is without function (Johnsson et al., 2014).

Figure 2.26 highlights how traditionally less effective seeds, such as the 6mer (2-7), are on average less conserved than the 7mer-a1 (1-7), 7mer-m8 (2-8) and 8mer (1-8).

Different gap types (BM, BT, LP) and G:U wobble arrangements (GUM, GUT) are also shown to have similar enrichment to a 6mer.



Source: Gaidatzis et al. (2007)

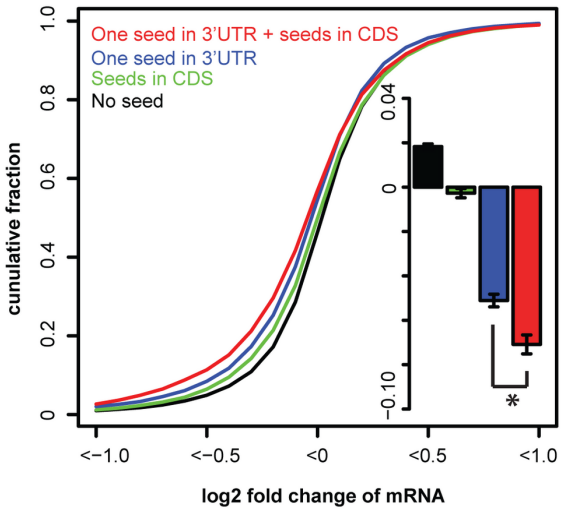
Figure 2.26: Conservation levels for different seed variations. (a) Breakdown of seed types: perfect (left), and imperfect as a result of gaps (right) and wobbles (bottom). (b) A comparison of seed type conservation across four species.

2.4.7 Target Site Abundance

A single mRNA molecule often contains multiple seed target sites (MTS) within its 3′ UTR. The same seed target sequence may also repeat multiple times throughout, particularly in the case of AU rich seeds (Garcia et al., 2011). In these instances, there is a strong correlation between the number of target sites for a given miRNA and the strength of the induced regulation from its binding, as each site can function redundantly (Doench et al., 2003). This increased down-regulatory effect is illustrated in Figure 2.18, where two 7mers perform more effectively than a single 8mer, which would otherwise be the biggest single determinant of efficacy.

The CDS is the primary targeting region for plant miRNA:mRNA interaction (Li et al., 2011), where near-perfect complementarity of the miRNA is common (Rhoades et al., 2002). This contrasts with animals, where meaningful targets are most prominent in the 3′ UTR. Nonetheless, there are approximately 1.1 million potential miRNA target sites in the 3′ UTR of *Homo sapiens*, less than the 1.6 million contained within the CDS (Zhou et al., 2009). Furthermore, miRNA target sites in the CDS of animals are functional (Hausser et al., 2013), though typically still less effective than those in the 3′ UTR (Schnall-Levin et al., 2011). Still, certain miRNAs are known to target the CDS specifically (Reczko et al., 2012).

Targets present in the CDS are able to provide redundancy to those located in the 3′ UTR. However, unlike MTS in the 3′ UTR, the effect is synergistic based on their presence rather than number (Fang and Rajewsky, 2011).



Source: Fang and Rajewsky (2011)

Figure 2.27: Synergistic effect of CDS targets on 3′ UTR targets. The bars indicate a CDS seed target is more effective than instances of no seed; however, it is significantly inferior to a 3′ UTR target. When the two exist together, a disproportionate effect is observed. This change is also represented using a cumulative plot (Section 2.3.2.1).

2.5 Computational Prediction of miRNA Targets

The prediction of miRNA:mRNA binding is a non-trivial computational problem due to the large number of potential mRNA targets available to each miRNA (Guo et al., 2014b). As discussed in Section 2.4, the existence of a seed target site is not solely sufficient in determining an effective binding, as there are many other factors to consider (Didiano and Hobert, 2006). This complexity is compounded by the many-to-many nature of the problem; a single target site may be targeted by miRNAs from the same family and an mRNA may contain a large number of different target sites within its 3′ UTR and CDS (Grimson et al., 2007).

Many prediction algorithms have been developed with the goal of predicting miRNA targets. These algorithms generally specialise in either plants or animals, or maintain separate versions, due to a lack of overlap in target recognition mechanisms between the two (Srivastava et al., 2014). Algorithms are typically published in the form of a command line program or web application and optionally provide a tabular data file containing pre-computed targets. Although early prediction efforts centred around

conditional rule sets with fixed thresholds, prediction tools increasingly utilise ML in their approaches.

2.5.1 Machine Learning

ML is the computational process of identifying underlying patterns in data through the development and application of algorithms (Bishop and Nasrabadi, 2006). More specifically, it refers to the training of a model on a set of data using an algorithm to analyse patterns and infer an output. ML solutions are largely automated, with little explicit programming (Samuel, 1959). This allows them to scale to large datasets, potentially recognising patterns that may otherwise be overlooked. The domain shares significant overlap with data mining, a parallel field centred around the understanding, processing and extracting of important 'features' from raw data to better suit analysis (Friedman, 1998). The extraction of such features is important in identifying characteristics and measurable properties from data in a form which the model can use to inform its decision-making process. ML also has a strong basis in statistics, as problems can benefit from both ML pattern prediction and statistical inference (Ij, 2018).

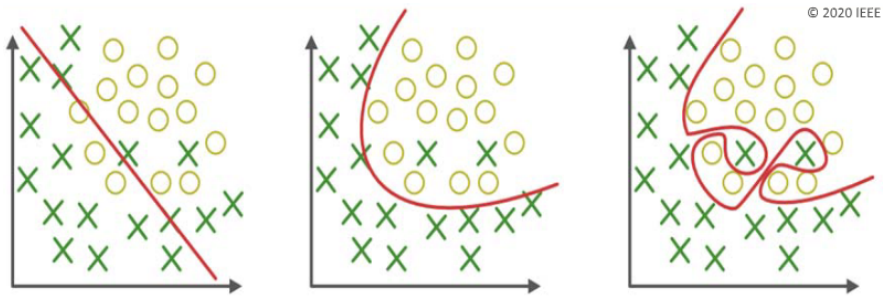
The process is broadly divided into supervised, unsupervised and reinforcement approaches. In supervised learning, data is 'labelled' with an answer, so the model can learn by comparison. This is contrary to unsupervised learning, where the model must maintain internal error correction metrics while attempting to reproduce provided examples. Similarly, in reinforcement learning, the model learns according to a feedback mechanism in the environment to reward positive learning (Kotsiantis et al., 2007).

In miRNA:mRNA target prediction, supervised learning is effective because the dataset can be built using experimentally verified examples (Riolo et al., 2020). Predictions may therefore be made by way of classification, in which a given miRNA is categorised as a target or non-target, or regression, where the model predicts a value such as the level of downregulation caused by an miRNA. There are many supervised learning algorithms; elements such as complexity, data availability, computational limitations and the cost of incorrect predictions each affect an algorithm's suitability to a given problem.

2.5.1.1 Sampling and Splitting

A dataset is typically split into training and test sets. During training, the model may only learn patterns from the training set to ensure that the test set remains an objective measure of performance. In supervised learning, the training set retains its label so it may learn by example. Conversely, this label is removed from test cases to prevent the model observing the answer during testing. A balance between these sets is required to build an optimised model and effectively evaluate its performance (Xu and Goodacre, 2018).

A common problem during training is 'overfitting', which occurs when the model learns its training data too closely, impeding its wider application to parallel problems. Inversely, there may also be 'underfitting', in which the model is unable to attain an in-depth understanding of the data.



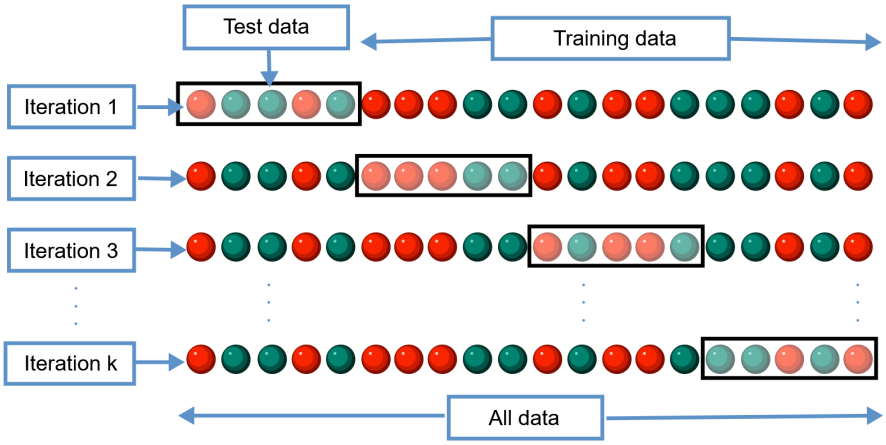
Source: Kolluri et al. (2020)

Figure 2.28: Model fitting types. A visualisation of a model's decision boundary while categorising two types of data points. (Left) Underfitting: the model's understanding of the problem is too simplistic; it cannot accurately solve it. (Center) Optimal fit: the problem is understood and outliers are ignored, a well-balanced solution. (Right) Overfitting: this specific problem is too well understood, the model will have issues scaling to other problems.

An optional subset, in addition to the training and testing sets, is the validation set. The use of validation allows the model a degree of self-correction during training through techniques such as 'early stopping', where the model stops training when overfitting is detected (Ying, 2019). Validation is also useful for optimising 'hyperparameters', high-level configuration settings unique to each ML algorithm that affect performance and generalisation. It may also be applied during feature selection to help reduce the bias that can otherwise occur (Fox et al., 2017). In this way, the validation set functions as a kind of 'internal' test set.

An alternative approach to using an explicitly defined validation set is cross-validation.

Cross-validation techniques sample the training set while withholding portions for internal testing, freeing up data that would have otherwise been reserved. The portion of data selected, and the number of iterations used to test it, depends on the method of cross-validation.



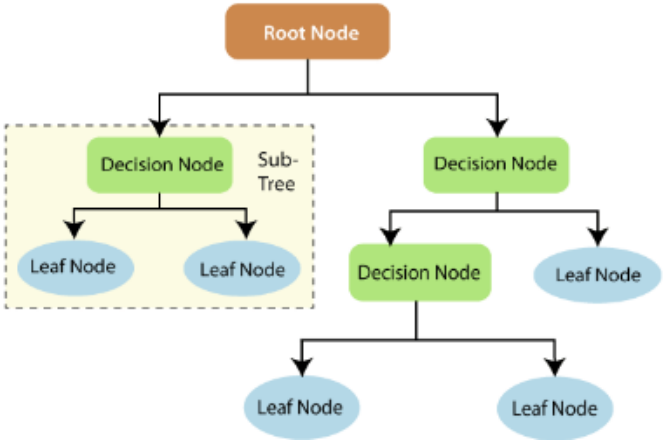
Source: Gufosowa, CC BY-SA 4.0

Figure 2.29: k-fold cross-validation. The data is iterated k times. Each time, $\frac{1}{k}$ of the data points are sampled for use in internal testing (validation) and the rest is used to train the model.

There is some consensus regarding data splitting ratios, but no explicit rules. Generally, the training set is composed of between 50% and 80% of the overall dataset. The test set should utilise the remainder of the data unless a validation set is used, in which case the validation and test sets should evenly share the remaining data.

2.5.1.2 Decision Tree

Decision tree (DT) uses a flowchart-like decision-making process, forming a graph (tree) built around edges (branches) and nodes. A DT begins with a singular node (root), consists of any number of layers (subtrees) containing internal decision nodes, and ends with one or more terminal nodes (leaves) (Safavian and Landgrebe, 1991). At each decision node, a dissection occurs with the intent of filtering data points toward a final classification.



Source: Charbuty and Abdulazeez (2021)

Figure 2.30: Visualisation of the DT algorithm. DT dissects data at each decision node, leading to leaf nodes of one or more data points.

With unrestricted depth and splitting rules, DTs tend towards overfitting, as each data point will eventually be individually categorised. The pruning of redundant subtrees is therefore a fundamental step in the DT algorithm (Mehta et al., 1995). There are many pruning algorithms for DT, for example CART (Breiman, 2017), which is also used in the random forest (RF) algorithm.

DT is a conceptually simple algorithm, which is likely a factor in its popularity (Charbuty and Abdulazeez, 2021). ML is highly stochastic and difficult to generalise, however individual DTs often have poor accuracy when applied to complex problems (Speiser et al., 2019).

2.5.1.3 Random Forest

Ensemble models utilise a collection of internal classifiers to produce an aggregate prediction using various weighting and resampling techniques. Despite the simplicity of the theory, ensembles commonly outperform single classifiers (Dietterich, 2000). A popular ensemble algorithm is RF, a homogeneous ensemble of CART-based DTs. RF maintains similar strengths to an individual DT, while achieving greater performance and a stronger inherent resistance to overfitting (Breiman, 2001).

At its core, RF constructs a number of randomised CART trees. Each tree is provided with different data subsets through bootstrap aggregating (bagging), a technique that reduces the instability of estimates in complex problems by maintaining bias while reducing variance (Biau and Scornet, 2016). Bagging creates random samples from a

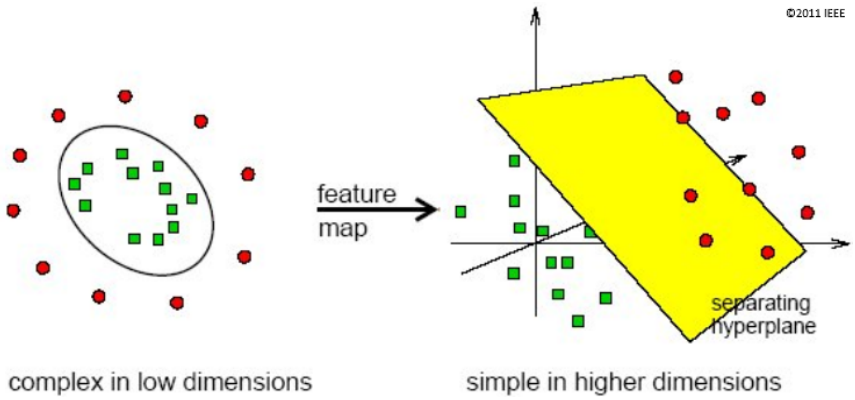
dataset 'with replacement', meaning that an instance of data may be selected more than once. Individual trees influence RF output by voting using unweighted (classification) or weighted (regression) scoring (Cutler et al., 2012).

RF is a consistently high-ranking algorithm when applied to general classification problems (Fernández-Delgado et al., 2014). It has seen use in bioinformatics because of its efficiency at handling large and complex data structures, resistance to overfitting, and effective generalisation due to bagging (Qi, 2012). Its good out-of-the-box performance and relative simplicity, due to the low number of hyperparameters requiring tuning, have made RF a popular ML algorithm (Biau and Scornet, 2016).

2.5.1.4 Support Vector Machine

Support vector machine (SVM) is a mathematically founded algorithm that aims to optimise the decision boundary (hyperplane) between data points (support vectors) (Karatzoglou et al., 2006). This is achieved through an iterative process of shifting the hyperplane against measurements taken from candidate support vectors to ensure the space (margin) is maximised.

The SVM kernel function is responsible for transforming data points into a higher dimension. In doing so, SVMs are able to form the hyperplane in a space allowing for a linear separation that would otherwise be impossible.



Source: Kwak (2013)

Figure 2.31: SVM kernel transformation to create linear separation. In 2D, the data types are not linearly separable. In 3D, the hyperplane (yellow) can more simply dissect the two groups.

SVMs have been used in biological problems to classify microarray gene expression profiles, proteins, and DNA sequences (Noble, 2006). However, SVMs have performance issues regarding runtime complexity and speed (Osuna and Girosi, 1998); in particular,

non-linear kernels cannot scale to problems with large datasets or feature counts (Lin and Lin, 2003). A moderately sized dataset may also encounter storage space issues, with constraint matrices reaching into the millions of cells with only a few thousand data points (Lee and Mangasarian, 2001).

2.5.1.5 Deep Neural Network

Artificial neural network (ANN) is an umbrella term for algorithms that simulate the learning processes of biological neural networks in order to capitalise on their high capacity for learning, adaptability, generalisation and massive parallelism (Jain et al., 1996).

ANNs are formed of a network of nodes (neurons), connected by edges (synapses) and arranged into layers. The first layer is referred to as the 'input layer' and the last is the 'output layer'; there may be any number of layers between them, known as 'hidden layers' (Figure 2.32). An ANN is said to be 'deep', or a deep neural network (DNN), when the number of hidden layers exceeds one, and 'very deep' at ten (Schmidhuber, 2015). The direction that information flows between layers determines whether it is feedforward or recurrent. While the former allows neurons to connect bidirectionally, feedforward ANNs are sequential; information may only transmit backwards in feedforward ANNs via backpropagation, where weights are progressively tuned according to a loss function.

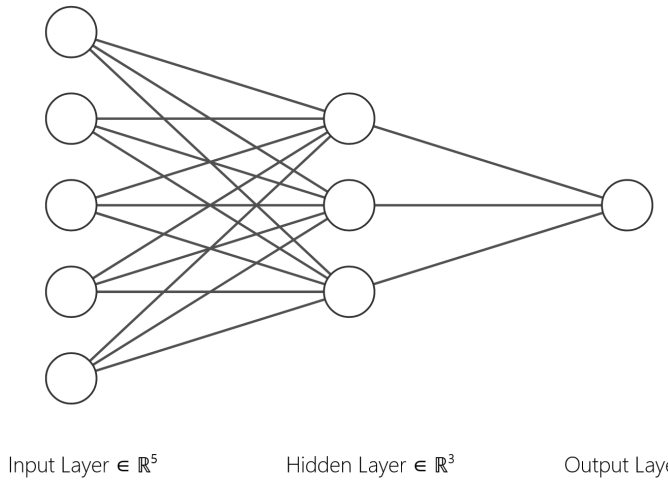
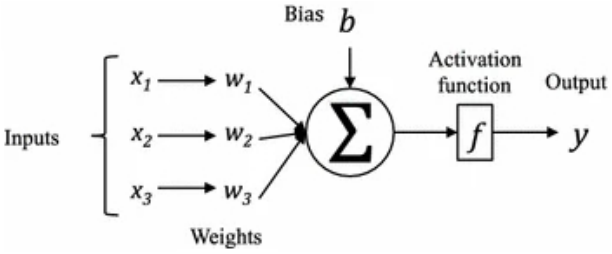


Figure 2.32: Densely connected DNN layers. The network is composed of three layers, one of them being a hidden layer. The layers are dense, as each neuron is fully connected to the next.

Each neuron assigns a weight to incoming synapses. These weights are collected and joined with a bias in the neuron's summation function (Equation 2.5.1). The overall

output of the neuron is then determined by feeding the net input into an activation function, which limits the result to a finite value (Dongare et al., 2012). This overall process is illustrated in Figure 2.33.

$$net = \sum_i (x_i w_i) + b. \tag{2.5.1}$$



Source: Koutsoukas et al. (2017)

Figure 2.33: Mathematical breakdown of an artificial neuron. The sum of the weights and inputs are combined with a bias in the summation function (Equation 2.5.1). This value is then supplied to the activation function, which determines the overall output of the neuron.

A single hidden layer is sufficient to solve most problems; however, a second hidden layer allows DNNs to represent functions of any shape at the cost of potential overfitting (Heaton, 2008). Applying dropout, where learned information is randomly omitted, at 50% between densely connected layers is an effective means of reducing overfitting, as it forces the network towards a more averaged understanding of the problem (Hinton et al., 2012). Although it is problem-specific, DNNs with two hidden layers generally perform more effectively than those with a single hidden layer (Thomas et al., 2017), and the use of a second hidden layer may optimise accuracy (Stathakis, 2009).

Deep learning has seen biological application specifically in miRNA research, as its ability to understand complex patterns is useful in problems that are otherwise intractable (Mahmud et al., 2018). Still, a significant setback to its application is that it requires considerable datasets to function effectively (Chen et al., 2018a). The importance DNN places on features is often described as being a 'black box' compared to other algorithms (Almeida, 2002). However, this can be somewhat mitigated by adjusting input groups to observe effects on the output (Li et al., 2019).

2.5.2 Prediction Tools

Table 2.1 lists by citation the most popular tools occupying the same problem domain as this study. For consideration, a tool must support prediction for *Homo sapiens* and not be a pure aggregator of other tools' outputs. Information is collected and cross-referenced between Tools4miRs (Lukasik et al., 2016), miRToolsGallery (Chen et al., 2018b), a published study of common features (Peterson et al., 2014) and each tool's publications and official website.

Table 2.1: Summary of popular target prediction tools 1/2

Name (Ver.*)	Versions Published	Use	Cited†	Active‡
TargetScan (8.0)	1.0 (Lewis et al., 2003) 2.0 (Lewis et al., 2005) 4.0 (Grimson et al., 2007) 5.0 (Friedman et al., 2009) 6.0 (Garcia et al., 2011) 7.0 (Agarwal et al., 2015) 8.0 (McGeary et al., 2019)	Both	42,514	✓
miRanda (mirSVR 3.3a)	1.0 (Enright et al., 2003) 2.0 (John et al., 2004) microRNA.org (Betel et al., 2008) miRanda-mirSVR (Betel et al., 2010)	CLI	15,410	×
PicTar	Original (Krek et al., 2005) Update (Lall et al., 2006)	Web	6,076	×
MirTarget (v4.0)	MirTarget (Wang and Wang, 2006) MirTarget2 (Wang and El Naqa, 2008) miRDB (Wang, 2008) miRDB 2015 (Wong and Wang, 2015) MirTarget3 (Wang, 2016) MirTarget v4.0 (Liu and Wang, 2019) miRDB 2020 (Chen and Wang, 2020)	Web	5,529	✓
RNAhybrid (2.1.2)	RNAhybrid (Rehmsmeier et al., 2004) Web update (Krüger and Rehmsmeier, 2006)	Both	4,208	×

* Version refers to the latest version of the algorithm, as opposed to the tool.

† Via Google Scholar. For tools with multiple versions, citations are summed across papers.

‡ Activity is determined by the presence of a maintained web application or update within five years.

Table 2.2: Summary of popular target prediction tools 2/2

Name (Ver.*)	Versions Published	Use	Cited†	Active‡
DIANA-microT (CDS)	microT (Kiriakidou et al., 2004) microT web (Maragkakis et al., 2009b) microT-v4.0 (Maragkakis et al., 2011) microT-CDS (Reczko et al., 2012) microT-v5.0 (Paraskevopoulou et al., 2013) microT 2023 (Tastsoglou et al., 2023)	Web	4,003	✓
MiRscan	Original (Lim et al., 2003a) Update (Lim et al., 2003b)	Web	3,548	×
PITA (6)	(Kertesz et al., 2007)	Both	2,693	×
RNA22 (2.0)	1.0 (Miranda et al., 2006) RNA22-GUI (Loher and Rigoutsos, 2012)	Both	2,602	×
IntaRNA (2.4.1) CopraRNA (2.1.4)	IntaRNA (Busch et al., 2008) Freiburg RNA Tools (Smith et al., 2010) CopraRNA (Wright et al., 2013) CopraRNA-IntaRNA (Wright et al., 2014) IntaRNA 2.0 (Mann et al., 2017) Freiburg RNA T. 2018 (Raden et al., 2018)	Web	1,947	✓
miRCode (11)	(Jeggari et al., 2012)	Web	627	×
ElMMo (3)	ElMMo (Gaidatzis et al., 2007) MirZ (Hausser et al., 2009)	Web	523	×
TargetRank	(Nielsen et al., 2007)	Web	511	×

* Version refers to the latest version of the algorithm, as opposed to the tool.

† Via Google Scholar. For tools with multiple versions, citations are summed across papers.

‡ Activity is determined by the presence of a maintained web application or update within five years.

The majority of popular tools were first published between 2003 and 2008 and received an average of three major publication updates. TargetScan, MirTarget and DIANA-microT are notable for being the only tools from the original batch to still receive regular updates, a likely factor in their comparatively high citation counts. Despite being first to publish, miRanda has not received an update since its mirSVR overhaul in 2010. PicTar, RNAhybrid and PITA were also relatively popular upon release, yet were only maintained for a short period of time.

TargetScan, miRanda-mirSVR, MirTarget and DIANA-microT are identified for further investigation due to their popularity and, except miRanda-mirSVR, status as actively maintained tools. Although it has not been recently updated, miRanda-mirSVR is instead chosen because of its research significance. Although all four tools initially used rule-based prediction, their current versions utilise ML.

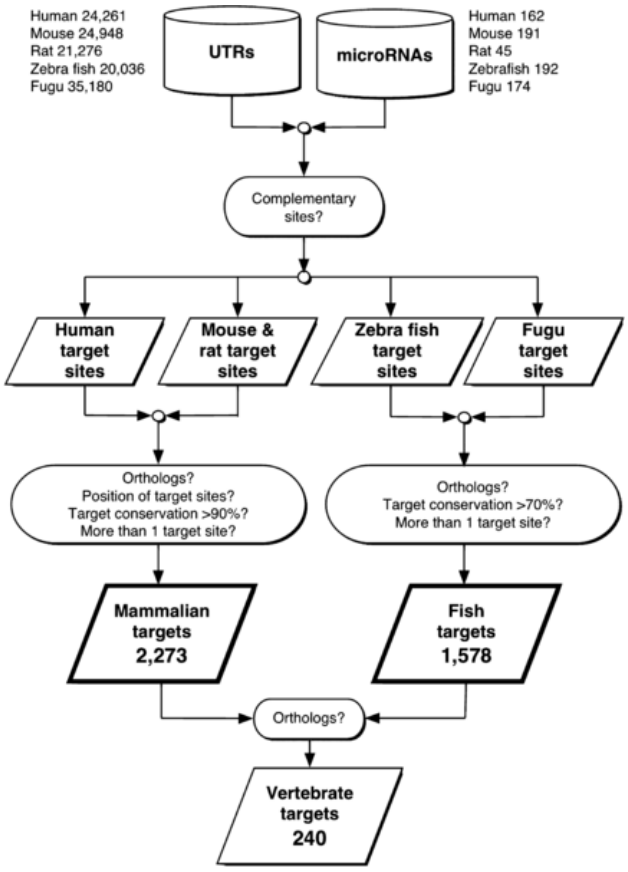
2.5.2.1 TargetScan

TargetScan has undergone significant changes throughout its eight primary versions, though each iteration generally builds on existing logic. The original version scans the 3′ UTR for perfect seed matches and extends out the window until a mismatch occurs. This is combined with several other metrics, such as thermodynamic stability, to generate a score (Lewis et al., 2003). Version 2.0, known as TargetScanS due to its reliance on targets built about 6mers, mostly simplifies this approach by removing redundant logic and focusing on conserved 6mer, 7mer and 8mer matches (Lewis et al., 2005). Version 4.0 introduced a primitive version of the context score, where seed type, supplementary pairing, local AU and positional contributions are summed to grant each target a confidence metric (Grimson et al., 2007). This was superseded in version 6.0 by context+, which also accounts for MTS and seed pairing stability (Garcia et al., 2011). In TargetScan 7.0, the context++ model was introduced. Here, an ensemble of four linear regression models are separately trained on 6mer, 7mer-a1, 7mer-m8 and 8mer (Agarwal et al., 2015). Of the 26 total features, 14 are selected based on their importance towards each category.

TargetScan 8.0 marks a fundamental shift from traditional miRNA:mRNA targeting by providing an additional option for ranking predictions using a novel biochemical approach (McGeary et al., 2019). Since miRNA is loaded into AGO to facilitate binding (Figure 2.15), the usage of RNA bind-n-seq to determine miRNA:AGO occupancy allows a more direct approach to target recognition. This method also utilises a convolutional neural network (CNN) trained on values derived from RNA bind-n-seq using TensorFlow (Abadi et al., 2015). According to TargetScan, the predictions made do not significantly differ from previous versions, as the change only affects its ranking system.

2.5.2.2 miRanda-mirSVR

miRanda-mirSVR trains a support vector regressor (SVR) (Smola and Schölkopf, 2004), a type of SVM, on predictions output by miRanda. The miRanda-mirSVR model also expands the feature set of the original miRanda with new site accessibility features. This allows miRanda to remain competitive with newer prediction tools without sacrificing the accuracy of the original algorithm (Betel et al., 2010). The miRanda portion of the algorithm uses a conditional rule set to determine targets according to a primitive feature set, including conservation and the existence of seeds (John et al., 2004). Predictions are then funnelled through mirSVR, where they are scored and ranked.



Source: John et al. (2004)

Figure 2.34: Rule flowchart of miRanda 2.0. The miRanda algorithm predicts targets in accordance with a set of rules. The output targets are used as input to miRanda-mirSVR.

2.5.2.3 MirTarget

The MirTarget algorithm is used to generate predictions for miRDB. In developing MirTarget v4.0, a large-scale RNA-seq dataset of 25 miRNA transfection experiments was produced to identify and rank key binding features (Table 3.1). An SVM is trained on a total of 96 features, largely centred around per-base nt identification. Although all

features are used in the final model, regardless of statistical significance, each feature is ranked by recursive feature elimination (RFE) using Weka (Eibe et al., 2016), where one feature is removed per iteration (Liu and Wang, 2019). The SVM's hyperparameters are optimised using LIBSVM (Wu et al., 2003), which is also used to output a probability score for each prediction. The use of RFE and probability scoring had first been introduced in MirTarget3 (Wang, 2016), with ML via SVM beginning in MirTarget2 (Wang and El Naqa, 2008). The original tool used a heuristic approach where feature filters were weighted by importance. These weights allowed stronger filters to bypass weaker filters, provided certain thresholds were met (Wang and Wang, 2006), which is otherwise a limitation of rule-based prediction.

2.5.2.4 DIANA-microT

DIANA-microT is an algorithm branch of the DIANA Tools suite. Predictions are made by combining positive and negative influencing features using linear models, internally referred to as miRNA-recognition elements (MREs) (Reczko et al., 2012). The feature set is optimised using the Akaike information criterion from the R MASS package (Venables and Ripley, 2013), an estimator of error for a prediction set. The overall MRE score is computed in DIANA-microT by comparison of SVM, ANN, RF and generalised linear models. Before ML, prior versions of the program formed their predictions by sliding sequence windows along the 3′ UTR to compute the binding energy of pairings measuring at least three nt. After extraction, these windows would be processed to remove overlaps and filtered into a set of predictions using feature rules (Kiriakidou et al., 2004).

DIANA-microT-CDS is unique in its additional handling for CDS targets. In this iteration of the tool, all optimisation is performed separately for both the CDS and 3′ UTR, as both are believed to use different MREs. In doing so, DIANA-microT-CDS can support CDS targets without lowering its accuracy when predicting 3′ UTR targets.

2.5.3 Seed Definition

In all tools except MirTarget, the traditional 6mer, 7mer-a1, 7mer-m8 and 8mer definitions are supported, while 6mer offset is monitored in some capacity (Table 2.3). The same tools also allow a maximum of one G:U wobble pair to be present within the

seed. MirTarget and DIANA-microT use slightly unorthodox seed definitions, as they are also concerned with the presence of a match at position 1 independently of an A (6merα). By extension, this means the alternative 7mer-m1 and 8mer-m1 seed types are also recorded. Despite testing these additional possibilities, MirTarget ultimately does not incorporate them into the algorithm, arguing that the enrichment is inferior to 7mer-a1, 7mer-m8 and 8mer (Liu and Wang, 2019).

Table 2.3: Seed type usage of several popular modern tools

	6merα	6mer	6mer Offset	7mer-a1	7mer-m8	8mer	G:U Wobble
TargetScan 8.0		✓	✓*	✓	✓	✓	✓
miRanda-mirSVR 3.3a		✓	✓	✓	✓	✓	✓
MirTarget v4.0				✓	✓	✓	
DIANA-microT-CDS	✓	✓	✓	✓	✓	✓	✓

* 6mer offset is counted for MTS in the 3′ UTR only, not as a primary seed type.

TargetScan regards each seed as an independent model, allowing feature importance to be determined relative to seed type. MTS scores 100% in all seed types; however, the identity of miRNA base 8 scores 0%, 0.8%, 100% and 100% for 8mer, 7mer-m8, 7mer-a1 and 6mer respectively. This suggests an 8mer binding is unconcerned with the identity of base 8, likely because the seed type already encodes this information. For situational features such as this, splintering the model adds context to training, which may improve prediction accuracy.

Tools which implement seed type as a feature may do so in different ways depending on the number and stringency of seed definitions used. For example, a 6mer can be represented as a boolean `6mer` or by encoding six `pair_2`, `pair_3`, `pair_4`, `pair_5`, `pair_6`, `pair_7` descriptors. Individual base flags allow a free-form approach to seed recognition, making support for alternative seed definitions easier. However, this complicates the feature set with intermediary ‘sub-features’, which must be combined to offer useful information. DIANA-microT and miRanda-mirSVR each use variations of

this approach, though miRanda-mirSVR does not record matches at position 1. MirTarget uses the fixed definitions 'seed 7A1', 'seed 7b' and 'seed 8A1', corresponding with 7mer-a1, 7mer-m8 and 8mer.

MirTarget is more discerning in what it categorises as a potential target; it has both the strictest seed definitions and does not include handling for wobbles or compensatory 3′ pairing (Figure 2.4). On the other hand, DIANA-microT has the loosest definitions, tracking the seed to an individual base level and even detecting 9mer and base 1 matching. Meanwhile, TargetScan's per-seed models mean a greater importance is placed on the type of seed relative to its features.

2.5.4 Common Target Recognition Features

Target prediction tools use a broad range of published features known to be biologically significant. In the current era, the use of ML has led to an increase in the number of interconnected features. Although prediction tools often make use of similar feature sets, there is no consistently recognised optimal set for target prediction. Figure 2.4 provides a breakdown of those which appear in at least two of the considered tools.

Table 2.4: Common features of several popular modern tools

	Binding Stability	Supplementary Sites	Conservation	Site Accessibility	Local AU Content	Dist to 3′ UTR end	3′ UTR MTS	3′ UTR Length	CDS MTS
TargetScan 8.0	✓	✓	✓	✓	✓	✓	✓	✓	✓
miRanda-mirSVR 3.3a	✓	✓	✓	✓	✓	✓		✓	
MirTarget v4.0	✓		✓	✓		✓		✓	
DIANA-microT-CDS	✓	✓	✓	✓	✓	✓	✓		✓

Chapter 3

Rule-based prediction of miRNA:mRNA Targets

3.1 Summary

This chapter describes the development of a naive target prediction rule set using fixed feature thresholds. The goal of this work is to build a proof of concept for established prediction methods in order to inform the development of more advanced algorithms and tooling. The structure of this algorithm resembles early attempts at target prediction, such as miRanda's conditional ruleset (Figure 2.34) and MirTarget's original heuristic filter approach.

The predictor is scripted using a combination of Bash (GNU, 2007) and R (R Core Team, 2018), for the bioinformatics and statistics modules, respectively. There are three key components to the algorithm: extraction of expression values from miRNA transfection datasets through the use of a bioinformatics pipeline, computation of binding features from seed target sites, and target prediction using fixed feature rules and thresholds. At each stage, a tabular output is produced for each sample, functioning as both an ad hoc caching system and a method of modular debugging. This simplified engineering allows the tool to grow in line with the project scope, without committing to any singular design pattern prematurely.

The end-to-end process of prediction from raw sequencing data is defined in Figure 3.1. Input RNA-seq reads from miRNA transfection experiments (Section 3.2.1) are

trimmed (Section 3.2.2), aligned against a reference gene set and used to generate gene abundances (Section 3.2.3). This output is piped into R for several layers of processing, including gene annotation (Section 3.2.4), locating of seed targets (Section 3.2.5), and calculation of expression values (Section 3.2.6). To determine factors relating to the structure of the binding, such as accessibility and paired bases, windows about the seed targets are fed through two folding prediction algorithms (Section 3.2.7). Finally, conservation scores are generated for the paired sequences (Section 3.2.8). Using a combination of these extracted features, transcripts are then categorised as either targets or non-targets.

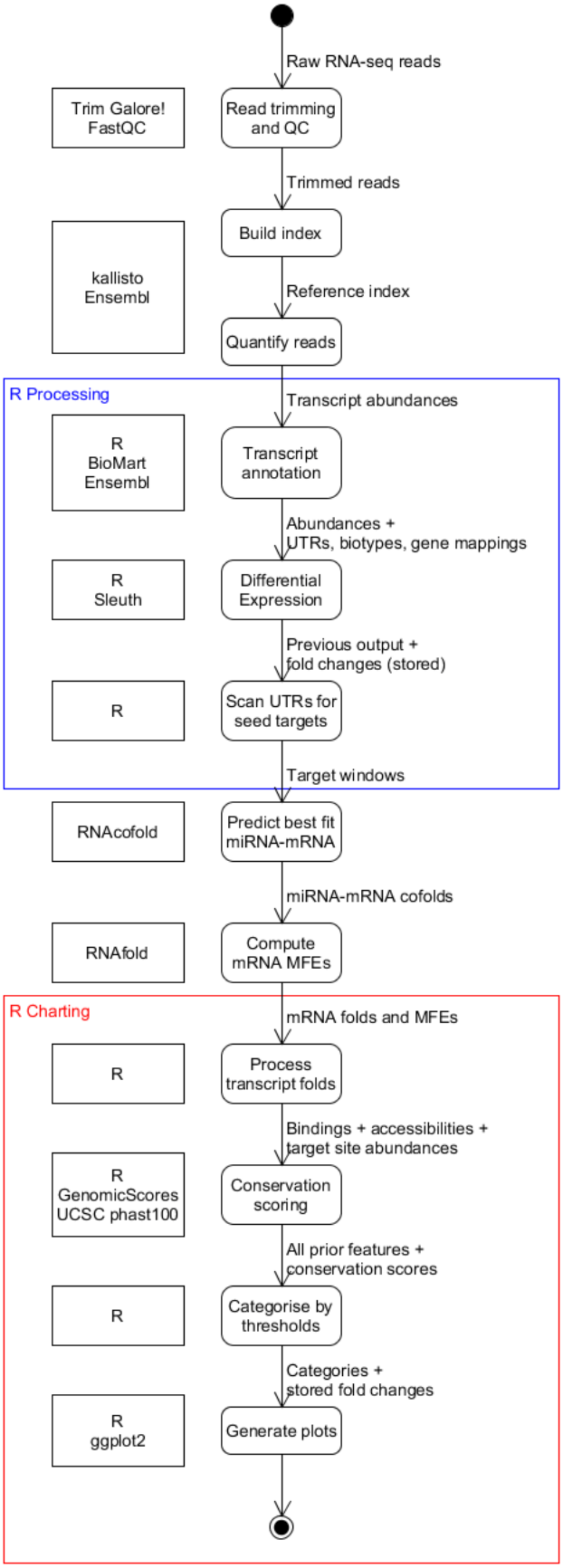


Figure 3.1: Rule-based prediction pipeline. The pipeline begins with the extraction of expression values and a differential expression analysis. This is followed by feature extraction and finally prediction through classification using rules.

3.2 Methods

3.2.1 Transfection Dataset

A large batch of sequencing data, recently published alongside the fourth iteration of the MirTarget prediction tool, is used for the development of this rule-based prediction algorithm (Liu and Wang, 2019). The dataset is designated the identifier '01-liu-HeLa' because it is the first dataset considered, made available courtesy of Liu and Wang (2019), and utilises the immortalised Henrietta Lacks cervical cancer cell line (HeLa).

Publicly available miRNA transfection datasets are uncommon and generally limited to one or two miRNA transfections per publication. 01-liu-HeLa is noteworthy in that it contains 25 transfections from a single cell line. This is beneficial because it limits potential batch effects, which may be evident in the low variance between samples (Figure 3.2). While a homogeneous dataset could lead to a tendency towards miRNA targeting features that are overly represented in HeLa, it has been shown that miRNA targeting is mostly unaffected by cell line (Nam et al., 2014). The biggest weakness of 01-liu-HeLa is arguably in its relatively low number of biological replicates for noise reduction. However, the dataset is proven for this purpose, as it was previously used as the sole dataset in the development of a popular target prediction algorithm.

Table 3.1: An overview of 01-liu-HeLa

Internal ID	01-liu-HeLa
Accession	PRJNA512378
Species	*Homo sapiens*
Data Type	RNA-seq
Procedure	25 miRNA transfections
Cell Line	HeLa
Biological Replicates	2
Sequence Type	Single-end
Source	Liu and Wang (2019)

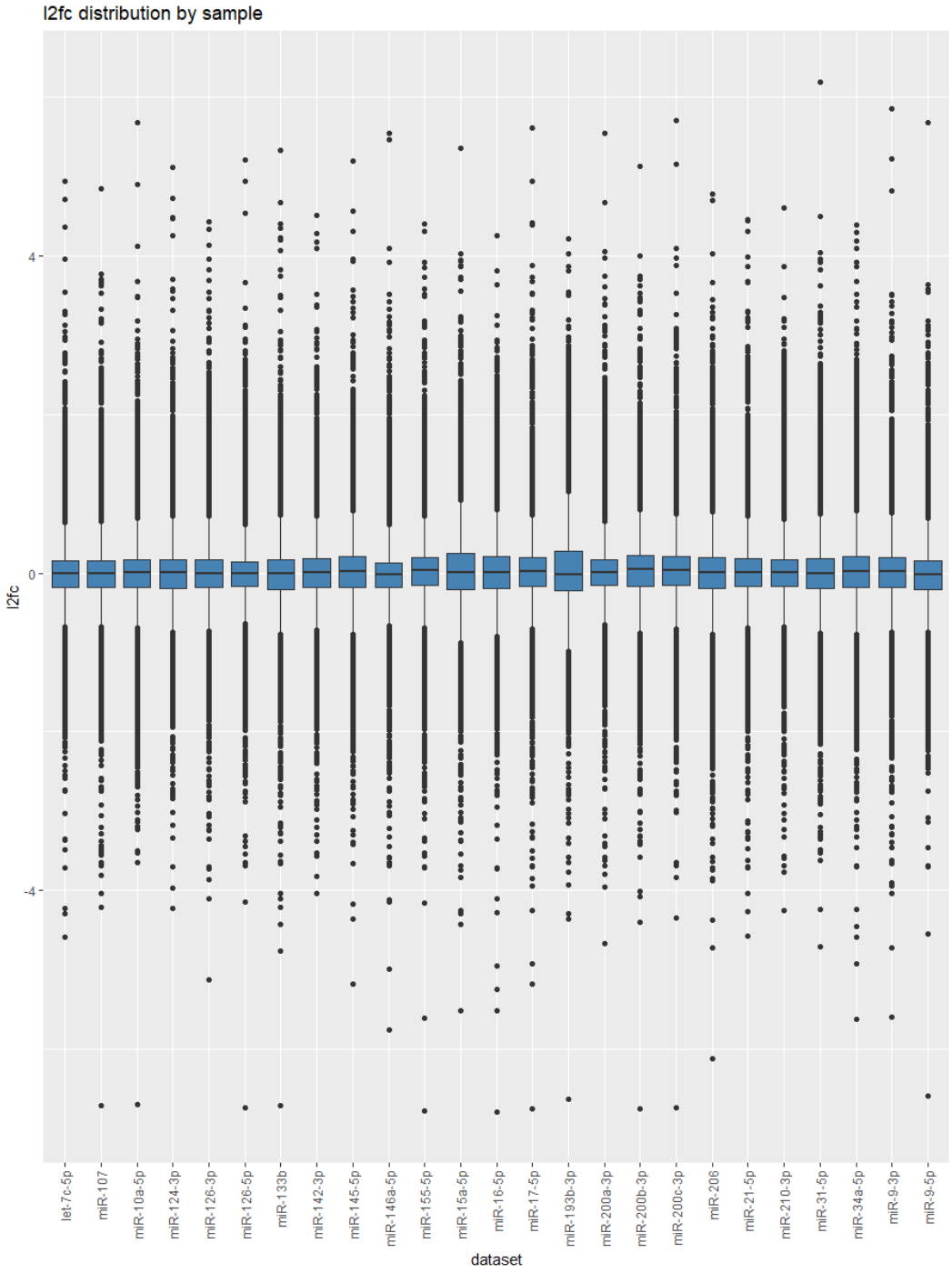


Figure 3.2: Unfiltered expression fold change variance between samples in 01-liu-HeLa. The distribution of unfiltered log_2 expression fold change between the 25 transfections of 01-liu-HeLa. The samples have relatively similar distributions.

3.2.2 Read Trimming

NGS techniques, such as RNA-seq, output a large number of short reads with varying quality and error rates (Section 2.3). Low-quality reads are not fully representative of their original sequence, making them less likely to align against a reference, particularly when perfect match settings are used. Trimming is therefore an important

pre-processing step to ensure a higher average quality, in turn leading to a better mapping rate, and functions as a method of assuring a standard across datasets.

When genetic data is sequenced in the FASTQ format, reads are accompanied by a Phred quality score (Ewing et al., 1998). This is a representation of the quality of each base at the time of sequencing. Using Phred scores, bases falling beneath a specified threshold can be truncated or discarded, improving the overall read quality. Beyond truncation, trimming can also offer improved mapping rates by way of removing 'adaptor content', specific base sequences inserted by many NGS procedures to facilitate sequencing. Since adaptor content is not present in the species genome, they complicate the task of alignment by appearing as mismatched segments.

Figure 3.3 illustrates how sequencing data from EX-guo-U20S (Appendix C) fares before and after trimming. Prior to trimming, a deterioration in average read quality can be seen towards the 3′ end of the reads. In this instance, trimming resulted in a reduction in errors across all bases of the sequenced reads, with the effect heightening from base 12.

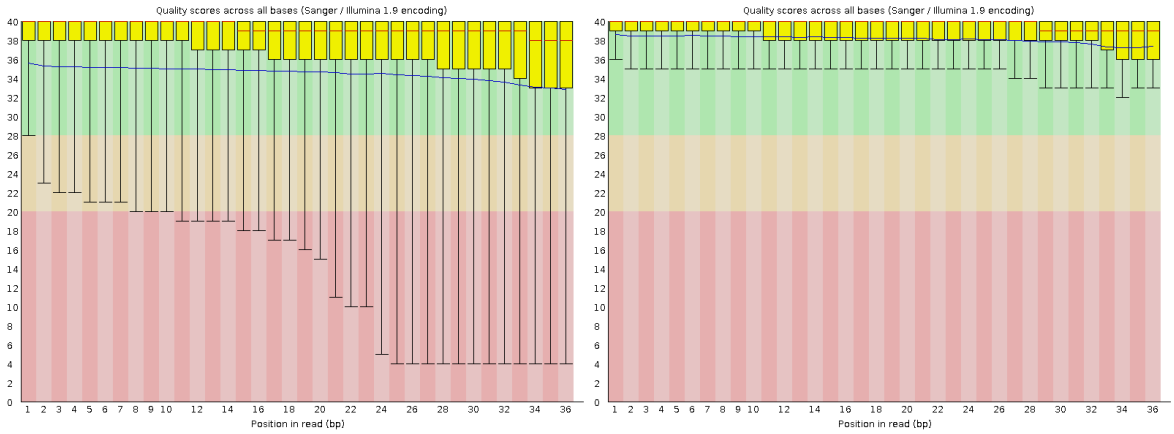


Figure 3.3: FastQC read quality before and after trimming. Yellow error bars indicate per-base Phred scores, where a longer bar is indicative of lower quality. Before trimming, the rate of error is higher.

Trimming is performed in this project using Trim Galore! (Krueger, 2015), a simplified wrapper for the Cutadapt trimming tool (Martin, 2011). It provides automatic quality-based trimming and effective default settings for general scenarios. Trim Galore! is executed in this pipeline using identical flags to previous work published in the lab (Bradley and Moxon, 2019). Read quality is evaluated both before and after trimming with FastQC (Andrews et al., 2010), a reporting tool used to generate

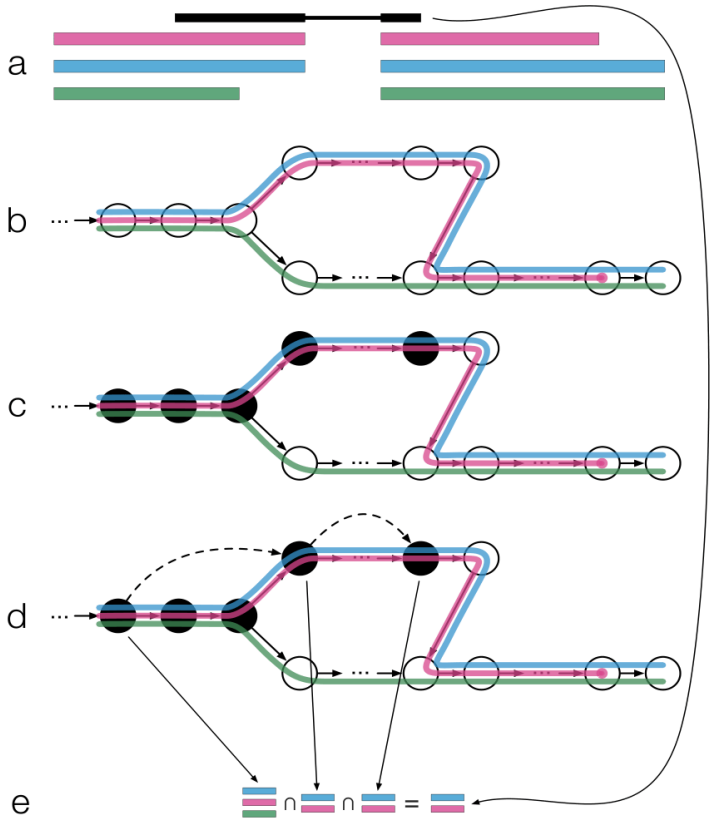
visualisations of various read attributes. No specific flags are used with FastQC.

```
trim_galore --length 35 --stringency 4 [input_files]
```

3.2.3 Sequence Alignment

kallisto (Bray et al., 2016) is used to quantify gene expression due to its streamlined 'pseudoalignment' approach to sequence alignment. Unlike other alignment tools, kallisto does not explicitly map reads to the genome itself, instead matching them against a set of transcript sequences. In addition to increased computational efficiency, this inherently favours granular transcript-level output as the analysis is performed transcriptome-wide compared to the gene-level of traditional genomic mapping.

In quantifying expression, kallisto first constructs an index structure similar to those used in assembly algorithms (Section 2.3.1). Within this structure, each *k*mer is mapped to its parent contig and given a relative position, allowing the program to take shortcuts during lookup procedures (Figure 3.4). With the index complete, kallisto uses a likelihood function to count RNA transcript abundances from reads.



Source: Bray et al. (2016)

Figure 3.4: Overview of kallisto pseudoalignment. (a-b) Creation of a graph where *k*mers (circles) are formed along reads (coloured lines). (c-d) *k*mer compatibility information (overlaps) reduces computational requirements by skipping redundant nodes. (e) The intersection of compatibility values from non-redundant *k*mers is taken to determine a read's *k*-compatibility, essentially indexing it.

A kallisto index is generated using Ensembl's GRCh38 release 101 (Cunningham et al., 2018) using the `index` command with default settings. Abundances are computed by passing the generated index and FASTQ files to kallisto's `quant` command. Command line flags used here are again adapted from prior work from the lab (Bradley and Moxon, 2019), with bootstraps `-b` also set to allow for differential expression analysis (Section 3.2.6).

```
kallisto quant -i [index_file] --single --bias -l 180 -s 20 -b 100 -t
    8 [input_file]
```

3.2.4 Gene Annotation

Annotations are pulled from Ensembl databases using the biomaRt package (Durinck et al., 2005) in R. Annotation is important in supplementing transcripts with the information required to facilitate further processing, such as the 3′ UTR sequences used in identifying potential target sites. Furthermore, because kallisto outputs count data relative to transcripts, gene mappings must be provided to support gene-level analysis (Soneson et al., 2015). Table 3.2 contains a full breakdown of annotations used to supplement transcript processing. At this stage, custom filters are used to remove any transcripts that are non-coding, contain *NA* values, or are missing annotations. Additionally, only transcripts from chromosomes 1-22, X and Y are considered.

Table 3.2: Annotations from biomaRt used to support target prediction

Annotation Name	Use in algorithm
`3utr`	Locate seed target sites
`3_utr_start`	Conservation scoring
`3_utr_end`	Conservation scoring
`biotype`	Protein-coding gene filter
`chromosome_name`	Conservation scoring
`ensembl_gene_id`	Transcript-gene map
`strand`	Conservation scoring
`transcript_mane_select`	Map representative transcript of each gene

The `transcript_mane_select` field is provided by the Matched Annotation from the NCBI and EMBL-EBI (MANE) project, which labels the representative transcript of each gene (Morales et al., 2022). This information allows transcript-level details,

such as 3′ UTR sequences, to be attained even when accessed at the gene level, where traditionally the relationship would be one-to-many.

3.2.5 Seed Target Detection

This work defines a potential target as one built around a minimum perfectly matched 6mer seed. Seed target sequences are computed by taking the reverse complement of the transfected miRNA's 6mer site using miRNA sequences provided by miRBase (Kozomara et al., 2019). This target sequence is then used to scan the 3′ UTR of each mRNA transcript for pattern matches. Since pattern matching cannot account for imperfect pairing, such as G:U wobbles or gaps, further specification of the target site is only determined during folding (Section 3.2.7.2). In instances where a 3′ UTR contains more than one target site, the associated transcript data is duplicated and treated as a unique case to support MTS features.

3.2.6 Differential Expression Analysis

Differential expression analysis is a statistical method for comparing the levels of expression in a sample against a control group. The difference in expression is often represented as log_2 fold change; the ratio difference between two expression values applied to a log_2 scale (Section 2.3.2). Two tools were considered for this task: DESeq2 (Love et al., 2014), due to its status as an industry standard tool, and Sleuth (Pimentel et al., 2017), as it shares a pipeline tie-in with kallisto and thus the option of transcript-level analysis.

Although both tools use statistical modelling to produce similar output, DESeq2 performs its analysis directly on RNA-seq counts, whereas Sleuth uses estimates output by kallisto's pseudoalignment. In order to account for uncertainty and variability in these estimates, kallisto generates multiple rounds of output for Sleuth using a resampling technique known as bootstrapping, where RNA-seq data is sampled randomly with replacement.

A key difference that emerges from these alternative approaches is that Sleuth is able to support transcript-level analysis, whereas DESeq2's analysis may only be performed relative to genes. A traditional differential expression analysis is more concerned with

functional gene output, making this distinction unimportant. However, Sleuth was chosen because genes can contain multiple targets, meaning the impact may be aggregated if examined at the gene level. This option of further granularity was considered useful early in the project, though ultimately not used due to the potential introduction of bias when comparing against gene-based tools.

Beyond their fundamentally different modelling procedures, both tools offer various bias correction methods, such as count normalisation, filters for the removal of low expression counts and shrinkage (Zhu et al., 2019), to reduce variability in lowly expressed genes by scaling their output.

Sleuth is prepared by supplying it with an experimental design object, a list of IDs to filter, and flags denoting that bootstraps should be used. The `gene_mode` flag and a set of gene mappings are also used to allow Sleuth to aggregate transcripts to the gene level as required. Finally, to estimate log_2 fold change, a transformation function is also applied to convert the tool's `b` output.

```
so <- sleuth_prep(
    s2c,
    ~condition,
    gene_mode = TRUE,
    aggregation_column = "ensembl_gene_id",
    target_mapping = gene_ids,
    extra_bootstrap_summary = TRUE,
    read_bootstrap_tpm = TRUE,
    filter_target_id=filtered_ids,
    transformation_function = function(x) log2(x + 0.5)
)
```

3.2.6.1 Filtering

Sleuth applies a `basic_filter` function to remove transcripts with less than 5 mapped reads in 47% of the samples. Removing poorly mapped reads improves the reliability of estimations for statistical relationships, such as mean and variance (Law et al., 2016). While filtering using a read threshold will reduce noise, it does not account for variations across samples, meaning those with more reads will proportionally have more transcripts removed. A typical solution is to instead take the mapped reads over a million bases: TargetScan uses a read filter of 10 reads per million (RPM) (Agarwal et al., 2015) and MirTarget has a floor of 5 RPM (Liu and Wang, 2019). Transcripts per

million (TPM) (Wagner et al., 2012) is another normalisation metric which accounts for read length and sequencing depth (Equations 3.2.1, 3.2.2, 3.2.3).

$$RPK = \frac{count}{lengthKB} \tag{3.2.1}$$

$$scalar = \frac{\sum_{n=1}^{samples} RPK_n}{1,000,000} \tag{3.2.2}$$

$$TPM = \frac{RPK}{scalar}. \tag{3.2.3}$$

As the goal of this research is to identify features of true targets, an effective filter can be defined as one leading to a cleaner separation between targets and non-targets. The chosen filter must also be consistent between samples to allow fair comparisons to be made. As outlined in section 2.3.2.1, the separation significance between categories can be evaluated through the combination of visualising log_2 fold change values in a cumulative plot and comparing the p-value derived from a one-sided MW test.

In Figure 3.5, a TPM filter is set at increasing thresholds of 0.5 to observe the impact of stricter filtering on the number and distribution of target candidates. As the filter becomes more stringent, the 6mer line shifts leftward and the corresponding p-value decreases on each plot, suggesting that a greater ratio of 6mers passing the filter are having a meaningful impact on expression. However, this comes at the cost of fewer overall transcripts passing the filter and, therefore, a potentially higher rate of false negatives. Depending on the transfection, a benefit is clear until 1.0 (p-value 1.4×10^{-43}) or 1.5 TPM (p-value 1.0×10^{-48}), becoming increasingly diminished after 2.0 (p-value 7.5×10^{-53}).

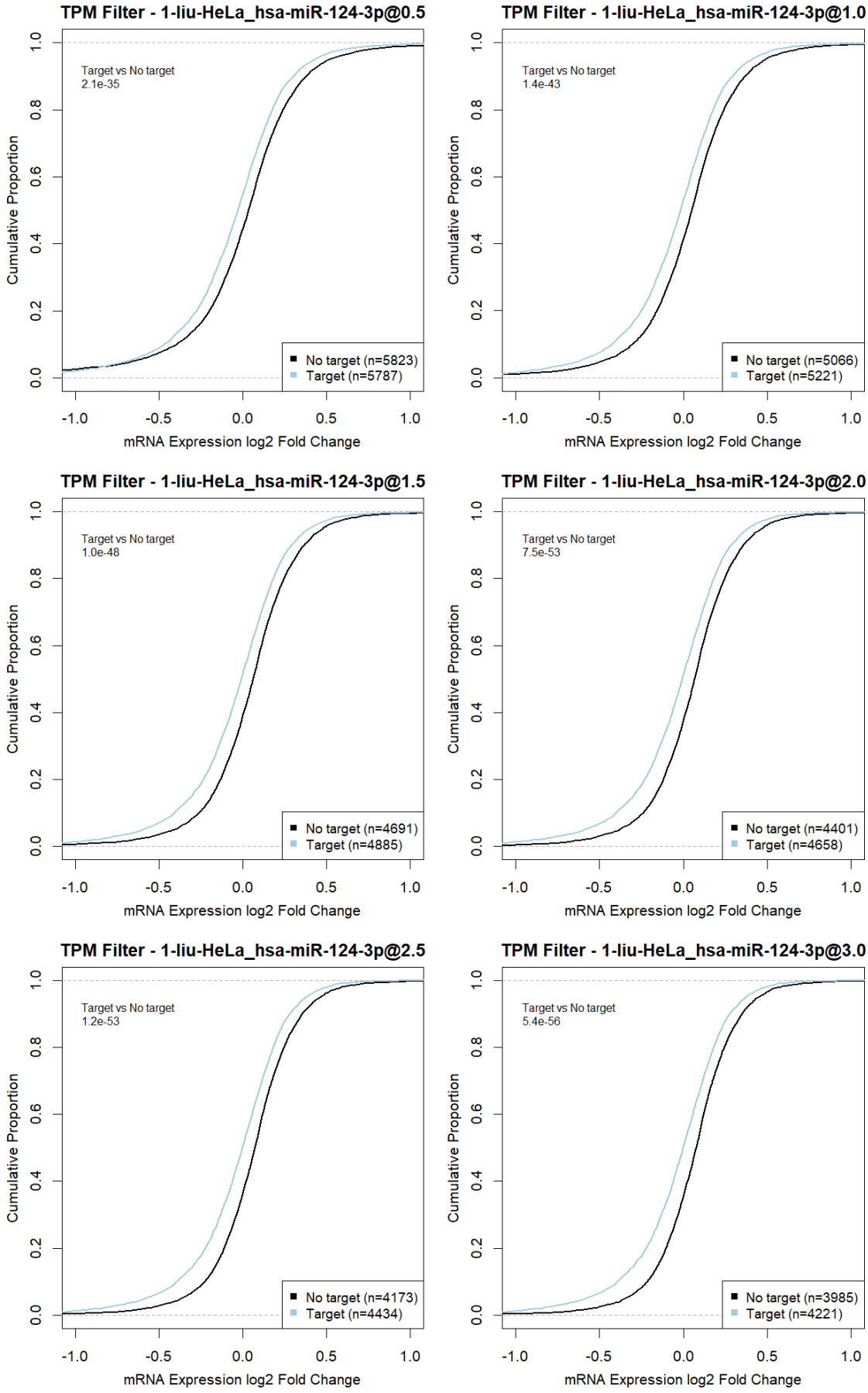


Figure 3.5: Comparison of different TPM filter thresholds on miR-124-3p. The effect of filtering low TPM reads is compared on miR-124-3p by increasing the TPM filter in increments of 0.5. The line separation between targets and non-targets increases as the TPM filter value increases. A target here refers to the presence of a 6mer, 7mer or 8mer target site.

This effect is reproducible at scale, as shown in Figure 3.6, where all transfections in the dataset are aggregated. It is difficult to assess the exact impact of the threshold at this level due to the p-value becoming inconsequential, though a minor improvement can be observed at -1.0 log_2 fold change. With this in mind, a relatively strict TPM threshold of 2.0 was selected because the dataset uses only two biological replicates, and the accuracy of data is prioritised over the transcript count. After filtering, the expression variance is reduced as a result of noise being removed (Figure 3.7).

```
tpm_filter <- function(row) {
    control_sample_tpms <- row[control_start:control_end]
    return(mean(control_sample_tpms) >= 2.0)
}
```

Beyond filtering, Sleuth also applies shrinkage to scale expression against read counts to favour reads with higher confidence. In this way, Sleuth offers additional protection against noise and outliers.

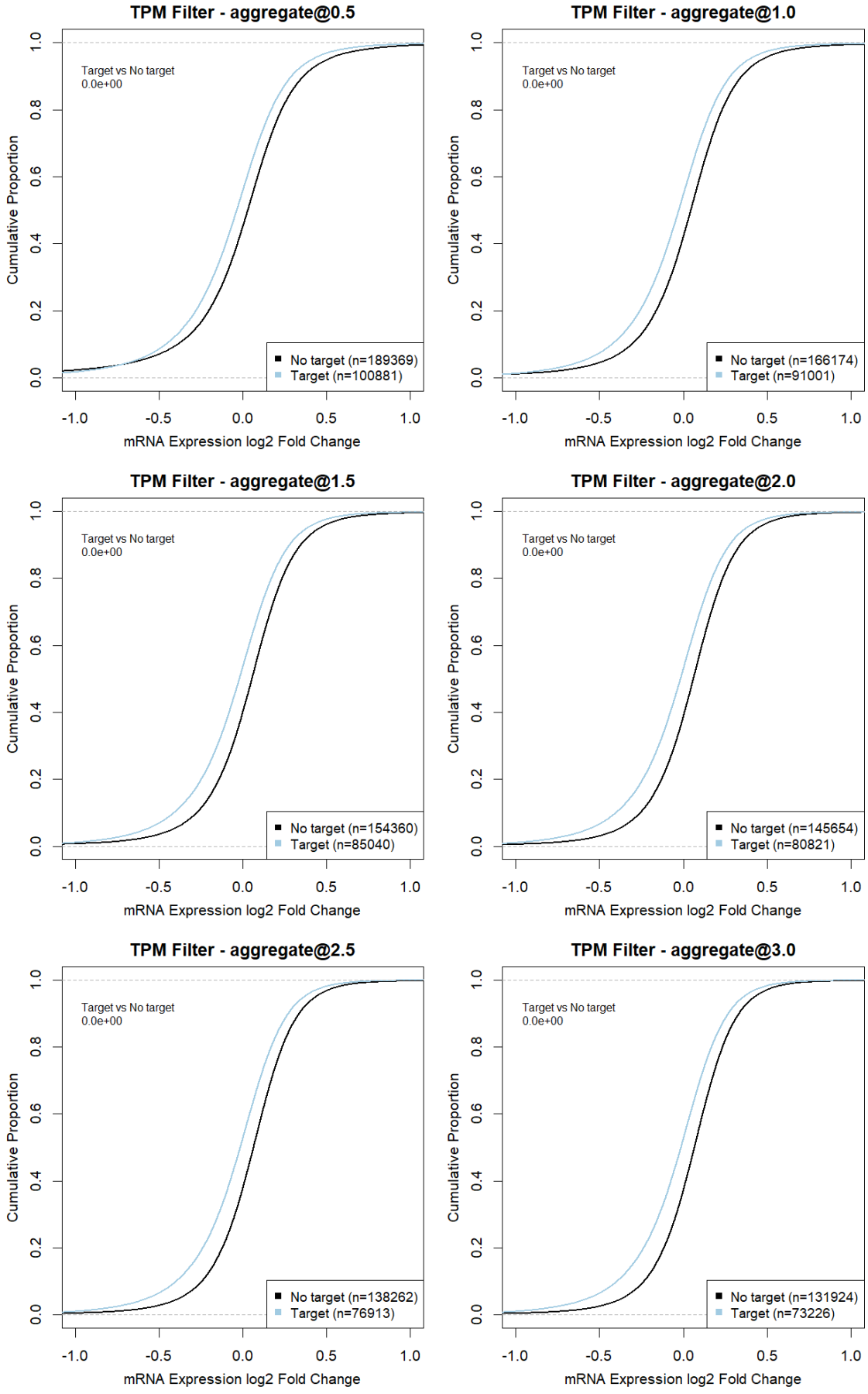


Figure 3.6: Comparison of different TPM filter thresholds across aggregated transfections. The effect of filtering low TPM reads is compared across all transfected samples by increasing the TPM filter in increments of 0.5. The line separation between targets and non-targets increases as the TPM filter value increases. A target here refers to the presence of a 6mer, 7mer or 8mer target site.

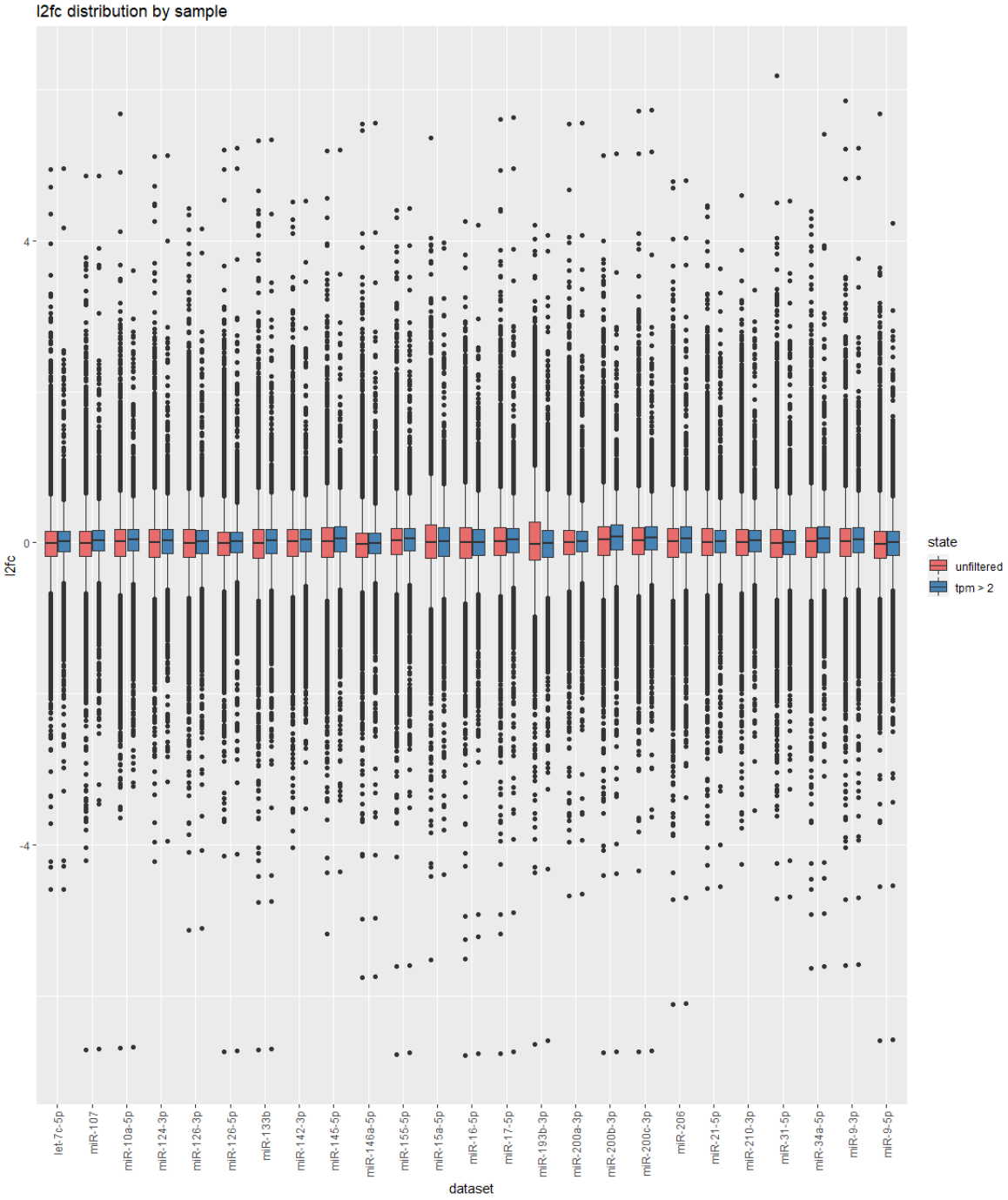


Figure 3.7: Expression fold change variance between samples in 01-liu-HeLa. The distribution of log_2 expression fold change before and after filtering between the 25 transfections of 01-liu-HeLa. The sample variance falls as a result of filtering.

3.2.6.2 Evaluating Transfection Quality

An MW test is employed on the expression fold change values of each sample to evaluate the quality of the transfection data (Table 3.3). In each case, the p-value represents the degree of significance in separation between transcripts containing at least a 6mer target site and transcript which do not. A clear separation between targets and non-targets is essential to identify target prediction rules. However, due to the large sample

size, the resulting p-values are naturally low. A strict filter is therefore applied, where any samples producing a p-value greater than 5.0×10^{-5} are discarded. In this instance, all tested transfections pass the filter.

Table 3.3: Mann-Whitney p-values for each transfection experiment

Transfection	MW p-value	Pass
hsa-let-7c-5p	2.4×10^{-96}	✓
hsa-miR-107	1.0×10^{-74}	✓
hsa-miR-10a-5p	8.1×10^{-56}	✓
hsa-miR-124-3p	3.9×10^{-102}	✓
hsa-miR-126-3p	3.6×10^{-14}	✓
hsa-miR-126-5p	7.5×10^{-108}	✓
hsa-miR-133b	4.7×10^{-105}	✓
hsa-miR-142-3p	8.1×10^{-105}	✓
hsa-miR-145-5p	7.2×10^{-147}	✓
hsa-miR-146a-5p	2.5×10^{-56}	✓
hsa-miR-155-5p	2.5×10^{-262}	✓
hsa-miR-15a-5p	3.9×10^{-82}	✓
hsa-miR-16-5p	2.2×10^{-114}	✓
hsa-miR-17-5p	4.2×10^{-47}	✓
hsa-miR-193b-3p	1.3×10^{-73}	✓
hsa-miR-200a-3p	1.2×10^{-135}	✓
hsa-miR-200b-3p	2.0×10^{-209}	✓
hsa-miR-200c-3p	9.8×10^{-224}	✓
hsa-miR-206	8.1×10^{-169}	✓
hsa-miR-21-5p	3.0×10^{-34}	✓
hsa-miR-210-3p	2.8×10^{-42}	✓
hsa-miR-31-5p	2.3×10^{-74}	✓
hsa-miR-34a-5p	6.4×10^{-129}	✓
hsa-miR-9-3p	4.7×10^{-191}	✓
hsa-miR-9-5p	8.7×10^{-62}	✓

3.2.7 RNA Folding

At this stage in the pipeline, transcripts are broadly categorised as those with a 6mer target site and those without. In practice, 6mers are relatively weak bindings compared to those containing more bases, so defining the exact binding type is important. RNA

folding tools offer insight into how strands of RNA will pair by predicting the most likely interactions between windows of bases.

The ViennaRNA package (Lorenz et al., 2011) provides a number of predictive folding tools for RNA structures. These tools use 'dot-bracket' notation to represent the relationship between given sequence strings, where '.' refers to a base with no binding and '(' or ')' represents a binding in that corresponding direction. Passing binding constraints '|' allows specific requirements to be set regarding which bases must pair (Figure 3.9). The two tools used in this chapter are RNAfold for target site accessibility calculation, and RNAcofold (Bernhart et al., 2006b) for seed and supplementary binding prediction.

3.2.7.1 Predictive Structural Accessibility

RNAfold is a secondary structure prediction tool. As part of its output, it produces an MFE stability value for the given sequence. Since a binding's MFE is a numerical representation of this stability, its inverse can be viewed as a quantification of how 'open' the bases are to external binding. As this is not a perfect relationship, being that it is built on a prediction of structure, a 'three-window' approach is used; of three windows constructed around different seed positions, the one producing the highest stability (lowest MFE) is used as the representative accessibility value.

A 30 nt window is extracted at different placements around the seed target site to produce three independent outputs (Figure 3.8). RNAfold does not require further parameters to determine secondary structure.

```
(A1)  UAUAUCAUUUUAAAAUGUCUUGGUCUUCUACUGCCUUG
(A2)  ......................(((........))).... ( -1.50)

(B1)  UGUCUUGGUCUUCUACUGCCUUGAAAAAUGACAAUUGU
(B2)  ..((..(((........)))..)).............. ( -3.40)

(C1)  CUGCCUUGAAAAAUGACAAUUGUGAACAUGAUAGUUAA
(C2)  ............((((.(((((....)))))).)))). ( -1.80)
```

Figure 3.8: Overview of accessibility computation using a three-window approach. The accessibility scores of three windows around a seed target are computed and compared. (A1, B1, C1) The three sequences to fold, where red bases denote a 6mer and blue a 7mer. All three are derived from the same mRNA, but the seed anchor point is placed in different positions. (A2, B2, C2) The RNAfold predicted structure output of the sequences. In parenthesis is the computed window's MFE, where lower is stronger. Sequence B2 has the most stable structure and would therefore be treated as the representative accessibility value by the three-window approach.

3.2.7.2 Predictive Base Pairing

RNAcofold is a tool for predicting the binding structure of two complementary RNA sequences. As part of its output, RNAcofold also computes the MFE of its prediction, where a lower value indicates a greater level of binding stability. Figure 3.9 illustrates how a constraint can be passed to RNAcofold to ensure the binding contains a 6mer. It also highlights the importance of RNAcofold when imperfections, such as gaps, are present outside the seed; a bulge would not be detectable by a simple pattern match of complementary sequences.

RNAcofold requires complementary sequences from both the miRNA and mRNA, in addition to a constraint instructing which bases must pair. In extracting the mRNA portion, the process is identical to A1 in Figure 3.8. For the miRNA, the entire sequence is used because miRNAs typically only measure around 22 nt. The constraint is generated by parsing the two complementary sequences and placing '|' 'must pair' symbols in line with the 6mer bases of the miRNA. No constraint is used outside the 6mer to allow RNAcofold to determine the most likely way that the remainder of the two sequences will bind. Beyond allowing RNAcofold to predict the type of seed binding, this method is effective in predicting any potential supplementary base pairing that my occur.

```
(1) UGAGGUAGUAGGUUGUAUGGUU&UCAUUUUGAAGUUGGAAUUGGUCUUCUGCCUACCUCU
(2) .||||||..............&............................||||||.
(3) .|6mer|..............&............................|6mer|.
(4) .((((((((((.........&.....................)))).)))))))).
```

Figure 3.9: Fold prediction using RNAcofold with a 6mer constraint. An example input and output using RNAcofold with a constraint. (1) An miRNA and mRNA sequence, respectively, separated by '&'. (2) Dot-bracket constraints to ensure a perfect 6mer match at minimum between the two sequences. (3) A visual representation of the dot-bracket constraints from (2). (4) The resulting RNAcofold output in dot-bracket notation. A bulge is predicted at the 9th base from the right side.

3.2.7.3 Target Window Extraction

Both tools require a window of bases beyond the six from the 6mer target, as local context is a factor in RNA folding. As previously discussed, there is often a one-to-many relationship between transcripts and target sites, meaning this process of extracting windows must be performed at the target level, instead of the transcript level. As a result, each target match must be processed independently. A transcript containing

n target sites therefore has n windows extracted. At 2 TPM, there are 145,654 non-targets and 80,821 targets (Figure 3.7). As a result of MTS handling, the number of target sites identified increases from 80,821 to 150,197.

Algorithm 1 Target window extraction algorithm

Input: miRNA sequence, mRNA transcripts + 3′ UTRs

```
extract 6mer site from sequence
reverse complement 6mer site for 6mer target
for all transcripts do
    search current transcript 3′ UTR for target
    for all target matches (MTS) do
        RNAfold: expand target match to form three RNAfold windows
        RNAcofold: take one RNAfold window for RNAcofold's mRNA half
        RNAcofold: Combine miRNA sequence with RNAcofold window
        RNAcofold: parse RNAcofold window for dot-bracket constraint
    end for
end for
```

Output: windows and dot-bracket parses

3.2.7.4 Supplementary Site Definition

There is no single definition for exactly which combination of bases form supplementary binding, though bases 13-16 relative to the miRNA are known to be particularly important (Section 2.4.2). A limitation of rule-based target prediction is in the difficulty of handling edge cases without significantly complicating the algorithm. This means that an effective supplementary site definition in the context of rule-based prediction should provide value without the need for conditional logic.

Using RNAcofold output, bp are counted within a set of categories formed by taking bases 13-16 and expanding outward by 1 nt at a time. In Figure 3.10, these results are aggregated across all transfections.

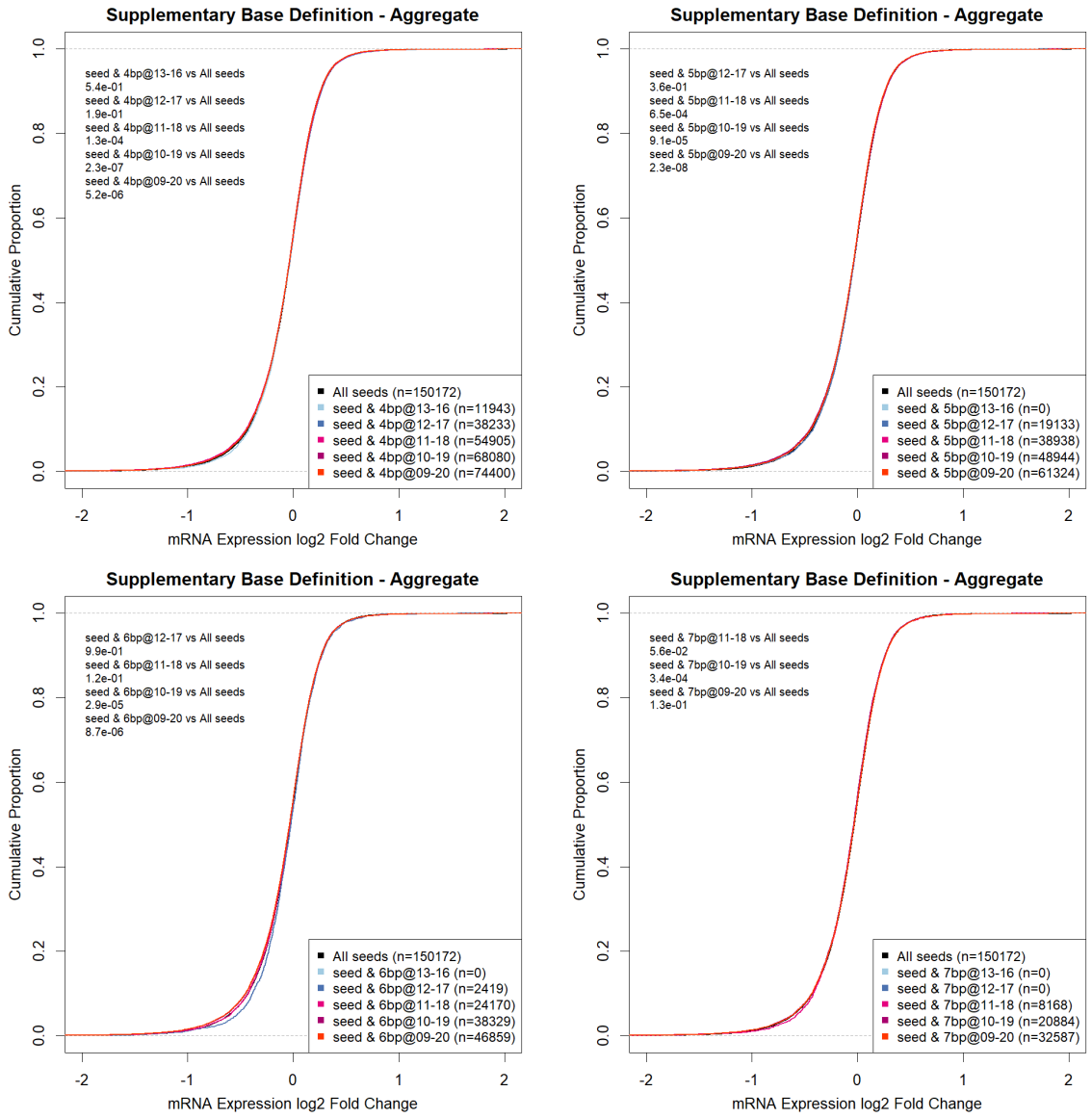


Figure 3.10: Comparison of supplementary definitions across aggregated transfections. The effect of different supplementary window definitions is compared across an aggregation of the 25 transfection experiments. The categories are created by starting with bases 13-16, then expanding out an additional base on both sides each time. (Top left) Minimum 4bp. (Top right) Minimum 5bp. (Bottom left) Minimum 6bp. (Bottom right) Minimum 7bp. Category bounds are inclusive, for example, 13-16 includes both bases 13 and 16.

In all categories, there is limited visual separation, though the majority of shifts are significant against the seed lines. This is expected, as effective supplementary pairing is dependent on factors not considered here, such as seed binding stability. Nonetheless, a trend can be seen where longer sequences with bounds not allowing for gaps tend to lead to shifts in the opposite direction (12-17 @ 6bp, 11-18 @ 7bp). This is consistent with the understanding that supplementary binding sequences are not as harshly affected by gaps as seed binding (Kiriakidou et al., 2004), while also suggesting that longer continuous sequences may be detrimental. The strongest definition is '09-20 @ 5bp'

(2.3×10^{-8}), which encompasses the first 11 nt of the 5′ flanking region of the mRNA from the seed target.

3.2.8 Conservation Scoring

Conservation scores are generated in R using GenomicScores (Puigdevall and Castelo, 2018) in combination with the phastCons100way.UCSC.hg38 track package (Siepel et al., 2005). GenomicScores allows a given track to be accessed at a per-base level via genomic coordinates. In this instance, the UCSC track package scores the similarity ratio of each base over 100 vertebrate sequences aligned in parallel.

For seed sites, the start coordinate is taken as the complementary first base of the miRNA and the end coordinate is the conditional 8th base. The seed target coordinates are relative to the 3′ UTR; to calculate their position in the wider genomic context, they are combined with the 3′ UTR start position.

```
start <- x3_utr_start + seed_start
end <- start + 8
```

For supplementary base coordinates, the extracted region uses the definition decided in Section 3.2.7.4, beginning at base 9 and ending at base 20 inclusive.

```
start <- x3_utr_start + seed_start + 9
end <- start + 12
```

In either case, these coordinates are passed to GenomicScores in the same way; the output received is the proportional representation of how conserved the given base sequence is across 100 vertebrates.

```
range <- GRanges(chromosome, IRanges(start:end), strand)
score <- gscores(track_v100, range)$default
```

There are a number of issues with this iteration of the conservation logic, namely that it is several magnitudes slower than all other aspects of the feature processing. As the output is dependent on transcripts and base coordinates, changes to filters or contingent elements also require the result to be fully regenerated, making caching difficult. It is ultimately overhauled in later work (Section 5.2.2.5), though the result between the two is unchanged.

3.3 Results

Each feature is first tested in isolation and then applied in a rule based prediction model. A p-value threshold of 0.05 is used to determine significance from a two-sample MW test.

3.3.1 Isolated Features

3.3.1.1 Seed Type

Seed types (Section 2.4.1) are identified by providing RNAcofold with complementary miRNA:mRNA sequences and a 6mer constraint (Section 3.2.7.2).

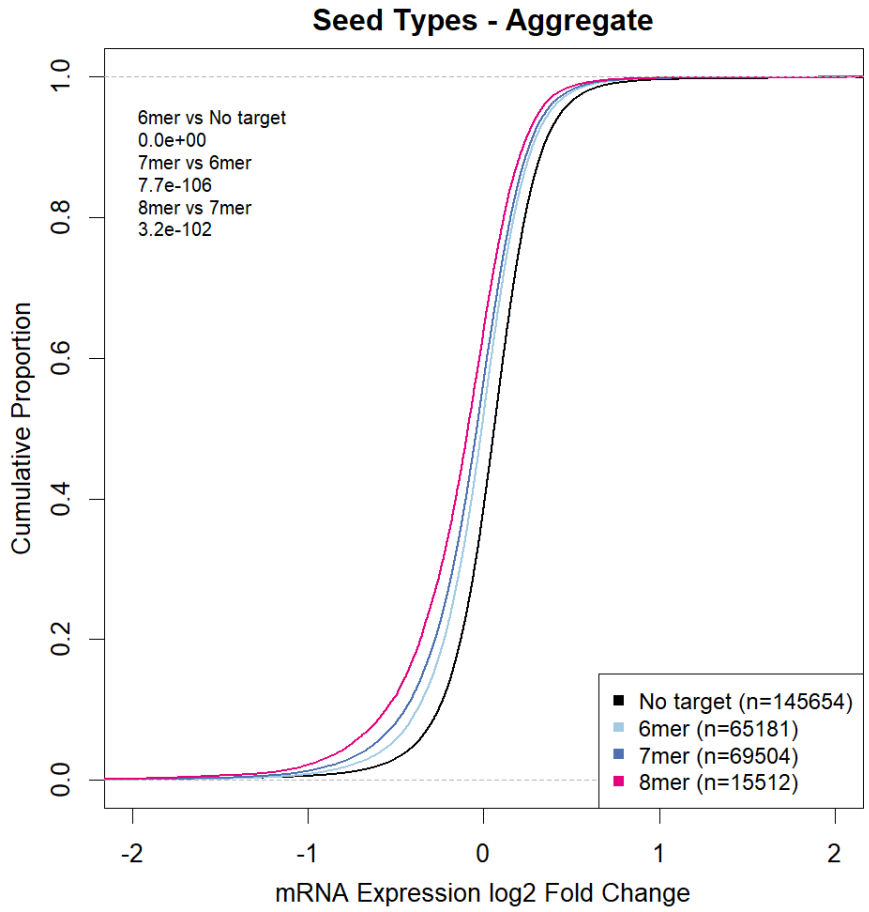


Figure 3.11: Comparison of aggregate seed type efficacy. Seed types are compared by aggregating the 25 transfection experiments.

The leftward shift in each line indicates that 8mer seed bindings have greater efficacy than 7mers, which are in turn more effective than 6mers. All seeds are significant in distinguishing targets from non-targets (p-values: 6mer rounded to 0, 7mer 7.7×10^{-106} and 8mer 3.2×10^{-102}), while also being visually removed from each other. As seed

sites are arguably the most important feature of a binding and the basis of many prediction algorithms, the low p-values of this result suggests the implementation of RNAcofold for categorising seed binding types is effective. This is further supported by the consistency of this feature, as the 8mer > 7mer > 6mer relationship is the same across all individual samples.

3.3.1.2 Multiple Seed Target Sites

MTS, also known as target site abundance (Section 2.4.7), is supported by performing RNA folding at the target level as opposed to the transcript level (Algorithm 1). The process is otherwise identical to seed-type identification.

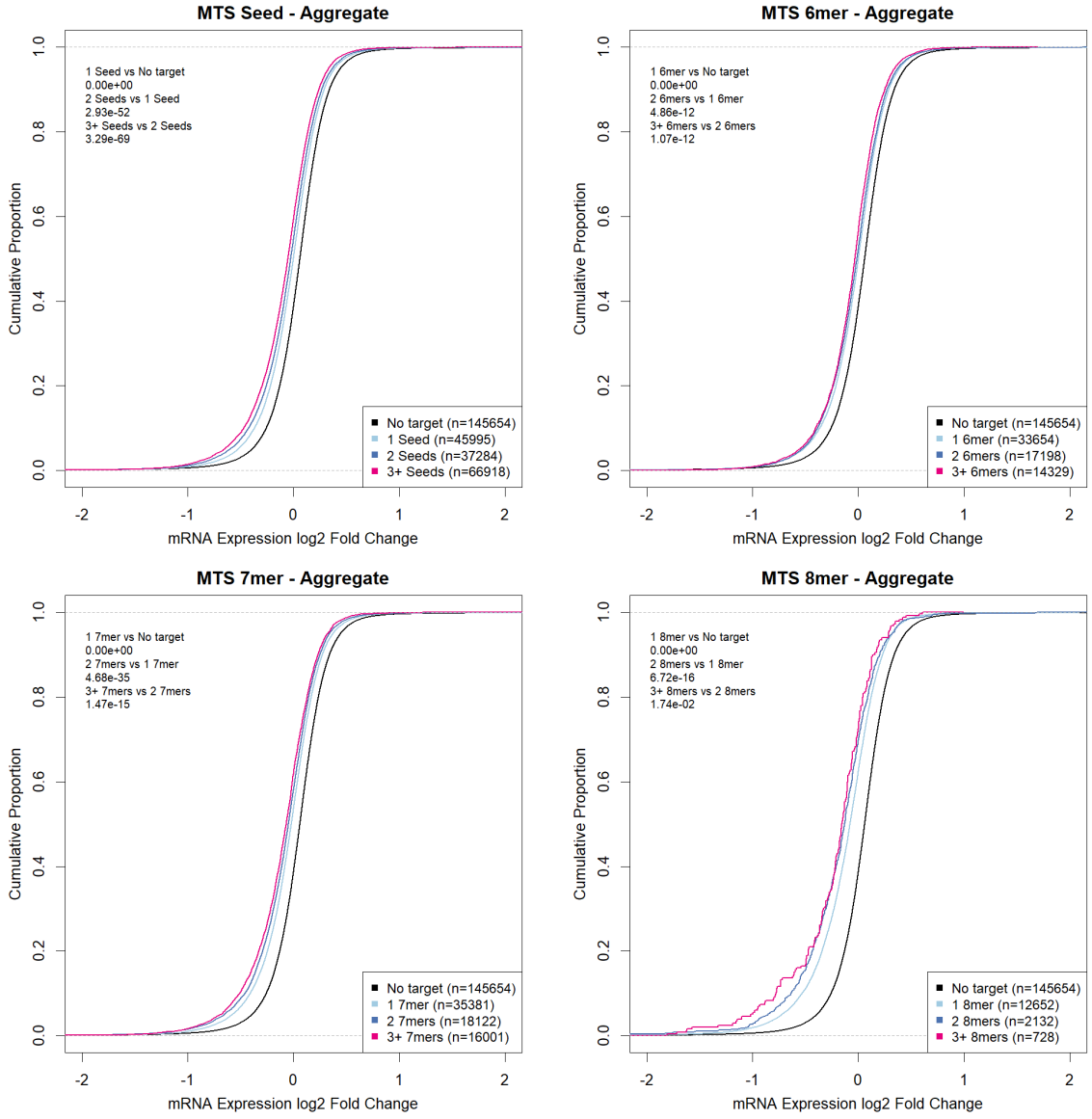


Figure 3.12: Comparison of aggregate MTS efficacy. MTS is compared by aggregating the 25 transfection experiments. (Top left) All seed type overview. (Top right) 6mer only. (Bottom left) 7mer only. (Bottom right) 8mer only.

A correlation can be seen between MTS and binding efficacy, as each successive target site leads to a greater leftward shift in all seed types. The effect is particularly pronounced in the 8mer category, where the 3+ line between -0.5 and -2 has an impact beyond that of any other isolated feature. Despite the downregulatory strength of multiple 8mer targets in the same 3′ UTR, the relative rarity of 8mers somewhat lessens its effectiveness as a single feature. The impact of this feature is also noticeably weaker for 6mers, as the separation between lines is minor even at 3+ target sites.

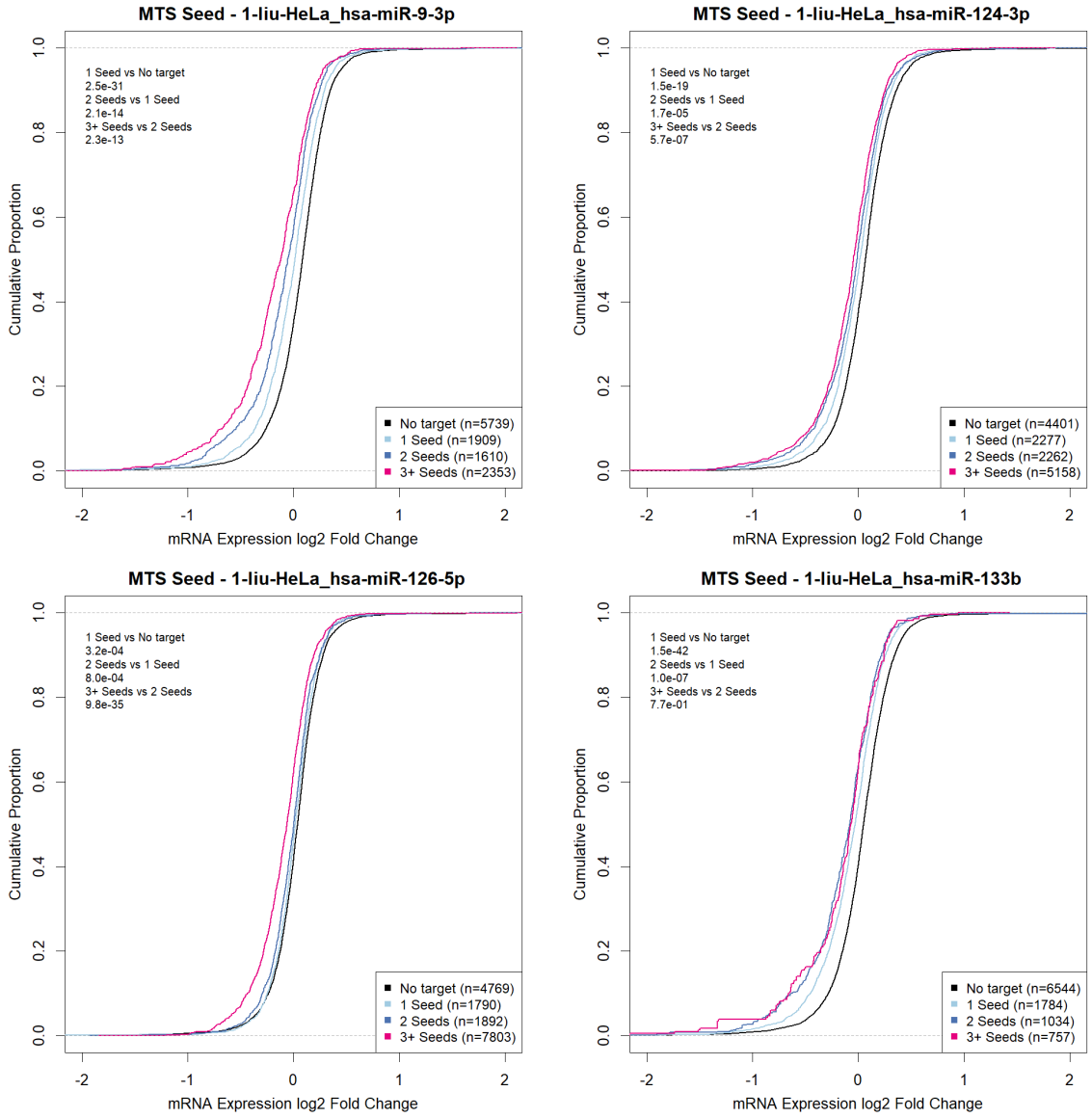


Figure 3.13: Comparison of MTS efficacy on four individual transfections. A subset of four MTS comparisons using all seed types to demonstrate result consistency. (Top left) miR-9-3p transfection: a strong result. (Top right) miR-124-3p transfection: limited separation between 2 and 3 seeds. (Bottom left) miR-126-5p transfection: limited separation between 1 and 2 seeds compared to 3. (Bottom right) miR-133b transfection: an outlier result for the 3+ line, likely due to limited occurrences.

There is a degree of inconsistency in individual experiments, stemming from the rarity

of 3+ bindings in specific seed-type breakdowns. In the combined seed category of individual transfections, this variation only causes the 3+ category to shift beyond another in transfections with low overall transcript counts.

3.3.1.3 Binding Stability

Binding stability (Section 2.4.1), represented by MFE, is computed as part of the output during complementary fold prediction by RNAcofold (Section 3.2.7.2).

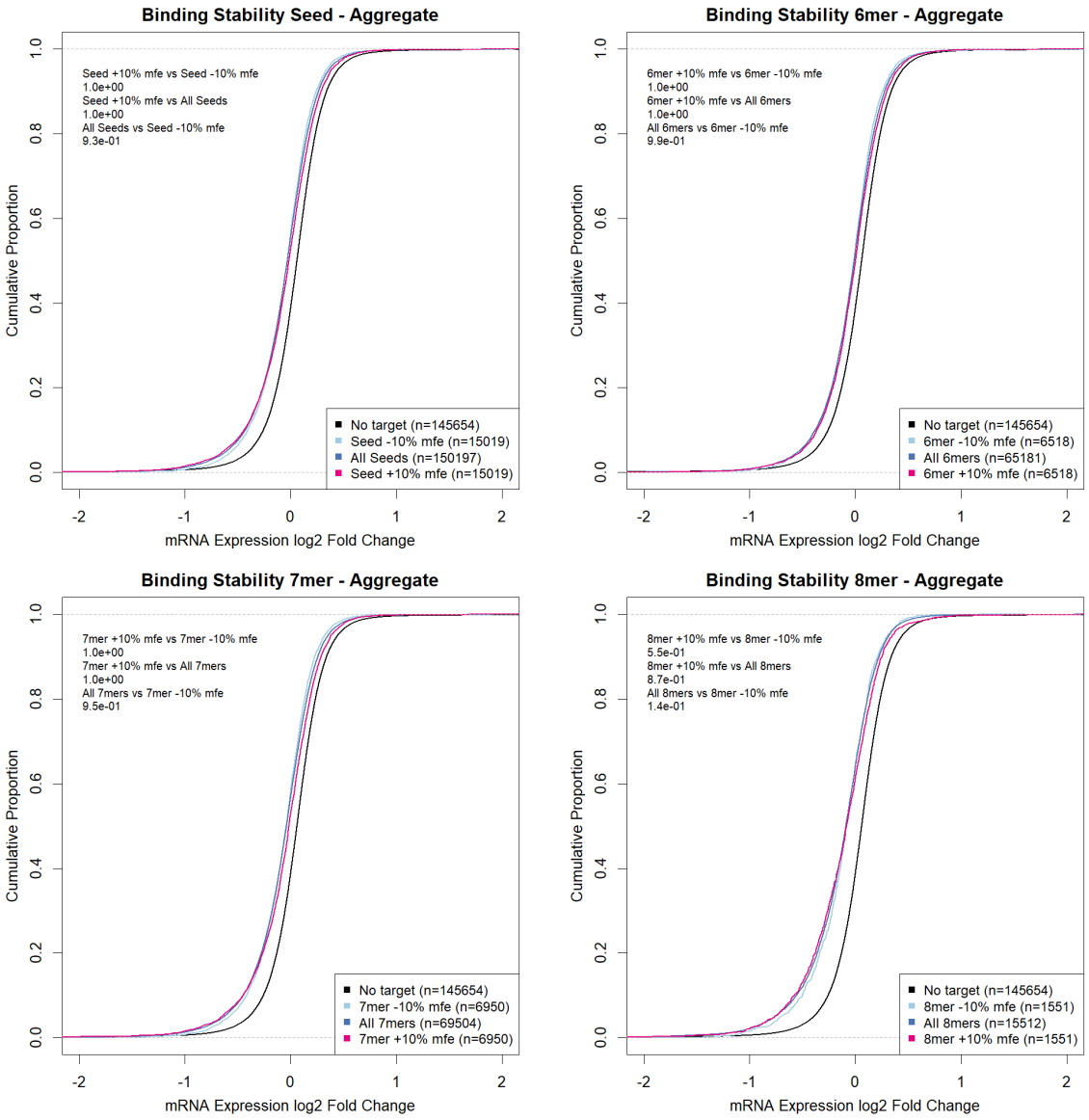


Figure 3.14: Comparison of aggregate binding stability efficacy. Binding stability is compared by aggregating the 25 transfection experiments. (Top left) All seed type overview. (Top right) 6mer only. (Bottom left) 7mer only. (Bottom right) 8mer only.

Binding stability is a generally ineffective feature in isolation, though it varies by the type of seed. When comparing the top 10% with the bottom 10%, there is significance only in the 8mer category. Although 6mer bindings are less stable on average, its

lower quartile is approximately the mean of an 8mer (Figure 3.15). Additionally, the interquartile range of a 7mer falls close to that of a general seed. This may explain why 6mers and 7mers are unable to attain significant separation by stability alone.

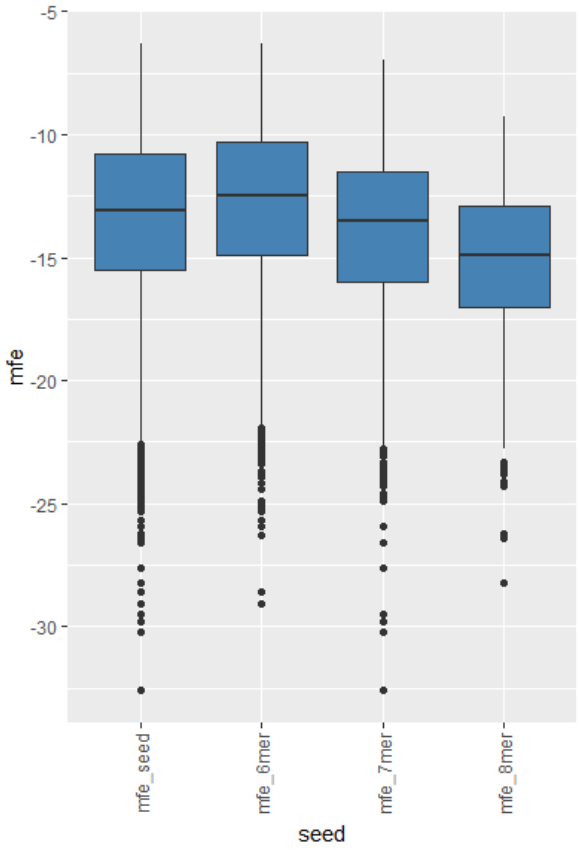


Figure 3.15: MFE distribution by seed type. (Left to right) The MFE of all seed types, 6mers only, 7mers only and 8mers only. Note that MFE is an inverse representation of stability, and lower values therefore indicate stronger bindings.

In all types of seed, low stability is more effective at distinguishing binding quality than high stability, as evidenced by the lower p-values of 'All Seeds vs Seed -10%' (9.3×10^{-1}) compared to 'Seed +10% vs Seed -10%' (rounded to 1.0) (Figure 3.14). There are no binding stability thresholds with a significant deviation from all seeds; therefore this feature is limited without context. Despite this, it should be noted that binding stability is shown to be marginally more effective at distinguishing non-targets than targets.

Binding stability is difficult to assess outside aggregate results as averages vary between samples, and seed targets are comprised of bases which confer different levels of stability.

3.3.1.4 Target Site Accessibility

Target site accessibility (Section 2.4.3) is quantified using the MFE output of the representative of three RNAfold windows (Section 3.2.7.1).

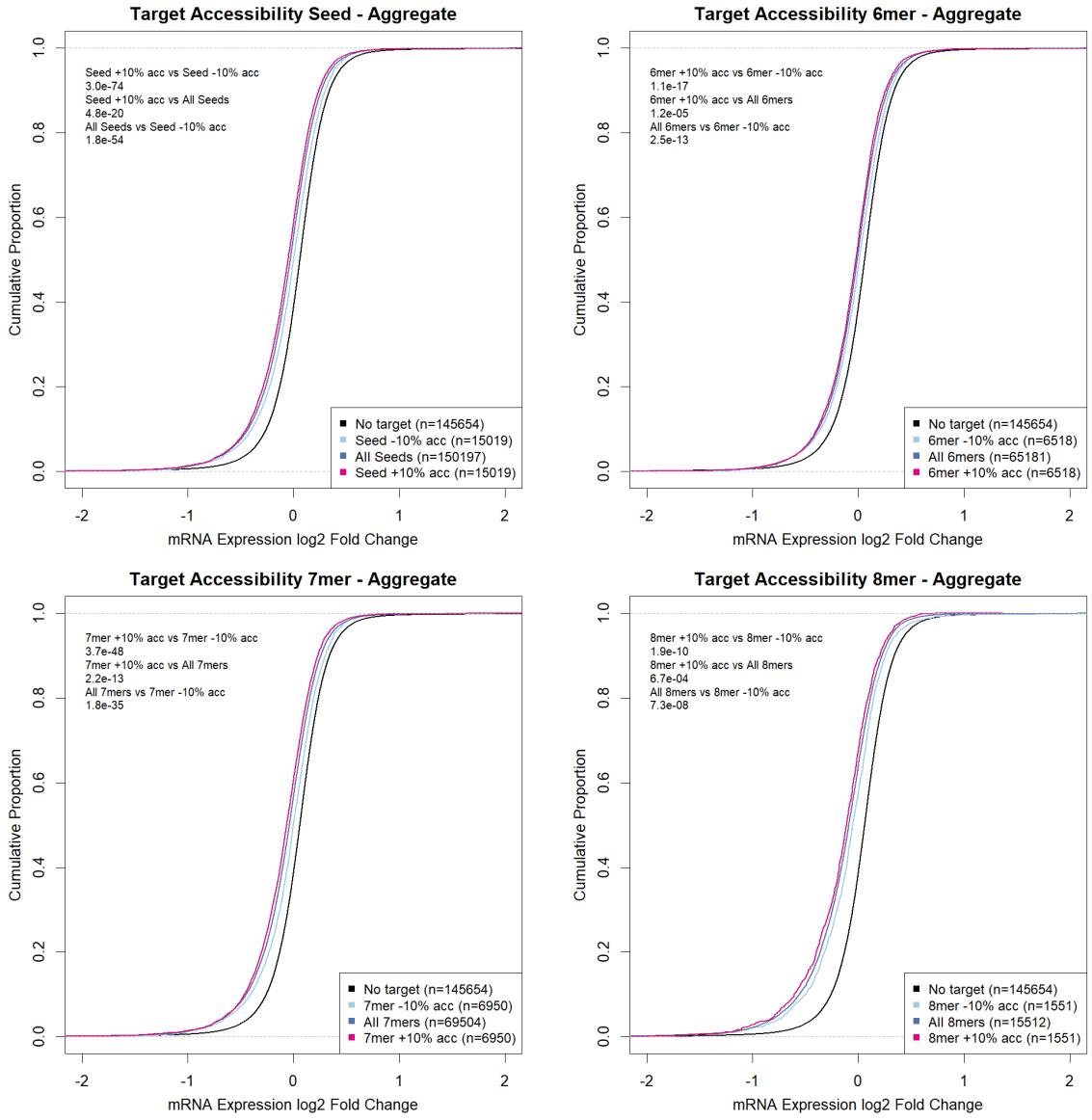


Figure 3.16: Comparison of aggregate target site accessibility efficacy. Target site accessibility is compared by aggregating the 25 transfection experiments. (Top left) All seed type overview. (Top right) 6mer only. (Bottom left) 7mer only. (Bottom right) 8mer only.

The result highlights a significant correlation between accessibility and binding efficacy at all thresholds ($p-$value 3.0×10^{-74}), with low accessibility ($p-$value 1.8×10^{-54}) being a slightly more impactful measure than high accessibility ($p-$value 4.8×10^{-20}).

An important element of this result is the variation between different seed categories ($p-$values: 6mer 1.1×10^{-17}, 7mer 3.7×10^{-48} and 8mer 1.9×10^{-10}). 6mers gain limited benefit from accessibility compared to 7mers and, while 8mers also have a low

p-value, the visual line separation indicates this could be due to the lower population size of the category. This may be an inherent bias of the methods used in this study, as a 6mer constraint is used with RNAcofold, potentially filtering a percentage of 6mers that would be deemed unlikely to pair due to inaccessible bases beforehand. This shortcoming would not affect 7mers, as the additional base is not part of the constraint.

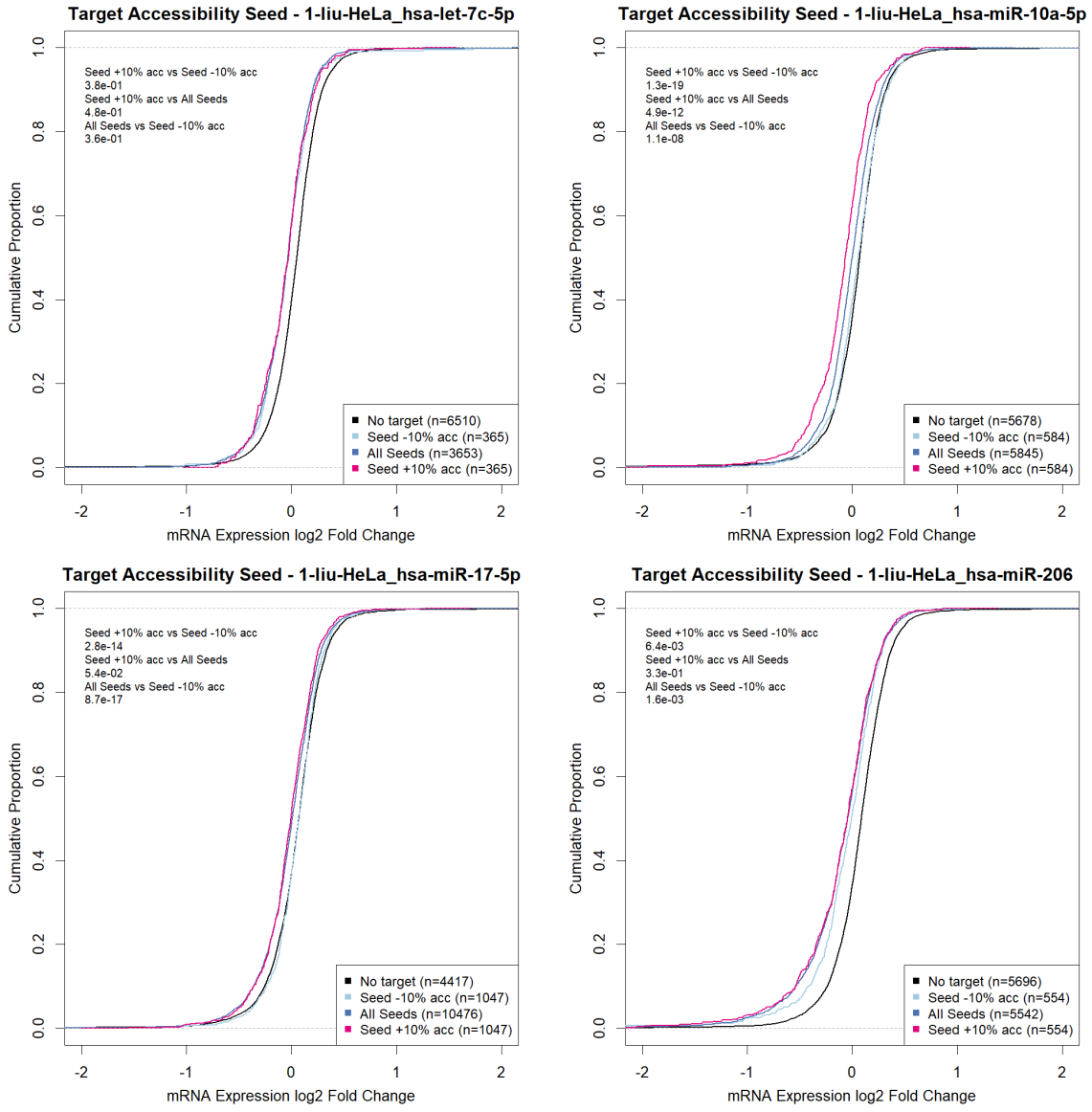


Figure 3.17: Comparison of target site accessibility efficacy on four individual transfections. A subset of four target site accessibility comparisons using all seed types to demonstrate result consistency. (Top left) miR-let-7c transfection: a poor result with limited separation. (Top right) miR-10a-5p transfection: a strong result, with both accessibility and inaccessibility leading to separation. (Bottom left) miR-17-5p transfection: separation only in the least accessible bases; however the shift is strong and almost as effective at determining a non-target as no seed target site. (Bottom right) miR-206 transfection: separation only in the least accessible bases.

There is a large variation in the results of individual experiments, although one of two categories always separates from 'All Seeds'. The overall seed result is generally

significant at a threshold of 0.05 for 'Seed +10% vs Seed -10%' in individual samples (Figure 3.17). However exceptions exist in samples with relatively few seed targets, for example let-7c-5p (top-left, p-value 3.8×10^{-1}).

3.3.1.5 Supplementary Binding

Supplementary binding (Section 2.4.2) is determined by using RNA folding to identify likely pairings between two complementary sequences and counting the total bp between a fixed base threshold (Section 3.2.7.4).

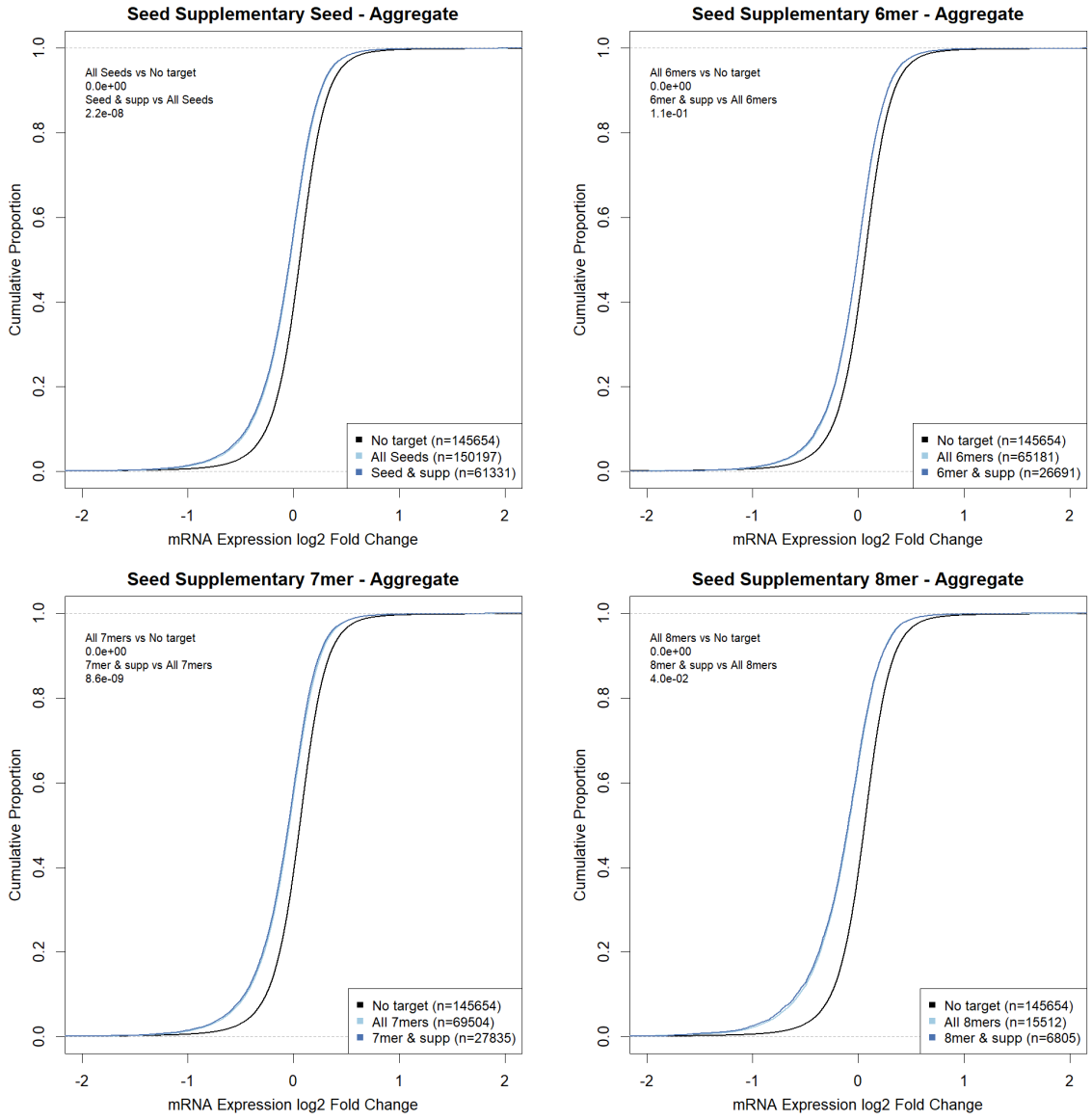


Figure 3.18: Comparison of aggregate supplementary binding efficacy. Supplementary binding is compared by aggregating the 25 transfection experiments. (Top left) All seed type overview. (Top right) 6mer only. (Bottom left) 7mer only. (Bottom right) 8mer only.

Supplementary binding in isolation provides negligible gains to targeting efficacy despite inflicting a steep cost to the transcript count (150,197 to 61,331). The p-values

indicate significance in the 7mer and 8mer categories, but not 6mer (p-values: 6mer 1.1×10^{-1}, 7mer 8.6×10^{-9} and 8mer 4.0×10^{-2}). A 7mer benefits the most, with 6mer bindings having a trivial influence on the overall result. This is perhaps contrary to expectations, as 6mers typically have a lower binding stability than 7mers (Figure 3.15), making them more likely candidates for support.

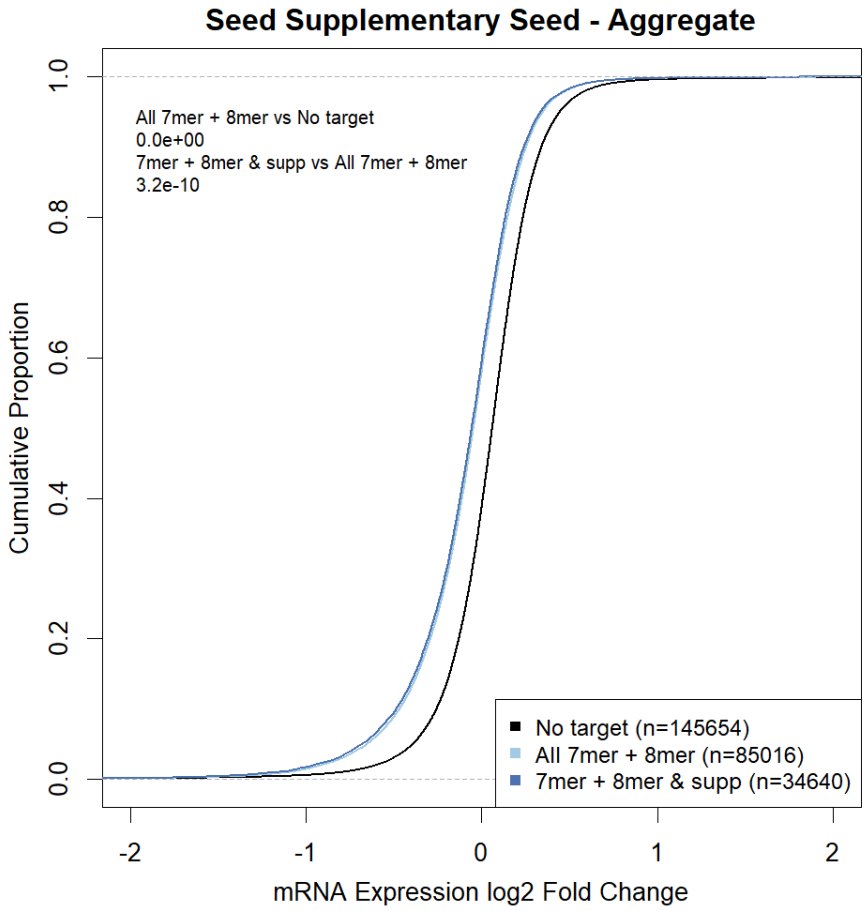


Figure 3.19: Comparison of aggregate supplementary binding efficacy on 7mers and 8mers. Supplementary binding is compared by aggregating the 25 transfection experiments. Only 7mers and 8mers are represented, as 6mers are removed.

After removing the statistically insignificant 6mer category from the overall result, supplementary binding produces a p-value of 3.2×10^{-10}. This makes it a more effective distinction than both the 7mer (8.6×10^{-9}) and 8mer (4.0×10^{-2}) supplementary categories, and the overall seed supplementary category (2.2×10^{-8}). There is a substantial reduction in the number of transcripts as a result, from 61,331 to 34,640.

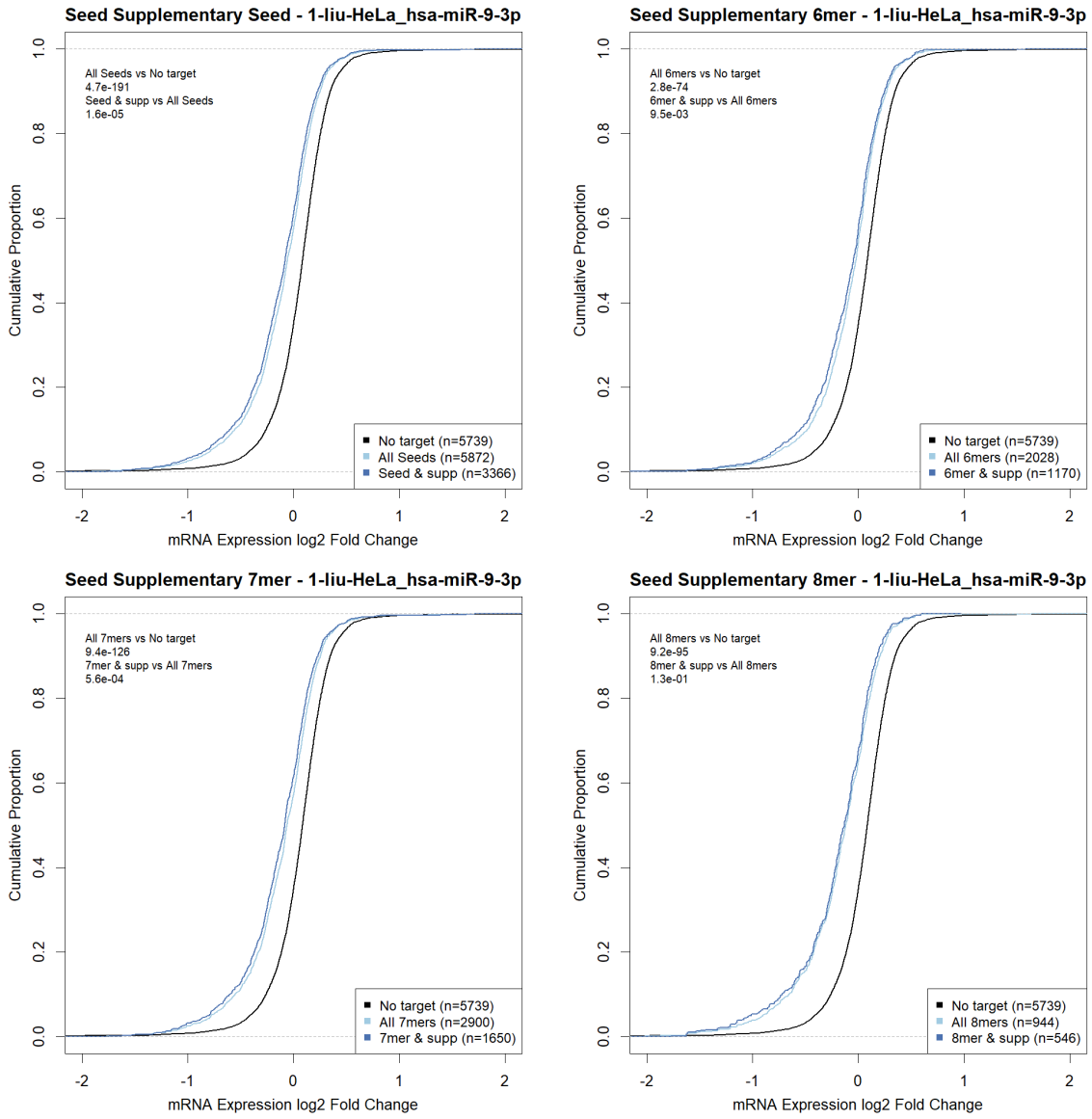


Figure 3.20: Comparison of supplementary binding efficacy on miR-9-3p. Supplementary binding is compared on miR-9-3p to demonstrate a positive individual result. (Top left) All seed type overview. (Top right) 6mer only. (Bottom left) 7mer only. (Bottom right) 8mer only.

Supplementary binding for 8mers is insignificant in all individual transfections, whereas it is only insignificant for 7mers in 18 of 25. Despite this, there are transfections where all three seed types are influenced by the presence of supplementary binding (Figure 3.20).

3.3.1.6 Evolutionary Conservation

Evolutionary conservation (Section 2.4.6) uses a track of 100 aligned vertebrate species sequences to determine the proportional representation of each base (Section 3.2.8). The overall score is calculated by taking the mean of a sequence's per-base proportional representation.

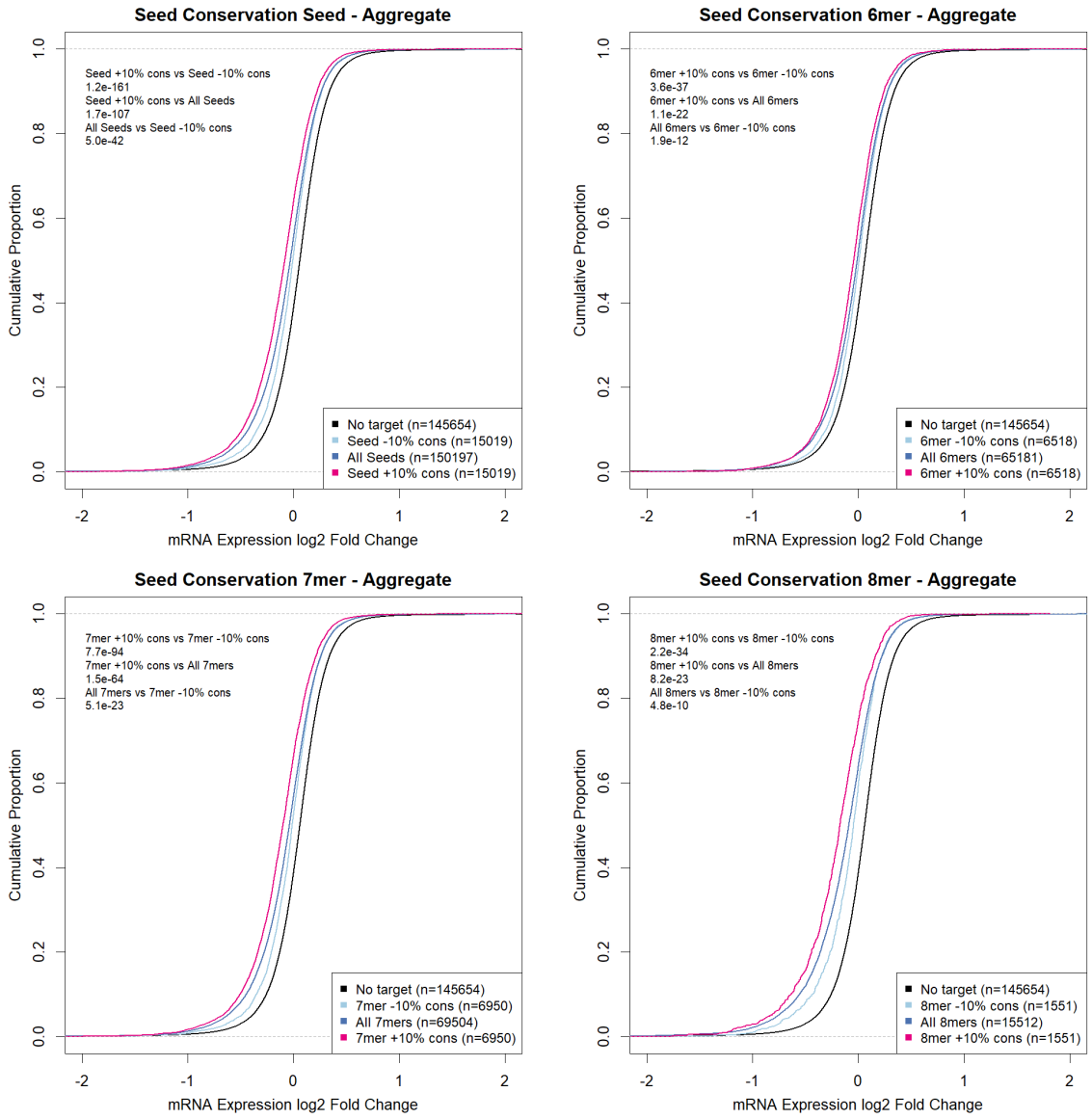


Figure 3.21: Comparison of aggregate seed conservation efficacy. Seed conservation is compared by aggregating the 25 transfection experiments. (Top left) All seed type overview. (Top right) 6mer only. (Bottom left) 7mer only. (Bottom right) 8mer only.

The efficacy gain of highly conserved target sites is significant across all seed types (p-values: 6mer 1.1×10^{-22}, 7mer 1.5×10^{-64} and 8mer 8.3×10^{-23}), as is the efficacy reduction in poorly conserved target sites (p-values: 6mer 1.9×10^{-12}, 7mer 5.1×10^{-23} and 8mer 4.8×10^{-10}). Results also show that high conservation is a significantly more effective method of dissection than low conservation; high conservation produces a p-value of 1.2×10^{-161} against all seeds, compared to 5.0×10^{-42} for low conservation.

Although there is some variation in individual transfections regarding the 8mer categories, the overall strength of the feature means this is only the case in the most sparsely populated of samples.

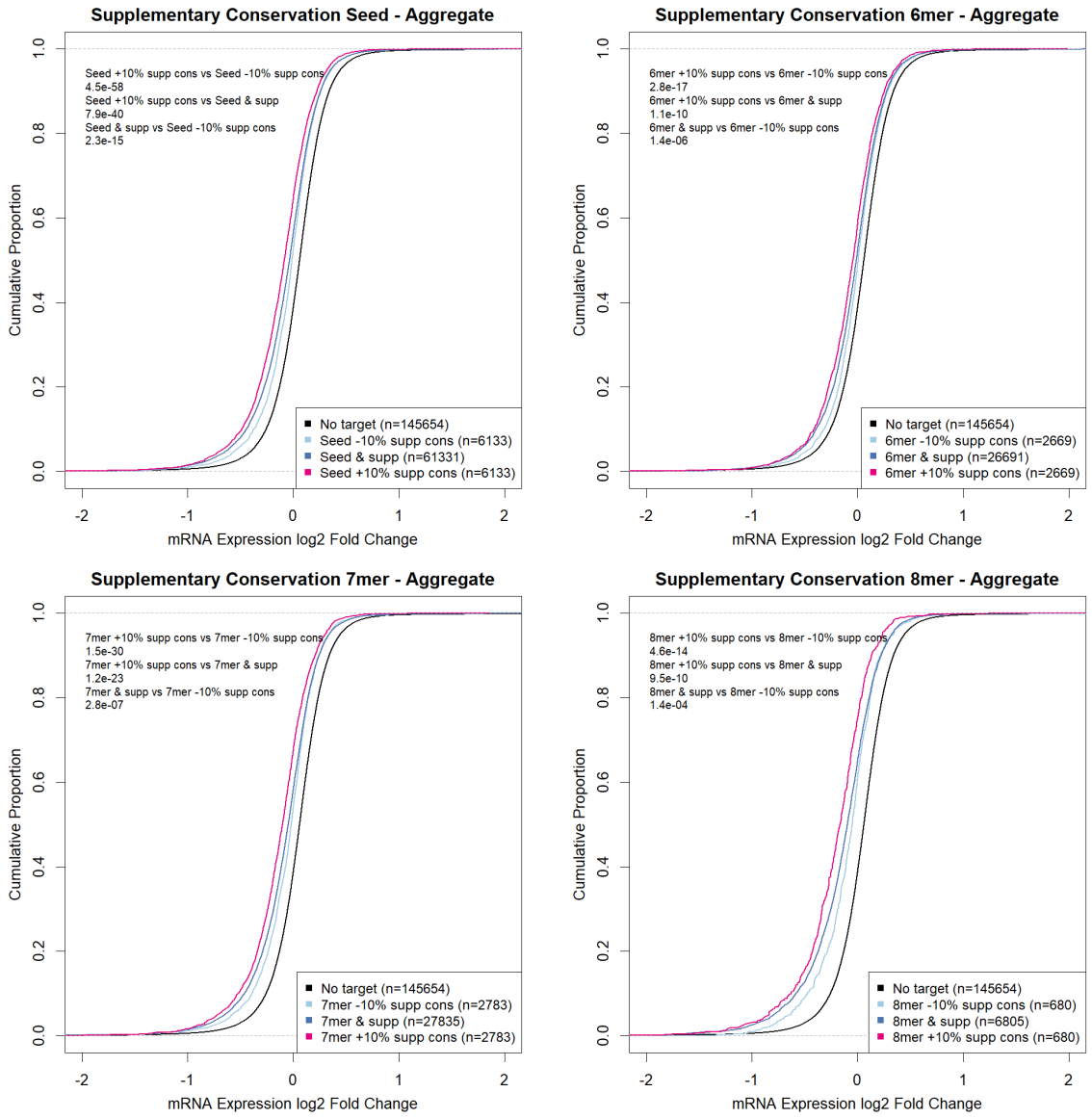


Figure 3.22: Comparison of aggregate supplementary conservation efficacy. Supplementary conservation is compared by aggregating the 25 transfection experiments. Seed conservation is not represented, despite the seed type being used in chart breakdowns. (Top left) All seed type overview. (Top right) 6mer only. (Bottom left) 7mer only. (Bottom right) 8mer only.

Similar line shifts are observed when conservation scoring is applied specifically to the supplementary portion of the binding, where significant p-values are produced in all seed types (p-values: 6mer 2.8×10^{-17}, 7mer 1.5×10^{-30} and 8mer 4.6×10^{-14}). Despite this, the comparatively low frequency of transcripts meeting the thresholds is likely to reduce the utility of the feature. Only 61,331 of the total 150,197 seed targets contain a minimum 5 nt supplementary binding, the bulk of which consists of 6mers and 7mers. This means that a looser threshold for supplementary conservation compared to seed conservation may be desirable.

As with other isolated features that have lower transcript counts, examining individual

samples is less informative because a strong pattern does not emerge until the results are aggregated.

3.3.2 Rule-based Prediction

Three feature rule sets are constructed to output predictions at different stringency levels. As these thresholds become stricter, a trade-off emerges between accuracy and the overall number of predictions made. The objectives are as follows:

- **Rule set 1:** produce a similar level of prediction accuracy to 8mers, while increasing the number of detected targets.
- **Rule set 2:** strike a balance between prediction accuracy and total predictions.
- **Rule set 3:** prioritise prediction accuracy at the cost of producing less overall predictions.

With the exception of supplementary base pairing, all features discussed in the isolated feature testing are used in some capacity. Supplementary site information is instead encoded by means of supplementary conservation scores, which is applied only in rule set 3 due to its substantial reduction to the number of transcripts. The prediction results are plotted against 8mers, a strong and basic feature that is comparatively easy to implement.

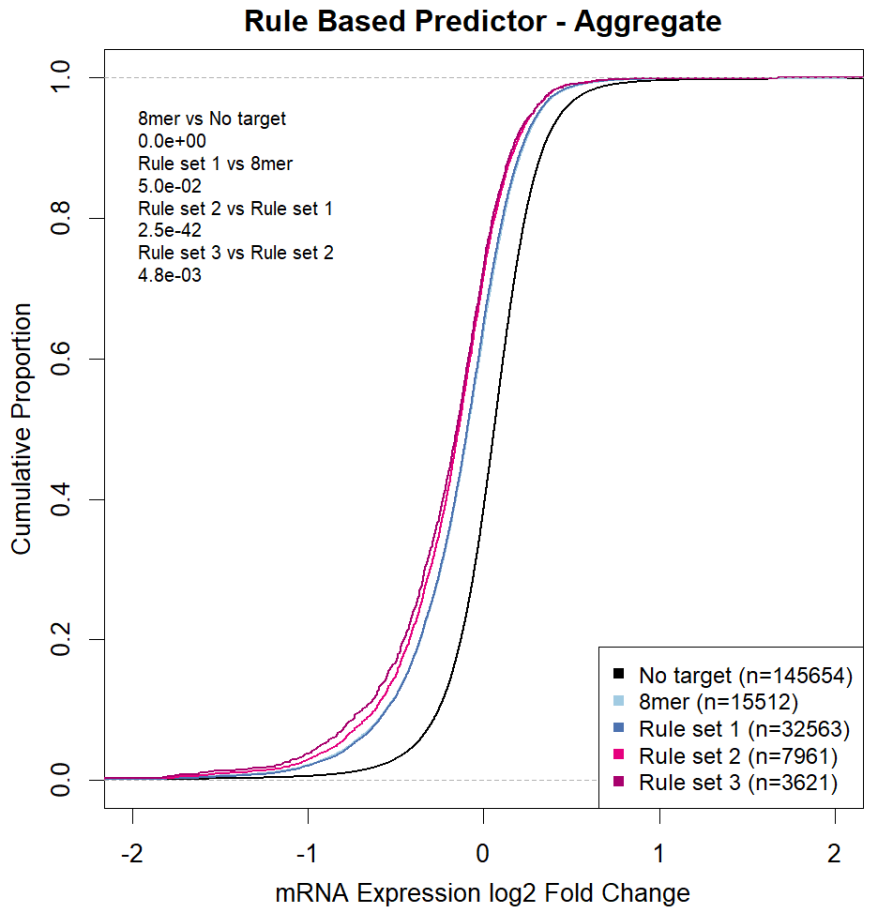


Figure 3.23: Comparison of aggregate rule-based prediction thresholds. Three rule sets constructed from a combination of target recognition features are compared by aggregating the 25 transfection experiments.

At each threshold, an improvement is seen in the predictor's ability to identify targets. Each successive line shift is significantly deviated from the previous, evidenced by the p-values: 5.0×10^{-2} rounded up (rule set 1 vs 8mer), 2.5×10^{-42} (rule set 2 vs rule set 1) and 4.8×10^{-3} (rule set 3 vs rule set 2). However, a lower number of predictions are made as a result: 32,563 in rule set 1, 7,961 in rule set 2 and 3,621 in rule set 3.

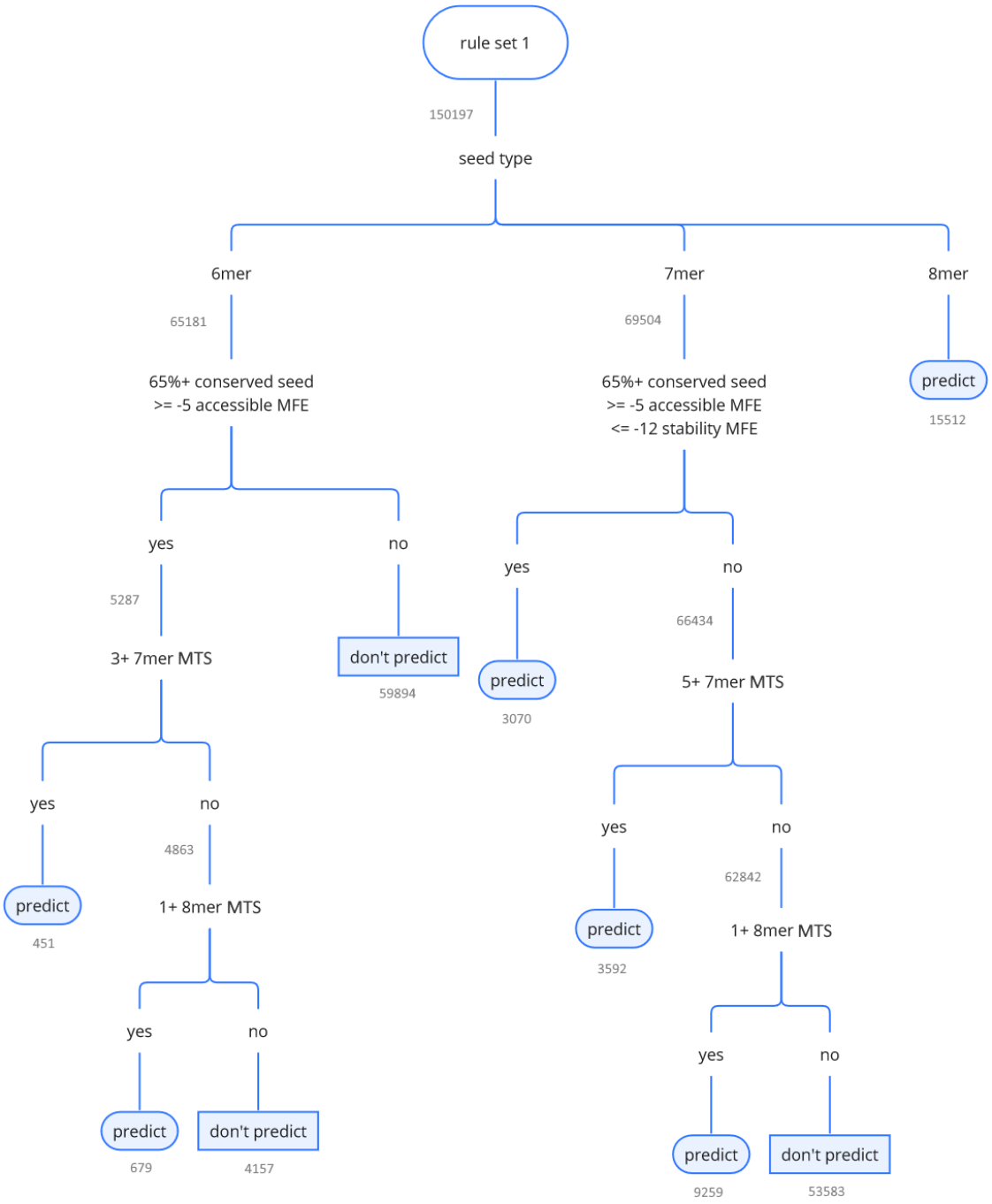


Figure 3.24: Rule set 1 expressed as a decision tree. A relatively loose set of filters for 6mers, 7mers and 8mers. The prediction of 6mers and 7mers is conditional to ensure their downregulatory effect is on a similar level to 8mers. A 6mer is only predicted if additional 7mers or an 8mer exists in the 3′ UTR through MTS.

Rule set 1 is significant in identifying targets at a higher rate to 8mers, despite producing over twice the number of predictions (p-value 5.0×10^{-2} rounded up).

As 6mers confer significantly less efficacy overall, they require the most stringent filters. 5,287 of 65,181 6mers have a 65% conserved seed and are deemed accessible using a representative MFE of -5. With the MTS condition met, only 1,130 (1.73%) of all 6mers are predicted under rule set 1.

7mers are predicted at a rate of 22.91%, making up 15,921 of the 32,563 predicted targets made by rule set 1, more than unfiltered 8mers. MTS is not required for a 7mer to downregulate at an 8mer average level. Instead, MTS offers a secondary path for non-conserved, inaccessible or less stable 7mers to pass the filter. This is similar to the concept of 'weak' and 'strong' filters utilised in the original MirTarget (Wang and Wang, 2006), and is responsible for 81.1% of 7mers which pass.

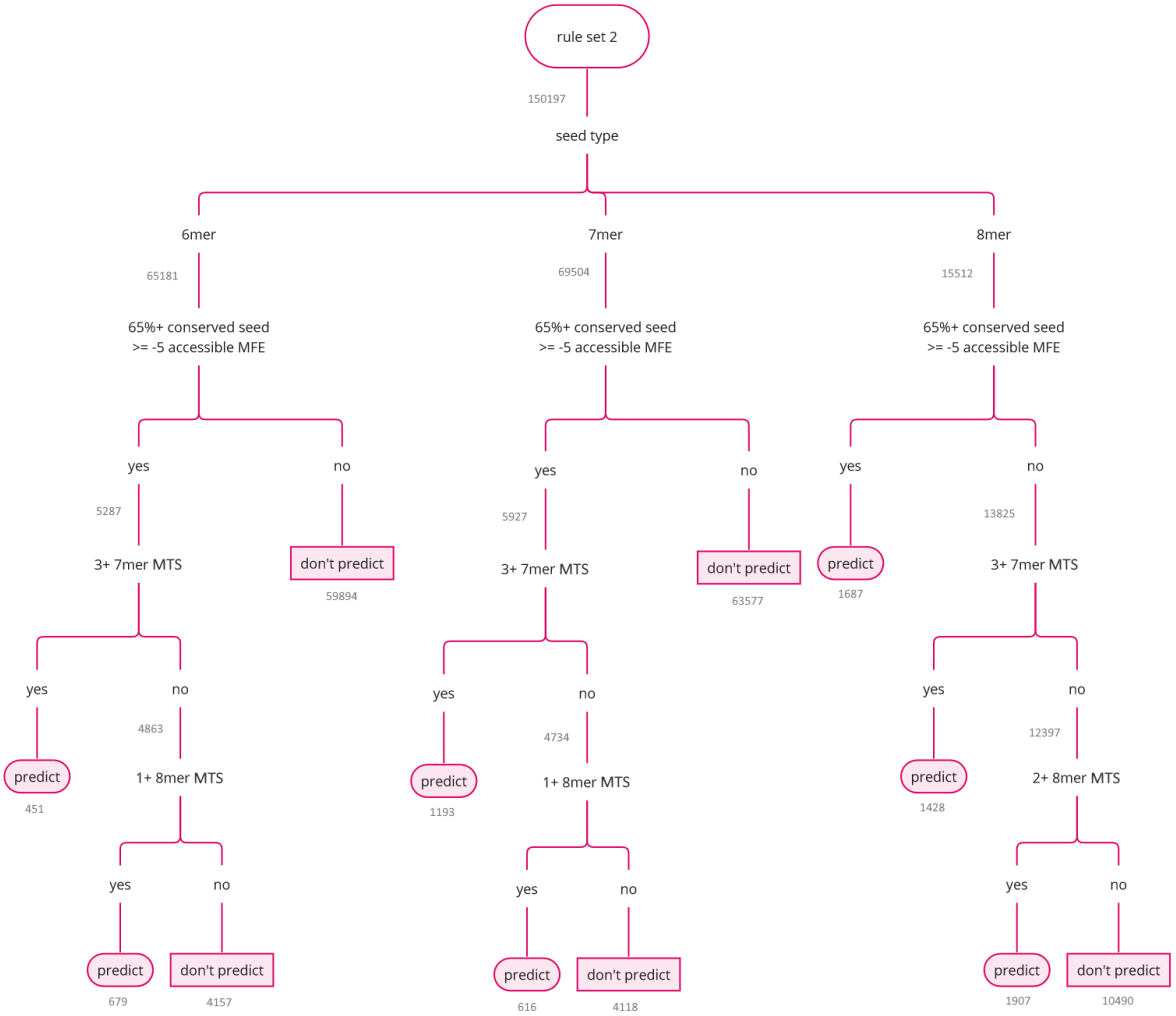


Figure 3.25: Rule set 2 expressed as a decision tree. A relatively balanced set of filters for 6mers, 7mers and 8mers. The prediction of 6mers and 7mers is highly conditional and dependent on MTS to ensure their downregulatory effect is on a similar level to a well-conserved, accessible 8mer.

Predicting all 8mers is generally not optimal, as they do not all downregulate gene expression equally. A new benchmark is formed around accessible 8mers with 65% conservation in rule set 2, meaning both 6mers and 7mers require MTS to compete.

The seed stability requirement of 7mers is removed to reduce stringency, allowing for an MTS filter. Isolated testing showed MTS is highly effective, but uncommon compared to other tested features (Section 3.3.1.2). As a result, 88.5% less 7mers pass rule set 2

compared to rule set 1. The 6mer requirements are unchanged from rule set 1.

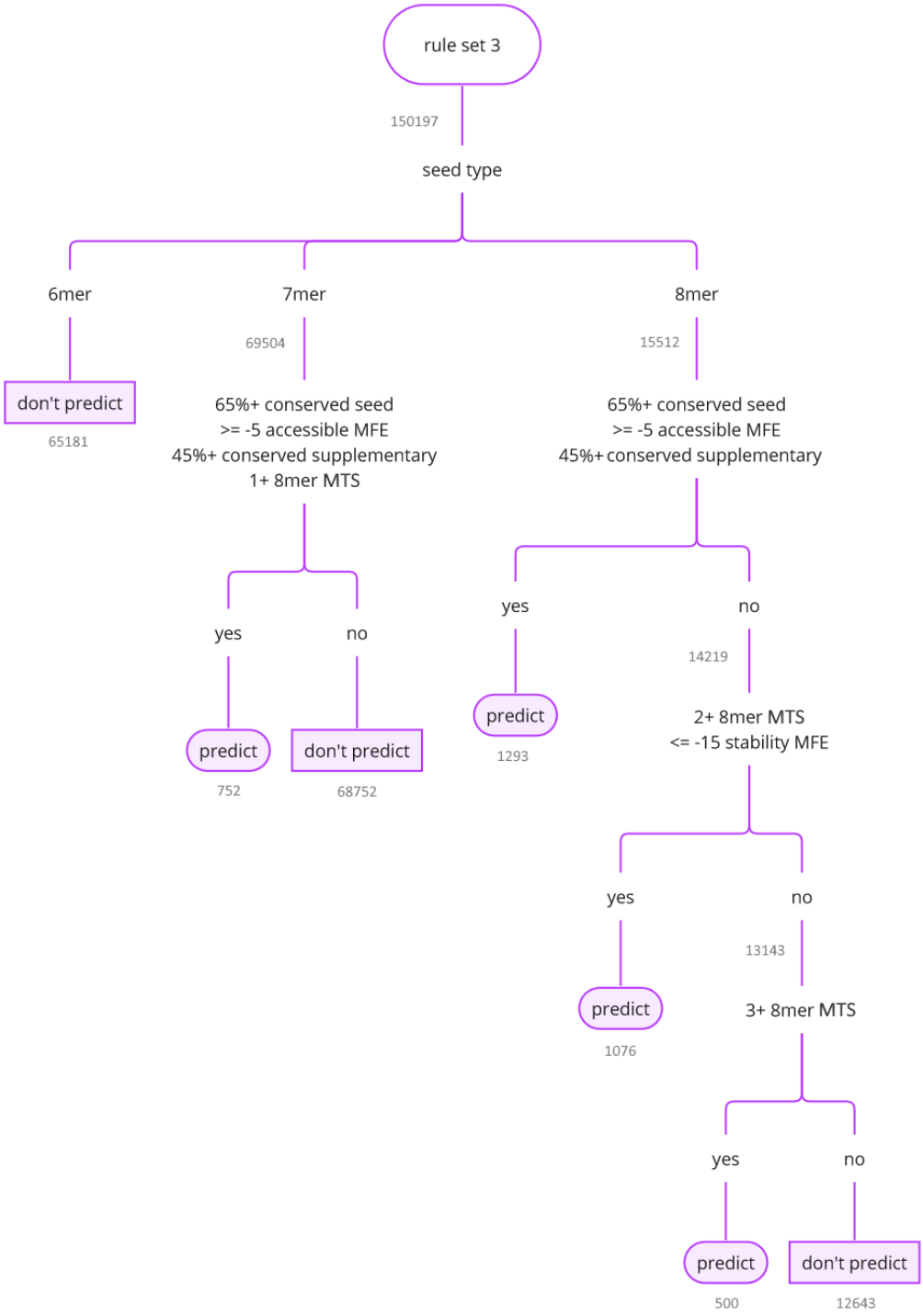


Figure 3.26: Rule set 3 expressed as a decision tree. A relatively harsh set of filters for 6mers, 7mers and 8mers. The prediction of 7mers is very stringent and dependent on 8mer MTS, while 6mers are not predicted. The filter for 8mers is highly conditional and contains several secondary paths.

Rule set 3 suffers a heightened level of diminishing returns compared to rule set 2, producing 55% fewer predictions at a proportionally reduced, yet significant, shift in the cumulative plot (p-value 4.8×10^{-3}).

As strict filtering is already in use at this level, the occurrence of seeds with conserved

supplementary pairing is proportionally higher in this population. This allows the feature to be used at a reduced cost to the number of predicted targets relative to the overall population seen in the isolated feature testing. In 8mers, 1,293 targets pass the filter prior to the integration of MTS.

Compared to rule set 2, 6mers are not predicted and the 7mer condition is simplified to a strict subset of previous features. In addition to the existing requirements, 7mer targets must have a 45% conserved supplementary binding and at least one 8mer site present in the 3′ UTR. The overall prediction is still comprised of 26.2% 7mers, despite the higher 8mer benchmark.

3.4 Discussion

The aim of this chapter was to develop a basic prediction model in order to trial established miRNA target prediction methods. The isolated testing results (Section 3.3.1) were consistent with prior background research into target recognition mechanics (Section 2.4), and each tested feature was proven capable of assisting prediction in some capacity. While performing these tests, results were aggregated to demonstrate that each feature was scalable beyond individual transfection experiments. As a result, the tooling developed throughout this chapter can be expanded upon in future work.

Prediction accuracy was improved through the integration and combination of these isolated features at different stringency levels in the rule-based predictor (Figure 3.23). However, a key issue that arose was the need for leniency in feature boundaries. Of note, a nuanced approach is likely to be required for MTS and supplementary pairing to make them viable prediction features due to their large cost to the overall number of predictions. As Figure 3.25 shows, the complexity of decision branches grows exponentially as more conditional paths are included. ML is likely popular in modern target prediction algorithms for this reason, as it has strong implicit handling for feature interdependency.

A trade-off is clear between the number of targets produced and the accuracy of the prediction in rule-based prediction. Target prediction tools are used in more than one context, necessitating that a balance be struck between the two. Depending on the use case, the optimal stringency is likely to fall somewhere between rule sets 2

and 3. As rule set 3 produces fewer, but more highly accurate predictions in its current state, it would probably be considered more useful unless a ranking system was introduced. Categories supporting both a 'top' and 'bottom' line, such as 'low conservation' and 'high conservation', tend to identify both a positive and negative shift. Where these correlations exist, the feature could theoretically be used to derive a primitive confidence metric.

It is apparent that seed type is an important factor affecting all tested features. For example, in cases such as seed stability and conservation, 6mers exhibit a more subdued reaction to those with stronger seed types (Figures 3.14 and 3.21). This potentially highlights an advantage in training individual models for different seed types, a method used by TargetScan (Section 2.5.2.1).

In terms of individual features, MTS was highly effective (Figure 3.12), and the rule-based predictor was largely built around it. A considerable number of 6mers and 7mers are able to benefit from this kind of redundancy because an additional target site is common in the 3′ UTR. After seed type and MTS, seed conservation was the most useful feature for building rule sets, which is reflected in its high significance (p-value 1.2×10^{-161}) in isolated testing (Figure 3.21). Supplementary binding was the weakest individual feature tested (Figure 3.18), although this may be a result of the rigid method of implementation; bp between predefined windows were simply counted without an examination of the wider binding context. A more effective method may need to use a case-by-case implementation based on seed type and other binding factors. For example, a binding with low seed stability may prefer supplementary binding as close to the seed as possible, regardless of the number of supplementary bases involved in pairing; on the other hand, an 8mer may simply benefit from any level of supplementary binding not in accordance with a single definition. However, it is also possible that supplementary binding is inherently conditional and, without considering other factors, its impact will be reduced. Testing conservation scores on paired supplementary bases produced substantially stronger results, even without accounting for seed conservation (Figure 3.22).

All the common features of target prediction tools were tested, except for AU content (Figure 2.4). In general, accessibility measures are an area of weakness in this chapter. While the three-window approach was able to identify low accessibility in 7mers and

8mers (Figure 3.3.1.4), it relies on nested prediction; the most likely window is used to infer the most likely secondary structure. Furthermore, the implementation is limited because RNAfold computes an MFE value over a window, as opposed to specific bases. A more ideal implementation would allow the seed and supplementary bases to be tested independently to provide more context. An argument could be made that its positive result in isolated testing may simply be due to the underlying strength of the feature. Improving the quality of site accessibility measures is therefore important prior to the feature set becoming fixed for machine learning.

Chapter 4

Site Accessibility Measures

4.1 Summary

This chapter describes research into a number of alternative methods for measuring target site accessibility, a staple feature of target prediction algorithms (Figure 2.4). While the rudimentary three-window approach to computing secondary structure was proposed and tested in Chapter 3, it was highlighted as a candidate for improvement due to its limitations, such as a lack of per-base precision and highly predictive nature (Section 3.4).

Three methods of computing site accessibility are introduced to the feature-set: base pairing prediction with RNAplfold, the measurement of local AU content, and a novel approach using SHAPE-seq data combined over five cell lines. Each method is able to compute per-base scores, allowing specific accessibility features to be extracted relative to both the seed and supplementary portions of a target site, improving the model's overall knowledge of the binding context.

4.2 Methods

4.2.1 Secondary Structure Stability

The proportion of AU bases present in secondary structure is indicative of weaker stability, which may be interpreted as a higher level of site accessibility in nearby bases (Section 2.4.3). In this way, it provides an alternative measure of accessibility to folding methods that rely on structural predictions.

Since mRNA sequences are processed extensively in this study, determining the presence of AU content is achieved by simply examining the windows extracted for RNAfold. A 30 nt window is taken in both the $5'$ and $3'$ flanking regions relative to the seed target site, based on a similar approach originally used by TargetScan (Grimson et al., 2007).

TargetScan weights the importance of bases progressively lower the further they are from the seed. As TargetScan also builds its model for each seed type independently, the exact weights chosen differ by seed type. For stronger seeds, a proportionally higher weighting is placed on bases 9 and 10 in particular. A simplified version of TargetScan's weighting system is used, which instead does not weight seed types separately. Figure 4.1 visualises these weightings as a sequence of progressively shrinking fractions. Although fractional representations are used to illustrate the sequence, the weights themselves are normalised by division against the sum total of the sequence.

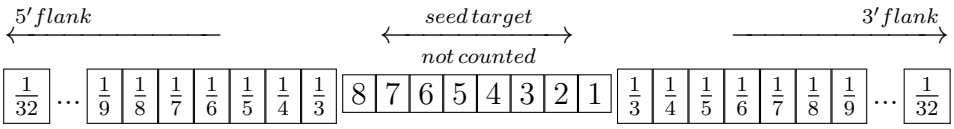


Figure 4.1: Visualisation of weights used in weighted AU content. Weights are applied in a decrementing sequence in the flanking regions around a seed target. As a result, emphasis is placed closer to the seed.

4.2.2 Local RNA Pairing Probability

RNAplfold (Bernhart et al., 2006a) computes the local unpaired probabilities of bp across a maximal span, with a further option to compute these probabilities consecutively for each successive base. The values are derived from the frequency of certain pairs in the local minimum energy structures that are constructed from a given window. A key benefit to using RNAplfold over both the three-window RNAfold and AU content approaches is that it provides a more direct measure of accessibility, being that MFE and AU proportion are measures of structure, as opposed to accessibility.

The window (`-W`) supplied to RNAplfold is extracted by taking 36 nt either side of the 8 nt potentially involved in a seed binding, for a total of 80 nt. Providing a large window to RNAplfold allows it to implicitly account for secondary structure that may affect the bases involved in the binding itself. However, a limit (`-L`) can also be set for the maximum span that secondary structure may be considered. RNAplfold computes a matrix containing the running likelihood of 1-`u` bases being unpaired, where `u` refers

to a user-defined consecutive base limit (`-u`). The values used for the window size and maximal span are based on published research for optimal miRNA metrics (Tafer et al., 2008), while the consecutive sequence limit follows TargetScan's arguments (Agarwal et al., 2015).

```
RNAplfold -L 40 -W 80 -u 14 --auto-id -o < input_file
```

Table 4.1 signifies what a limited output of RNAplfold may look like when using an 8 nt window with a 6 nt running limit. In the resulting 6×8 matrix, the walking probability is output as *NA* prior to the correct number of bases becoming available for computation. For example, cell (1, 1) can be computed, as a spanning probability will only require a single base. However, (1, 2) would require a second base to exist. In the second row, up to cell (2, 2) can be computed, but (2, 3) requires a third. The accessibility of a theoretical 6mer ending at base 7 can be calculated by taking the 6th value of the 7th row, making the unpaired probability 0.091. In this example, increasing the number of bases available in the window would allow further accessibility scores to be calculated for the supplementary portion of the target.

Table 4.1: Example RNAplfold output

	1	2	3	4	5	6
1	0.591	*NA*	*NA*	*NA*	*NA*	*NA*
2	0.949	0.541	*NA*	*NA*	*NA*	*NA*
3	0.139	0.137	0.135	*NA*	*NA*	*NA*
4	0.181	0.132	0.130	0.129	*NA*	*NA*
5	0.996	0.181	0.131	0.130	0.129	*NA*
6	0.955	0.950	0.147	0.121	0.119	0.118
7	0.130	0.117	0.116	0.094	0.093	0.091
8	0.091	0.090	0.085	0.085	0.080	0.080

4.2.3 RNA Structure Analysis

SHAPE-seq is a sequencing-based approach to quantifying structure information (Section 2.3.3). SHAPE-seq functions by chemically modifying RNAs to probe secondary structure, making it non-predictive in nature. The implementation of SHAPE-seq used in this study is provided by icSHAPE-pipe (Li et al., 2020), a comprehensive toolkit for the end-to-end computation of reactivity values with quality control and reporting. Published SHAPE-seq data for the HeLa, HEK-293, K562, HepG2 and H9 cell lines (Sun et al., 2021) is also used to generate SHAPE-seq accessibility scores.

Table 4.2: An overview of SH-sun-HS

Internal ID	SH-sun-HS
Accession	PRJNA608297
Species	*Homo sapiens*
Data Type	SHAPE-seq
Procedure	icSHAPE
Cell Line	HEK293, HeLa, K562, HepG2, H9
Biological Replicates	2
Sequence Type	Single-end
Source	Sun et al. (2021)

Sequences are extracted from SHAPE-seq as a set of 0-1 reactivity values. Unlike other considered accessibility approaches, SHAPE-seq coverage is limited, as only 21,396 of 150,197 (14%) reactivity values for HeLa target sequences are obtainable. This is an issue for ML because it complicates an ML model's ability to understand the feature's importance, particularly if it is high-performing when data is available. To counteract this, sequences are logged as *NA* if there is no available base data, while missing values in an otherwise populated series are padded to somewhat salvage the remainder of the sequence.

Using SHAPE-seq data from HeLa only may potentially limit the feature's application to RNA-seq data originating from other sources. On the other hand, while published

SHAPE-seq data does include data from other cell lines, this may lower the effectiveness of the feature in targeting HeLa. Since increasing coverage is important in making the method viable for features, individual base reactivity scores are combined across the same transcripts of different cell lines to observe the extent of their correlation.

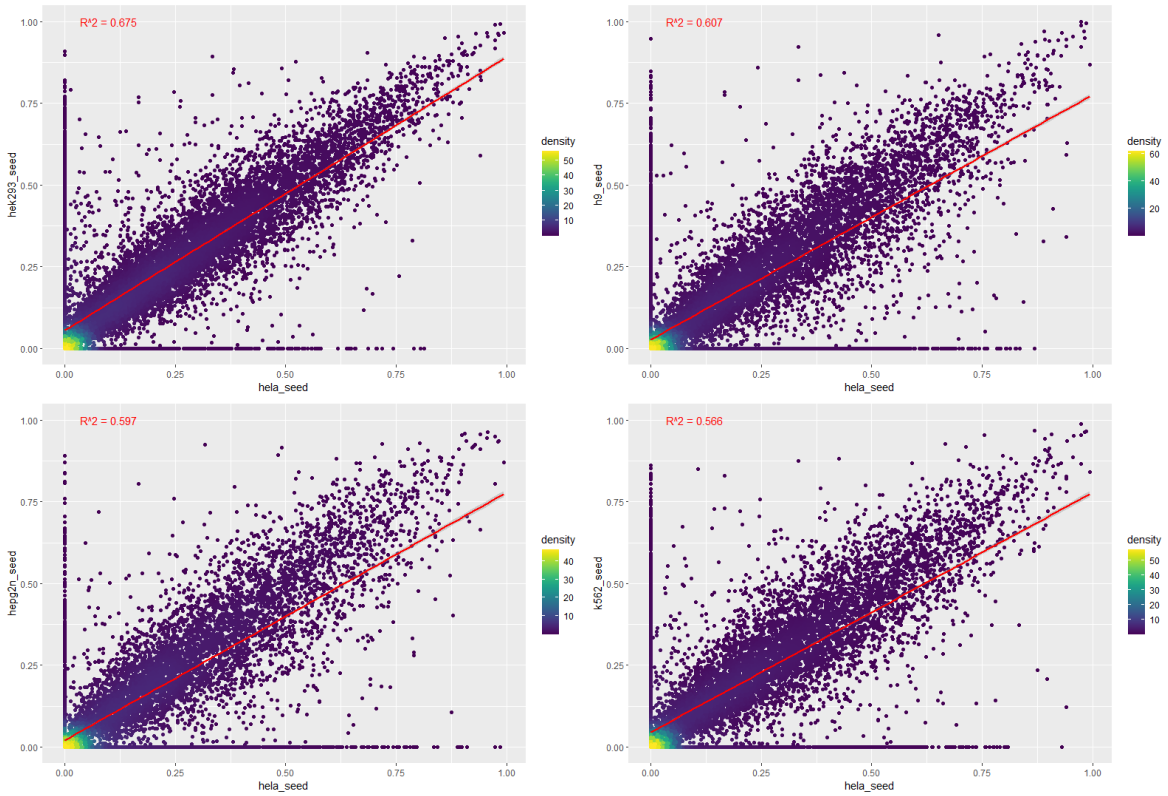


Figure 4.2: Correlation between SHAPE-seq reactivity in HeLa and other cell lines. A scatter plot of each transcript containing a reactivity value in both of the compared cell lines. (Top left) HeLa vs HEK-293. (Top right) HeLa vs H9. (Bottom left) HeLa vs HepG2. (Bottom Right) HeLa vs K562.

Plotting HeLa reactivity values against those of HEK-293, H9, HepG2 and K562 shows a positive correlation in all cases, producing a mean R^2 of 0.611. There is a build-up of scores in all cell lines around 0, though the density scale indicates the majority occur at (0, 0). Nonetheless, a substantial number occur along the x and y axis, which may be due to disparities in secondary structure between different cell lines. In these cases, a consensus value may prove beneficial to the feature accuracy, as target prediction is independent of cell line.

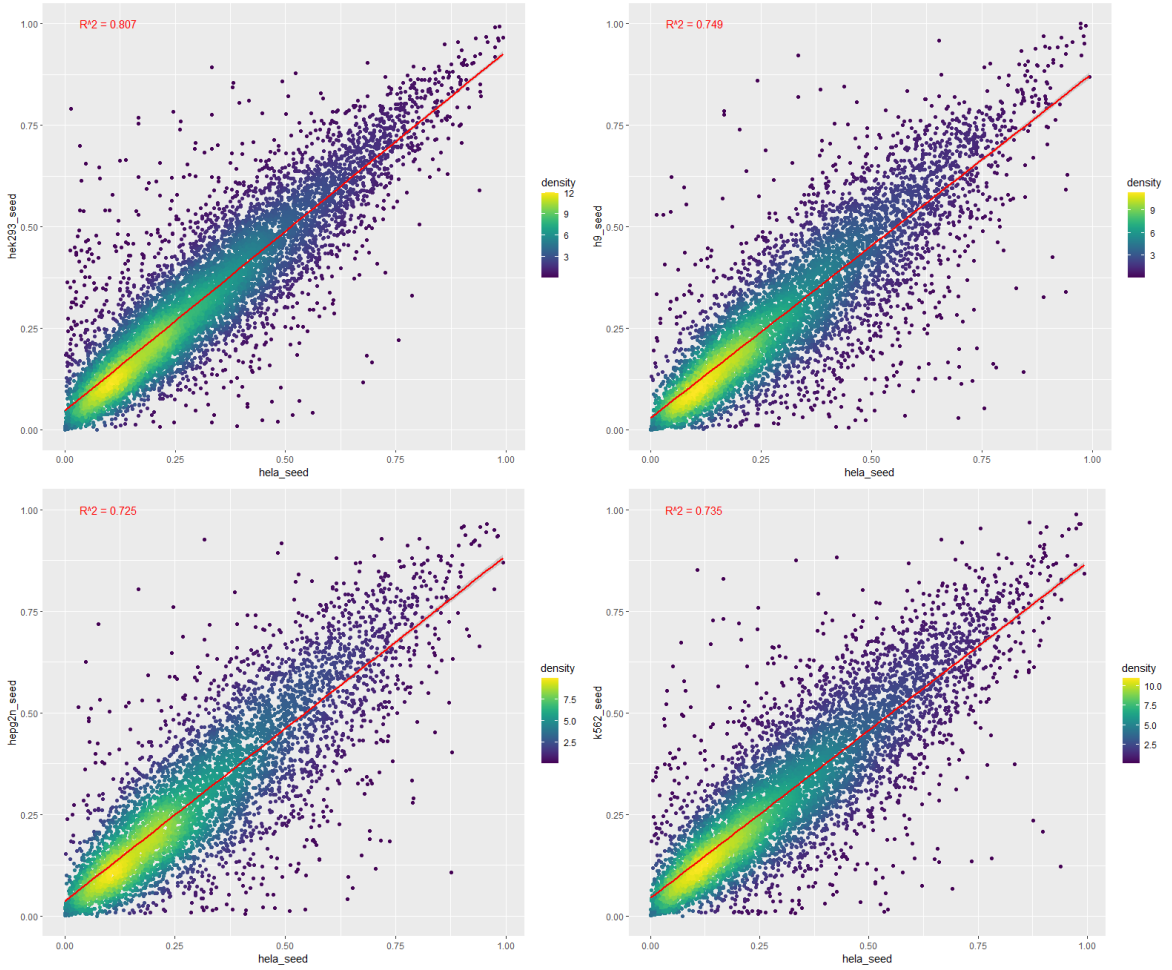


Figure 4.3: Correlation between non-zero SHAPE-seq reactivity in HeLa and other cell lines. A scatter plot of each transcript containing a reactivity value in both of the compared cell lines. Reactivity values of 0 are removed. (Top left) HeLa vs HEK-293. (Top right) HeLa vs H9. (Bottom left) HeLa vs HepG2. (Bottom Right) HeLa vs K562.

In order to determine the correlation of values which are more closely related, 0 is filtered from the results. This increases the mean R^2 value to 0.75, which is a relatively strong correlation between similar values of different cell lines. As previously discussed, the inclusion of 0 is still beneficial to mitigate skewing towards a single cell line; when calculating the mean between a variety of sources, these values will reduce the weighting of a single distorting value.

4.3 Results

The significance of each accessibility measure is tested using a p-value threshold of 0.05, derived from a one-sided two-sample MW test.

4.3.1 Local AU Content

Local AU content is tested in the 5′ and 3′ flanking regions, defined by TargetScan, with and without weights. Similarly, an AU score is computed for the seed.

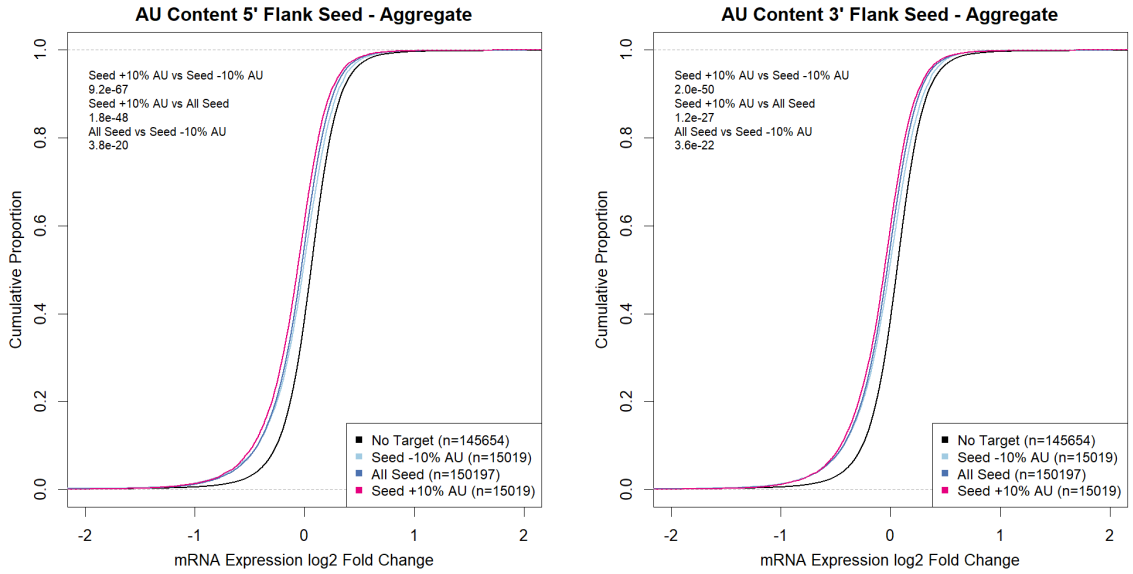


Figure 4.4: Comparison of flanking AU content efficacy. Target site accessibility using AU content is compared by aggregating the 25 transfection experiments. (Left) AU content in the 5′ flanking region. (Right) AU content in the 3′ flanking region.

The significance in separation between the top 10% and bottom 10% of AU content scores for the 5′ flanking region (p-value 9.2×10^{-67}) is slightly higher than that of the 3′ (p-value 2.0×10^{-50}). This is likely due to the greater importance of the 5′ region of the mRNA, as a result of its role in miRNA 3′ supplementary binding. Higher accessibility in this area allows the bases to pair, strengthening the core binding and improving efficacy. Despite this, both regions provide new context on unrelated bases, so their inclusion as features is not mutually exclusive.

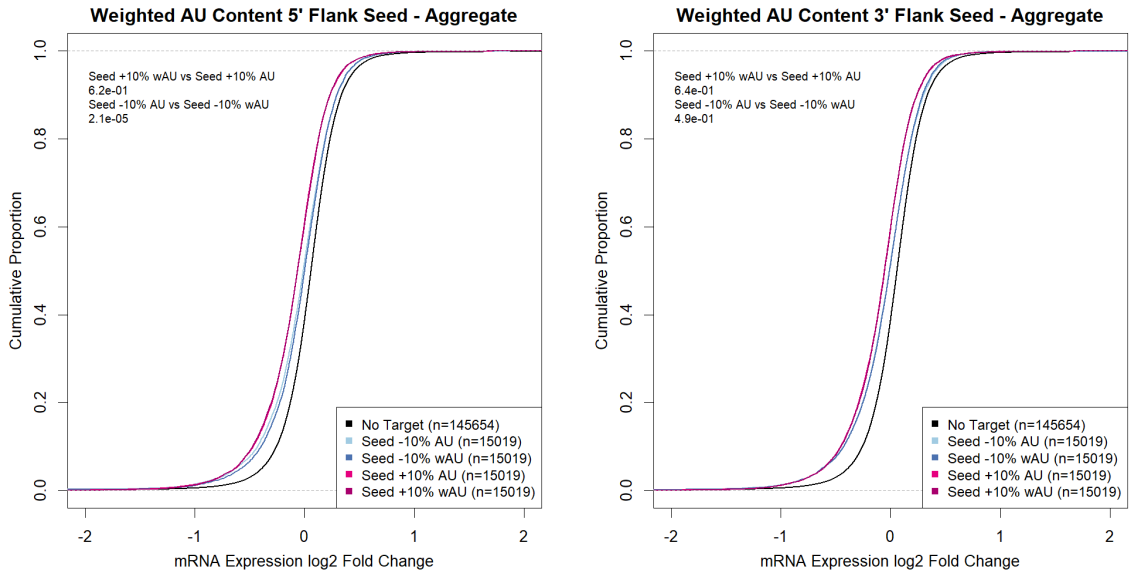


Figure 4.5: Comparison of weighted and unweighted flanking AU content. Target site accessibility using weighted AU content is compared by aggregating the 25 transfection experiments. (Left) Weighted and unweighted AU content in the 5′ flanking region. (Right) Weighted and unweighted AU content in the 3′ flanking region.

The base-weighted 5′ flank shows a significant improvement compared to its unweighted result in the bottom 10% of cases (p-value 2.1×10^{-5}), though this significance does not extend to the top 10% (p-value 6.2×10^{-1}). There is a logical basis in applying weightings here, as paired supplementary bases are of greater importance closer to the seed. The 3′ flanking region does not gain significant benefit from weighting (p-values: 6.4×10^{-1} and 4.9×10^{-1}), as such a mechanism is not present on the 3′ side of the binding.

The effect of AU content in the seed is tested using two different implementations. In the first, the seed is treated dynamically; the bases used to calculate the proportional representation of AU content vary depending on the seed definition. In the second, only the traditional 6mer bases 2-7 are used.

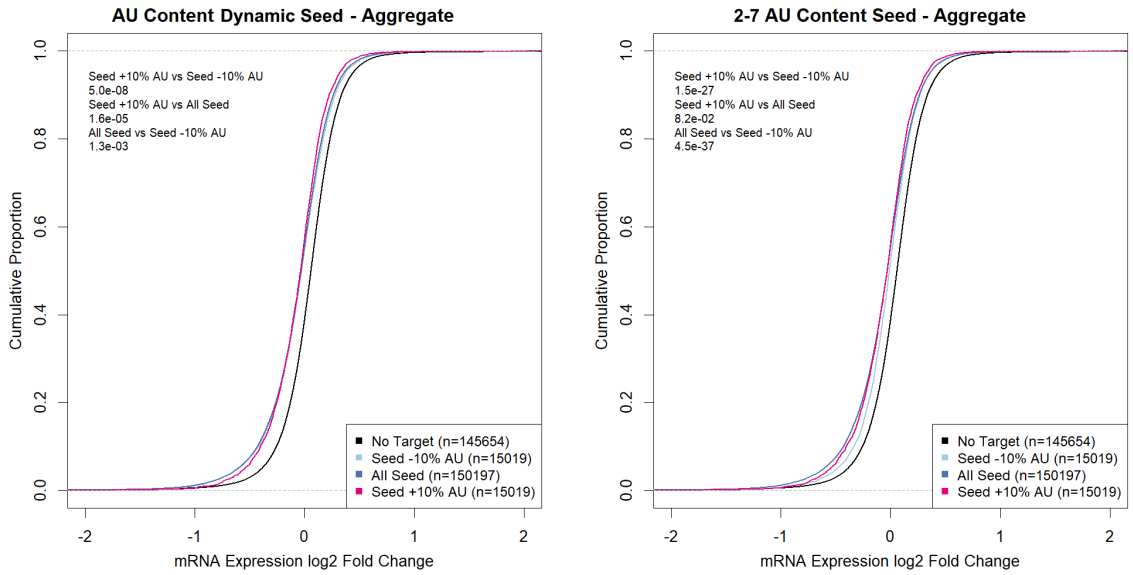


Figure 4.6: Comparison of dynamic and fixed seed AU content. Target site accessibility using AU content is compared by aggregating the 25 transfection experiments. (Left) The seed region is defined dynamically, according to the type of binding that occurs. (Right) The seed is defined as a 6mer of bases 2-7.

Dynamic handling for the seed bases produces an inferior p-values compared to the static definition, 5.0×10^{-8} vs 1.5×10^{-27}. The difference is mostly observable in the bottom 10% AU lines, 1.3×10^{-3} vs 4.5×10^{-37}, implying the difference is due to the critical nature of bases 2-7 to a successful binding. Since this is the primary 6mer definition that other seed types are derived from, this feature may be considered an indicator of '6mer quality'.

4.3.2 RNAplfold

Using RNAplfold, a separate feature is extracted for the seed and supplementary portions of the binding. A dynamic 6-8 base window is used for the former, depending on the seed type. The supplementary portion consists of bases 9-20, regardless of the seed window used.

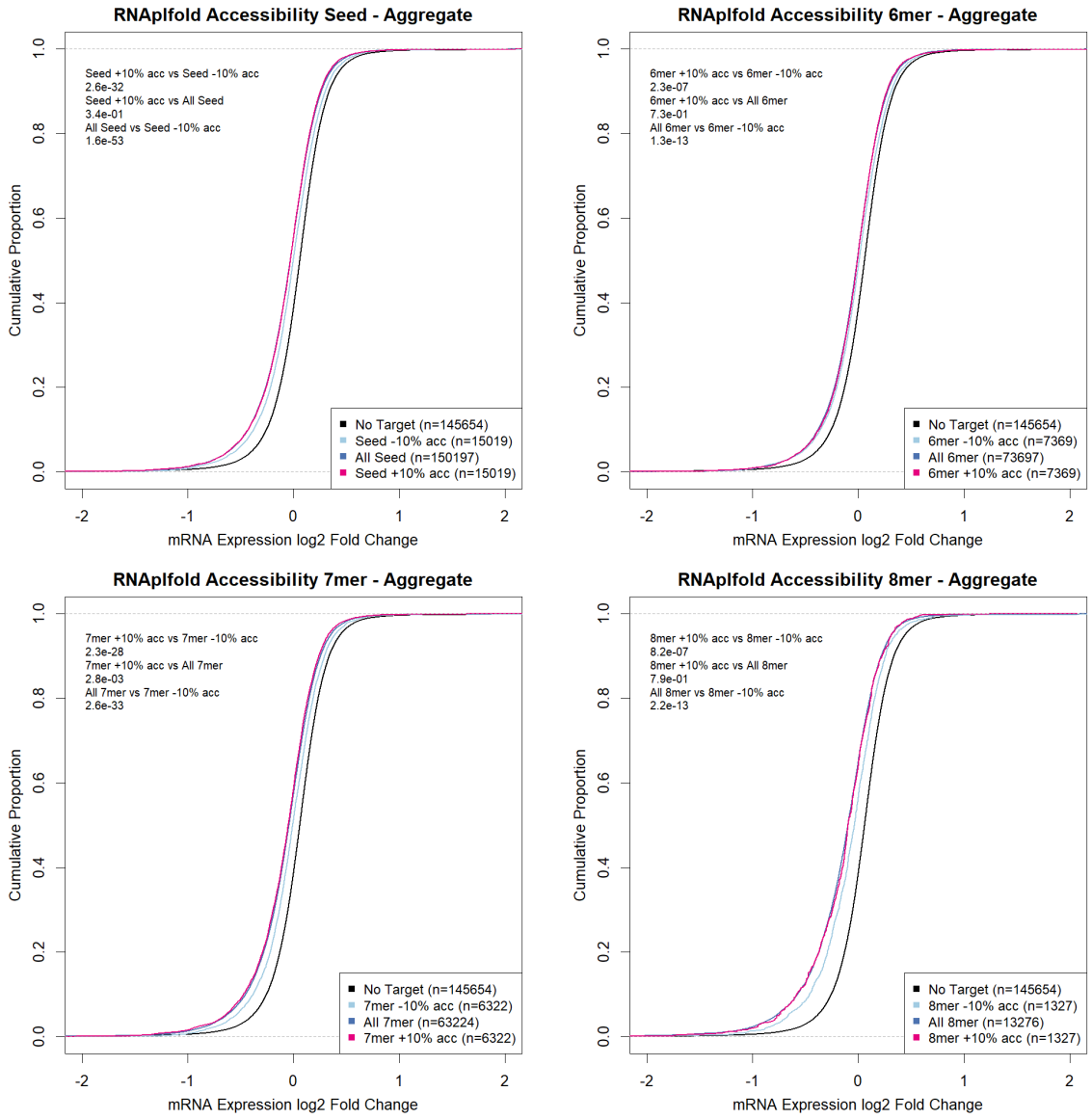


Figure 4.7: Comparison of aggregate RNAplfold efficacy. Target site accessibility using RNAplfold is compared by aggregating the 25 transfection experiments. (Top left) All seed type overview. (Top right) 6mer only. (Bottom left) 7mer only. (Bottom right) 8mer only.

A statistically significant shift can be observed in all seed types when comparing the top and bottom 10% accessibility lines (p-value 2.6×10^{-32}). As with all tested accessibility methods in this study, non-targets (p-value 1.6×10^{-53}) are more distinguished than targets (p-value 3.4×10^{-1}). For AU seed content, a substantial improvement can be seen in RNAplfold's ability to detect inaccessible seeds, producing a p-value of 1.6×10^{-53}, compared to 4.5×10^{-37} (Figure 4.6).

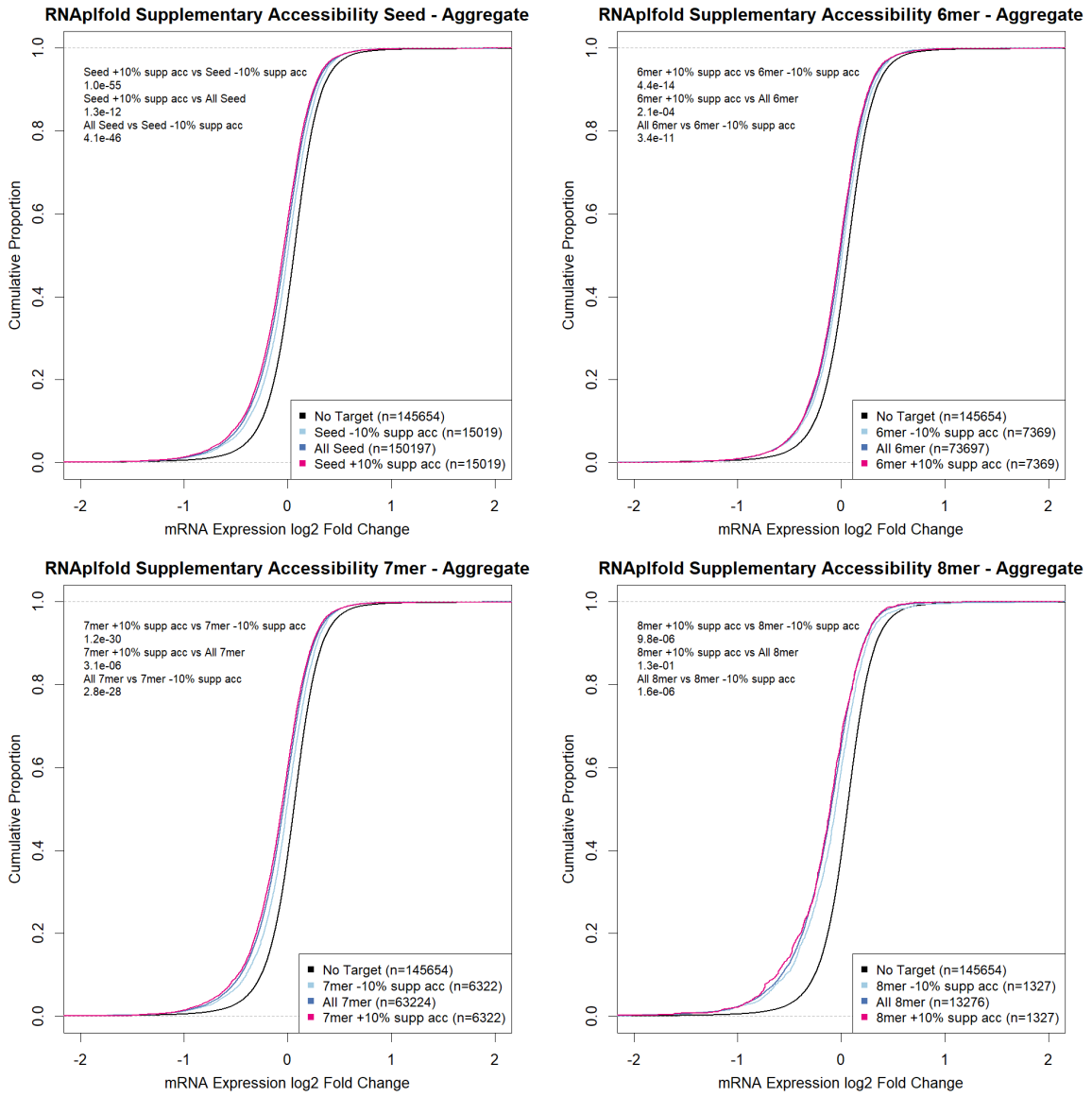


Figure 4.8: Comparison of aggregate RNAplfold supplementary efficacy. Target site supplementary accessibility using RNAplfold is compared by aggregating the 25 transfection experiments. (Top left) All seed type overview. (Top right) 6mer only. (Bottom left) 7mer only. (Bottom right) 8mer only.

Bases 9-20 were previously found to produce the strongest results when testing alternative supplementary base definitions centred around 13-16 (Section 3.2.7.4). Applying RNAfold at this window with an 11-base spanning accessibility score produces significant separation in the least accessible cases (p-value 4.1×10^{-46}), though it is less than seed accessibility (p-value 1.6×10^{-53}). An interesting element to this result is that it produces a slight leftward shift in the top 10% cases for 7mers and 8mers. The shift is significant at a p-value of 1.3×10^{-12}, despite being insignificant in seed accessibility 3.4×10^{-1}.

4.3.3 SHAPE-seq

The same seed and supplementary sequence values are used with SHAPE-seq as they were with RNAplfold, extracted separately to generate two unique features. Results are first generated exclusive to HeLa, and then again using a consensus value derived from the mean of available scores across the five cell lines.

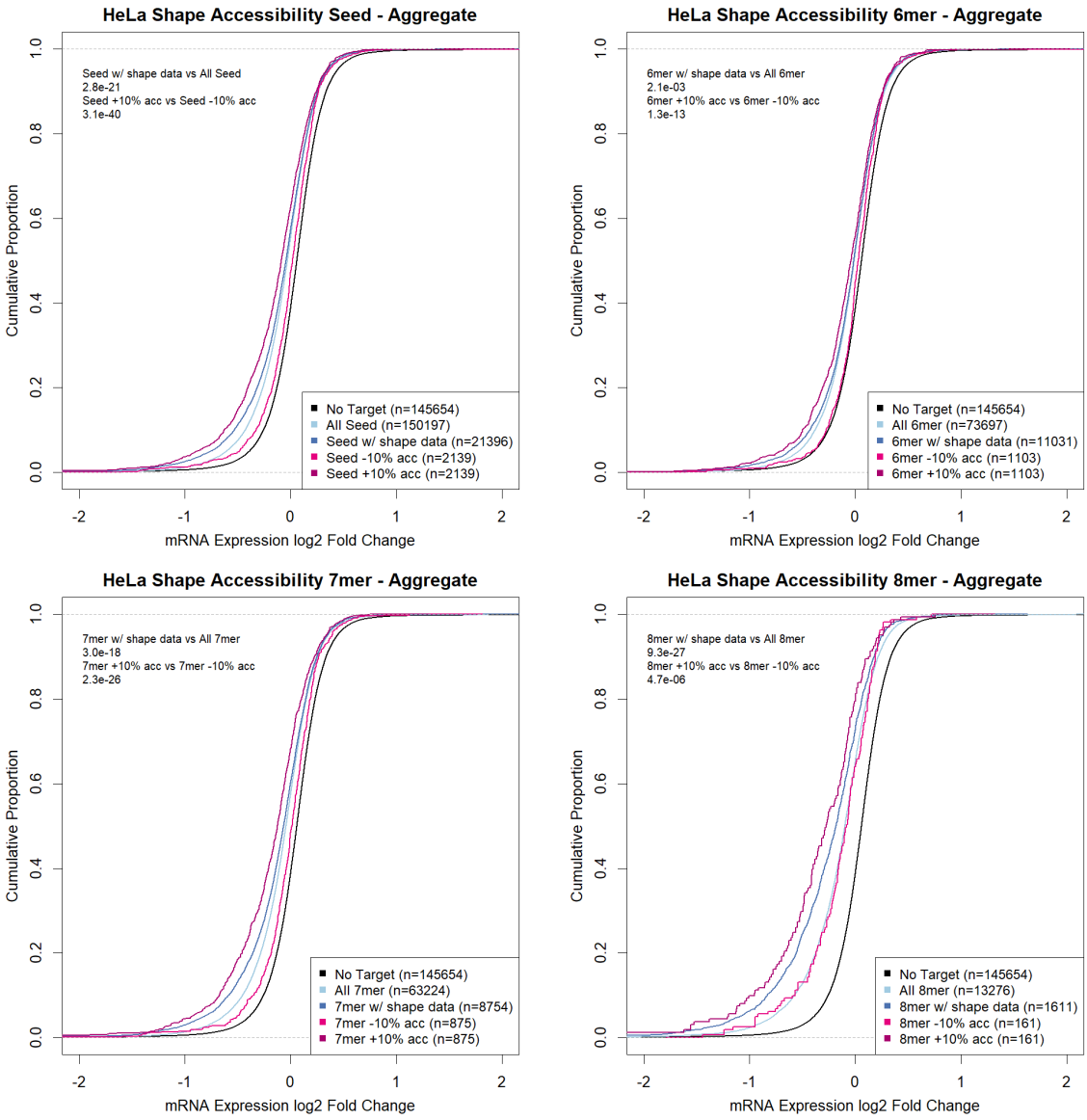


Figure 4.9: Comparison of aggregate SHAPE-seq efficacy from HeLa. Target site accessibility using SHAPE-seq is compared by aggregating the 25 transfection experiments. (Top left) All seed type overview. (Top right) 6mer only. (Bottom left) 7mer only. (Bottom right) 8mer only.

The plots display a radical shift in the high and low accessibility lines (p-value 3.1×10^{-40}). In particular, 6mers with low accessibility under this measure are comparable to transcripts without a seed target, shifting rightward close to the control line. The 8mer result is not as effective in determining non-targets using low accessibility, as it overlaps heavily with the line for 8mers labelled with SHAPE-seq data. This may

be due to bias, as there is a significant separation between all 8mers and 8mers with SHAPE-seq coverage (p-value 9.3×10^{-27}). This value is lower for both 7mer (p-value 3.0×10^{-18}) and 6mer (p-value 2.1×10^{-3}).

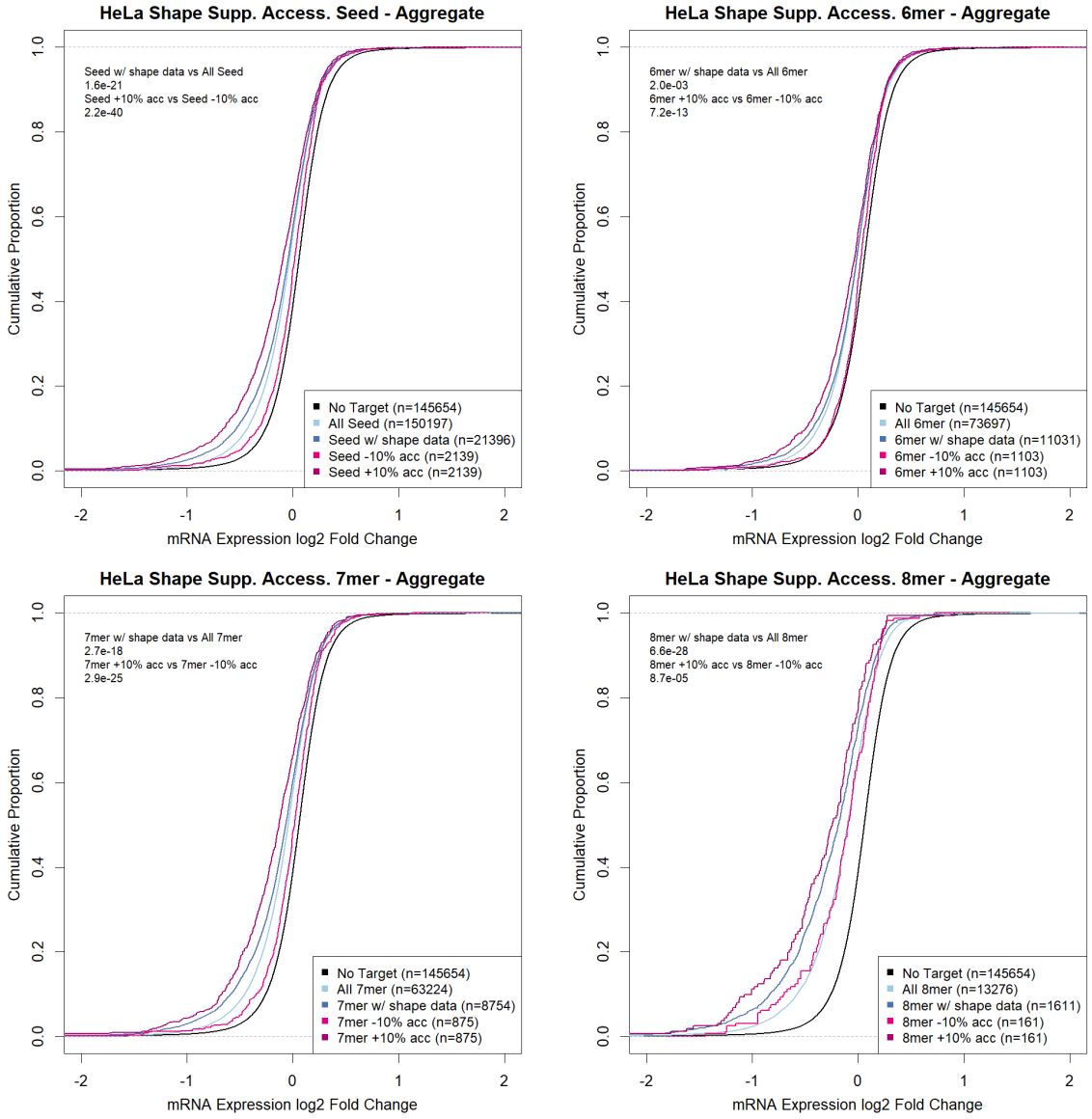


Figure 4.10: Comparison of aggregate SHAPE-seq supplementary efficacy from HeLa. Target site supplementary accessibility using SHAPE-seq is compared by aggregating the 25 transfection experiments. (Top left) All seed type overview. (Top right) 6mer only. (Bottom left) 7mer only. (Bottom right) 8mer only.

A similar pattern can be seen when applied exclusively to the supplementary region, where the shift is again significant (p-value 2.2×10^{-40}). In general, the separation between the most and least accessible categories are some of the strongest isolated feature shifts observed, including those in Section 3.3.1.

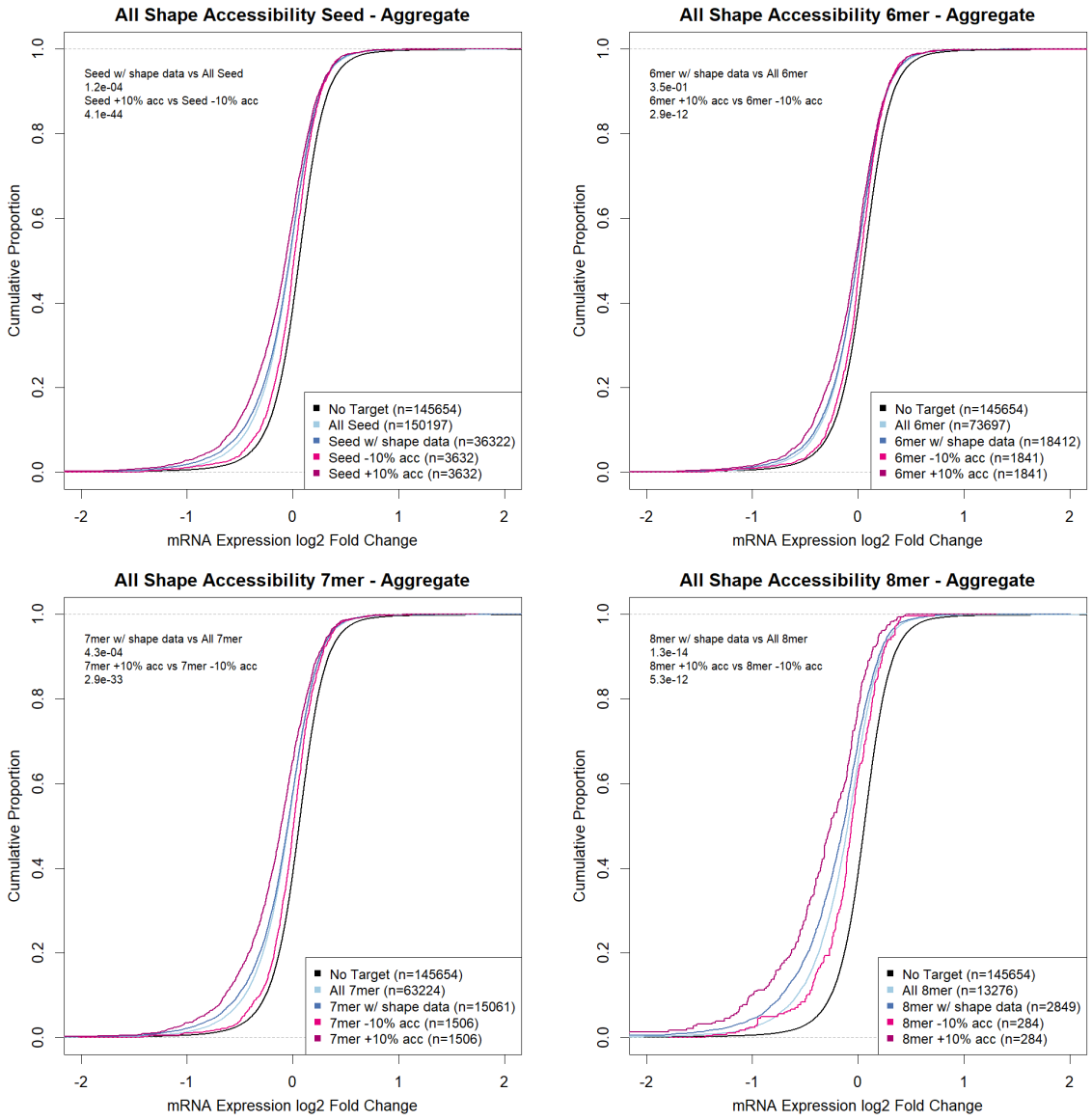


Figure 4.11: Comparison of aggregate SHAPE-seq efficacy from five cell lines. Target site accessibility using SHAPE-seq is compared by aggregating the 25 transfection experiments. (Top left) All seed type overview. (Top right) 6mer only. (Bottom left) 7mer only. (Bottom right) 8mer only.

Although the aggregated transfections originate exclusively from HeLa, averaging reactivity values from all available sources both improves coverage and the overall result. The bias of labelled values is reduced at all levels, evidenced by the smaller gap between 'All Seed' and 'Seed with shape data' lines and reduction of p-value from 1.6×10^{-21} to 1.2×10^{-4}. There is also an improvement in the detection of non-targets and targets, with the significance improving from 3.1×10^{-40} to 4.1×10^{-44}. Additionally, the most accessible 8mers shift leftward at a high rate around the -1.0 log_2 fold change mark. The number of labelled targets after adding data sources is 24% compared to 14% when using HeLa alone, meaning the consensus value increases coverage by a total of 71%.

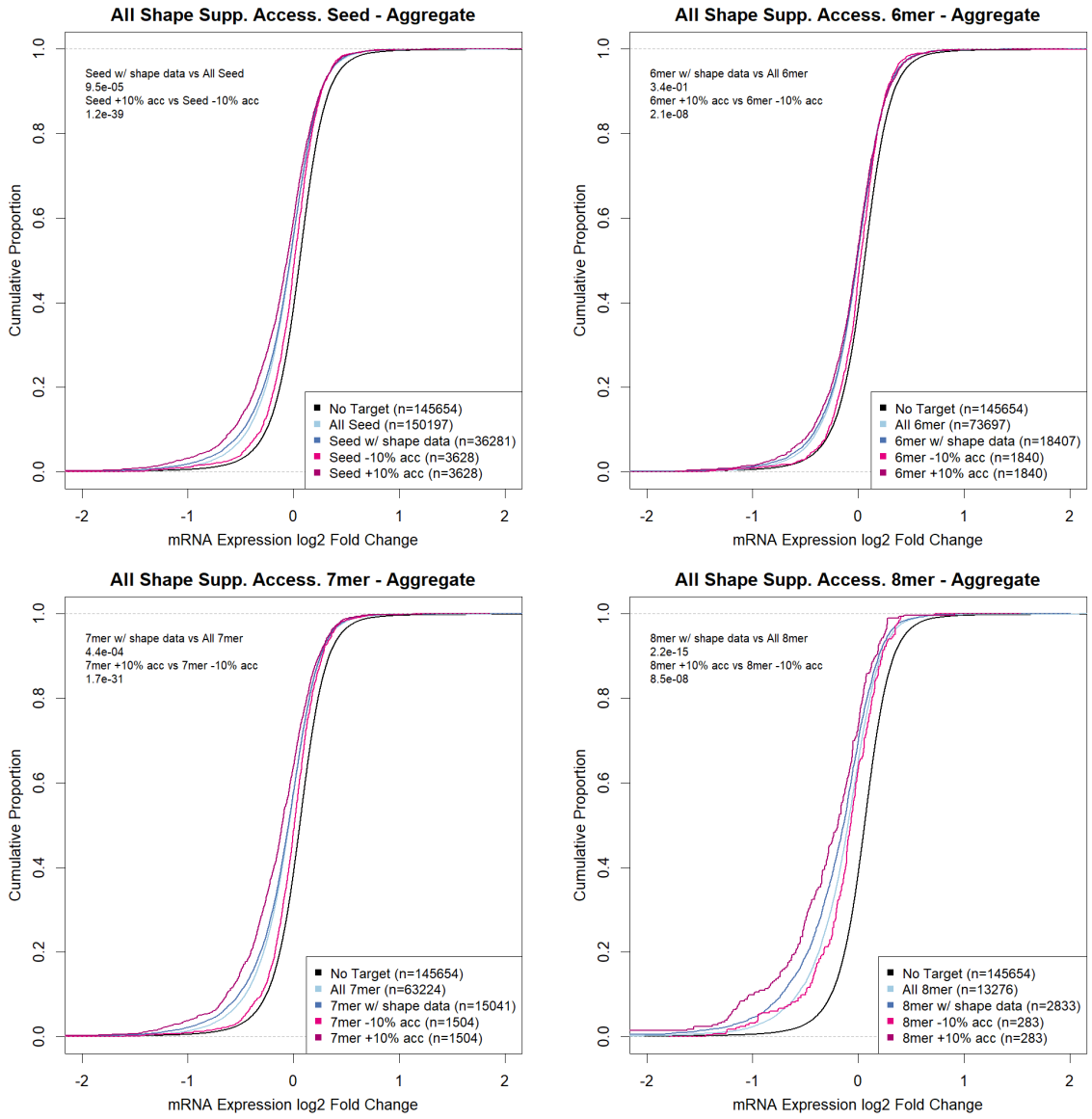


Figure 4.12: Comparison of aggregate SHAPE-seq supplementary efficacy from five cell lines. Target site supplementary accessibility using SHAPE-seq is compared by aggregating the 25 transfection experiments. (Top left) All seed type overview. (Top right) 6mer only. (Bottom left) 7mer only. (Bottom right) 8mer only.

In terms of the supplementary portion, the pattern in results is similar to those of the seed. This further suggests that the improvement from aggregating cell lines is not coincidental and can potentially reduce noise from erroneous reactivity values.

4.4 Discussion

The methods tested in this section each display some merit as potential targeting features, both in identifying true positive targets and filtering those without sufficient accessibility to facilitate a binding.

The strongest overall result is the novel usage of SHAPE-seq structure analysis, both in terms of seed and supplementary features, and its ability to provide strong positive and negative filters. As previously discussed, the lack of feature coverage calls its utility into question. Even with a 71% increase to 24% target coverage, the majority of targets will not be represented in the data. While this can be somewhat offset by the choice of ML algorithm, or mitigated through data mining techniques, it is the only feature to require such measures. Nevertheless, the feature's performance increasing in line with data from different cell lines, even when exclusively testing a specific cell line, is a surprising result warranting further research. Implementing SHAPE-seq data from more cell lines should continue to increase coverage and reduce skew towards specific cell lines, though it should be noted that SHAPE-seq data is not as widely available as RNA-seq.

RNA pairing probability using RNAplfold may be the most useful accessibility feature after factoring in the limitations of SHAPE-seq data. While the shift is smaller by comparison, the feature is consistent and capable of more specific targeting than AU content. Pairing probability has the smallest leftward shifts of the three methods, instead favouring identification of non-targets with low accessibility. This arguably makes it a more direct measure of accessibility, as the 10% most inaccessible bases should reduce binding efficacy more than the 10% most accessible bases increase it, a pattern observable in both AU content and SHAPE-seq results.

The presence of AU content offers a relatively consistent set of features, though the use of large flanking windows means it lacks precision compared to the other methods tested. The seed AU content result is arguably the poorest overall performer, and the AU content value in this case is not cleanly correlated with target efficacy. The weighted $5'$ result is the strongest of the AU content features, which is expected given its application in both TargetScan and miRanda-mirSVR.

A total of 8 feature candidates are extracted as a result of this testing: AU content scores for the $3'$ flanking, $5'$ flanking, weighted $5'$ flanking and seed regions, in addition to pairing probabilities and RNA structure reactivity scores for both the seed and supplementary bases. Combined with the original three-window approach (Section 3.2.7.1), this brings accessibility scoring features up to a total of 9. There is a

variation in the quality and utility of these features, in addition to a degree of redundancy. However, there is no limit on the number of features that may be used to build a model. The inclusion of all potential features at this stage allows an ML model to draw its own conclusions as to their importance, as it may exclude or lower the weighting of poorer performers.

Chapter 5

miRNA:mRNA Target Prediction using Machine Learning

5.1 Summary

This chapter describes the development of an ML model for predicting miRNA targets by evaluating several supervised learning algorithms and benchmarking the results against popular prediction tools. The goal of this work is to build upon prior results by utilising ML to allow for more fluid prediction boundaries and ultimately expand the feature set.

The model and associated scripts are managed by Python (Van Rossum and Drake Jr, 1995), with R and Bash subroutines used for feature extraction. Where possible, original feature extraction logic from Chapters 3 and 4 is retained, though the tool itself is overhauled to function as a more coherent end-to-end piece of software.

5.2 Methods

5.2.1 Tool Architecture

The primary deliverable of Chapter 3 was presented in the form of R scripts with limited architecture and cohesion. This encouraged a leaner and more exploratory development process, which could pivot as discoveries were made. As ML approaches are substantially more complex, improved code structure is a prerequisite to further development to reduce runtime scaling and potential bugs. The engineering philosophy

adopted at this stage emphasises scalability and rapid feature prototyping.

A key adjustment to the tool architecture is the integration of Python to manage script execution and link otherwise independent modules. Python is selected due to its role as a general-purpose programming language, interoperability with R, Linux support (required by ViennaRNA) and widespread adoption in academic research. In addition, access to the packages scikit-learn (Pedregosa et al., 2011), for general ML, and TensorFlow (Abadi et al., 2015), for deep learning, are also determining factors.

The tool is composed of three thematic parts: setup (Figure 5.1), responsible for acquiring annotations and caching conservation scores; feature extraction (Figure 5.2), in which data relating to target recognition is mined and processed; and ML (Figure 5.3), where models are trained and evaluated to determine a 'best' predictor for future work.

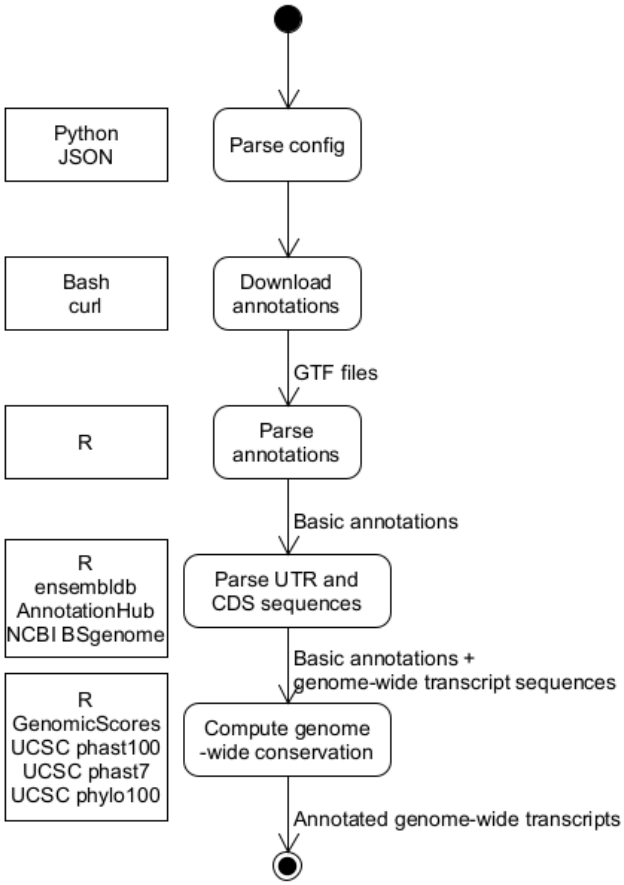


Figure 5.1: Setup module activity diagram. The setup module is primarily responsible for parsing settings, acquiring annotations and building the conservation cache.

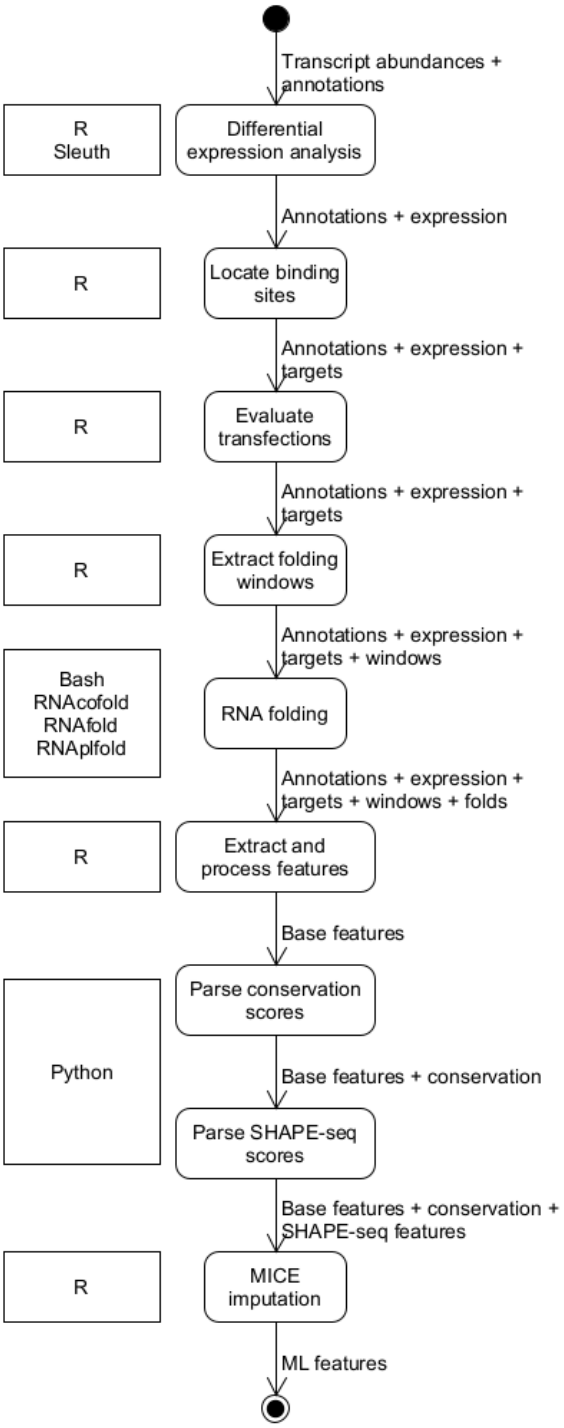


Figure 5.2: Feature extraction module activity diagram. The feature extraction module is primarily responsible for differential expression analysis, target location and feature preparation.

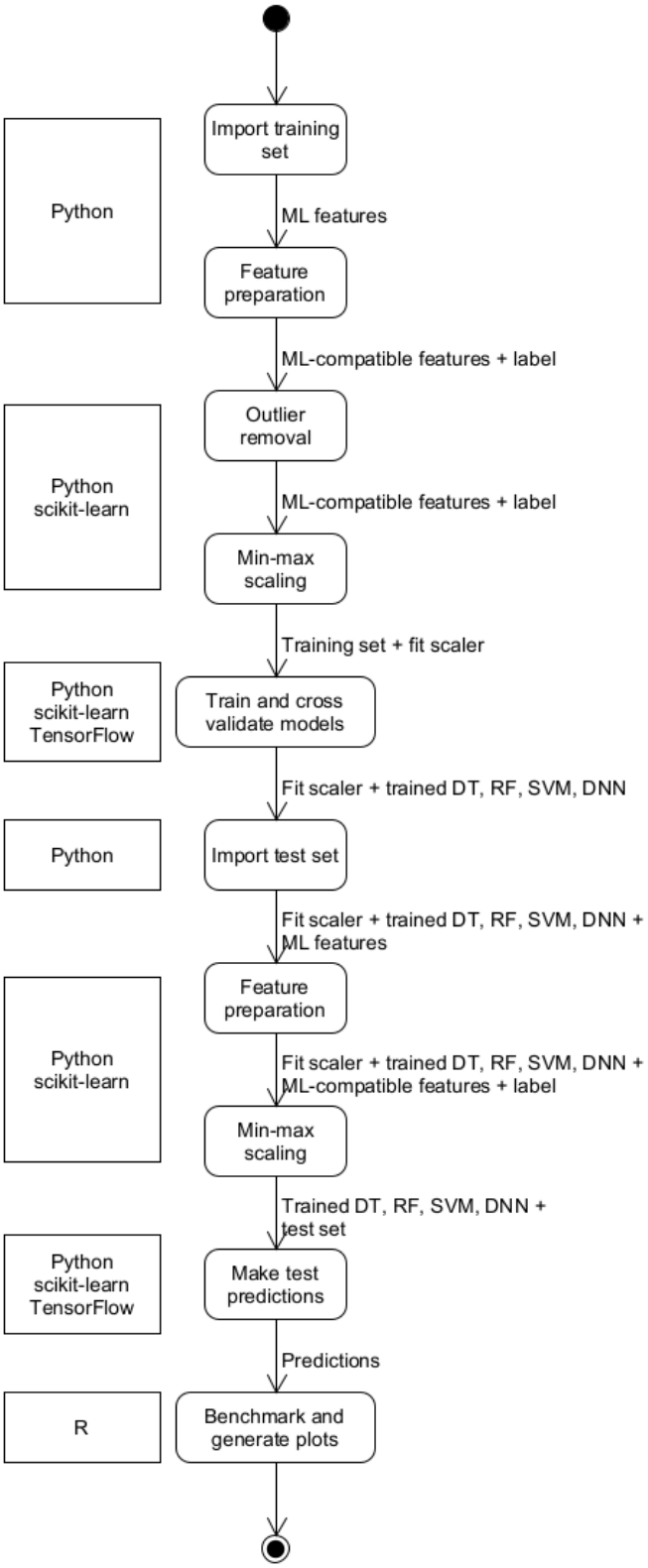


Figure 5.3: Machine learning module activity diagram. The machine learning module is primarily responsible for feature processing, model training, making predictions and result evaluation.

5.2.1.1 Bioinformatics Pipeline

The bioinformatics pipeline remains unchanged from its initial flow (Figure 5.4). However, as the deliverable software does not utilise RNA-seq data, and there is little overlap in technologies, it is now isolated from the rest of the tool.

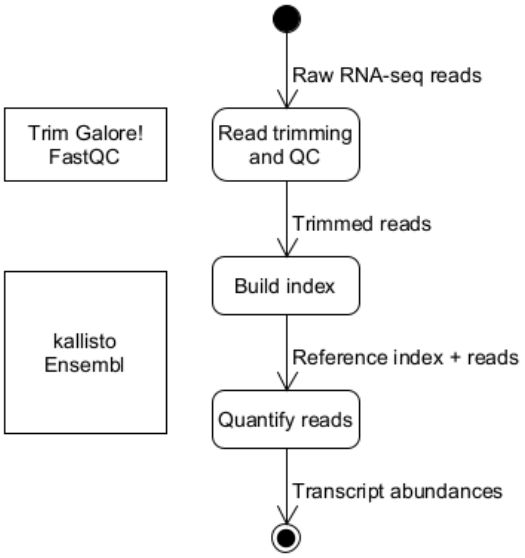


Figure 5.4: Bioinformatics pipeline. RNA-seq data is trimmed and checked for quality. An index is then constructed for the specified genome version and reads are quantified using kallisto. RNA-seq data is only required for model training and evaluation, meaning the end user software does not need to include this module.

5.2.2 Functionality Enhancements

5.2.2.1 Runtime Settings

A formal method for adjusting runtime settings via external configuration files is introduced to simplify testing flags and file paths. Settings had previously been managed through global constants, but this becomes more cumbersome when maintained between scripting languages.

Settings are applied through a JavaScript Object Notation (JSON) file (Ecma International, 2017), parsed at runtime by Python into a dictionary object and passed as command line arguments to Bash scripts. Using the jsonlite package (Ooms, 2014), the JSON file can also be read directly into R.

5.2.2.2 Output Caching

Intermediary outputs relative to transfection experiments are generated at each stage of feature extraction (Figure 5.5). When the `use_caching` setting is enabled, the tool

will attempt to load these files before the execution of the associated stage. This design pattern is useful in debugging, as each module can be tested in isolation against its expected behaviour.

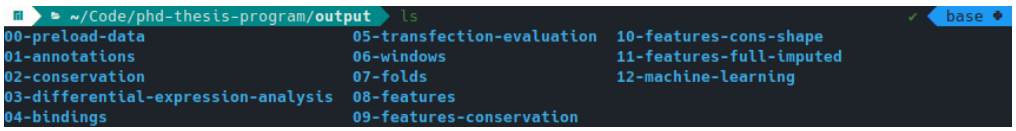


Figure 5.5: Cached output directory structure. The tool's output directory structure. Each directory stores intermediary outputs which allow the tool to resume from a desired stage.

5.2.2.3 Dynamic Experiment Loading

Experiment metadata is used at several processing stages, such as during differential expression analysis. Experiment data is supplied manually in Chapter 3 because the RNA-seq data used at that stage originated from the same source. As more data is integrated, a method of managing unique identifiers, samples accessions and varying numbers of biological replicates between batches becomes necessary. This information is also used by the caching system to create intermediary outputs as required.

Metadata files are dynamically generated using information provided to a JSON array at runtime.

```
[
    {
        "batch_id": "01-liu-HeLa",
        "mirna": "hsa-let-7c-5p",
        "transfected_samples": ["SRR8382192", "SRR8382193"],
        "control_samples": ["SRR8382242", "SRR8382243"]
    },
    {
        ...
    }
]
```

5.2.2.4 Local Annotation Parsing

Annotations were previously fetched using biomaRt queries in R (Section 3.2.4). As requirements grew, annotation queries increasingly fell outside biomaRt's recommended parameters. This created a limitation where only expressed transcripts were annotated to reduce overhead. The process therefore became per-sample, meaning the cache was invalidated if a new miRNA was transfected or the TPM filter was adjusted.

In addition to biomaRt, Ensembl publishes annotations in General Transfer Format (GTF) tabular files with each genome release. Parsing is necessary to convert these files into R tables due to the non-standard `attribute` fields, which contain different numbers of semicolon separated metadata. The `transcript_mane_select` had previously been pulled as part of the biomaRt query, but is now acquired from an additional third party GTF file published by NCBI. As the filter context of the original query is lost, a MANE version mapping ensures the correct genome version is used.

Using coordinate annotations from the GTF file, 3′ UTR and CDS sequences are extracted for all transcripts using the ensembldb (Rainer et al., 2019) R package. After mapping the local annotation database to the BSgenome.Hsapiens.NCBI.GRCh38 (The Bioconductor Dev Team, 2014) genome track, it is queried using GenomicFeatures (Lawrence et al., 2013) to extract sequences at given coordinates.

```
genome <- BSgenome.Hsapiens.NCBI.GRCh38
filter <- protein_coding_filter & chromosome_filter
utrs <- ensembldb::threeUTRsByTranscript(ensdb, filter = filter)

utr_seqs <- GenomicFeatures::extractTranscriptSeqs(genome, utrs)
```

Annotations parsed with this method are ultimately identical to those described in Table 3.2. CDS coordinates are also included using extraction methods parallel to that of the 3′ UTR to support the inclusion of CDS-related features.

5.2.2.5 Genome-wide Conservation Caching

Conservation scoring is a computationally expensive process because sequence scores must be generated, accessed and processed for a large number of targets per transfection. Previously, this created a bottleneck during conservation scoring. By decoupling per-genome and per-sample annotation logic, and removing annotation download caps (Section 5.2.2.4), conservation scoring can be partially cached.

It is not possible to know where target sites lie on a 3′ UTR before examining the transfected miRNA's seed sequence. However, the combination of expressed and unexpressed transcripts, and therefore their 3′ UTR sequences, do not change between samples. With annotations being expanded to include unexpressed transcripts, a conservation score can be pre-computed for each base in all the 3′ UTRs in preparation for when the seed target sequences are known. While this comes at the cost of up-front

computation, the cache is valid until the genome version is changed.

GenomicScores is used to compute a conservation score for every 3′ UTR sequence using the original method described in Section 3.2.8. A minor upgrade here is that an interval of 1 nt is used to enable per-base scoring, as opposed to the averaged window scoring used previously.

```
range <- GRanges(chromosome, IRanges(utr_start:utr_end, 1), strand)
```

The dimensions of a matrix cache for conservation scores across a genome can be defined as 'genome transcript count' × 'genome longest 3′ UTR length', or $84,419 \times 270,375$. An empty vector of this size has a data allocation of 125.9 GB in R, making it unfeasible to retain in memory. Utilising a non-uniform data structure, such as a jagged array, reduces this memory requirement. However, R's native support for non-uniform structures is limited. Truncating or filtering the longest 3′ UTRs is another option for reducing the memory footprint, as the majority fall shorter than 10,000 nt.

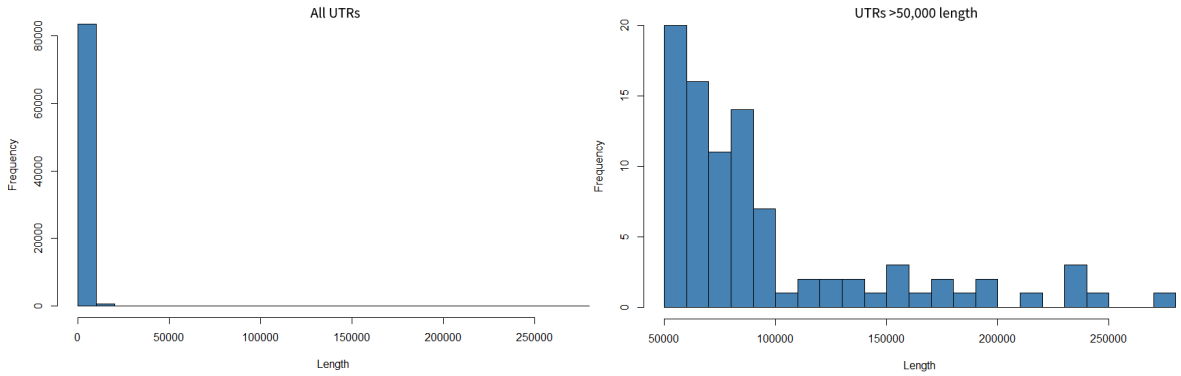


Figure 5.6: Histogram of 3′ UTR lengths. The vast majority of 3′ UTR lengths are shorter than 10% of the binned scale. (Left) All 3′ UTRs in *Homo sapiens*. (Right) Filtered 3′ UTRs to those longer than 50,000 nt.

A 100,000 nt base limit results in a 46.3% (58.3 GB) reduction in memory, with a loss of 0.00004% (36) transcripts (Table 5.1). At 12,500 nt, there is a loss of 0.00713% (602) transcripts for a resulting 8.5 GB matrix. Although this reduction is beneficial, a loss of transcripts could have a significant impact in samples where a truncated or removed transcript contains a true positive target. MTS also means a removed transcript could potentially host more than one target.

Table 5.1: Matrix size as a trade-off against transcript loss

3′ UTR Length Limit	Matrix Size (GB)	Transcripts Lost	Transcript Loss (%)
No limit	125.9	0	0
100,000	67.6	36	0.00004
50,000	31.5	104	0.00123
25,000	16.9	214	0.00254
12,500	8.5	602	0.00713
10,000	6.8	860	0.01019
7,500	5.1	1497	0.01773
5,000	3.4	3240	0.03839

Another approach to improve memory efficiency is to simplify the data structure by collapsing the scoring columns into a single sequence. Since R is used to extract conservation scores, yet Python governs the ML component, this can be achieved by writing the scores directly to file as they are generated, row-by-row. In this way, neither language holds more than a single row of the data structure in memory at a time.

Algorithm 2 Genome-wide per-base conservation score logging algorithm

Input: transcripts and full annotations

```
open file for writing
for all transcripts do
    log transcript ID
    log tab
    if 3′ UTR start or end coordinate is NA then
        log NA
    else
        compute and log tab-separated per-base scores
    end if
    start new line
end for
close file
```

Output: conservation scores tabular file

The cache is accessed once the seed target sequences and corresponding coordinates have been identified for a sample. The loss of columns is counteracted by parsing the

sequences of each row using Python's string-character iteration (Algorithm 3). With this method, the data structure is reduced to 218 MB at rest, expanding to 228 MB when loaded into a Python Pandas (Wes McKinney, 2010) `DataFrame`.

Algorithm 3 Conservation score fetching algorithm

Input: targets and annotations

```
for all targets do
    locate target using transcript ID
    for all abundant target sites do
        seed score = conservation[start_seed:end_seed]
        supplementary score = conservation[start_sup:end_sup]
        3′ flanking score = conservation[start_sup:start_sup + 30]
        5′ flanking score = conservation[start_seed - 31:start_seed - 1]
        calculate mean of each score, skip NAs
    end for
end for
store results
```

Output: per-target conservation scores

Owing to these enhancements, two new conservation tracks are added to support additional ML features: phastCons7way.UCSC.hg38 (Siepel et al., 2005) and phyloP100way.UCSC.hg38 (Pollard et al., 2010). Additionally, the flexibility in base extraction means conservation is also recorded for the 30 bases immediately flanking the seed, as conservation in these regions has been shown to impact target prediction accuracy (Ohler et al., 2004).

5.2.3 Dataset Construction

5.2.3.1 Additional Datasets

The 01-liu-HeLa dataset described in Table 3.1 is expanded using EBI Search (Madeira et al., 2022), and crawling the NCBI GEO (Edgar et al., 2002) and BioProject (Wheeler et al., 2007) databases with filter queries. Increasing the number of transfections available at this stage is important to facilitate ML, as data used to train models cannot be used to tweak algorithm parameters or produce test results without introducing biases. A full list of additional datasets is provided in Appendix A.

5.2.3.2 Data Quality Evaluation

A comparison of expression change distribution shows a degree of variation between transfection datasets, likely due to differences in lab techniques (Figure 5.7). In particular, 14-nam-Hela_hsa-miR-155-5p and 09-tam-U251_hsa-miR-137-3p have large deviations. 01-liu-HeLa remains the largest dataset source, with 25 individual transfections from a single cell line. This is substantially more than the second largest, containing seven transfections from four sources (datasets 11-14).

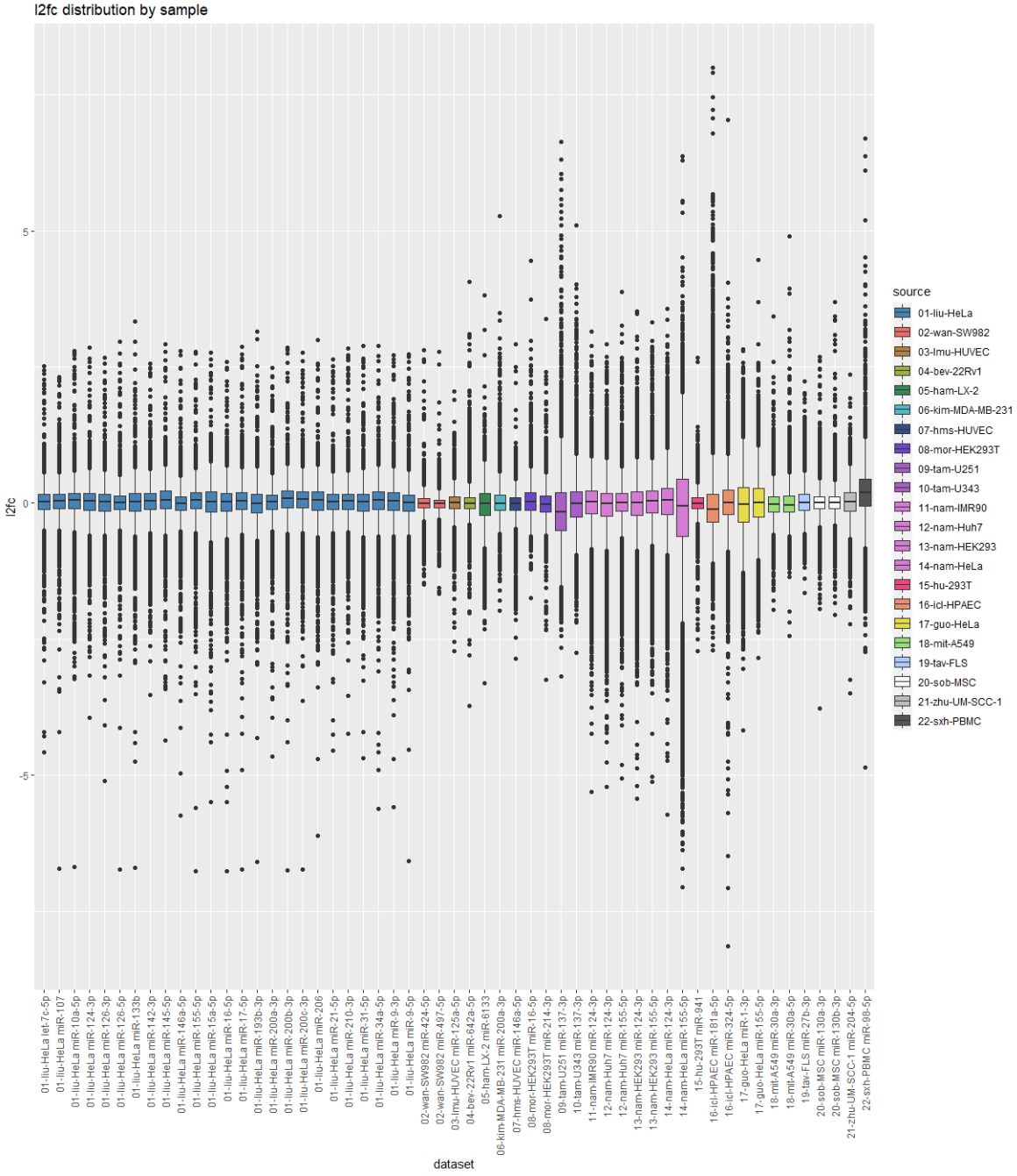


Figure 5.7: Filtered expression fold change variance between all datasets. The distribution of filtered log_2 expression fold change within each sample. There is more variation in samples outside of 01-liu-HeLa.

In Section 3.2.6.2, an MW test was performed to compare the expression fold change of targets and non-targets in order to filter samples with a p-value greater than 5.0×10^{-5}. Applying this filter to the additional datasets, three fail to show significant separation and are excluded from the ML datasets.

Table 5.2: Mann-Whitney p-values for each new transfection experiment

Transfection	MW p-value	Pass
02-wan-SW982_hsa-miR-424-5p	6.2×10^{-28}	✓
02-wan-SW982_hsa-miR-497-5p	6.9×10^{-22}	✓
03-lmu-HUVEC_hsa-miR-125a-5p	2.0×10^{-19}	✓
04-bev-22Rv1_hsa-miR-642a-5p	1.8×10^{-39}	✓
05-ham-LX-2_hsa-miR-6133	4.1×10^{-222}	✓
06-kim-MDA-MB-231_hsa-miR-200a-3p	2.3×10^{-1}	×
07-hms-HUVEC_hsa-miR-146a-5p	2.0×10^{-84}	✓
08-mor-HEK293T_hsa-miR-16-5p	6.6×10^{-168}	✓
08-mor-HEK293T_hsa-miR-214-3p	6.8×10^{-13}	✓
09-tam-U251_hsa-miR-137-3p	9.5×10^{-49}	✓
10-tam-U343_hsa-miR-137-3p	2.1×10^{-136}	✓
11-nam-IMR90_hsa-miR-124-3p	2.9×10^{-24}	✓
12-nam-Huh7_hsa-miR-124-3p	6.5×10^{-14}	✓
12-nam-Huh7_hsa-miR-155-5p	1.1×10^{-41}	✓
13-nam-HEK293_hsa-miR-124-3p	2.4×10^{-61}	✓
13-nam-HEK293_hsa-miR-155-5p	2.5×10^{-231}	✓
14-nam-HeLa_hsa-miR-124-3p	3.3×10^{-42}	✓
14-nam-HeLa_hsa-miR-155-5p	1.4×10^{-111}	✓
15-hu-293T_hsa-miR-941	4.2×10^{-2}	×
16-icl-HPAEC_hsa-miR-181a-5p	1.1×10^{-68}	✓
16-icl-HPAEC_hsa-miR-324-5p	9.9×10^{-1}	×
17-guo-HeLa_hsa-miR-1-3p	5.3×10^{-35}	✓
17-guo-HeLa_hsa-miR-155-5p	2.2×10^{-42}	✓
18-mit-A549_hsa-miR-30a-3p	1.4×10^{-163}	✓
18-mit-A549_hsa-miR-30a-5p	2.0×10^{-43}	✓
19-tav-FLS_hsa-miR-27b-3p	1.6×10^{-81}	✓
20-sob-MSC_hsa-miR-130a-3p	2.3×10^{-35}	✓
20-sob-MSC_hsa-miR-130b-3p	1.1×10^{-36}	✓
21-zhu-UM-SCC-1_hsa-miR-204-5p	2.5×10^{-156}	✓
22-sxh-PBMC_hsa-miR-98-5p	5.1×10^{-36}	✓

5.2.3.3 Train and Test Split

A pool of 55 individual miRNA transfections are available to form the training and test sets, though there are some limitations. miRNA transfection data has an implicit grouping based on the transfected miRNA sequence that should be maintained during splitting. Allowing targets from the same miRNA transfection to populate both sets will potentially allow test cases partial access to the label, particularly in the case of MTS (Figure 5.8). This may also be an issue for same-family miRNAs, as the model could become biased towards efficacy factors in these over-represented seed sequences.

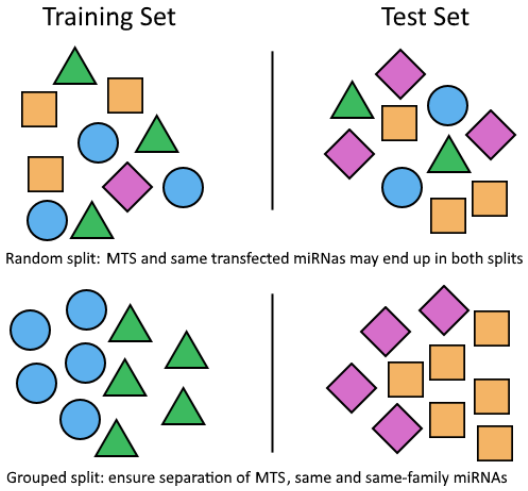


Figure 5.8: Grouped vs random data splits. In a random split, data is arbitrarily placed in each category. In a grouped split, related data is prevented from entering other sets.

The presence of mislabelled or noisy training data has a detrimental effect on a model's ability to learn (Brodley and Friedl, 1999). As 01-liu-HeLa has been tested extensively in this study, it is chosen to comprise the entirety of the training set. Although there is a concern that this data originates exclusively from HeLa, cell lines generally do not affect miRNA targeting (Nam et al., 2014). Nonetheless, to maintain fair test conditions during model evaluation and benchmarking, HeLa data is not used in testing.

The validation set is formed using the scikit-learn library's `StratifiedGroupKFold` cross-validation function. In k-fold cross-validation, the training set is subset k times while withholding a portion of the data for validation (Section 2.5.1.1). Grouped k-fold cross-validation is a derivative technique for ensuring groupings are retained across these splits, while stratification preserves class distributions.

In order to determine groupings for the train and test split, experiments are organised based on the 6mer seed sequence of the transfected miRNA.

Table 5.3: Transfected miRNAs with fully unique seeds

miRNA	Seed	Count*	Present In
hsa-miR-107	A**GCAGCA**U	1	01-liu-HeLa
hsa-miR-10a-5p	U**ACCCUG**U	1	01-liu-HeLa
hsa-miR-124-3p	U**AAGGCA**C	5	01-liu-HeLa, 11-nam-IMR90, 12-nam-Huh7, 13-nam-HEK293, 14-nam-HeLa
hsa-miR-126-3p	U**CGUACC**G	1	01-liu-HeLa
hsa-miR-126-5p	C**AUUAUU**A	1	01-liu-HeLa
hsa-miR-133b	U**UUGGUC**C	1	01-liu-HeLa
hsa-miR-142-3p	U**GUAGUG**U	1	01-liu-HeLa
hsa-miR-145-5p	G**UCCAGU**U	1	01-liu-HeLa
hsa-miR-146a-5p	U**GAGAAC**U	2	01-liu-HeLa, 07-hms-HUVEC
hsa-miR-155-5p	U**UAAUGC**U	5	01-liu-HeLa, 12-nam-Huh7, 13-nam-HEK293, 14-nam-HeLa, 17-guo-HeLa
hsa-miR-17-5p	C**AAAGUG**C	1	01-liu-HeLa
hsa-miR-193b-3p	A**ACUGGC**C	1	01-liu-HeLa
hsa-miR-200a-3p	U**AACACU**G	2 (1)	01-liu-HeLa, 06-kim-MDA-MB-231
hsa-miR-21-5p	U**AGCUUA**U	1	01-liu-HeLa
hsa-miR-210-3p	C**UGUGCG**U	1	01-liu-HeLa
hsa-miR-31-5p	A**GGCAAG**A	1	01-liu-HeLa
hsa-miR-34a-5p	U**GGCAGU**G	1	01-liu-HeLa
hsa-miR-9-3p	A**UAAAGC**U	1	01-liu-HeLa
hsa-miR-9-5p	U**CUUUGG**U	1	01-liu-HeLa
hsa-miR-125a-5p	U**CCCUGA**G	1	03-lmu-HUVEC
hsa-miR-642a-5p	G**UCCCUC**U	1	04-bev-22Rv1
hsa-miR-6133	U**GAGGGA**G	1	05-ham-LX-2
hsa-miR-214-3p	A**CAGCAG**G	1	08-mor-HEK293T
hsa-miR-137-3p	U**UAUUGC**U	2	09-tam-U251, 10-tam-U343
hsa-miR-941	C**ACCCGG**C	1 (0)	15-hu-293T
hsa-miR-181a-5p	A**ACAUUC**A	1	16-icl-HPAEC
hsa-miR-324-5p	C**GCAUCC**C	1 (0)	16-icl-HPAEC
hsa-miR-30a-3p	C**UUUCAG**U	1	18-mit-A549
hsa-miR-30a-5p	U**GUAAAC**A	1	18-mit-A549
hsa-miR-27b-3p	U**UCACAG**U	1	19-tav-FLS
hsa-miR-204-5p	U**UCCCUU**U	1	21-zhu-UM-SCC-1

* Parentheses account for those which did not pass the p-value filter.

Table 5.4: Transfected same-family miRNAs with the AGCAGC 6mer

miRNA	Seed	Count	Present In
hsa-miR-15a-5p	U**AGCAGC**A	1	01-liu-HeLa
hsa-miR-16-5p	U**AGCAGC**A	2	01-liu-HeLa, 08-mor-HEK293T
hsa-miR-424-5p	C**AGCAGC**A	1	02-wan-SW982
hsa-miR-497-5p	C**AGCAGC**A	1	02-wan-SW982

Table 5.5: Transfected same-family miRNAs with the AAUACU 6mer

miRNA	Seed	Count	Present In
hsa-miR-200b-3p	U**AAUACU**G	1	01-liu-HeLa
hsa-miR-200c-3p	U**AAUACU**G	1	01-liu-HeLa

Table 5.6: Transfected same-family miRNAs with the AGUGCA 6mer

miRNA	Seed	Count	Present In
hsa-miR-130a-3p	C**AGUGCA**A	1	20-sob-MSC
hsa-miR-130b-3p	C**AGUGCA**A	1	20-sob-MSC

Table 5.7: Transfected same-family miRNAs with the GGAAUG 6mer

miRNA	Seed	Count	Present In
hsa-miR-206	U**GGAAUG**U	1	01-liu-HeLa
hsa-miR-1-3p	U**GGAAUG**U	1	17-guo-HeLa

Table 5.8: Transfected same-family miRNAs with the GAGGUA 6mer

miRNA	Seed	Count	Present In
hsa-let-7c-5p	U**GAGGUA**G	1	01-liu-HeLa
hsa-miR-98-5p	U**GAGGUA**G	1	22-sxh-PBMC

In the 01-liu-HeLa dataset, there are two same-family overlaps between hsa-miR-15a-5p and hsa-miR-16-5p, and hsa-miR-200b-3p and hsa-miR-200c-3p. To prevent these seed sequences becoming over-represented in the training set, hsa-miR-15a-5p and hsa-miR-200b-3p are excluded as they had higher p-values, and therefore a less significant separation between targets and non-targets, in the transfection evaluation (Table 5.2).

There are 30 potential transfection experiments to build the test set, 13 after accounting for same-family miRNAs in the training set. Of the remainder, hsa-miR-137-3p is transfected twice, making one a redundant test. The 09-tam-U251 iteration of this transfection is therefore removed due to its inferior p-value, bringing the total to 12. Finally, hsa-miR-130a-3p and hsa-miR-130b-3p share a seed sequence, but are independent miRNAs. Since there is no feedback mechanism to the model that could introduce bias, and this sequence is not present in the training set, they are both included in the test set.

Table 5.9: Dataset splits after accounting for same-family miRNAs

Training Set	Test Set
01-liu-HeLa_hsa-let-7c-5p	03-lmu-HUVEC_hsa-miR-125a-5p
01-liu-HeLa_hsa-miR-107	04-bev-22Rv1_hsa-miR-642a-5p
01-liu-HeLa_hsa-miR-10a-5p	05-ham-LX-2_hsa-miR-6133
01-liu-HeLa_hsa-miR-126-3p	08-mor-HEK293T_hsa-miR-214-3p
01-liu-HeLa_hsa-miR-126-5p	10-tam-U343_hsa-miR-137-3p
01-liu-HeLa_hsa-miR-142-3p	16-icl-HPAEC_hsa-miR-181a-5p
01-liu-HeLa_hsa-miR-146a-5p	18-mit-A549_hsa-miR-30a-3p
01-liu-HeLa_hsa-miR-155-5p	18-mit-A549_hsa-miR-30a-5p
01-liu-HeLa_hsa-miR-16-5p	19-tav-FLS_hsa-miR-27b-3p
01-liu-HeLa_hsa-miR-17-5p	20-sob-MSC_hsa-miR-130a-3p
01-liu-HeLa_hsa-miR-193b-3p	20-sob-MSC_hsa-miR-130b-3p
01-liu-HeLa_hsa-miR-200a-3p	21-zhu-UM-SCC-1_hsa-miR-204-5p
01-liu-HeLa_hsa-miR-200c-3p	
01-liu-HeLa_hsa-miR-206	
01-liu-HeLa_hsa-miR-21-5p	
01-liu-HeLa_hsa-miR-210-3p	
01-liu-HeLa_hsa-miR-31-5p	
01-liu-HeLa_hsa-miR-34a-5p	
01-liu-HeLa_hsa-miR-9-3p	
01-liu-HeLa_hsa-miR-9-5p	
01-liu-HeLa_hsa-miR-124-3p	
01-liu-HeLa_hsa-miR-145-5p	
01-liu-HeLa_hsa-miR-133b	

TPM filtering was used to improve the quality of the training set by removing poorly

mapped reads (Section 3.2.6.1). Filtering is not applied to the test set in order to maintain fair test conditions.

5.2.4 Data Preparation

Following the examination of popular prediction tools (Section 2.5.2), and testing of isolated features (Section 3.3) and site accessibility measurement methods (Section 4.3), the feature set is expanded to take advantage of ML's heightened feature capacity. Unless explicitly discussed prior, new features are derivative of an existing feature and did not require noteworthy development.

5.2.4.1 Collinearity Reduction

Throughout this study, many target recognition features have been investigated. Consequently, there is some overlap between the information gained from these features, such as alternate site accessibility measures. This feature collinearity does not necessarily influence overall prediction accuracy (Mason and Perreault Jr, 1991); rather, it destabilises and complicates the model's ability to draw conclusions about the distinct influence of individual variables. This makes it problematic to regression approaches in particular (Midi et al., 2010). Furthermore, in both classification and regression problems, redundant features increase the computational complexity of the model.

Figure 5.9: Training set feature correlation heatmap. The correlation of each feature pair is compared in a hierarchically clustered heatmap, where a dark colour indicates higher correlation. A number of closely related feature clusters exist, notably those relating to evolutionary conservation and supplementary binding.

An examination of feature-to-feature correlation highlights three kinds of collinearity. In the first type, two features simply describe the same piece of information, possibly represented in a different form. These features are the easiest to remove, as a two-way correlation means little information is lost by removing either variable.

Table 5.10: Two-way collinear features

Collinear features	Description
`binding_site_pos` `rel_utr_pos`	The position of the miRNA target relative to the 3′ UTR, the relative position gives the result as a proportion rather than an nt count.
`au_content_6mer` `rnacofold_seed_mfe`	The presence of AU content indicates weaker seed stability, and predicted seed MFE is also a measure of this stability.
`perfect_pair_8` `seed_type`	Between 7mer-m8 and 8mer, the existence of a perfect pair at base 8 is encoded in `seed_type`.

Another observable type of collinearity is when one feature partially encodes another, for example, `site_abundance_7mer` and `site_abundance_7m8`. While information is still gained by maintaining both states, a third variable, `site_abundance_7a1`, leads to an overlap between the three features. In these cases, one of the features can be removed without a significant reduction in the information available to the model.

Table 5.11: Three-way collinear features

Collinear features	Description

`perfect_pair_1`
`au_1`
`seed_type`
	A:U is a type of perfect binding, and at base 1 this is described by the 7mer-a1 `seed_type`.

`au_content_5`
`au_content_5_weighted`
`au_content_sup`
	AU content to the 5′ side of the mRNA target is measured in three different ways; the supplementary and weighted methods favour bases close to the seed.

`site_abundance_7mer`
`site_abundance_7a1`
`site_abundance_7m8`
	7mer-a1 and 7mer-m8 are the two subtypes of 7mer.

`site_abundance_7cds`
`site_abundance_7a1cds`
`site_abundance_7m8cds`
	7mer-a1 and 7mer-m8 are the two subtypes of 7mer.

The third type of collinearity present in the matrix is overlap between feature clusters. Features revolving around sequence windows are expected to have some collinearity because there is a correlation between adjacent base sequences. However, parallel

features, such as alternate conservation scoring methods, were initially used to avoid ruling out potentially valuable features during development, but are instead now a source of unnecessary complexity.

Table 5.12: Clustered collinear features

Collinear features	Description
Conservation using the phast7, phast100 and phylo100 tracks.	The three tracks are all metrics for conservation, however phast100 and phylo100 are derived from the same 100 species, whereas phast7 is a 7 species subset.
Supplementary features using bases 12-17 and 9-20.	Both feature clusters describe subsets of the supplementary bases, of which 12-17 follows established literature on the importance of continuous pairings in the region, whereas 9-20 uses the definition found to be useful in this study for pair count features.

The two-way collinear relationships are solved by removing the lesser informative feature. For three-way correlations, preference is given to utilising as few variables as possible to fully encode the relationship. The exception to this rule is AU content, where both the weighted and supplementary iterations of the feature are maintained despite some redundancy. This preserves a popular feature in target prediction algorithms, while also retaining information regarding the supplementary portion.

The cluster of evolutionary conservation features are simplified by preferring comparisons over 100 species. phylo100 is selected because it is a derivative of phast100 which uses per-base scoring, as opposed to phast's window average approach. Finally, when considering supplementary binding features, base definition 12-17 is important because contiguous pairing of 3-5 bp between bases 12-14 are more effective (Grimson et al., 2007). Features revolving around continuous pairings and positioning therefore use the base 12-17 definition, whereas abstract supplementary features use the base 9-20 definition that was previously found effective in this study (Section 3.2.7.4).

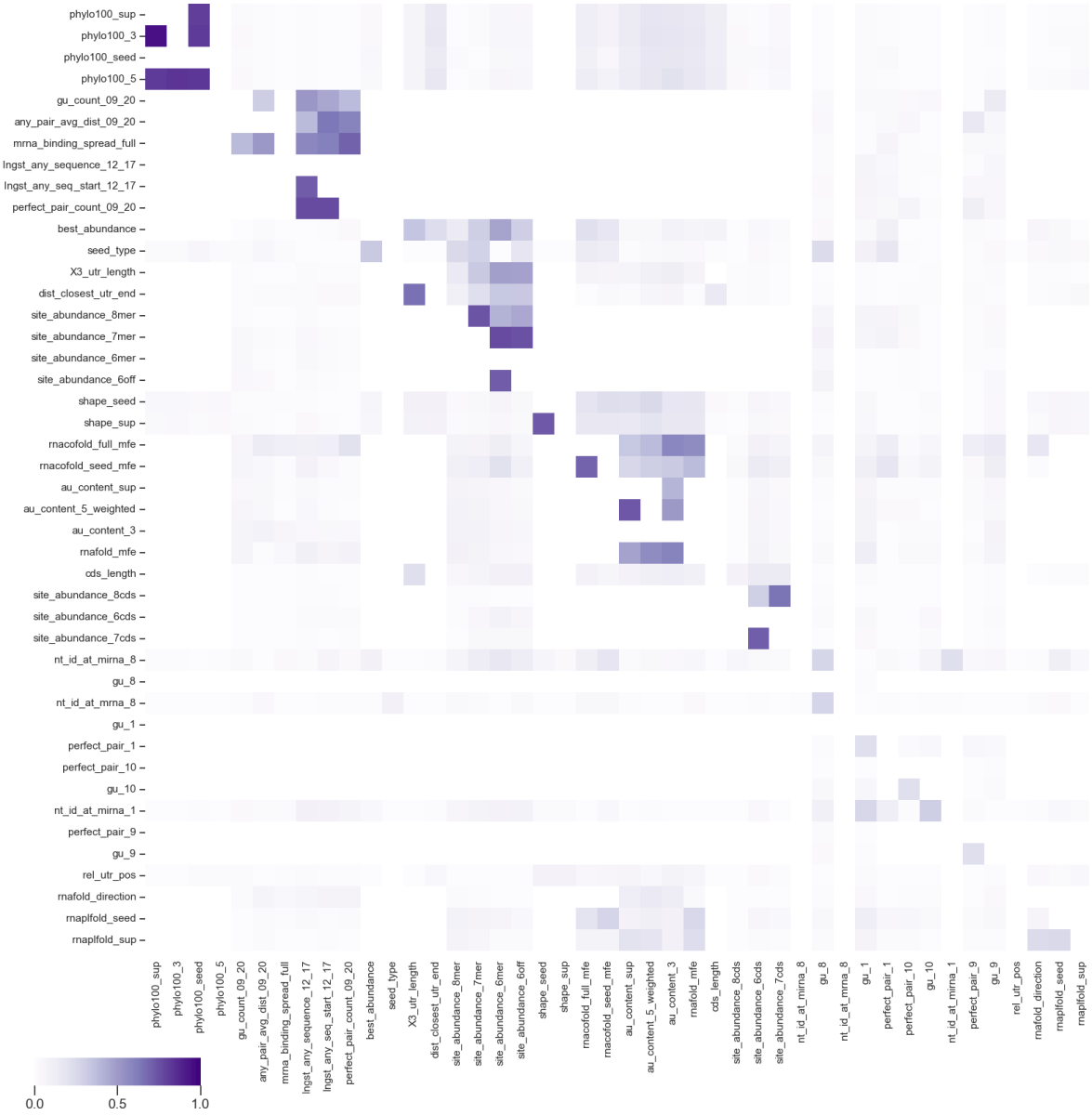


Figure 5.10: Training set feature correlation heatmap after reducing collinearity. The correlation of each feature pair is compared in a hierarchically clustered heatmap, where a dark colour indicates higher correlation. After the removal of closely related feature clusters, there is a reduction in the general feature correlation.

5.2.4.2 Common Features

Seed site implementations vary between TargetScan, miRanda-mirSVR, MirTarget and DIANA-microT (Section 2.5.3), though the use of 6mer, 7mer and 8mer is consistent. In this study, alternative definitions such as 6merα are implemented through the tracking of base matches in uncommon positions such as 1 and 9. G:U wobbles are also permitted in this way, provided the 6mer is perfectly matched. Similar to TargetScan, 6mer offset is recorded as an alternative seed type in MTS.

Table 5.13: Seed type usage in comparison to other tools

	6merα	6mer	6mer Offset	7mer-a1	7mer-m8	8mer	G:U Wobble
TargetScan 8.0		✓	✓*	✓	✓	✓	✓
miRanda-mirSVR 3.3a		✓	✓	✓	✓	✓	✓
MirTarget v4.0				✓	✓	✓	
DIANA-microT-CDS	✓	✓	✓	✓	✓	✓	✓
This tool	✓†	✓	✓*	✓	✓	✓	✓†

* 6mer offset is counted for MTS in the 3′ UTR only, not as a primary seed type.

† Tracked by features, not treated as a primary seed type.

All common features discussed in Section 2.5.4 are implemented.

Table 5.14: Common features in comparison to other tools

	Binding Stability	Supplementary Sites	Conservation	Site Accessibility	Local AU Content	Dist to 3′ UTR end	3′ UTR MTS	3′ UTR Length	CDS MTS
TargetScan 8.0	✓	✓	✓	✓	✓	✓	✓	✓	✓
miRanda-mirSVR 3.3a	✓	✓	✓	✓	✓	✓		✓	
MirTarget v4.0	✓		✓	✓		✓		✓	
DIANA-microT-CDS	✓	✓	✓	✓	✓	✓	✓		✓
This tool	✓	✓	✓	✓	✓	✓	✓	✓	✓

5.2.4.3 Full Feature List

A total of 44 features are extracted for use in ML.

Table 5.15: Features relating to seed definitions

Feature	Potential Information Gain	Basis
Seed binding type*	Core feature of miRNA:mRNA interaction	Lewis et al. (2003), Lewis et al. (2005), Grimson et al. (2007)
Base 1 perfect match	Encodes alternative seeds: 6merα, 7mer-m1 and 8mer-m1	Ellwanger et al. (2011)
Base 1 G:U	G:U wobbles may be tolerated in seeds, track 7mer-m1 wobble	Didiano and Hobert (2006)
Base 8 G:U	G:U wobbles may be tolerated in seeds, track 7mer-m8 wobble	Didiano and Hobert (2006)
Base 9 perfect match	Encodes alternative 9mer seed beginning at base 1	Maragkakis et al. (2009a)
Base 9 G:U	Track potential 9mer wobble	Didiano and Hobert (2006), Maragkakis et al. (2009a)
Base 10 perfect match	Encodes alternative 9mer seed beginning at base 2	Maragkakis et al. (2009a)
Base 10 G:U	Track potential 9mer wobble	Didiano and Hobert (2006), Maragkakis et al. (2009a)
miRNA base 1 A	Base context for 7mer and 8mer	Agarwal et al. (2015)
miRNA base 8 identity	Base context for 7mer and 8mer	Agarwal et al. (2015)
mRNA base 8 identity	Base context for 7mer and 8mer	Agarwal et al. (2015)

* 6mer, 7mer-a1, 7mer-m8, or 8mer.

Table 5.16: Features relating to binding stability

Feature	Potential Information Gain	Basis
RNAcofold seed MFE	Measure of seed stability	Garcia et al. (2011)
RNAcofold full binding MFE	Measure of binding stability	Garcia et al. (2011)

Table 5.17: Features relating to supplementary sites

Feature	Potential Information Gain	Basis
Base 12-17 longest any pair sequence	3-5 contiguous pairs beginning at bases 12-14 offer greater benefit	Grimson et al. (2007)
Base 12-17 longest any pair seq. start	3-5 contiguous pairs beginning at bases 12-14 offer greater benefit	Grimson et al. (2007)
Base 9-20 perfect pair count	9-20 encompasses a larger window of bases expanded from 12-17	Grimson et al. (2007)
Base 9-20 avg. distance between pairs	Gaps are tolerated in the supplementary portion to around 5 bases	Kiriakidou et al. (2004)
Base 9-20 G:U count	Frequent wobbles may destabilise bindings	Didiano and Hobert (2006)

Table 5.18: Features relating to conservation

Feature	Potential Information Gain	Basis
Base 1-8 PhyloP100	Seed conservation score	Lewis et al. (2005)
Base 9-20 PhyloP100	Supplementary conservation score	Lewis et al. (2005)
$3'$ flank PhyloP100	Flanking conservation score	Ohler et al. (2004)
$5'$ flank PhyloP100	Flanking conservation score	Ohler et al. (2004)

Table 5.19: Features relating to site accessibility

Feature	Potential Information Gain	Basis
RNAfold MFE of three-window*	Basic measure of seed accessibility	Kertesz et al. (2007)
RNAfold position of three-window*	Context on which side of the binding is least accessible	Kertesz et al. (2007)
RNAplfold unpairing seed	Predictive measure of sequence accessibility	Kertesz et al. (2007), Agarwal et al. (2015)
RNAplfold unpairing supplementary	Predictive measure of sequence accessibility	Kertesz et al. (2007), Agarwal et al. (2015)
AU content 3′ flank	Effective sites often reside within rich AU content	Grimson et al. (2007)
Weighted AU content 5′ flank	Effective sites often reside within rich AU content	Grimson et al. (2007)
AU content supplementary	Effective sites often reside within rich AU content	Grimson et al. (2007)
SHAPE-seq reactivity seed	Objective measure of sequence accessibility	Kertesz et al. (2007)
SHAPE-seq reactivity supplementary	Objective measure of sequence accessibility	Kertesz et al. (2007)

* Three windows are taken to the left, right, and centre of a binding. The strongest MFE predicted by RNAfold is taken as representative (Section 3.2.7.1).

Table 5.20: Features relating to MTS

Feature	Potential Information Gain	Basis
Best MTS*	Context on current target relative to others in the same transcript	Garcia et al. (2011)
3′ UTR 6mer MTS	Track 6mer MTS	Garcia et al. (2011)
3′ UTR 6mer offset MTS	Track 6mer offset MTS	Garcia et al. (2011)
3′ UTR 7mer MTS	Track 7mer MTS	Garcia et al. (2011)
3′ UTR 8mer MTS	Track 8mer MTS	Garcia et al. (2011)
Length of 3′ UTR	MTS is higher in long 3′ UTRs, but sites are less effective	Stark et al. (2005), Hausser et al. (2009)
CDS 6mer MTS	Track 6mer MTS	Reczko et al. (2012)
CDS 7mer MTS	Track 7mer MTS	Reczko et al. (2012)
CDS 8mer MTS	Track 8mer MTS	Reczko et al. (2012)
Length of CDS	CDS equivalent to 3′ UTR length	Agarwal et al. (2015)

* Refers to the target with the most bases paired and strongest MFE when there are multiple targets on the same 3′ UTR.

Table 5.21: Features relating to target positioning

Feature	Potential Information Gain	Basis
Relative binding position in 3′ UTR	Bindings are generally more effective close to 3′ UTR ends	Grimson et al. (2007)
Minimum distance to 3′ UTR end	Bindings are generally more effective close to 3′ UTR ends	Grimson et al. (2007)

Table 5.22: Features relating to the entire miRNA and complementary mRNA bases

Feature	Potential Information Gain	Basis
mRNA binding spread	Indicates the extent of gaps and bulges	Ding et al. (2016)

5.2.4.4 Categorical Data Encoding

With the exception of seed type, all extracted features are numerical or binary. To support non-binary categorical data, the four possible seed type values ‘6mer’, ‘7mer-a1’, ‘7mer-m8’ and ‘8mer’ are encoded as 0, 1, 2 and 3. This allows the model to

maintain four independent non-continuous numerical states for seed types.

5.2.4.5 Incomplete Data Imputation

Imputation is a technique for populating missing data with reasonable values. Imputation ranges from a simple replacement of *NA* values with 0, to predicting missing values using complex regression models. Alternative approaches to imputation include the removal of data points or features, or the use of binary flags for indicating missing fields. Due to the high rate of missing SHAPE-seq derived data (76%), these alternative methods are unlikely to be effective.

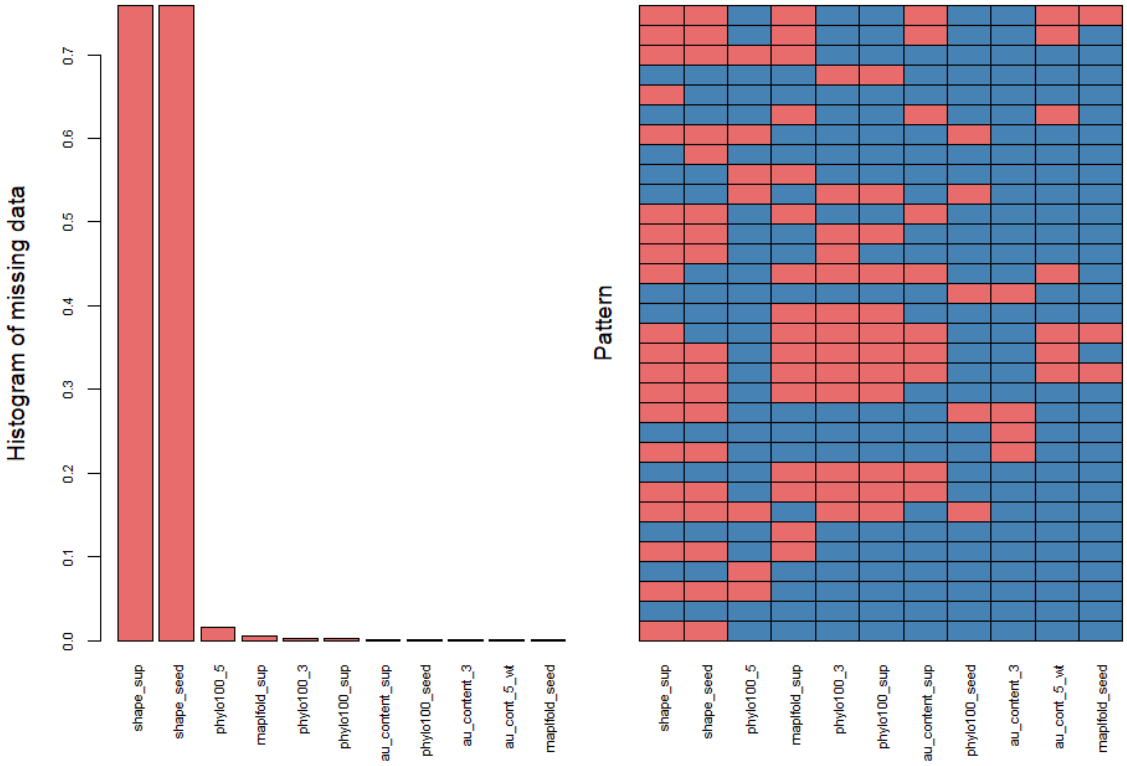
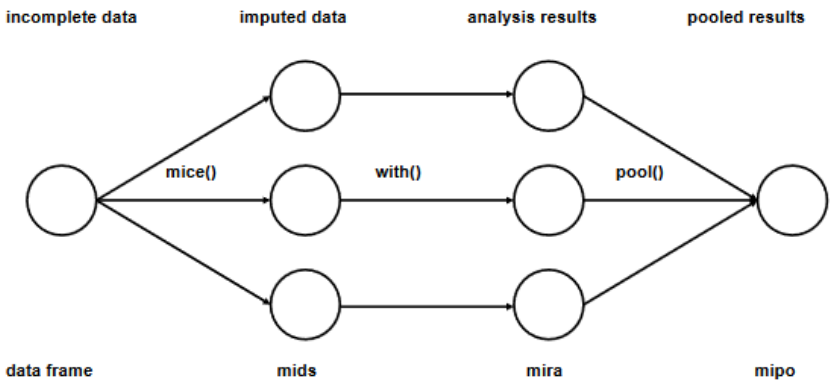


Figure 5.11: Training set missing data visualisations. (Left) Histogram of missing data by feature. Except for SHAPE-seq features, unavailability occurs at a rate of less than 0.5% in other features. (Right) Missing data patterns. The pattern of unavailability for both SHAPE-seq fields simultaneously is the most frequent, followed by full data availability. Generally, seed features are more available than supplementary features.

Multivariate imputation by chained equations (MICE) is a multiple imputation approach which generates and analyses numerous potential imputations, and pools them to produce an output (Van Buuren and Oudshoorn, 1999). The potential imputations are first seeded randomly, before a more sophisticated prediction is made using strongly correlated features. MICE is integrated in this study through the use of its R packages (Van Buuren and Groothuis-Oudshoorn, 2011), and internal predictions are made using the CART algorithm.



Source: Van Buuren and Groothuis-Oudshoorn (2011)

Figure 5.12: Multiple imputation steps in MICE. (Left to right) Taking incomplete data from the data frame, a user-defined number of multiply imputed dataset (**mids**) values are generated (3 pictured). The **mids** are analysed and assigned an interest coefficient to transform them into multiply imputed repeated analysis (**mira**) objects. Finally, they are recombined to produce a multiple imputed pooled outcomes (**mipo**) object.

The effectiveness of MICE is situational, but it has been to shown to be potentially effective even at a rate of 80% incomplete data (Poyatos et al., 2018). More specifically, it is capable of mostly preserving data patterns with up to 60% missing values, after which the rate of error increases exponentially, though the value does not entirely disappear (Penone et al., 2014).

An examination of distributions between a feature with a low (Figure 5.13) and high (Figure 5.14) missing rate highlights this relationship with the overall result. In the former, it is difficult to observe a distinction between the two distributions, though a small build-up of red pixels can be observed to the right of the mean line below 4. In the latter, MICE is able to somewhat reproduce the original distribution, although it is a vastly inferior result.

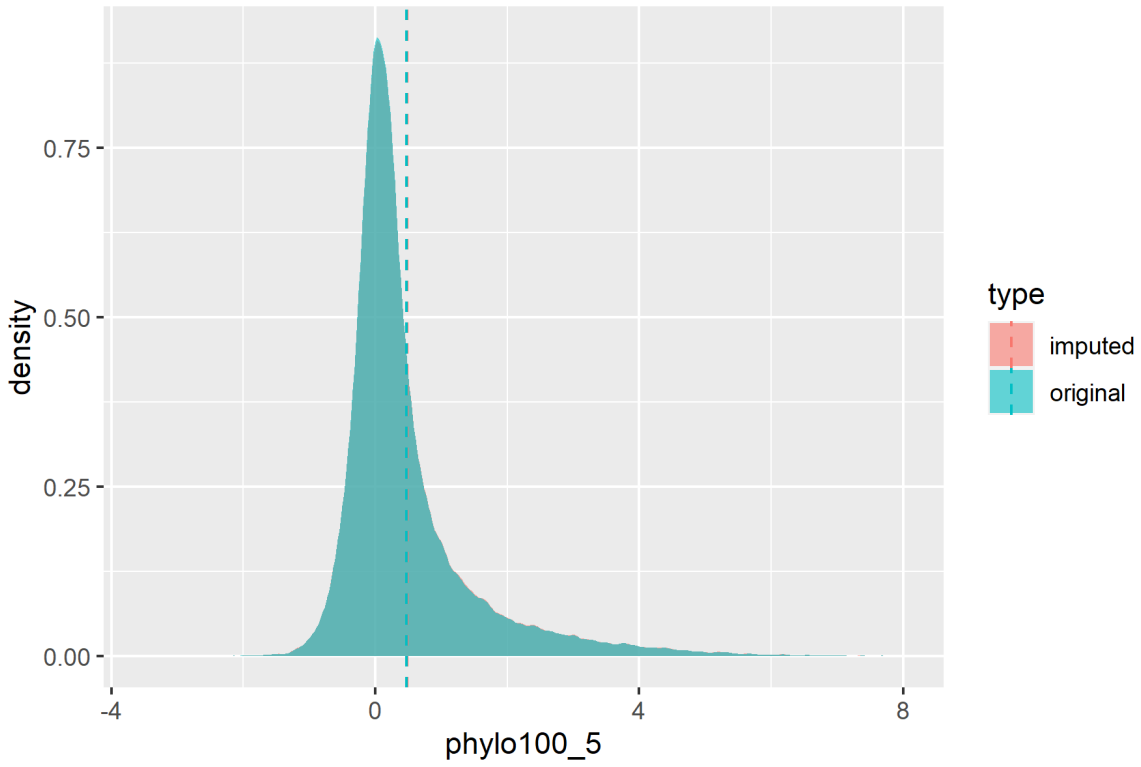


Figure 5.13: Comparison of original and MICE-imputed phylo100 5′ conservation scores. MICE can impute new values for the feature without changing the original distribution. A small deviation in imputed values can be seen to the right of the mean line.

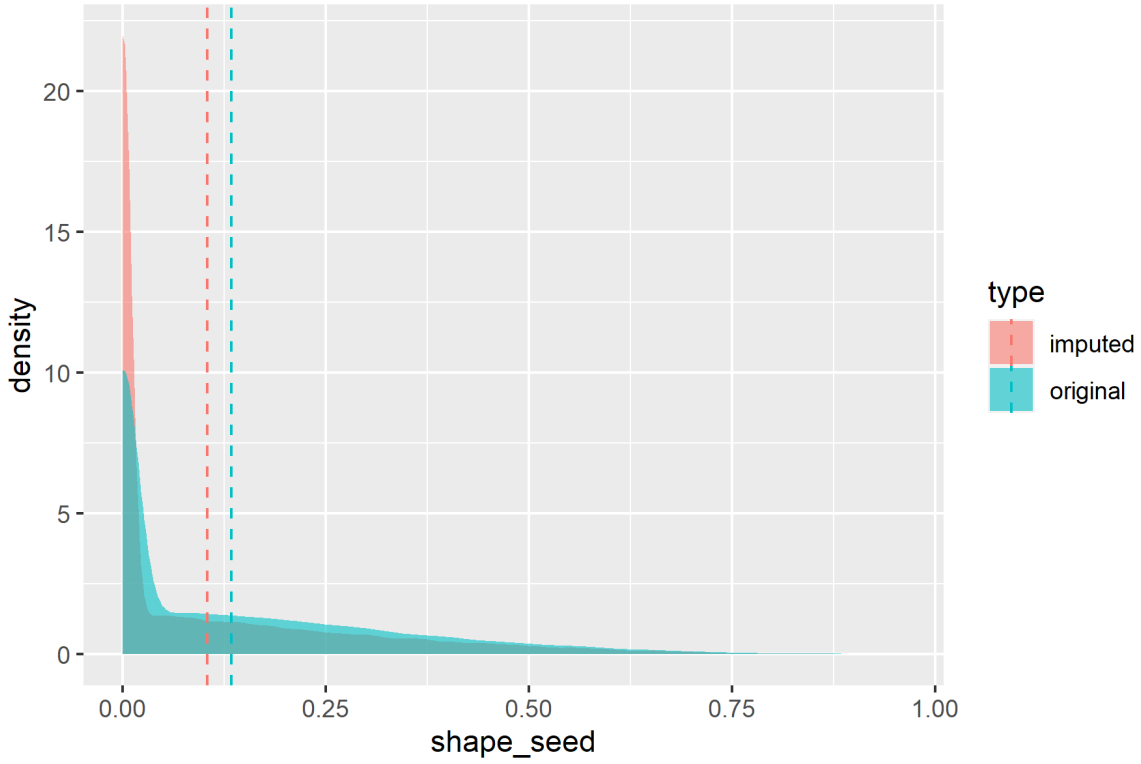


Figure 5.14: Comparison of original and MICE-imputed SHAPE-seq reactivity values. MICE can mostly reproduce the original distribution, with a large build-up around 0. As a result, the mean is lower in the imputed set.

In Figure 5.15, aggregate imputed SHAPE-seq values for seed targets in 01-liu-HeLa are plotted against the original values in Figure 4.9. The 10% most and least accessible values with imputation, represented by the 'Seed +10% imp' and 'Seed -10% imp' lines, are able to maintain a separation. However, the degree of separation is visually lower than when the categories contain only non-imputed values, as in the 'Seed -10% acc' and 'Seed -10% acc' lines. In both categories, the imputed line performs worse than the original (p-values: rounded to 1), though the total number of data points is increased by 75.8%. As a result, while the feature gains utility, its effectiveness is somewhat diminished.

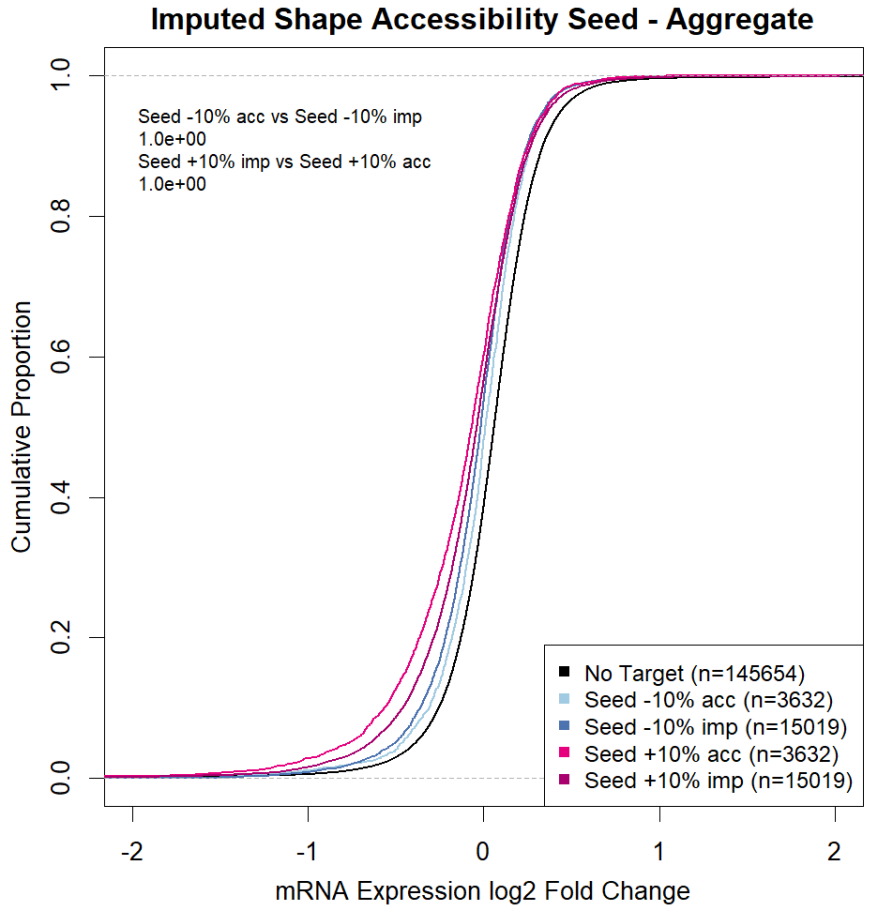


Figure 5.15: Comparison of aggregate SHAPE-seq efficacy before and after imputation. Imputation is compared by aggregating the 25 transfection experiments. The ability of SHAPE-seq data to distinguish high and low accessibility targets is reduced, but a separation still exists.

5.2.4.6 Scale Transformation

Transformation to a *log* scale allows features with values occupying a wide spectrum to be compacted without a loss of information. In the training data, there is a large difference in magnitude between the three features tracking 3′ UTR and CDS spans compared to the rest of the feature set (Figure 5.16).

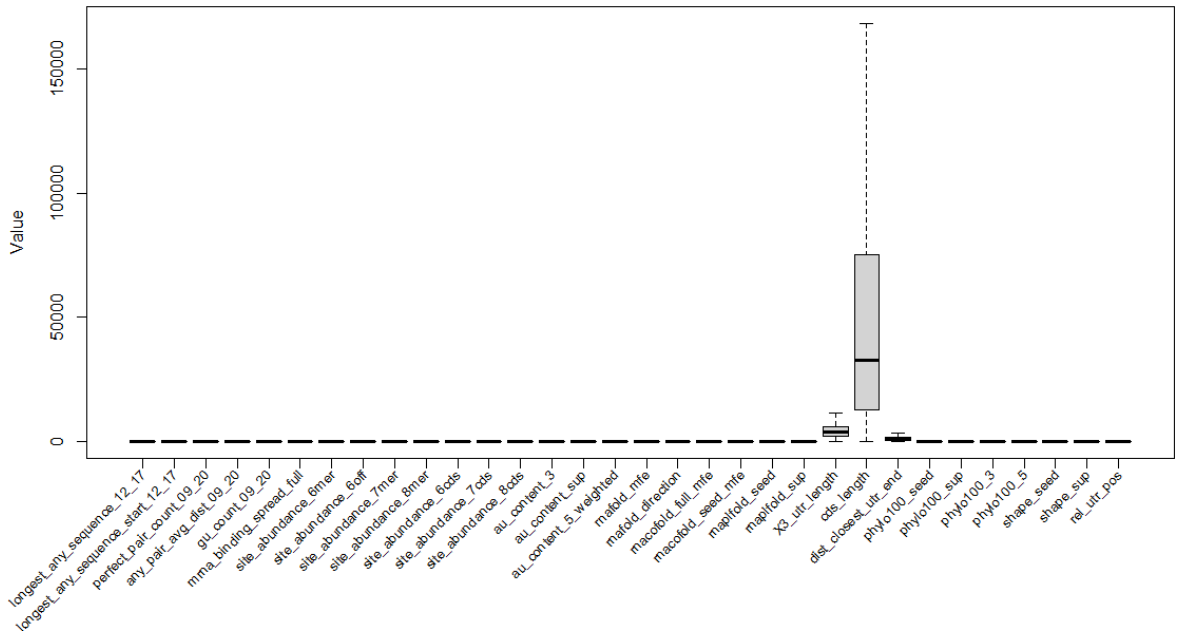


Figure 5.16: Feature scale prior to transformation. The natural value range has a high degree of variation due to features regarding sequence length causing scale distortions.

After transforming skewed features to log_{10} scale, the distribution is more visibly balanced. This standardisation prevents models that are not invariant to distance from becoming biased towards features of greater magnitude.

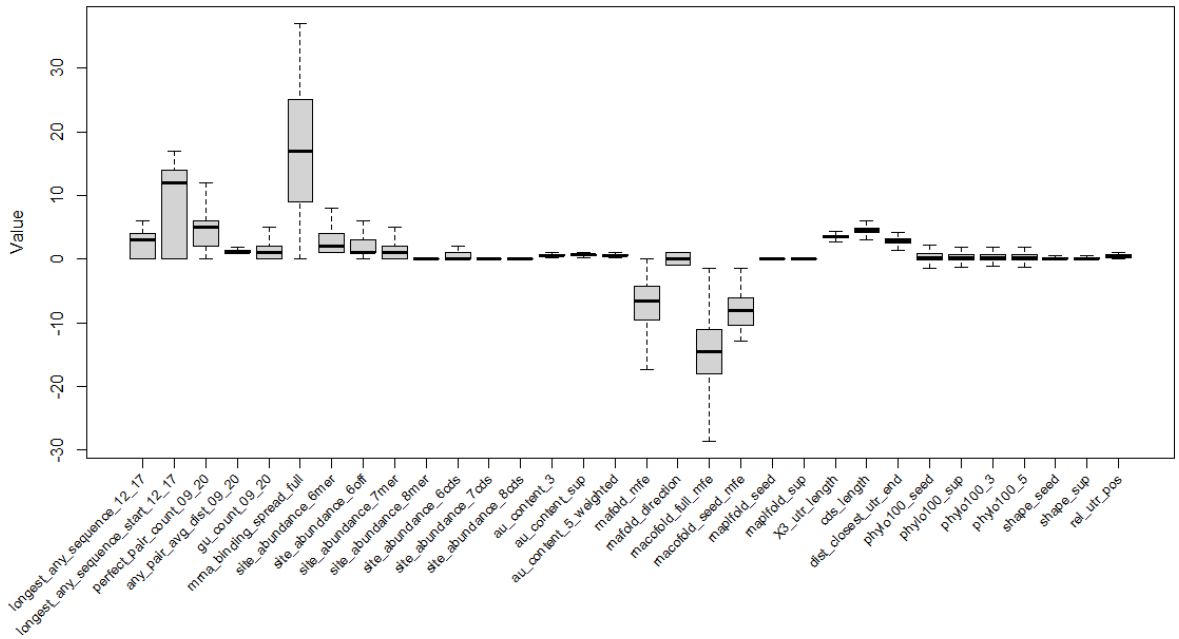


Figure 5.17: Feature scale after transformation. After transforming the length features to log_{10} scale, the values are distributed over a narrower range.

5.2.4.7 Outlier Removal

The removal of outliers is performed using scikit-learn's `LocalOutlierFactor` function, which compares the local deviation of a sample relative to its neighbours using k-nearest neighbours clustering. Removing outliers from the training set allows the model to better learn from the data by reducing the number of noise-influenced samples. As a result, the number of training samples is reduced from 150,197 to 149,480, or a 0.005% total reduction.

5.2.4.8 Normalisation

Normalisation is a technique for removing variations in magnitude between features. In tree-based algorithms such as DT and RF, normalisation is unnecessary as the decision nodes split based on isolated features, meaning the process is unaffected by feature scaling. Conversely, normalisation improves the performance of SVM and ANN models, due to their data point distance computation involving feature comparisons.

Normalisation is applied using scikit-learn's `MinMaxScaler` implementation of min-max scaling (Equation 5.2.1). Using the scalar, features are transformed so that all values fall between 0 and 1.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{5.2.1}$$

While the log_{10} scale transformation is applied only to UTR and CDS span features (Section 5.2.4.6), normalisation is applied to all features, causing the feature scale to become uniform.

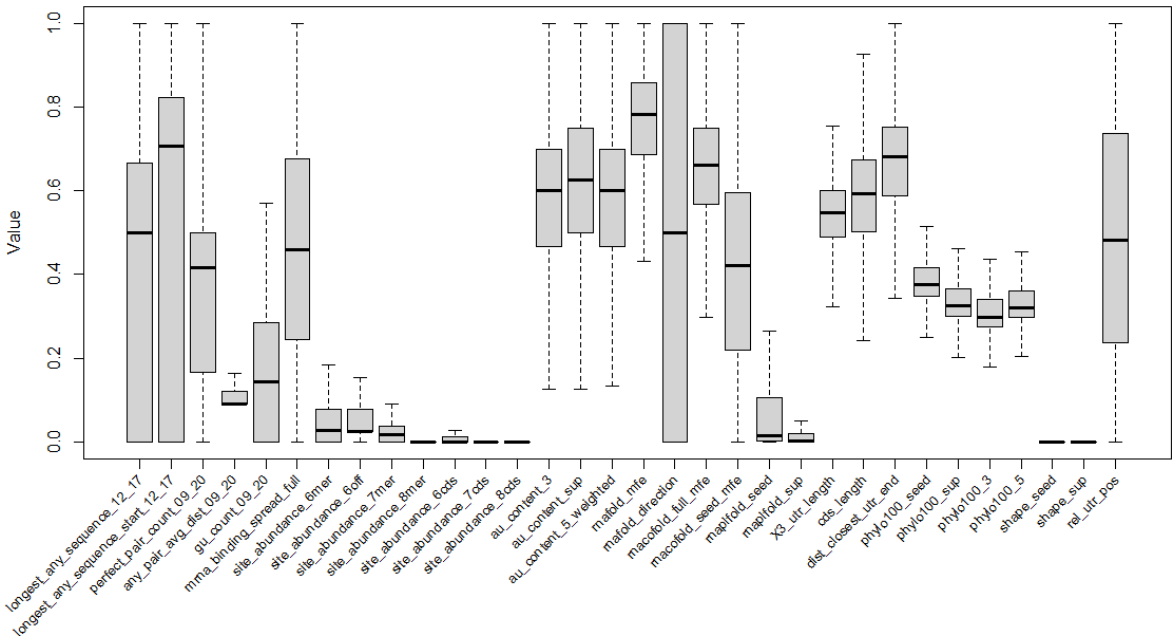


Figure 5.18: Feature scale after normalisation. With the application of min-max scaling, the range of feature values is normalised to 0-1.

5.2.5 Machine Learning

ML is performed by training a model on a set of data using an algorithm (Section 2.5.1). Four algorithms are compared to determine which offers the greatest performance:

- **DT:** used as a baseline for classifier performance, but not expected to perform well (Section 2.5.1.2).
- **RF:** popular for biological applications, one of four algorithms used by DIANA-microT (Section 2.5.1.3).
- **SVM:** generally popular in many fields, used by DIANA-microT, miRanda-mirSVR, MirTarget and previously TargetScan (Section 2.5.1.4).
- **DNN:** cutting-edge performance for some problems, but substantially harder to train, used by TargetScan and DIANA-microT (Section 2.5.1.5).

All ML algorithms are implemented using their associated scikit-learn packages (Pedregosa et al., 2011), except for DNN, which is instead provided by TensorFlow's Keras library (Abadi et al., 2015).

5.2.5.1 Class Label Assignment

The goal of classification is to categorise data through an assignment of class labels. In binary classification, there are only two class labels, one corresponding with positive

(P) and the other negative (N). In the context of this work, the expression log_2 fold change of a sample can be taken at a threshold to produce a binary categorisation of downregulated (P) and not downregulated (N).

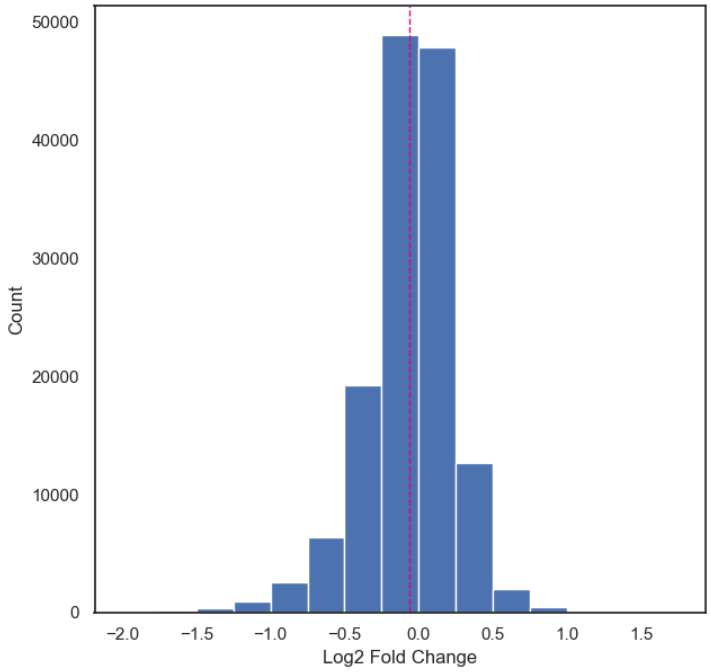


Figure 5.19: Distribution of training set log_2 fold change. A histogram breakdown of the expression log_2 fold change over the training set. The dotted line indicates the mean.

Using these labels, a real miRNA target is a true positive (TP) if predicted, or a false negative (FN) if not predicted. Conversely, a non-target is a false positive (FP) if predicted, or a true negative (TN) if not predicted. A naive metric for classifier performance can be computed using these statistics.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{P + N} \tag{5.2.2}$$

Trialling class labels at different downregulation thresholds highlights a potential class imbalance at lower values (Figure 5.20). When a class is underrepresented, the model is unable to learn effectively from the limited training samples and may incorrectly favour predictions towards the majority class (Guo et al., 2008). Although there is no commonly accepted definition for an imbalanced dataset, a severe imbalance is highly detrimental to model performance and will often necessitate re-balancing techniques such as sampling (Buda et al., 2018). In these cases, accuracy (Equation 5.2.2) is unable to offer effective scoring because the majority class becomes disproportionately

easy to classify.

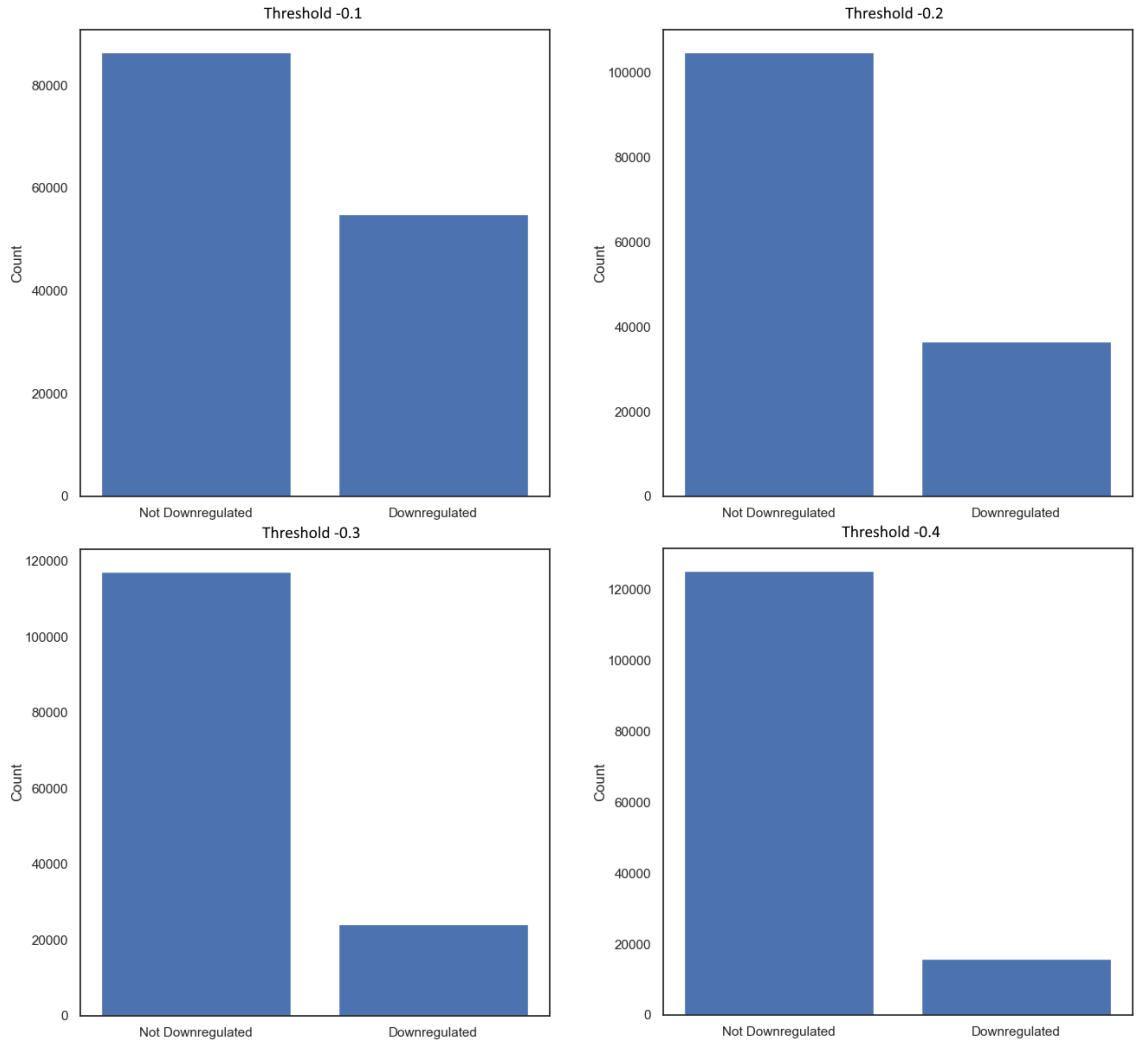


Figure 5.20: Training set class balance at different log_2 fold change thresholds. As the class label threshold becomes more negative, the class imbalance becomes larger.

Assigning a threshold too close to 0 introduces a disproportionately high rate of FP due to experimental noise and magnitude differences between samples, however it also increases the rate of P in general. On the other hand, assigning too negative of a threshold leads to class imbalance, although this stricter categorisation allows the model to better learn the pattern of TP at the cost of overall P. At -0.2 log_2 fold change, transcripts showing at least a 13% reduction in expression are considered downregulated, amounting to 1:2.5 ratio of downregulated to not downregulated transcripts. At -0.3, the 19% reduction leads to a ratio of 1:5. With this in mind, -0.2 log_2 fold change is selected as the class label threshold as it favours P, without being subjected to the high rate of diminishing returns that increases in line with lower values.

5.2.5.2 Evaluative Metrics

In miRNA target prediction, the identification of miRNA targets is preferred to explicitly ruling out non-targets. Depending on the use case, a researcher may have preferences as to how they wish to use the predicted targets. For research centred around a set of genes, it is generally more useful to examine all of a gene's potential targets. On the other hand, when examining targets of a particular miRNA, testing fewer but higher confidence predictions is more feasible. In either case, P is held in higher regard than N, while missed targets FN are more costly than incorrectly predicted targets FP. A method of balancing these use cases, common in prediction tools, is to produce many predictions and utilise confidence metrics to allow the user to fine-tune results. This may compensate for a lower prediction accuracy, provided the total number of predictions is sufficient.

A well-tuned predictor of miRNA targets therefore strikes a balance between recall (Equation 5.2.3) and precision (Equation 5.2.4); recognition of TP against P, and recognition of TP against all predictions, respectively. While both metrics favour P, recall prioritises a large number of P, whereas precision aims to prevent FP from entering the prediction set. In other words, a lower log_2 fold change threshold emphasises precision, whereas recall is favoured closer to 0.

$$recall = \frac{TP}{TP + FN} = \frac{TP}{P} \tag{5.2.3}$$

$$precision = \frac{TP}{TP + FP} \tag{5.2.4}$$

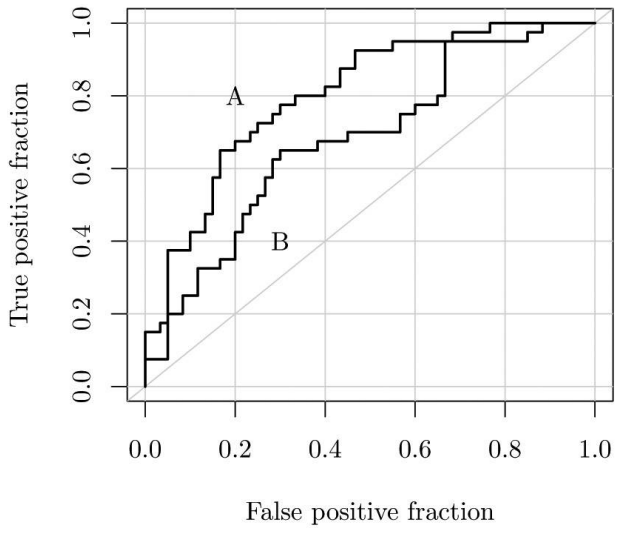
The F_1 score calculates the harmonic mean between recall and precision, allowing them to be represented under a single metric (Equation 5.2.5). This makes the F_1 score useful as an internal metric for optimising models towards identifying targets over non-targets. A downside of using F_1 score is that it does not account for TN, making it an incomplete metric to evaluate a trained classifier's overall performance with imbalanced classes.

$$F_1 = \frac{2TP}{2TP + FP + FN} = 2\left(\frac{precision * recall}{precision + recall}\right) \tag{5.2.5}$$

The area under the curve (AUC) represents the probability that a classifier will correctly determine a randomly selected positive instance from a randomly selected negative instance (Fawcett, 2006). The metric refers to the area under the receiver operating characteristic (ROC) curve (Figure 5.21), charted using the FP rate (FPR) and TP rate (TPR) (Equations 5.2.7, 5.2.6). In an ROC curve, the FPR and TPR make up the x-axis and y-axis respectively, and the physical area to the right of the line as a proportion of the chart (AUC), is indicative of performance. AUC is a particularly effective evaluative metric because it is invariant to class weight (Airola et al., 2008).

$$FPR = \frac{FP}{FP + TN} = \frac{FP}{N} \tag{5.2.6}$$

$$TPR = \frac{TP}{TP + FN} = \frac{TP}{P} \tag{5.2.7}$$



Source: Sachs (2017)

Figure 5.21: ROC chart. An example ROC chart plotting the performance of two models. Model A has a higher AUC than B, approximately 0.8 compared to 0.7. This means in 80% of samples, model A will correctly classify a positive instance over a negative one regardless of class balance. The diagonal line is used to indicate 50% AUC.

Finally, a confusion matrix provides an overview of how predicted labels are categorised compared to correct labels (Table 5.22). As the dataset is imbalanced towards negative

cases, this skew is likely to be represented in the confusion matrices, as it is more difficult for the model to identify TP without introducing FP; in stronger results, the positive labels will therefore be more precisely allocated.

		Predicted	
		P	N
Actual	P	TP	FP
	N	FN	TN

Figure 5.22: Layout of a confusion matrix. A matrix is formed by categorising P and N predictions according to where they were predicted compared to what the actual result is. An ideal confusion matrix shows a high number of values in the top left (TP) and bottom right (TN) boxes, while the top right (FP) and bottom left (FN) values should be as low as possible.

5.3 Results

The hyperparameters of DT, RF, SVM and DNN are tuned by optimising against the F_1 score. The fitted models are then compared using a one-sided two-sample MW test to determine significance at a p-value threshold of 0.05. After selecting the best model for use in miRsight, the trained model is compared with TargetScan, MirTarget and DIANA-microT over the 12 transfection test set.

5.3.1 Hyperparameter Tuning

Hyperparameters are tuned by supplying a list of values to scikit-learn's `GridSearchCV` function, which computes scoring metrics for each feature-value combination. The best fit for the classifier is determined by the mean F_1 score over five cross-validation folds. In each case, an F_1 score is generated for both the training and validation (internal test) sets. The use of validation is useful in identifying when the model becomes overfit, which is observable when the training line rises but the validation line falls. The hyperparameters and values used are unique to each model.

5.3.1.1 Decision Tree

In DT, the hyperparameters to be tuned are `class_weight` and a group of parameters centred around preventing overfitting. `class_weight` sets weights inverse to class proportions, thereby preventing the classifier from favouring overpopulated classes. The classifier exposes `max_depth`, `min_samples_split` and `min_samples_leaf` for overfitting prevention. `max_depth` simply limits the number of splits the tree can make, whereas

`min_samples_split` and `min_samples_leaf` limit tree growth according to conditions relating to the number of data points required to split.

```
DecisionTreeClassifier: {
    "class_weight": [None, "balanced"],
    "max_depth": range(1, 20),
}
```

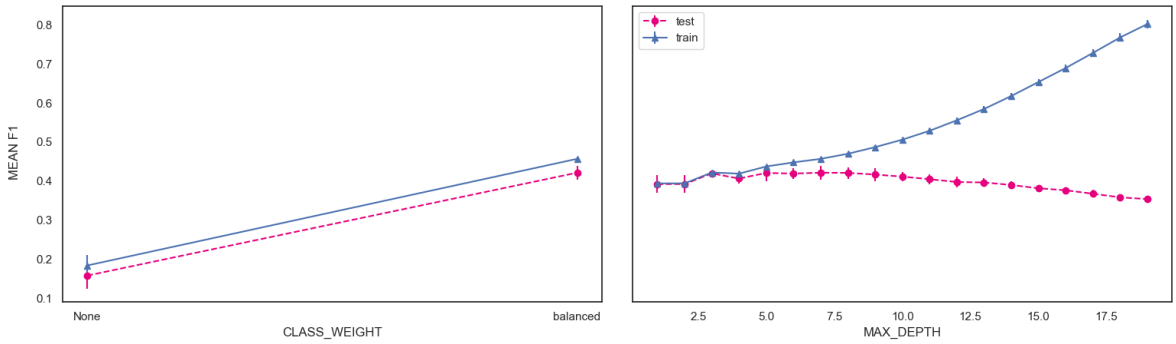


Figure 5.23: Hyperparameter tuning results for DT. The mean F_1 score of five cross-validation folds is computed using different hyperparameter values.
Optimal values - `class_weight`: balanced, `max_depth`: 7.

Re-balancing `class_weight` leads to a substantial improvement in DT's ability to classify samples, doubling the score in both the training and validation set. The `max_depth` chart shows underfitting at a depth of three and below, with an optimal range of fitting between four and seven. From eight, the classifier begins to overfit on the training set, steadily worsening validation performance.

5.3.1.2 Random Forest

RF shares the same base hyperparameters as DT, with an additional `max_features` parameter for placing restrictions on the number of features that may be considered when splitting. The accuracy of RF increases in line with the `n_estimators` (number of trees) in the ensemble, with diminishing returns which eventually level out performance gain (Oshiro et al., 2012).

```
RandomForestClassifier: {
    "class_weight": [None, "balanced"],
    "n_estimators": [200, 500],
    "max_features": [5, 8, 11, 14, 17, 20],
    "max_depth": [5, 8, 11, 14, 17, 20],
}
```

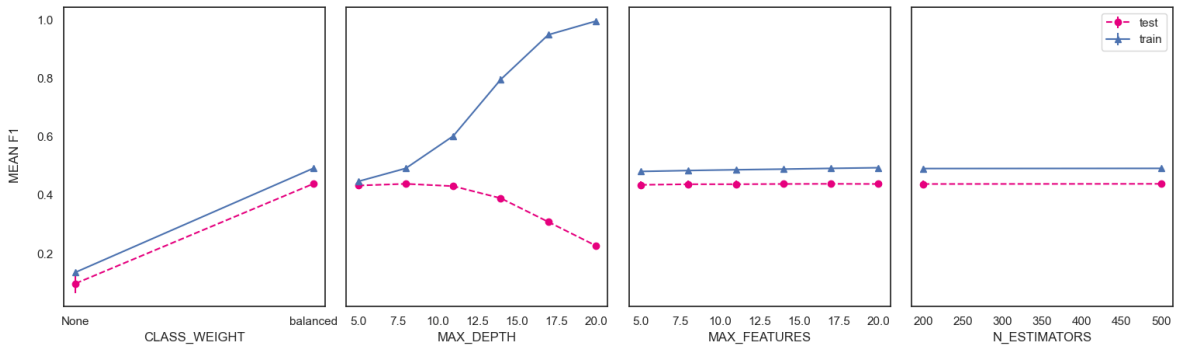


Figure 5.24: Hyperparameter tuning results for RF. The mean F_1 score of five cross-validation folds is computed using different hyperparameter values.
Optimal values - `class_weight`: balanced, `max_depth`: 8, `max_features`: 17, `n_estimators`: 500.

Similarly to DT, RF favours a balanced `class_weight` and low `max_depth`. In this instance, the grid search reduced overfitting using `max_depth`, though `max_features` could have been employed to obtain a similar result. Increasing `n_estimators` from 200 to 500 resulted in a trivial improvement in performance, as the initial performance gain had been reached before 200 trees. Since this number of trees did not substantially reduce prediction speed, it was not re-tested with a lower number.

5.3.1.3 Support Vector Machine

The high dimensionality of the dataset means that only a linear kernel is viable for SVM. As a result, the only hyperparameters to be tuned are `C` and `class_weight`. `C` manages regularisation and must be positive. At higher values, it pushes the hyperplane to favour categorisation over margin optimisation. Tuning `C` is best achieved using an exponential sequence of potential values, for example: 2.0×10^{-2}, 2.0×10^{-1}, 2×10, 2.0×10^{1}, 2.0×10^{2} (Hsu et al., 2003).

```
LinearSVC: {
    "class_weight": [None, "balanced"],
    "c": [0.002, 0.002, 0.02, 0.2, 2, 20]
}
```

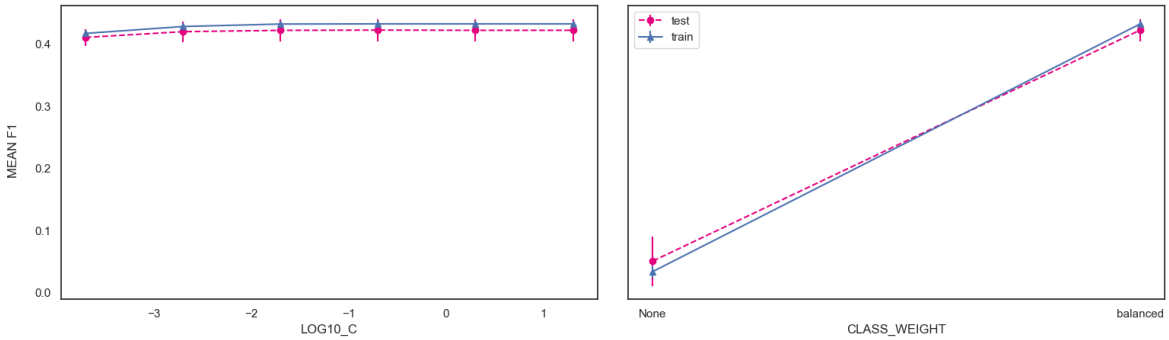


Figure 5.25: Hyperparameter tuning results for SVM. The mean F_1 score of five cross-validation folds is computed using different hyperparameter values.
Optimal values - `C`: 0.2, `class_weight`: balanced.

Using a balanced `class_weight` substantially improves SVM's performance (Figure 5.25), at a rate higher than DT and RF (Figures 5.23 and 5.24). This is likely because SVMs are highly sensitive to class imbalances (Tang et al., 2008). Increasing the `C` regularisation causes a marginal improvement in the F_1 score, mostly levelling out at 0.02 and reaching a peak value at 0.2.

5.3.1.4 Deep Neural Network

Testing on DT, RF and SVM indicates that performance is significantly increased by rebalancing class weights. Class imbalance has a detrimental effect on the accuracy of deep learning classifiers (Buda et al., 2018). Therefore, as DNN has a larger number of variable components compared to other models, the class weight is balanced in advance to reduce training computation.

The number of nodes in each layer is tuned using `unit1` for the first hidden layer, and `unit2` for the second. The learning process of ANNs is iterative; the optimiser continually adjusts the model over numerous passes through the dataset (epochs). `nb_epoch` manages the number of epochs used to train the classifier, while `batch_size` sets the number of samples required for an adjustment to take place. Finally, `learn_rate` refers to magnitude of these learning adjustments.

```
KerasClassifier: {
    "batch_size": [10, 20, 30, 40],
    "nb_epoch": [50, 200, 350],
    "unit1": [8, 16, 32, 64],
    "unit2": [8, 16, 32, 64],
    "learn_rate": [0.0001, 0.001, 0.01]
}
```

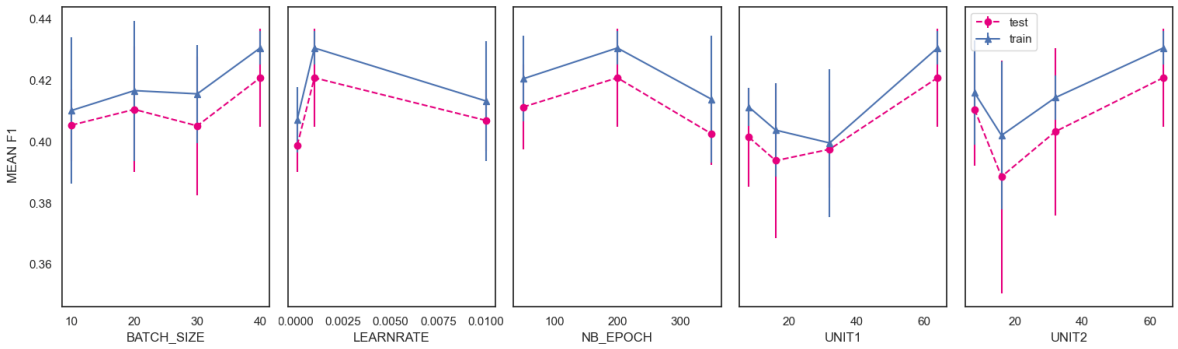


Figure 5.26: Hyperparameter tuning results for DNN. The mean F_1 score of five cross-validation folds is computed using different hyperparameter values.
Optimal values - `batch_size`: 40, `learnrate`: 0.001, `nb_epoch`: 200, `unit1`: 64, `unit2`: 64.

Figure 5.26 shows that the hyperparameter tuning identified 64 as the optimal number of nodes in `unit1` and `unit2`, meaning both hidden layers should use the same layout. Balancing DNN hyperparameters is difficult, as changes tend to destabilise the optimal settings of other hyperparameters, particularly between `batch_size` and `learnrate`. Figure 5.27 illustrates the constructed DNN model. Notably, two hidden layers are used with a 50% dropout layer placed between them to reduce overfitting (Section 2.5.1.5).

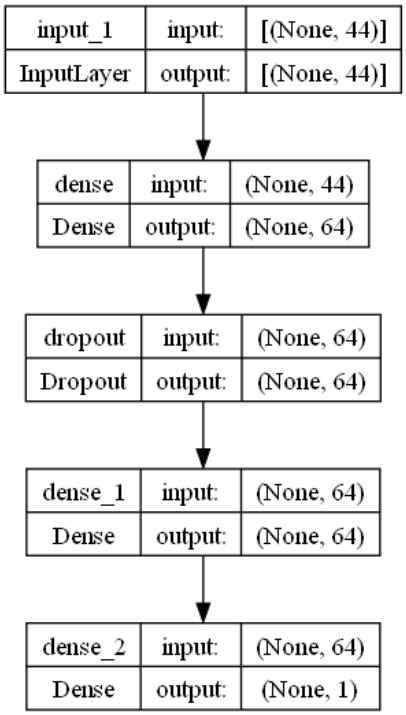


Figure 5.27: Tuned DNN layers. (Top to bottom) The topmost layer is the input layer, sized according to the number of features. At the bottom is the output layer, which uses a sigmoid function to enforce a binary prediction. The hidden layers are comprised of two dense layers set according to hyperparameter tuning, and a 50% dropout.

5.3.2 Classifier Performance

In a critical difference (CD) diagram (Demšar, 2006), the rank of each classifier is averaged over several datasets. The CD refers to the range at which a classifier's average rank must deviate to be meaningfully separated. In this instance, the CD diagrams are created using AUC and F_1 scores over the 12 experiment test set.

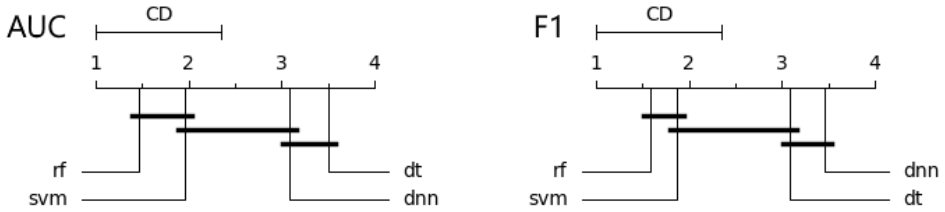


Figure 5.28: CD diagram for AUC and F_1 rankings on the test set. Average rankings of each classifier over the 12 miRNA transfections in the test set. The ranking is the same under both metrics.

Under both metrics, RF has the highest performance and is critically different from both DNN and DT. However, it is not critically different from SVM. DNN ranks above DT in AUC, but below DT in F_1. In terms of AUC, RF ranks first in 8 of 12 tests compared to 5 of 12 for SVM, with one result being a shared first place tie. The mean AUC scores are 0.599 for RF and 0.592 for SVM.

An examination of ROC charts shows a high level of consistency across experiments (Figure 5.29). The most significant variation is in miR-125a-5p, where there is a large line separation between all four classifiers. Generally, for DNN and DT to score highly in comparison to RF and SVM, the latter classifiers under-perform rather than the former over-performing. Examples of this are miR-181-5p (AUC 0.567) and miR-130a-3p (AUC 0.557), where the AUC scores are lower than average.

RF and SVM are consistently the highest scoring classifiers under both metrics, with RF ranking higher overall. The trained RF classifier is therefore selected as the foundation of the miRsight target prediction algorithm.

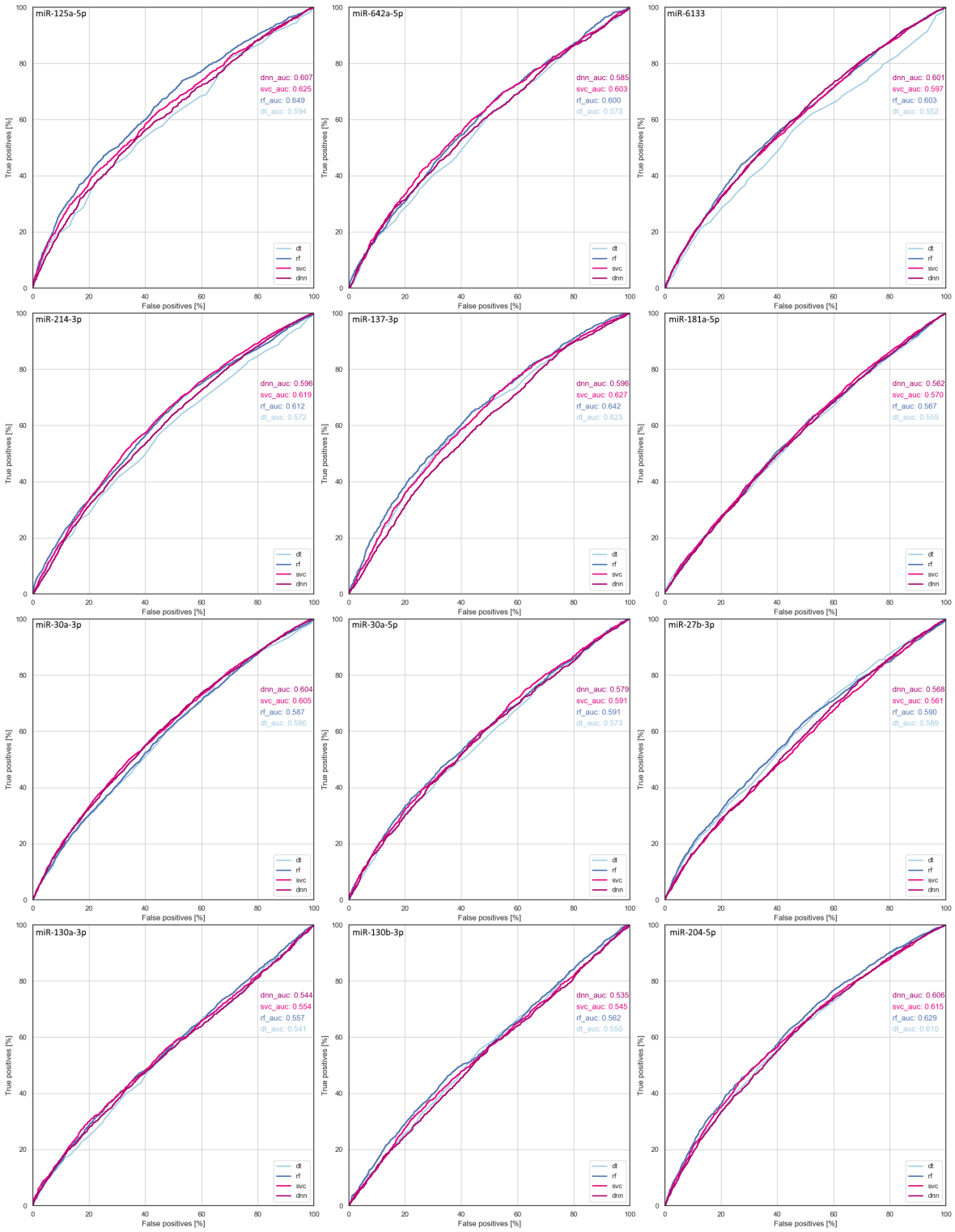


Figure 5.29: ROC chart collage for the test set. Each of the 12 miRNA transfections which comprise the test set are plotted on an independent ROC chart. Where a line shifts further to the left, the area under the curve increases. The AUC scores of each classifier are listed on the right side of each chart.

5.3.3 Benchmarking

Predictions output by miRsight are compared against TargetScan, miRDB's MirTarget and DIANA-microT. These tools were previously identified as the most popular target

prediction tools to receive active maintenance (Section 2.5.2). TargetScan results are obtained by supplying the tool with the same list of unfiltered transcripts used in this study. As MirTarget and DIANA-microT are web-only tools, pre-computed predictions for *Homo sapiens* are obtained directly from the data download pages on their respective websites.

Table 5.23: Prediction tool data summary

Tool Version	Prediction Algorithm	Release Date
TargetScan 8.0	TargetScan 7.2	September 2021
miRDB 6.0	MirTarget V4	June 2019
DIANA-microT 2023	DIANA-microT-CDS	April 2023

As discussed, target prediction tools cater to multiple use cases, generally favouring either a large number of predictions, or a smaller number of more confident results. Benchmarking is therefore performed with respect to all targets, followed by the top 500, 300 and 100 predictions, ranked by confidence score. The results are first visualised using an aggregate cumulative plot for each prediction threshold, then elaborated in a heatmap comparison of p-values from individual experiments.

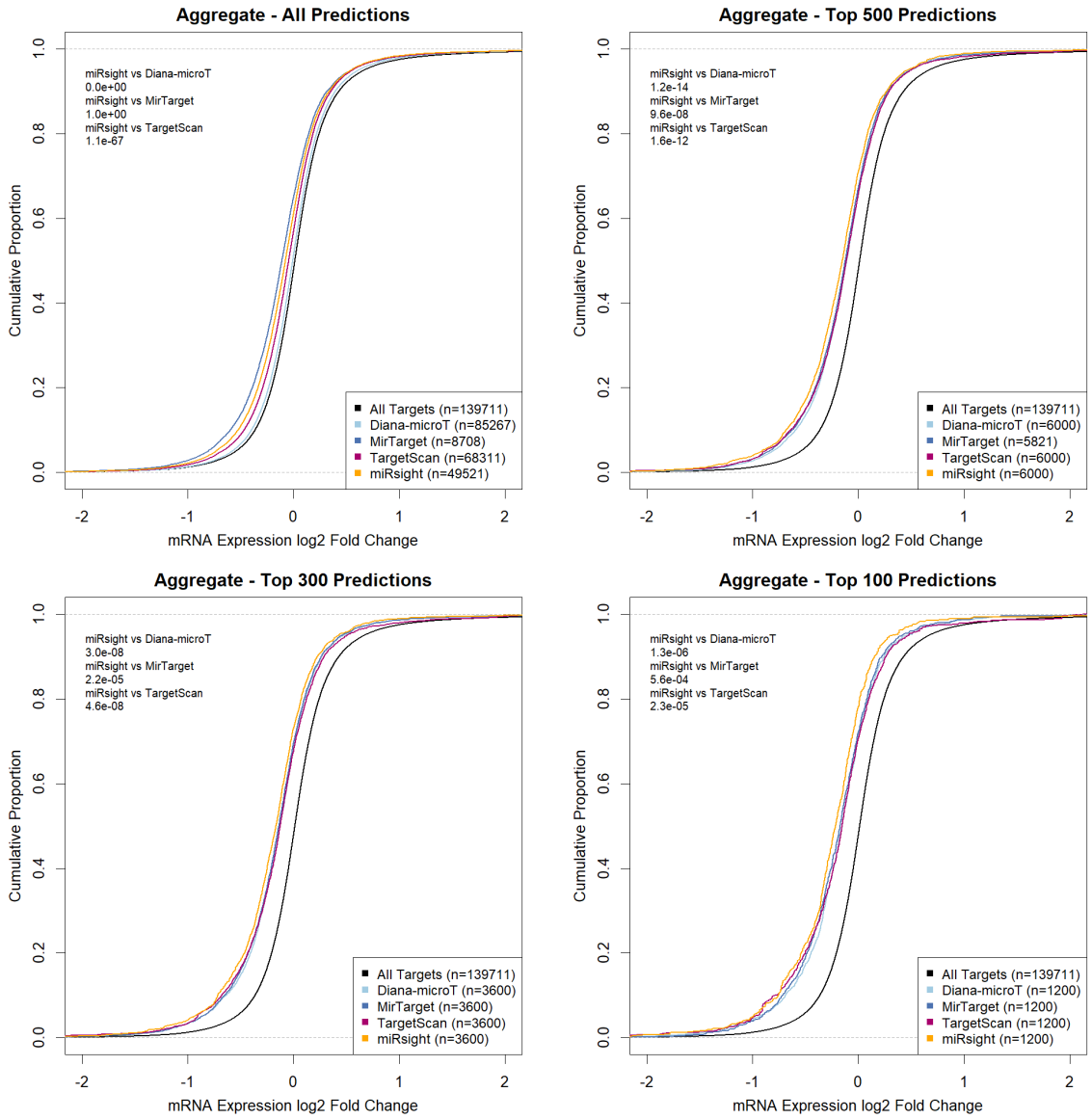


Figure 5.30: Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT. Prediction accuracy is compared by aggregating the 12 test transfection experiments. In each prediction threshold chart, the number of predicted targets are 12 times higher due to aggregation, while the control line simply plots all 139,711 log_2 values output by Sleuth. (Top left) All predictions. (Top right) Top 500 predictions per experiment. (Bottom left) Top 300 predictions per experiment. (Bottom right) Top 100 predictions per experiment. In 1 of the 12 transfections, MirTarget does not produce the 500 total predictions required for the top 500 category.

Fewer overall predictions are made by miRsight (49,521) than DIANA-microT (85,267) and TargetScan (68,311), but substantially more than MirTarget (8,708). MirTarget's low prediction output may be due to its stricter method of seed identification compared to other tools (Section 2.5.3). It should be noted that the 'All Predictions' plot does not account for a tool's confidence score, and therefore favours tools that make less overall predictions, particularly in the case of MirTarget's significantly lower number of predictions.

miRsight has the highest degree of separation in all three aggregate breakdowns, with the clearest line separation in the top 500 category. In this category, the p-values are significant against DIANA-microT (1.2×10^{-14}), MirTarget (9.6×10^{-8}) and TargetScan (1.6×10^{-12}). MirTarget places second in all three categories (p-values: 9.6×10^{-8}, 2.2×10^{-5} and 5.6×10^{-4}), followed by TargetScan (p-values: 1.6×10^{-12}, 4.6×10^{-8} and 2.3×10^{-5}) and DIANA-microT (p-values: 1.2×10^{-14}, 3.0×10^{-8} and 1.3×10^{-6}). DIANA-microT places higher than TargetScan in the top 300 prediction category.

miRsight is most effective in identifying true targets between -0.75 and 0. In all three plots, the line separation is most clearly visible in this range. This sensitivity to log_2 fold change values close to 0 is likely to be a result of setting the classification label at -0.2 instead of a more extreme value.

A heatmap summary of p-values is constructed from the 12 individual cumulative plots which comprise these aggregated results (Figure 5.31). A further breakdown is provided in Appendix B, where each individual cumulative plots is presented alongside miRsight's confusion matrix.

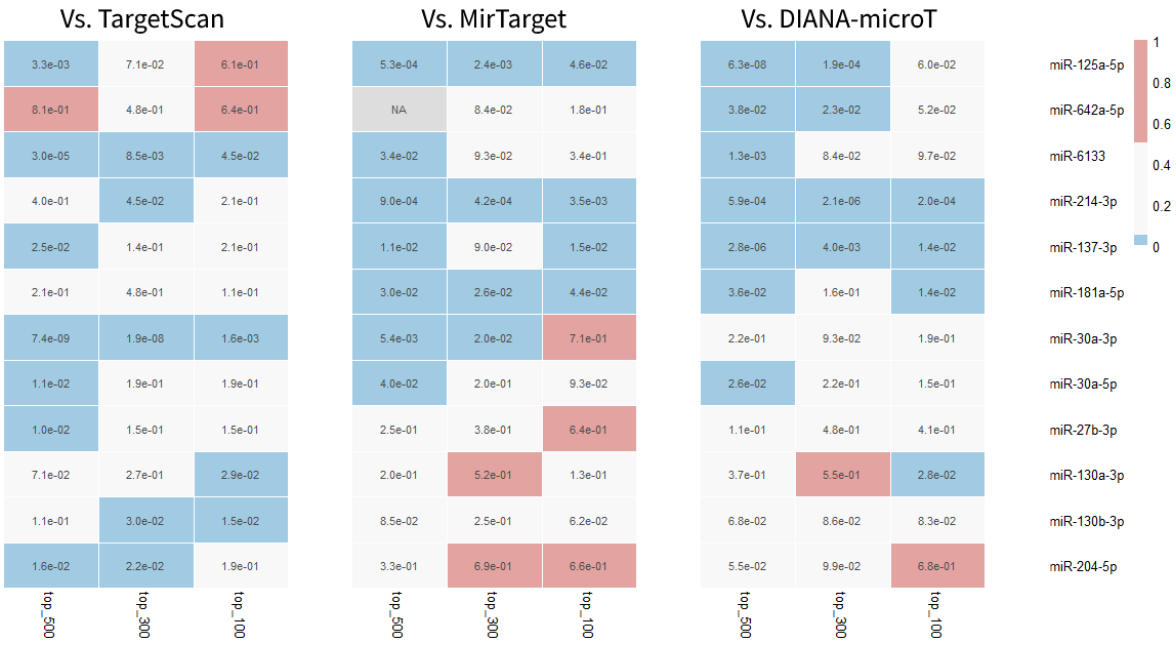


Figure 5.31: Heatmap of miRsight predictions against TargetScan, MirTarget and DIANA-microT. The MW p-value quantifies the separation in lines between miRsight and each tool in the 12 individual test transfections (listed right-hand side. These values are presented as a heatmap. (Blue) p-value 0-0.05: significant and positive result compared to the other tool. (White) p-value 0.05-0.5: insignificant but positive result compared to the other tool. (Red) p-value 0.5-1.0: poor result compared to the other tool. (Grey) Insufficient predictions to make a fair comparison.

miRsight predicts true targets at a significantly higher rate (blue) than a compared

tool in 46 tests, an insignificantly higher rate (white) in 50 tests, and a much lower rate (red) in 11 tests. There is a single incompatible comparison (grey), due to MirTarget outputting an insufficient number of predictions.

The significant results are most notable in the top 500 category, where there is only one instance of another tool attaining a greater leftward line separation (TargetScan in miR-642a-5p). miR-642a-5p may be an unusual test, as this p-value is statistically miRsight's worst result across all comparisons (8.1×10^{-1}), for which MirTarget can also only identify 321 potential targets. In stricter categories, particularly the top 100 predictions by confidence, significance is harder to achieve due to the lower number of data points. A white result still favours miRsight, but the line shift is generally less defined when the p-value increased beyond 0.2.

MirTarget scores the highest number of significant results over miRsight, particularly in miR-204-5p (top 300 p-value 6.9×10^{-1} and top 100 p-value 6.6×10^{-1}), miR-27b-3p (top 100 p-value 6.4×10^{-1}) and miR-130a-3p (top 300 p-value 5.2×10^{-1}). In miR-204-5p, miRsight appears to have difficulty in classifying transcripts with expression log_2 fold change values lower than -1.0 (Figure B.12), particularly in the top 300 and 500 categories.

5.3.4 Feature Importance

Feature importance scores can be extracted from trained RF and SVM models. While limited, particularly in untangling feature interdependency, these scores offer insight into the weights assigned during the training process.

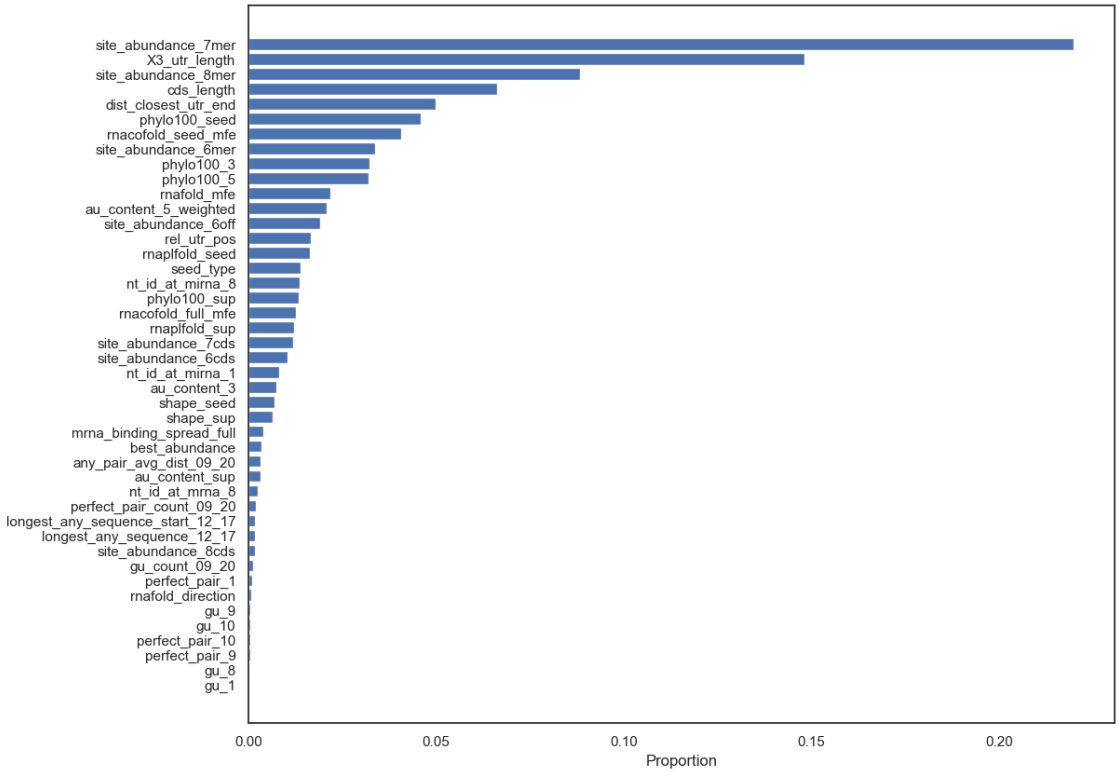


Figure 5.32: Feature coefficients in the trained RF. Ranked feature scores from the RF model using its trained importance weights. MTS, binding positioning, conservation and accessibility features score highly.

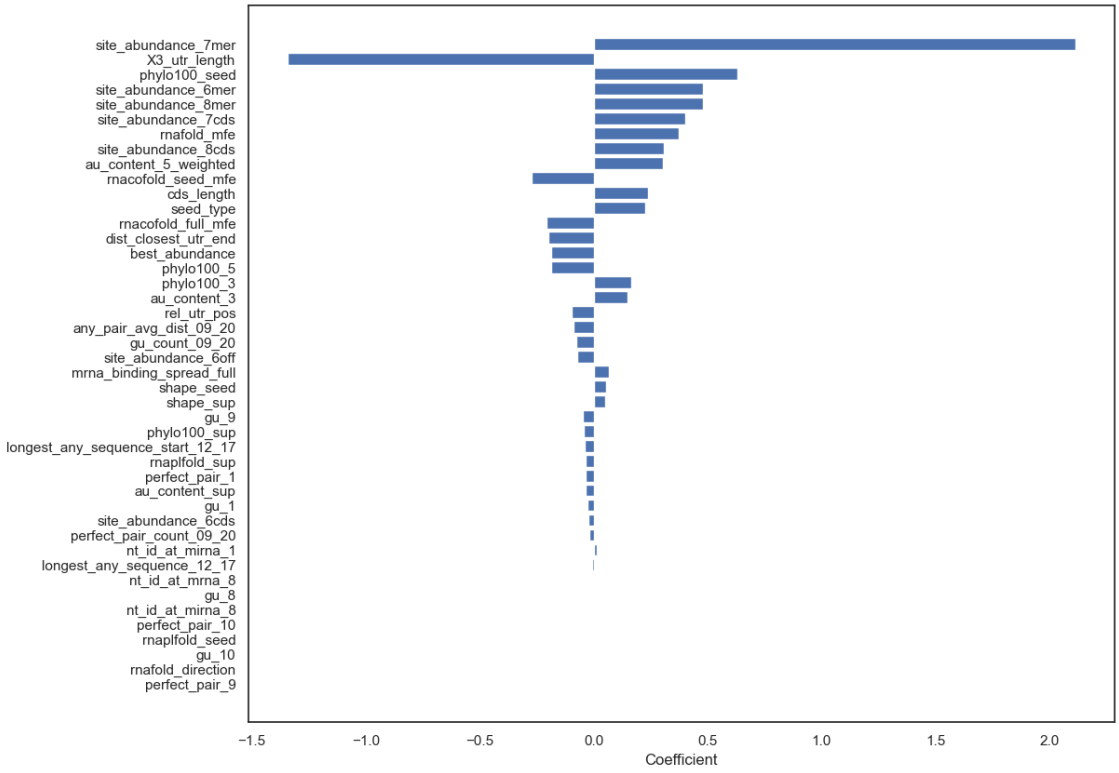


Figure 5.33: Feature coefficients in the trained SVM. Ranked feature scores from the SVM model using its trained coefficients. MTS, conservation, accessibility, and binding positioning features score highly.

MTS is considered the most valuable feature by both models, particularly 7mer 3′ UTR MTS (`site_abundance_7mer`). In RF, 7mer 3′ UTR MTS comprises 22% of the overall feature importance and has the highest coefficient in SVM with 2.11. The length of the 3′ UTR (`X3_utr_length`) is also rated second highest (14.8%, -1.34), which is intrinsically linked to MTS, as longer 3′ UTRs contain more target sites (Stark et al., 2005). The parallel CDS MTS features do not score as highly, however CDS length ranks 4th in RF and 11th in SVM.

After MTS, RF ranks binding positioning (`dist_closest_utr_end`) and conservation features (`phylo100_seed`) in 5th and 6th. SVM also favours seed conservation in 3rd, although it does not value binding positioning as highly (14th). RF rates seed stability (`rnacofold_seed_mfe`) in 7th, whereas SVM places more weight on accessibility, being that it places in 7th (`rnafold_mfe`) and 9th (`au_content_5_weighted`).

Features which track specific bp in the supplementary portion score poorly in both models, however supplementary base accessibility (`au_content_5_weighted`) and conservation (`phylo100_5`) each place in the top half of both tools; this is consistent with results from Chapters 4 and 3. Of the features specifically engineered to track supplementary bp, a novel feature recording the average distance between pairs ranks highest in both models (`any_pair_avg_dist_09_20`).

Features encoding alternative seeds, such as wobbles (`gu_1`, `gu_8`), and 9mer and 10mer matching (`perfect_pair_9`, `perfect_pair_10`) are the lowest scoring in both tools. The novel feature that tracks the placement of the three-window method's representative window (`rnafold_direction`), used in accessibility measurement, is also deemed unimportant to both models. In general, features that rank above these very bottom placements are able to provide some degree of value to the model.

5.4 Discussion

In this chapter, four machine learning models were trained by collating methods and features researched in Chapters 3 and 4. The trained miRsight model was demonstrated to be capable of making accurate predictions to the standard of established target prediction tools. Across all 12 miRNA transfections, miRsight consistently produced a significant p-value for line separation in at least one category (Figure 5.31).

Furthermore, there are many cases of miRsight attaining a significant p-value in multiple categories against all four compared tools. Over 12 tests, miRsight produced an unfiltered output of 49,521 predictions, more than five times that of MirTarget (8,708), but less than TargetScan (68,311) and DIANA-microT (85,267) (Figure 5.30). As a result, when combined with the confidence score, this will be sufficient in allowing a user to tailor miRsight towards their intended use.

The two strongest results relative to TargetScan are miR-6133 (p-values: top 500 3.0×10^{-5}, top 300 8.5×10^{-3} and top 100 4.5×10^{-2}) and miR-30a-3p (p-values: top 500 7.4×10^{-9}, top 300 1.9×10^{-8} and top 100 1.6×10^{-3}), as a significant separation can be observed at all thresholds. In both cases, this is reflected by the relatively higher number of TP in the confusion matrices (Figures B.3 and B.7), where miR-6133 is indicated to be the stronger result, as there are 6.8% more TP predictions. While miRsight predicts miR-30a-3p well compared to TargetScan, the second-worst overall result occurs against MirTarget in the top 100 category of this test (p-value 7.1×10^{-1}). Furthermore, miRsight's miR-30a-3p results are not significant against DIANA-microT at any threshold. The AUC values may explain this inconsistency (Figure 5.29); miR-6133 has an AUC of 0.603 for RF, 0.004 above the mean, however the AUC of miR-30a-3p is only 0.587, making it one of the worst scores. This means that while miR-6133 is a legitimately good result for miRsight, miR-30a-3p may be more of an instance of TargetScan under-performing.

miRsight's worst results are arguably in the miR-204-5p and miR-130a-3p transfections (Figures B.12 and B.10). In both cases, miRsight performs worse than DIANA-microT and MirTarget in multiple categories, in addition to shifting further to the right of the 'All Targets' control line in miR-130a-3p. A potential reason for this is that the miR-130a-3p transfection has an unusually steep gradient in the cumulative log_2 fold change compared to other transfections. Nonetheless, while these results are weaker, the performance of miRsight across the remaining 10 transfections is the highest and most consistent of the tested tools.

MirTarget is generally the second-highest performing tool, followed by TargetScan. The placement of miRsight and MirTarget highlights the value of a large, low-noise dataset for model training, as both tools are exclusively trained using data from 01-liu-HeLa. It should be noted that none of the data used in testing originates from

01-liu-HeLa, the HeLa cell line, or same-family miRNAs transfected in 01-liu-HeLa, so it is unlikely that this is a result of dataset bias. As both tools were trained on HeLa, it is more likely that their application to the unique cell lines of the test set would suffer as a result.

The classification models were internally optimised using F_1 score to maintain a balance between the number and quality of target predictions. There is an argument to be made that, based on the number of predictions produced, the models could have been weighted more towards precision, as in MirTarget's case. This would lead to an improvement in prediction accuracy and, while some users may prefer a large number of predictions, 500 predictions or more may be excessive even with confidence scoring.

Setting the class label close to 0 caused the miRsight to be more accurate than the compared tools between -0.75 and 0 log_2 fold change, possibly at the cost of being worse at predicting targets beyond -1.0. This decision was made to address the trade-off between class imbalance and noise (Figure 5.20). An alternative approach could be to use regression, as it would remove the need for a class label altogether. Using a regression model, the level of downregulation could be predicted directly using patterns learned in training. However, it is also possible that, similar to the classifier, the regressor would struggle to infer values below -1.0 due to these training examples forming a smaller proportion of the dataset.

Based on the CD rankings of each classifier (Figure 5.28), DNN is likely to be underperforming; a well-trained DNN would be expected to perform at a similar level to RF and SVM, as opposed to DT (Koutsoukas et al., 2017). In addition to having the most hyperparameters to tune, layer adjustments invalidate any previously optimised hyperparameter values, meaning DNN may have not been tuned as effectively as the other models under the process that was used. Another potential reason could be that because DNN requires a large amount of training data to function optimally (Chen et al., 2018a), the amount supplied was not sufficient. Had DNN placed closely to RF and SVM in the AUC CD rankings, constructing a heterogeneous ensemble of DNN, RF and SVM may have led to a more effective model, as an ensemble of even three accurate classifiers can improve performance over a single classifier by averaging out erroneous predictions (Dietterich, 2000).

RF was useful as a baseline classifier because the algorithm was capable of producing

competent predictions before tuning. A large factor of ML complexity stems from its stochastic nature and vast number of variables. RF could have been better employed during the data mining phases to help make more informed decisions on feature collinearity and testing. Inversely, treating DNN as a regular ML algorithm increased the complexity of the project due to its poor performance under default settings, coupled with its disproportionate complexity. A more efficient process would have been to use RF to establish baseline methods and performance, then progress to deep learning once extraneous factors were reduced.

A consensus was found between high importance features in the trained models (Figures 5.32 and 5.32), and high-ranking features in the isolated feature tests (Section 3.3.1). In particular, MTS and conservation were found to be highly effective in identifying targets in both scenarios, whereas features using fixed bp thresholds to determine supplementary pairing were mostly unable to distinguish targets from non-targets. In both instances, utilising accessibility and conservation features on the 5′ flanking region of the mRNA target site was more valuable in measuring traits of the supplementary region directly.

Over one third of feature importance consists of MTS. This is similar to findings discussed in Section 3.4 where it was most of the rule-based prediction model's focus. As previously discussed, the 3′ UTR length is related to MTS because longer 3′ UTRs typically contain more target sites (Stark et al., 2005), which means this proportion is possibly higher than it appears. This theory is corroborated by the negative coefficient that the 3′ UTR (and CDS) length is given by SVM (figure 5.33), as this may indicate they are internally scaling the effect of MTS. Merging these two feature groups by scaling MTS against the length of the region may help to reduce the overall weighting these features receive.

The seed type is not as important of a feature as expected, given its central role in miRNA:mRNA binding. As potential targets require a minimum 6mer binding, some information this feature encodes will have been used to filter the target pool before the prediction model is involved. Furthermore, the assigned classification label is -0.2 log_2 fold change. At such a threshold, it is possible that further categorisation of seeds beyond the 6mer does not lead to a radical change in the prediction label. Therefore, if the class label assignment were to be more negative, this feature may become more

important to the classification model regardless of the 6mer target requirement.

The two SHAPE-seq features ranked midway in importance for both RF and SVM, despite having collinearity with alternative accessibility measures and less than 25% data coverage. As discussed, MICE is exponentially less effective as a greater percentage of data is missing, particularly past 60% (Penone et al., 2014). This is therefore a promising result in showing the potential of SHAPE-seq and MICE, provided more data can be assembled.

The miRNA transfection datasets collated for use in the training and test sets may prove useful to further research in this area. In general, a lack of available transfection data is a limiting factor in miRNA target prediction, particularly in training ML models.

Chapter 6

Software Development

6.1 Summary

This chapter outlines the finalisation of tooling created throughout this thesis into the miRsight command line application. In addition, it discusses the development of a web application for hosting and querying its pre-computed predictions. The goal of this work is to increase the accessibility of miRsight by providing simple methods for its use.

The miRsight command line tool is a combined subset of modules discussed in Section 5.2.1, notably encompassing the trained RF model of Section 5.3. As a user-facing tool does not require performance evaluation, modules relating to RNA-seq are removed at this stage (Section 6.2.1). Additionally, the ML component is simplified as the model is already trained following Chapter 5. The tool can be downloaded from `https://github.com/ryanjp18/mirsight`.

For the web application, miRsight predictions are pre-computed for each human miRNA and ingested into a database (Section 6.2.2.1). The database is managed and accessed by the back-end (Section 6.2.2.2) in response to requests from the front-end user interface (Section 6.2.2.3). Following deployment (Section 6.2.2.4), the miRsight web application is hosted at `https://mirsight.info`.

6.2 Methods

Both tools utilise Git (Chacon and Straub, 2014) for source control, enabling the tracking of changes and flexible feature development using code branches.

6.2.1 Command Line Application

In Chapter 3, architectural decisions were deferred as the scope of the project was unclear. In Chapter 5, the software became more sophisticated with the integration of Python, and cohesive functionality was grouped into modules. As a result of this iterative development, producing the final miRsight software is a mostly superficial conversion of any modules tightly coupled with RNA-seq and model training.

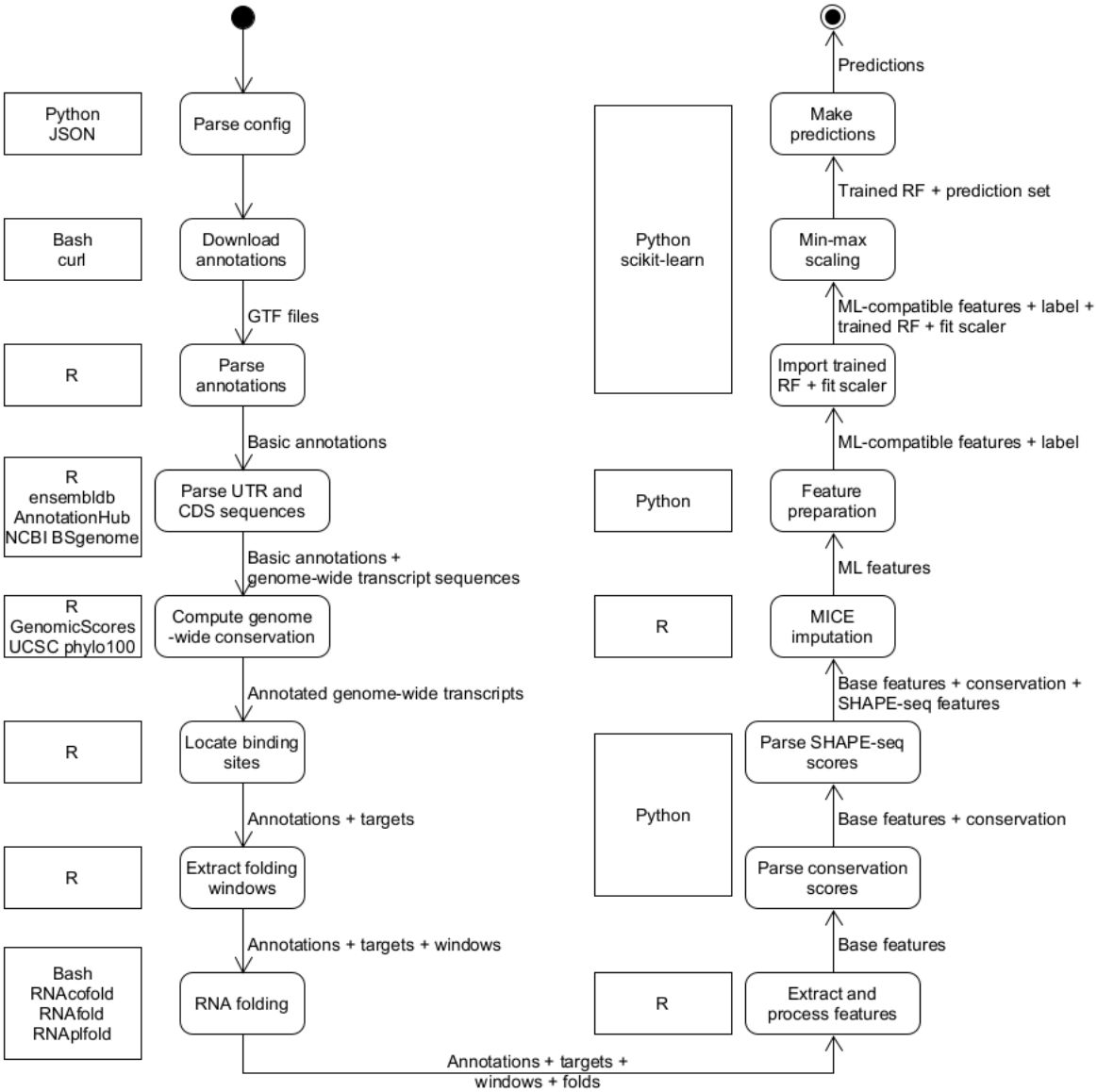


Figure 6.1: miRsight activity diagram. miRsight is an amalgam of previous tooling. The application flow begins with the setup module, which generates annotations and caches. Next is feature extraction, which is simplified by the removal of RNA-seq. Finally, ML is used to make predictions. As the RF is already trained, these ML steps are also simplified.

Since expression log_2 fold change values are integral to training a model for ML, miRNA transfection experiments were previously used to drive the tool. For each transfected miRNA, the seed target sequence was used to extract features. As RNA-seq data is no longer required, a list of all potential human miRNAs are now pulled from miRBase. The process of locating the seed sequence and extracting features is otherwise identical.

The version of the miRsight software is expressed using semantic versioning (Preston-Werner, 2009), where the syntax `MAJOR.MINOR.PATCH` denotes three independent tracking numbers dependent on the type of update. In semantic versioning, `MAJOR` is incremented in response to breaking functionality changes. Of backwards-compatible changes, `MINOR` is incremented in response to additional functionality and `PATCH` is incremented for bug fixes.

6.2.2 Web Application

Docker is used to simplify the deployment of the miRsight web application through containers; executable software bundles that host an application and its dependencies in a virtual environment. Each core module of the web application is granted a container: `database`, `backend` and `frontend`. The `frontend` communicates with the `backend` via application programming interface (API) routes tied to user actions, while the `backend` is responsible for interacting with the `database` and sending responses back to the `frontend`.

6.2.2.1 Database Layout

The MySQL-managed database (Oracle, 2023) contains a single `predictions` table for storing miRsight predictions. The initial version of `predictions` has five visible columns (Table 6.1). This can potentially be extended to utilise the expansive annotation and feature data collected by miRsight. However, minimising database content is important in reducing load times and storage requirements.

Table 6.1: Prediction table data type declaration

Column	Type	Visible
`id`	`unique auto-incrementing ID`	
`mirna_id`	`string`	✓
`ensembl_transcript_id`	`string`	✓
`position`	`integer`	✓
`seed_type`	`string`	✓
`score`	`float`	✓
`created_at`	`datetime`	
`updated_at`	`datetime`	

6.2.2.2 Back-end Architecture

Laravel (Otwell, 2023) is a back-end systems framework for PHP (Bakken et al., 2000) that provides lightweight templating and database interaction tooling. It performs three key functions for the web application: database population, front-end communications management and testing.

The `predictions` table is created using a custom schema defined by extending Laravel's `Migration` class. Migrations allow a database to undergo structural changes without being reset, as they define the actions required to transition it between two states. This is useful to miRsight because the `predictions` table contains millions of rows, and the schema is likely to require adjustments based on user feedback. A custom seeder, which extends the `Seeder` class, is responsible for populating the `predictions` table by parsing predictions from miRsight output. The seeder utilises chunking, a method of compiling small pieces of information into larger units to reduce memory usage (Thalmann et al., 2019). A combination of migrations and seeders is useful in preventing data loss as the database setup is reduced to a set of reproducible instructions; where a row already exists, the seeder does not attempt to add it twice.

When a user searches on the website, the search term is passed from front-end to back-end by way of the `search_mirna` and `search_transcript` API routes. A query is then constructed to collect database rows matching the search pattern, for which indexing is applied to optimise lookup speed. In B-Tree indexing, the number of steps required to reach a search result is reduced by splitting the data and ensuring each decision branch

points to a higher or lower index (Schwartz et al., 2012). Once the query is complete, the result set is sent back to the front-end in JSON format and used to populate the web page.

6.2.2.3 Front-end Design and Development

The front-end is programmed using TypeScript (Microsoft, 2023), and formatted using HTML (W3C, 2024) and CSS (W3C, 2023). TypeScript is a programming language that compiles into JavaScript (Ecma International, 2024) code as part of an application's build process. It provides several benefits compared to native JavaScript, notably static typing and tighter scoping (Bierman et al., 2014). Vue.js (You, 2023) is a front-end framework for building data-driven apps, providing out-of-the-box support for single-page applications (SPAs), reusable elements (components) and automatic updating of page content to reflect data changes (reactivity). Finally, Vite (You, 2024) is a Vue.js build tool that provides optimised asset bundling and basic code obfuscation.

Target prediction websites typically employ a minimal and professional aesthetic, albeit they are often grounded in older design philosophy. miRsight attempts to update this style through the use of reactive and dynamic page elements, in addition to a subtler colour palette compared to traditional web-safe HTML colour schemes. Developing the website as an SPA is another such modernisation, as asynchronous updates and the avoidance of page loading offer a more seamless user experience to static pages. Furthermore, both mobile and tablet layouts are supported through responsive design patterns and media queries. While the former is unlikely to see significant use in miRNA research, tablet use is relatively widespread and rapidly growing in tangential industries such as healthcare (Sclafani et al., 2013).

Search terms are first collected on the website through a unified search box, which infers whether the user is searching by miRNA or transcript. Debouncing prevents bandwidth issues by ensuring searches do not occur more than once per second. Following a search, the request is sent asynchronously to the associated API path using JavaScript's `fetch` API. When a response is received, the `results` are stored in a Vue.js `ref` object to enforce reactivity, visually updating elements that use the `results` `ref` automatically.

A component refers to a group of HTML elements with tightly coupled layout, logic,

or styling. Using Vue.js, a `Table` component is constructed to display miRsight predictions. `Table` is reactively linked to search `results`, meaning it is automatically redrawn to reflect new response data from the server. Using **v-for** list rendering, a row is created for each predicted target of `results`, populating child elements according to corresponding display fields. In this way, the `Table` component can be easily extended as user requirements grow.

```
<div class="row" v-for="result in filteredResults">
    <p class="cell">{{ result["mirna_id"] }}</p>
    <p class="cell">{{ result["ensembl_transcript_id"] }}</p>
    <p class="cell">{{ result["transcript_start"] }}</p>
    <p class="cell">{{ result["transcript_end"] }}</p>
    <p class="cell">{{ result["score"] }}</p>
</div>
```

A range slider allows users to filter `results` by confidence. The slider is two-way data bound to a `filter ref`, synchronising the slider and `filter`. This `filter` is automatically applied to the results table using a `computed` property. In summary, the `results` of a search are mapped to the confidence `filter` (and slider), which in turn is mapped to the `Table` component. As a result, the page automatically re-renders to display new data following user actions.

```
const filteredResults = computed(
    () => results.value.filter(result => result.score > filter.value)
);
```

For more precise filtering, a user can click on a prediction to have it removed from the results. Filtered results can be printed, exported to a document, or downloaded as a tabular file by clicking on the corresponding button.

6.2.2.4 Deployment

When running on a web server, common communication ports can be exposed by Docker containers to enable network connections. Most notably, the `frontend` server must be exposed to external connections to allow users to access the website. Similarly, opening ports for internal connections allow the `backend` API routes to be made accessible to the `frontend` container, providing a proxy for `database` communication.

Network communications occur over the Hypertext Transfer Protocol (HTTP) protocol

by default. However, these communications can be encrypted to instead utilise Hypertext Transfer Protocol Secure (HTTPS) through Transport Layer Security (TLS). Beyond the benefits of encryption, HTTPS is a requirement in modern web browsers to prevent security warnings that may deter users from accessing the website. A robust server host is required for TLS, which is provided here by Nginx (Reese, 2008).

An automated deployment pipeline is built by providing Docker with a set of commands specific to each of the containers. In all cases, Docker is instructed to download and install dependencies, execute respective tooling and expose necessary network ports. The `frontend` container is additionally required to produce a front-end build with Vite, serve the build to Nginx and provide Nginx with TLS certification.

6.3 Results

The miRsight web application is hosted at `https://mirsight.info`, while the command line tool itself can be downloaded directly from its GitHub repository, located at `https://github.com/ryanjp18/mirsight`.

6.3.1 Testing

The integrity of the database and API routes is tested using Laravel's unit testing. In each test, a factory is used to generate randomised fake prediction data according to a set of realistic seeding rules.

```
[ryan@ryan-lt backend]$ sudo bash pa.sh test
php artisan test

  PASS  Tests\Feature\TargetTest
  ✓ search targets by mirna id                          0.08s
  ✓ search targets by mirna id when empty               0.01s
  ✓ search targets by transcript id                     0.01s
  ✓ search targets by transcript id when empty          0.01s
  ✓ targets data types are correct                      0.01s

  Tests:    5 passed (36 assertions)
  Duration: 0.13s
```

Figure 6.2: Unit testing for miRsight. A set of unit tests executed using Laravel. (1) Assert that the correct miRNA targets are returned when searching by miRNA ID. (2) Assert that no results are found when searching for a non-existent miRNA. (3) Assert that the correct transcript results are returned when searching by transcript ID. (4) Assert that no results are found when searching for a non-existent transcript. (5) Assert that the database schema is correct.

6.3.2 User Interface

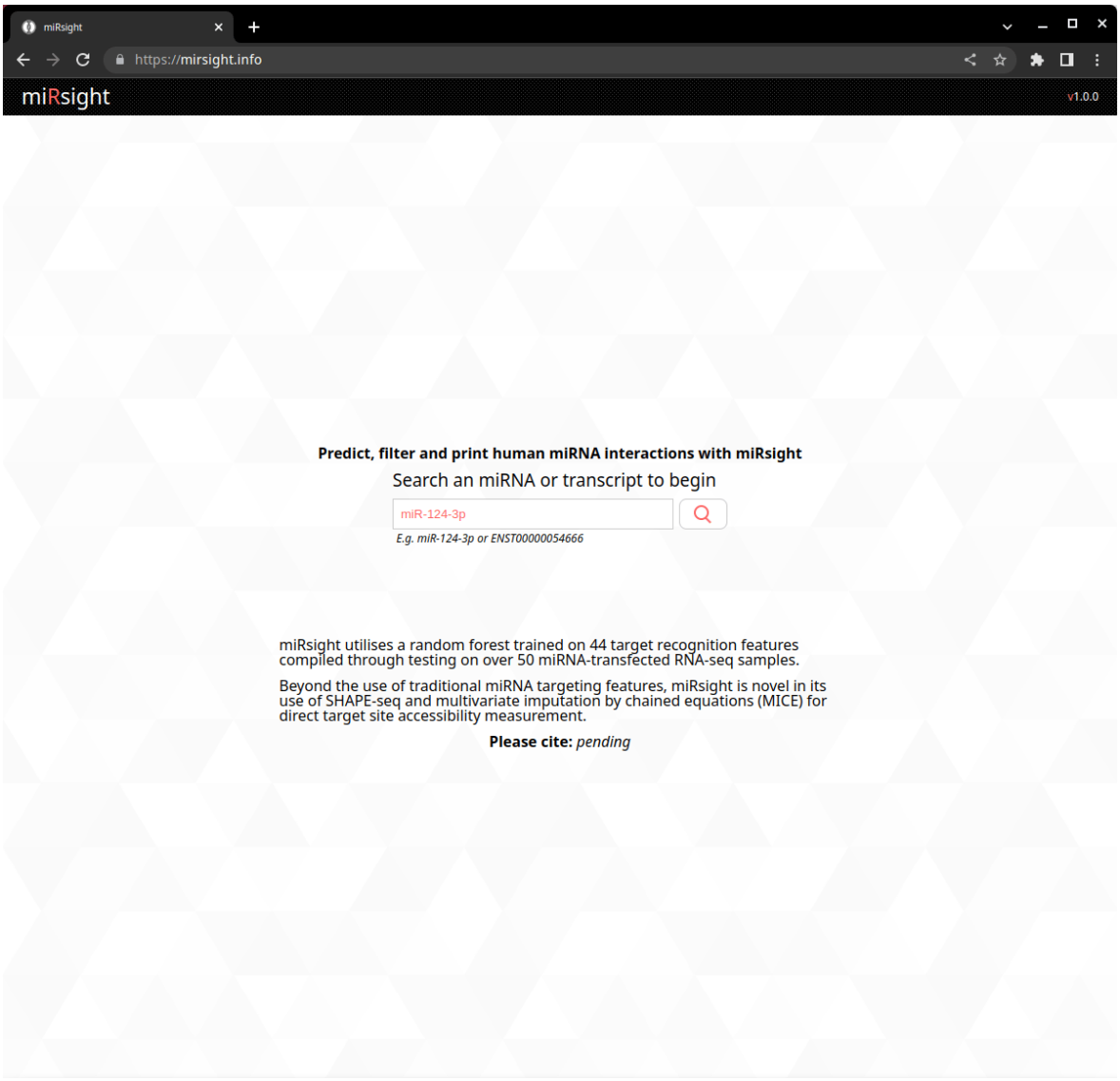


Figure 6.3: miRsight website search bar. When miRsight is loaded, the home page displays usage information and the unified search field. An example miRNA query is provided for ease of use.

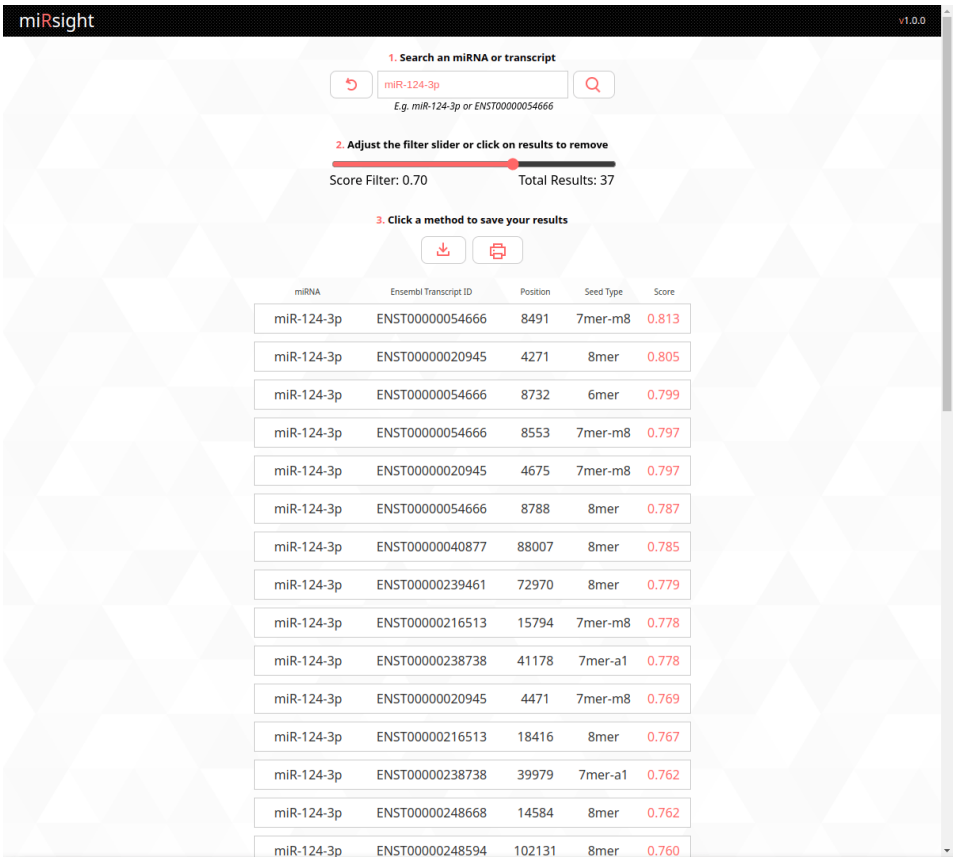


Figure 6.4: miRsight website results table. Searching for either an miRNA or transcript brings up the results table. By default, these results are filtered to 0.7 confidence.

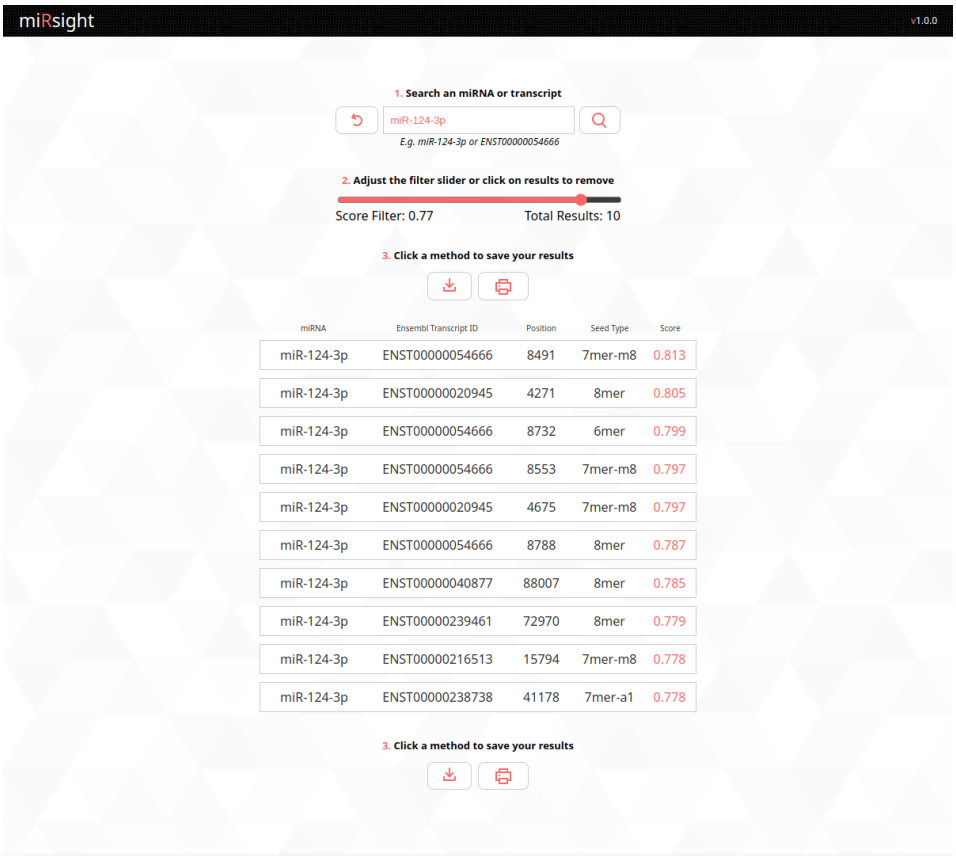


Figure 6.5: miRsight website results filtering. Dragging the confidence score slider above the results table, or manually clicking predictions to remove them, filters the predictions on the page.

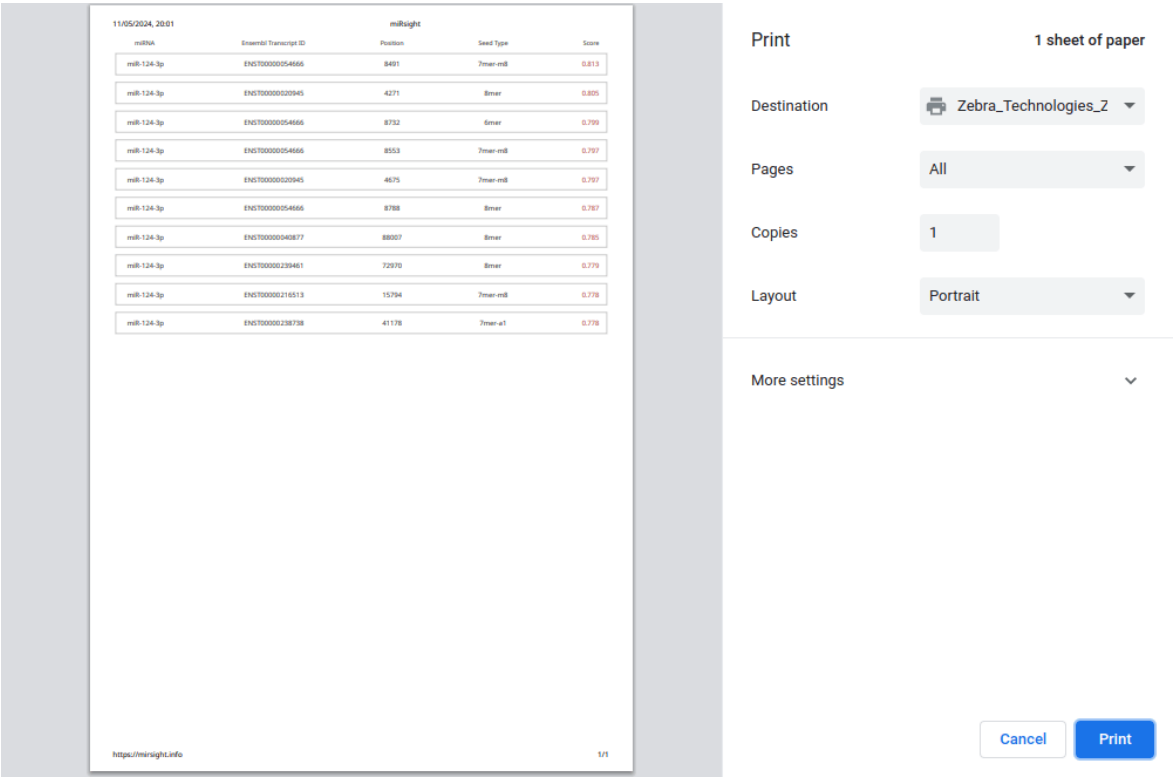


Figure 6.6: miRsight website printing filtered results. Selecting the print icon sends the filtered results to the browser's print window, where they can be exported to file or printed.

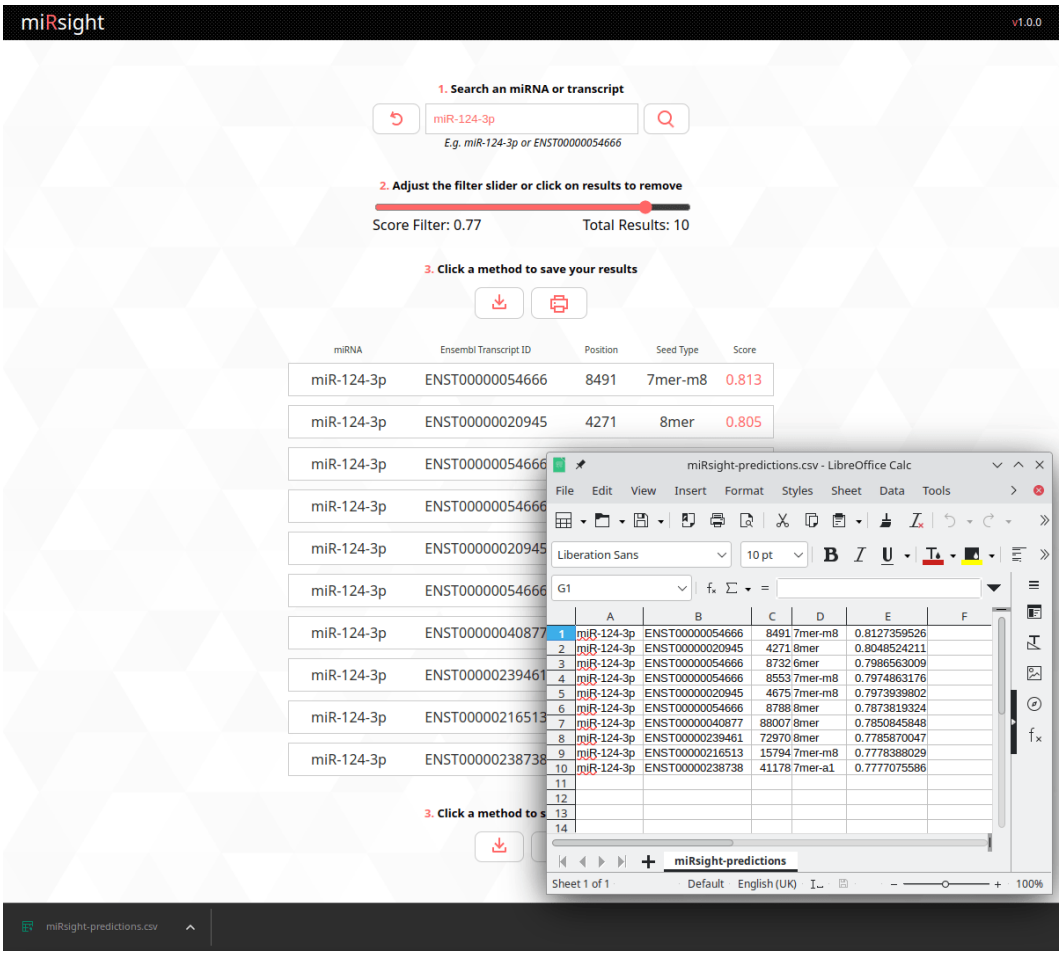


Figure 6.7: miRsight website downloading filtered results. Selecting the download icon sends the filtered results to a tabular file and prompts the browser to download it.

6.3.3 Responsive Design

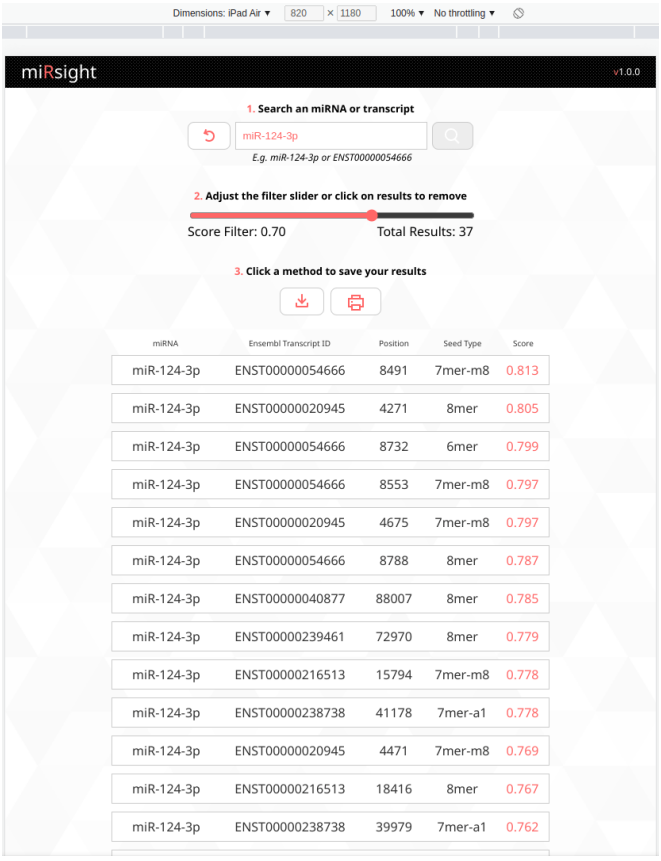


Figure 6.8: miRsight website scaled to tablet dimensions. At tablet size, miRsight can show all of its content on the screen without resizing.

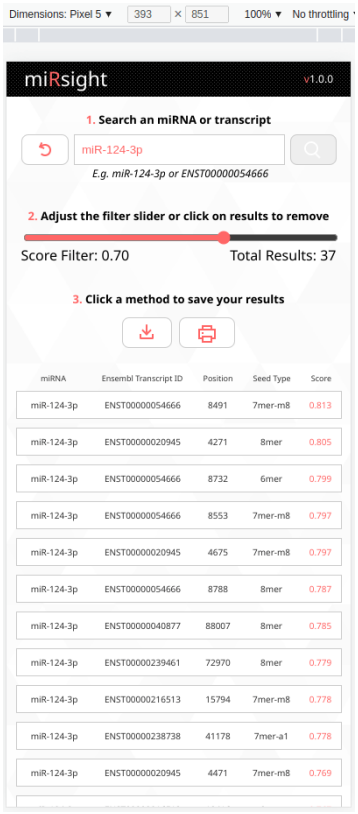


Figure 6.9: miRsight website scaled to mobile dimensions. At mobile size, media queries are used to resize padding and font sizes to fit the screen.

6.4 Discussion

In this chapter, a command line tool and web application have been developed with the aim of improving the accessibility of miRsight predictions. As software development is an iterative process, steps have been taken to ensure that code is maintainable and extensible.

Both tools adhere to modern development practices; in particular, the separation of concerns between modules means that enhancements and fixes can be performed without causing a ripple effect. Communication between modules is restricted and interaction instead occurs through inputs and outputs, which reduces the risk of regression bugs and simplifies development. While the RF model would need to be retrained following new features being added, the command line tool is not opinionated in which features it uses between modules, as the ML component will simply use any provided in its input. An additional benefit of this approach is code abstraction, as other modules are not required to be deeply understood to engineer adjacent functionality.

A key motivation behind the web server's design philosophy was in minimising the need for maintenance. Section 2.5.2 highlighted that prediction tools have long product lifespans, yet commonly lack upkeep. By utilising a simple design and feature set, issues stemming from browser and device support are reduced. Furthermore, the use of responsive design, as opposed to targeted device support, means that new screen sizes will continue to be naturally supported. Finally, using the component and reactivity of Vue.js, much of the web functionality is encapsulated in a way that allows new fields and display methods to be added without complicating the existing code base.

The use of Git for source control also assists in mitigating potential bugs through its version history and branch functionality. As a result, feature development can occur in discrete version releases, simplifying testing and shrinking the surface of potential bugs. As parallel development is also simplified owing to the tool's modular architecture, miRsight may also benefit from the open-source collaboration afforded by Git platforms. Regardless of direct collaboration, it is at least expected that user enhancement requests and bug reports will be periodically submitted in this way.

Docker was particularly beneficial in deploying the web application. Following development work, changes are pulled on the server using Git, then deployed by resetting the

associated Docker image once unit tests are run. This could be further simplified using a workflow management tool to create a fully automated pipeline, although this may be excessive at this stage. An element of Docker that is possibly underutilised here is in its ability to simplify dependencies and provide multi-platform support. Docker is presently only used in the web component of miRsight. However, the command line application has numerous dependencies and an operating system limitation due to ViennaRNA's Linux requirement. Supplying a Docker method of installation would reduce the software dependency to Docker alone, in addition to allowing the tool to run on any operating system that supports Docker.

Chapter 7

Conclusion and Future Work

7.1 Summary

This thesis presents a new miRNA target prediction algorithm in miRsight, the pre-computed predictions of which can be accessed at `https://mirsight.info`, while the tool can be downloaded directly from `https://github.com/ryanjp18/mirsight`. Across the 12 datasets tested, miRsight is shown to perform to a consistently higher standard of accuracy than TargetScan, MirTarget and DIANA-microT. As demonstrated by its performance in the top 100, 300 and 500 categories ranked by confidence, miRsight is also capable of effectively sorting and filtering its predictions to fit the needs of individual users. In addition to delivery of miRsight, some observations were made throughout this thesis:

- **Target recognition features:** Many targeting features that affect the efficacy and specificity of miRNA:mRNA bindings have been discovered throughout the course of miRNA research. However, relatively few prediction tools implement more than a small subset of common features. In general, there is a strong acceptance in modern tools for features centred around seed types, conservation, and site accessibility. However, MTS is perhaps not as well utilised as its impact in Chapters 3 and 5 would suggest. Furthermore, there is a lack of consistency in the implementation of even these prolific features, including seed type, for which research is well-established and consistent as to the importance of the three canonical sites compared to marginal and alternative sites.

- **Site accessibility using SHAPE-seq:** There is potential for SHAPE-seq as a novel measure of accessibility, if an appropriate amount of data can be assembled. The discovery that reactivity values can be merged between cell lines to improve both the feature coverage and overall efficacy may help to mitigate the relative rarity of SHAPE-seq compared to RNA-seq data (Section 4.3.3). Still, the current implementation of the feature provides value to miRsight according to both feature importance rankings (Figures 5.32 and 5.33), despite having less than 25% coverage.

- **Supplementary pairing definitions:** It is unclear exactly which method is best used to capture information for the supplementary portion of bindings. Many alternative definitions for recording fixed bp combinations were tested throughout this thesis, yet little impact was observed unless supported with additional features. For example, weighted AU content scoring of the seed target's $5'$ flanking region is often used in prediction tools as a means of encoding supplementary pairing. Nonetheless, of those tested, conservation scoring in this region was shown to be particularly effective (Figure 3.22), ranking as the 10th most important feature in the RF model (Figure 5.32). On the other hand, calculating the average distance between paired bases in the 11 nt downstream of the seed was the most effective isolate supplementary pairing feature tested.

- **Machine learning:** The shift in target prediction from rule-based models to ML models is advantageous due to the number of target recognition features and degree of their interdependence. The 01-liu-HeLa dataset published by MirTarget (Liu and Wang, 2019) was invaluable in developing this tool, and highlights the importance of a large and consistent dataset for ML model training. Furthermore, the strategy of grouping this data by miRNA transfection, as opposed to individual data points, was used to great effect in this thesis. In terms of ML models, there is little consensus in the use of specific algorithms among prediction tools; SVMs are widespread in miRNA target prediction, while deep learning is likely to become more popular in the field as a result of recent innovations and adoption in many industries. However, it should be noted that RF is highly effective as an out-of-the-box classifier (Biau and Scornet, 2016), which was particularly useful in this thesis for exploring the subtleties of targeting mechanics.

- **miRNA transfection dataset:** Over 50 miRNA transfections were collated from various databases throughout this project. As the performance of ML algorithms is directly correlated with the quality and quantity of available data, this dataset may provide value as a starting point for further development of miRNA targeting software.

7.2 Future Work

Several potential improvements were uncovered during the course of this thesis, which may be expanded upon in future releases of miRsight:

- **Handling for imperfect seeds and CDS targets:** The decision to use a fixed 6mer seed target constraint to exclusively target the $3'$ UTR was made early in development. Tools such as TargetScan have found success in allowing G:U wobbles and gaps at specific positions within the seed (Agarwal et al., 2015), while DIANA-microT-CDS trains an independent model to locate CDS target sites (Reczko et al., 2012). Accounting for these scenarios would allow the tool to predict targets that are currently not able to enter the prediction pool.

- **Machine learning improvements:** The use of an ensemble model often offers improved performance over single classifiers (Dietterich, 2000). DIANA-microT-CDS compares four models to determine its MRE score (Reczko et al., 2012), and TargetScan trains an individual model for each type of seed (Agarwal et al., 2015). More unique models will be needed to utilise a similar approach to the former, as the current implementation of DNN is under-performing, likely due to limited training data (Chen et al., 2018a), or insufficient optimisation. All four models were trained using the same process to promote fair test conditions; however, DNN training is significantly more complex than DT, RF and SVM, and has a number of deviations to warrant special treatment, such as the need to optimise layers and a disproportionate number of hyperparameters. With DNN functioning effectively, a combination of RF, SVM and DNN may be sufficient in forming a heterogeneous ensemble. Alternatively, building a homogeneous of RF classifiers according to each seed typein a manner similar to TargetScan. This is a viable option because feature importance variations in accordance with seed type was a phenomenon noted in isolated feature testing (Section 3.3.1). Finally, there

is a relationship between the assigned class label of -0.2 and miRsight's sensitivity to targets between 0 and -0.5 log_2 fold change (Appendix B). However, the relatively inconsistent correlation between AUC and ROC (Figure 5.29), and the model's overall performance in these cumulative plots, suggests the relationship between this threshold and the underlying biology is not perfect. A greater prediction accuracy may there for achieved by changing the evaluative metric, class label threshold, or using a regression approach.

- **Integrating more datasets:** The present SHAPE-seq data originates from a single source. As aggregating SHAPE-seq data between cell lines did not lead to a reduction in data quality (Figure 4.11), the integration of further datasets should improve the quality of these features. In terms of RNA-seq, the dataset is composed entirely of miRNA transfection data. miRNA knockouts could be used to substantially increase the amount of data available for use in ML. Furthermore, the use of knockout typically leads to lower noise, as the absence of an miRNA shows a stronger signal than increasing the amount of an miRNA that may already be expressed in a particular cell.

- **Package management and operating system support:** The website is intended to be the primary use case for the miRsight algorithm; however, some users will prefer to use the command line tool with their own specific configuration. The feature extraction algorithm has several dependencies which complicate both the set-up process and maintenance over the tool's life-cycle. Furthermore, miRsight is Linux-only, due to a restriction imposed by the ViennaRNA suite. A simple solution to these problems would be to provide pre-configured Docker images for miRsight. This method has been used by some recent target prediction tools, such as isoTar (Distefano et al., 2019).

- **External target verification:** DIANA-microT verifies predictions against TargetScan in their prediction results. A similar system would reduce risk for the end user, as lab verification or high confidence targets predicted by more than one tool are more likely to be true positive predictions.

7.3 Conclusion

miRNA target prediction remains a difficult problem despite substantial methodological advancement over the past two decades. A promising recent innovation is the use of RNA bind-n-seq to directly probe the miRNA:AGO interaction, a technique first applied in TargetScan 8.0 (McGeary et al., 2019). Such a forthright approach may encourage more efficient solutions by eliminating some of the numerous intermediary features that complicate miRNA:mRNA binding prediction. Nonetheless, following the progression from comparatively simple rule-based approaches to the sophisticated ML models of modern tools, future efforts are prone to become even more complex in their attempts to model miRNA interaction, particularly in light of widespread industry trends towards deep learning. With this being the case, high-quality datasets, such as the primary 01-liu-HeLa miRNA transfection dataset used in this study (Liu and Wang, 2019), are likely to become increasingly important in training these data-hungry algorithms.

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Agarwal, V., Bell, G. W., Nam, J.-W., and Bartel, D. P. (2015). Predicting effective microrna target sites in mammalian mrnas. *elife*, 4:e05005.

Airola, A., Pyysalo, S., Björne, J., Pahikkala, T., Ginter, F., and Salakoski, T. (2008). All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC bioinformatics*, 9:1–12.

Ali Syeda, Z., Langden, S. S. S., Munkhzul, C., Lee, M., and Song, S. J. (2020). Regulatory mechanism of microrna expression in cancer. *International journal of molecular sciences*, 21(5):1723.

Alles, J., Fehlmann, T., Fischer, U., Backes, C., Galata, V., Minet, M., Hart, M., Abu-Halima, M., Grässer, F. A., Lenhof, H.-P., et al. (2019). An estimate of the total number of true human mirnas. *Nucleic acids research*, 47(7):3353–3364.

Almeida, J. S. (2002). Predictive non-linear modeling of complex data by artificial neural networks. *Current opinion in biotechnology*, 13(1):72–76.

Andrews, S. et al. (2010). Fastqc: a quality control tool for high throughput sequence data.

Baek, D., Villén, J., Shin, C., Camargo, F. D., Gygi, S. P., and Bartel, D. P. (2008). The impact of microrna on protein output. *Nature*, 455(7209):64–71.

Bakken, S. S., Suraski, Z., and Schmid, E. (2000). *PHP Manual: Volume 2*. iUniverse, Incorporated.

Barillot, E., Calzone, L., Hupe, P., Vert, J.-P., and Zinovyev, A. (2012). *Computational systems biology of cancer*. CRC Press.

Bartel, D. P. (2004). Microrna: genomics, biogenesis, mechanism, and function. *cell*, 116(2):281–297.

Bartel, D. P. (2009). Microrna: target recognition and regulatory functions. *cell*, 136(2):215–233.

Bazzini, A. A., Lee, M. T., and Giraldez, A. J. (2012). Ribosome profiling shows that mir-430 reduces translation before causing mrna decay in zebrafish. *Science*, 336(6078):233–237.

Bentley, D. R., Balasubramanian, S., Swerdlow, H. P., Smith, G. P., Milton, J., Brown, C. G., Hall, K. P., Evers, D. J., Barnes, C. L., Bignell, H. R., et al. (2008). Accurate whole human genome sequencing using reversible terminator chemistry. *nature*, 456(7218):53–59.

Bernhart, S. H., Hofacker, I. L., and Stadler, P. F. (2006a). Local rna base pairing probabilities in large sequences. *Bioinformatics*, 22(5):614–615.

Bernhart, S. H., Tafer, H., Mückstein, U., Flamm, C., Stadler, P. F., and Hofacker, I. L. (2006b). Partition function and base pairing probabilities of rna heterodimers. *Algorithms for Molecular Biology*, 1:1–10.

Betel, D., Koppal, A., Agius, P., Sander, C., and Leslie, C. (2010). Comprehensive modeling of microrna targets predicts functional non-conserved and non-canonical sites. *Genome biology*, 11:1–14.

Betel, D., Wilson, M., Gabow, A., Marks, D. S., and Sander, C. (2008). The microrna. org resource: targets and expression. *Nucleic acids research*, 36(suppl_1):D149–D153.

Beveridge, D. J., Richardson, K. L., Epis, M. R., Brown, R. A., Stuart, L. M.,

Woo, A. J., and Leedman, P. J. (2021). The tumor suppressor mir-642a-5p targets wilms tumor 1 gene and cell-cycle progression in prostate cancer. *Scientific Reports*, 11(1):18003.

Biau, G. and Scornet, E. (2016). A random forest guided tour. *Test*, 25:197–227.

Bierman, G., Abadi, M., and Torgersen, M. (2014). Understanding typescript. In *ECOOP 2014–Object-Oriented Programming: 28th European Conference, Uppsala, Sweden, July 28–August 1, 2014. Proceedings 28*, pages 257–281. Springer.

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

Bradley, T. and Moxon, S. (2019). Filtar: Using rna-seq data to improve microrna target prediction accuracy in animals. *BioRxiv*, page 595322.

Brancati, G. and Großhans, H. (2018). An interplay of mirna abundance and target site architecture determines mirna activity and specificity. *Nucleic acids research*, 46(7):3259–3269.

Bray, N. L., Pimentel, H., Melsted, P., and Pachter, L. (2016). Near-optimal probabilistic rna-seq quantification. *Nature biotechnology*, 34(5):525.

Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32.

Breiman, L. (2017). *Classification and regression trees*. Routledge.

Brennecke, J., Stark, A., Russell, R. B., and Cohen, S. M. (2005). Principles of microrna–target recognition. *PLoS biology*, 3(3).

Brodley, C. E. and Friedl, M. A. (1999). Identifying mislabeled training data. *Journal of artificial intelligence research*, 11:131–167.

Broughton, J. P., Lovci, M. T., Huang, J. L., Yeo, G. W., and Pasquinelli, A. E. (2016). Pairing beyond the seed supports microrna targeting specificity. *Molecular cell*, 64(2):320–333.

Buda, M., Maki, A., and Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106:249–259.

Busch, A., Richter, A. S., and Backofen, R. (2008). Intarna: efficient prediction of bacterial srna targets incorporating target site accessibility and seed regions. *Bioinformatics*, 24(24):2849–2856.

Calo, E. and Wysocka, J. (2013). Modification of enhancer chromatin: what, how, and why? *Molecular cell*, 49(5):825–837.

Chacon, S. and Straub, B. (2014). *Pro git*. Springer Nature.

Charbuty, B. and Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, 2(01):20–28.

Chen, H., Engkvist, O., Wang, Y., Olivecrona, M., and Blaschke, T. (2018a). The rise of deep learning in drug discovery. *Drug discovery today*, 23(6):1241–1250.

Chen, L., Heikkinen, L., Wang, C., Yang, Y., Knott, K. E., and Wong, G. (2018b). mirtoolsgallery: a tag-based and rankable microrna bioinformatics resources database portal. *Database*, 2018:bay004.

Chen, Y. and Wang, X. (2020). mirdb: an online database for prediction of functional microrna targets. *Nucleic acids research*, 48(D1):D127–D131.

Chowdhury, B. and Garai, G. (2017). A review on multiple sequence alignment from the perspective of genetic algorithm. *Genomics*, 109(5-6):419–431.

Cloonan, N. (2015). Re-thinking mirna-mrna interactions: Intertwining issues confound target discovery. *Bioessays*, 37(4):379–388.

Corbett, A. H. (2018). Post-transcriptional regulation of gene expression and human disease. *Current opinion in cell biology*, 52:96–104.

Crick, F. (1970). Central dogma of molecular biology. *Nature*, 227(5258):561–563.

Cunningham, F., Achuthan, P., Akanni, W., Allen, J., Amode, M. R., Armean, I. M., Bennett, R., Bhai, J., Billis, K., Boddu, S., et al. (2018). Ensembl 2019. *Nucleic acids research*, 47(D1):D745–D751.

Cutler, A., Cutler, D. R., and Stevens, J. R. (2012). Random forests. *Ensemble machine learning: Methods and applications*, pages 157–175.

DeHaseth, P. L., Zupancic, M. L., and Record Jr, M. T. (1998). Rna polymerase-promoter interactions: the comings and goings of rna polymerase. *Journal of bacteriology*, 180(12):3019–3025.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, 7:1–30.

Didiano, D. and Hobert, O. (2006). Perfect seed pairing is not a generally reliable predictor for mirna-target interactions. *Nature structural & molecular biology*, 13(9):849–851.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.

Ding, J., Li, X., and Hu, H. (2016). Tarpmir: a new approach for microrna target site prediction. *Bioinformatics*, 32(18):2768–2775.

Distefano, R., Nigita, G., Veneziano, D., Romano, G., Croce, C. M., and Acunzo, M. (2019). isotar: consensus target prediction with enrichment analysis for micrornas harboring editing sites and other variations. *MicroRNA Target Identification: Methods and Protocols*, pages 211–235.

Doench, J. G., Petersen, C. P., and Sharp, P. A. (2003). sirnas can function as mirnas. *Genes & development*, 17(4):438–442.

Doench, J. G. and Sharp, P. A. (2004). Specificity of microrna target selection in translational repression. *Genes & development*, 18(5):504–511.

Dongare, A., Kharde, R., Kachare, A. D., et al. (2012). Introduction to artificial neural network. *International Journal of Engineering and Innovative Technology (IJEIT)*, 2(1):189–194.

Drury, R. E., O'Connor, D., and Pollard, A. J. (2017). The clinical application of micrornas in infectious disease. *Frontiers in immunology*, 8:295828.

Durinck, S., Moreau, Y., Kasprzyk, A., Davis, S., De Moor, B., Brazma, A., and Huber, W. (2005). Biomart and bioconductor: a powerful link between biological databases and microarray data analysis. *Bioinformatics*, 21(16):3439–3440.

Ecma International (2017). Information technology — the json data interchange syntax. Standard ISO/IEC TR 21778:2017, International Organization for Standardization, Geneva, CH.

Ecma International (2024). *ECMA-262 Language Specification*. `https://tc39.es/ecma262/` [Accessed: 11-05-2024].

Edgar, R., Domrachev, M., and Lash, A. E. (2002). Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic acids research*, 30(1):207–210.

Eibe, F., Hall, M. A., and Witten, I. H. (2016). The weka workbench. online appendix for data mining: practical machine learning tools and techniques. In *Morgan Kaufmann*. Morgan Kaufmann Publishers San Francisco, California.

Ellwanger, D. C., Büttner, F. A., Mewes, H.-W., and Stümpflen, V. (2011). The sufficient minimal set of mirna seed types. *Bioinformatics*, 27(10):1346.

Engström, P. G., Steijger, T., Sipos, B., Grant, G. R., Kahles, A., Rätsch, G., Goldman, N., Hubbard, T. J., Harrow, J., et al. (2013). Systematic evaluation of spliced alignment programs for rna-seq data. *Nature methods*, 10(12):1185–1191.

Enright, A., John, B., Gaul, U., Tuschl, T., Sander, C., and Marks, D. (2003). Microrna targets in drosophila. *Genome biology*, 4:1–27.

Ewing, B., Hillier, L., Wendl, M. C., and Green, P. (1998). Base-calling of automated sequencer traces usingphred. i. accuracy assessment. *Genome research*, 8(3):175–185.

Fang, Y. and Fullwood, M. J. (2016). Roles, functions, and mechanisms of long non-coding rnas in cancer. *Genomics, proteomics & bioinformatics*, 14(1):42–54.

Fang, Z. and Rajewsky, N. (2011). The impact of mirna target sites in coding sequences and in 3′ utrs. *PloS one*, 6(3).

Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874.

Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *The journal of machine learning research*, 15(1):3133–3181.

Finotello, F. and Di Camillo, B. (2015). Measuring differential gene expression with rna-seq: challenges and strategies for data analysis. *Briefings in functional genomics*, 14(2):130–142.

Fox, E. W., Hill, R. A., Leibowitz, S. G., Olsen, A. R., Thornbrugh, D. J., and Weber, M. H. (2017). Assessing the accuracy and stability of variable selection methods for random forest modeling in ecology. *Environmental monitoring and assessment*, 189:1–20.

Friedman, J. H. (1998). Data mining and statistics: What's the connection? *Computing science and statistics*, 29(1):3–9.

Friedman, R. C., Farh, K. K.-H., Burge, C. B., and Bartel, D. P. (2009). Most mammalian mrnas are conserved targets of microrna. *Genome research*, 19(1):92–105.

Gaidatzis, D., van Nimwegen, E., Hausser, J., and Zavolan, M. (2007). Inference of mirna targets using evolutionary conservation and pathway analysis. *BMC bioinformatics*, 8:1–22.

Gao, M., Fritz, D. T., Ford, L. P., and Wilusz, J. (2000). Interaction between a poly (a)-specific ribonuclease and the 5′ cap influences mrna deadenylation rates in vitro. *Molecular cell*, 5(3):479–488.

Garcia, D. M., Baek, D., Shin, C., Bell, G. W., Grimson, A., and Bartel, D. P. (2011). Weak seed-pairing stability and high target-site abundance decrease the proficiency of lsy-6 and other microrna. *Nature structural & molecular biology*, 18(10):1139.

Gharizadeh, B., Herman, Z. S., Eason, R. G., Jejelowo, O., and Pourmand, N. (2006). Large-scale pyrosequencing of synthetic dna: A comparison with results from sanger dideoxy sequencing. *Electrophoresis*, 27(15):3042–3047.

GNU, P. (2007). Free software foundation. bash (3.2. 48)[unix shell program].

Grimson, A., Farh, K. K.-H., Johnston, W. K., Garrett-Engele, P., Lim, L. P., and Bartel, D. P. (2007). Microrna targeting specificity in mammals: determinants beyond seed pairing. *Molecular cell*, 27(1):91–105.

Guo, H., Ingolia, N. T., Weissman, J. S., and Bartel, D. P. (2010). Mammalian microrna predominantly act to decrease target mrna levels. *Nature*, 466(7308):835–840.

Guo, J. U., Agarwal, V., Guo, H., and Bartel, D. P. (2014a). Expanded identification and characterization of mammalian circular rnas. *Genome biology*, 15(7):409.

Guo, L., Zhao, Y., Yang, S., Zhang, H., and Chen, F. (2014b). Integrative analysis of mirna-mrna and mirna-mirna interactions. *BioMed research international*, 2014.

Guo, X., Yin, Y., Dong, C., Yang, G., and Zhou, G. (2008). On the class imbalance problem. In *2008 Fourth international conference on natural computation*, volume 4, pages 192–201. IEEE.

Ha, M. and Kim, V. N. (2014). Regulation of microrna biogenesis. *Nature reviews Molecular cell biology*, 15(8):509–524.

Hamada-Tsutsumi, S., Onishi, M., Matsuura, K., Isogawa, M., Kawashima, K., Sato, Y., and Tanaka, Y. (2020). Inhibitory effect of a human microrna, mir-6133-5p, on the fibrotic activity of hepatic stellate cells in culture. *International Journal of Molecular Sciences*, 21(19):7251.

Hammell, M., Long, D., Zhang, L., Lee, A., Carmack, C. S., Han, M., Ding, Y., and Ambros, V. (2008). mirwip: microrna target prediction based on microrna-containing ribonucleoprotein–enriched transcripts. *Nature methods*, 5(9):813–819.

Hausser, J., Berninger, P., Rodak, C., Jantscher, Y., Wirth, S., and Zavolan, M. (2009). Mirz: an integrated microrna expression atlas and target prediction resource. *Nucleic acids research*, 37(suppl_2):W266–W272.

Hausser, J., Syed, A. P., Bilen, B., and Zavolan, M. (2013). Analysis of cds-located mirna target sites suggests that they can effectively inhibit translation. *Genome research*, 23(4):604–615.

Heather, J. M. and Chain, B. (2016). The sequence of sequencers: The history of sequencing dna. *Genomics*, 107(1):1–8.

Heaton, J. (2008). *Introduction to neural networks with Java*. Heaton Research, Inc.

Hebert, P. D., Braukmann, T. W., Prosser, S. W., Ratnasingham, S., DeWaard, J. R., Ivanova, N. V., Janzen, D. H., Hallwachs, W., Naik, S., Sones, J. E., et al. (2018). A sequel to sanger: amplicon sequencing that scales. *BMC genomics*, 19(1):219.

Higgs, P. G. (2000). Rna secondary structure: physical and computational aspects. *Quarterly reviews of biophysics*, 33(3):199–253.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Hirose, Y. and Manley, J. L. (2000). Rna polymerase ii and the integration of nuclear events. *Genes & development*, 14(12):1415–1429.

Hobert, O. (2008). Gene regulation by transcription factors and micrornas. *Science*, 319(5871):1785–1786.

Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al. (2003). A practical guide to support vector classification.

Hu, H., Liu, J.-M., Hu, Z., Jiang, X., Yang, X., Li, J., Zhang, Y., Yu, H., and Khaitovich, P. (2018). Recently evolved tumor suppressor transcript tp73-as1 functions as sponge of human-specific mir-941. *Molecular biology and evolution*, 35(5):1063–1077.

Ij, H. (2018). Statistics versus machine learning. *Nat Methods*, 15(4):233.

Jain, A. K., Mao, J., and Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer*, 29(3):31–44.

Jain, M., Olsen, H. E., Paten, B., and Akeson, M. (2016). The oxford nanopore minion: delivery of nanopore sequencing to the genomics community. *Genome biology*, 17(1):239.

Jeggari, A., Marks, D. S., and Larsson, E. (2012). mircode: a map of putative microrna target sites in the long non-coding transcriptome. *Bioinformatics*, 28(15):2062–2063.

John, B., Enright, A. J., Aravin, A., Tuschl, T., Sander, C., and Marks, D. S. (2004). Human microrna targets. *PLoS biology*, 2(11).

Johnsson, P., Lipovich, L., Grandér, D., and Morris, K. V. (2014). Evolutionary conservation of long non-coding rnas; sequence, structure, function. *Biochimica et Biophysica Acta (BBA)-General Subjects*, 1840(3):1063–1071.

Karatzoglou, A., Meyer, D., and Hornik, K. (2006). Support vector machines in r. *Journal of statistical software*, 15:1–28.

Kertesz, M., Iovino, N., Unnerstall, U., Gaul, U., and Segal, E. (2007). The role of site accessibility in microrna target recognition. *Nature genetics*, 39(10):1278–1284.

Khvorova, A., Reynolds, A., and Jayasena, S. D. (2003). Functional sirnas and mirnas exhibit strand bias. *Cell*, 115(2):209–216.

Kim, H.-K., Park, J. D., Choi, S. H., Shin, D. J., Hwang, S., Jung, H.-Y., and Park, K.-S. (2020). Functional link between mir-200a and elk3 regulates the metastatic nature of breast cancer. *Cancers*, 12(5):1225.

Kim, S.-K., Nam, J.-W., Lee, W.-J., and Zhang, B.-T. (2005). A kernel method for microrna target prediction using sensible data and position-based features. In *2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 1–7. IEEE.

Kiriakidou, M., Nelson, P. T., Kouranov, A., Fitziev, P., Bouyioukos, C., Mourelatos, Z., and Hatzigeorgiou, A. (2004). A combined computational-experimental approach predicts human microrna targets. *Genes & development*, 18(10):1165–1178.

Kolluri, J., Kotte, V. K., Phridviraj, M., and Razia, S. (2020). Reducing overfitting problem in machine learning using novel l1/4 regularization method. In *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, pages 934–938. IEEE.

Kolovos, P., Knoch, T. A., Grosveld, F. G., Cook, P. R., and Papantonis, A. (2012). Enhancers and silencers: an integrated and simple model for their function. *Epigenetics & chromatin*, 5(1):1–8.

Kotsiantis, S. B., Zaharakis, I., Pintelas, P., et al. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1):3–24.

Koutsoukas, A., Monaghan, K. J., Li, X., and Huan, J. (2017). Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. *Journal of cheminformatics*, 9(1):1–13.

Kozomara, A., Birgaoanu, M., and Griffiths-Jones, S. (2019). mirbase: from microrna sequences to function. *Nucleic acids research*, 47(D1):D155–D162.

Krek, A., Grün, D., Poy, M. N., Wolf, R., Rosenberg, L., Epstein, E. J., MacMenamin, P., Da Piedade, I., Gunsalus, K. C., Stoffel, M., et al. (2005). Combinatorial microrna target predictions. *Nature genetics*, 37(5):495–500.

Krol, J., Loedige, I., and Filipowicz, W. (2010). The widespread regulation of microrna biogenesis, function and decay. *Nature Reviews Genetics*, 11(9):597.

Krueger, F. (2015). Trim galore. *A wrapper tool around Cutadapt and FastQC to consistently apply quality and adapter trimming to FastQ files*, 516:517.

Krüger, J. and Rehmsmeier, M. (2006). Rnahybrid: microrna target prediction easy, fast and flexible. *Nucleic acids research*, 34(suppl_2):W451–W454.

Kumar, K. R., Cowley, M. J., and Davis, R. L. (2019). Next-generation sequencing and emerging technologies. In *Seminars in thrombosis and hemostasis*, volume 45, pages 661–673. Thieme Medical Publishers.

Kwak, N. (2013). Nonlinear projection trick in kernel methods: An alternative to the kernel trick. *IEEE transactions on neural networks and learning systems*, 24(12):2113–2119.

Lall, S., Grün, D., Krek, A., Chen, K., Wang, Y.-L., Dewey, C. N., Sood, P., Colombo, T., Bray, N., MacMenamin, P., et al. (2006). A genome-wide map of conserved microrna targets in c. elegans. *Current biology*, 16(5):460–471.

Law, C. W., Alhamdoosh, M., Su, S., Dong, X., Tian, L., Smyth, G. K., and Ritchie, M. E. (2016). Rna-seq analysis is easy as 1-2-3 with limma, glimma and edger. *F1000Research*, 5.

Lawrence, M., Huber, W., Pagès, H., Aboyoun, P., Carlson, M., Gentleman, R., Morgan, M. T., and Carey, V. J. (2013). Software for computing and annotating genomic ranges. *PLoS computational biology*, 9(8):e1003118.

Lee, S. W. L., Paoletti, C., Campisi, M., Osaki, T., Adriani, G., Kamm, R. D., Mattu, C., and Chiono, V. (2019). Microrna delivery through nanoparticles. *Journal of Controlled Release*, 313:80–95.

Lee, Y., Ahn, C., Han, J., Choi, H., Kim, J., Yim, J., Lee, J., Provost, P., Rådmark, O., Kim, S., et al. (2003). The nuclear rnase iii drosha initiates microrna processing. *Nature*, 425(6956):415–419.

Lee, Y., Kim, M., Han, J., Yeom, K.-H., Lee, S., Baek, S. H., and Kim, V. N. (2004). Microrna genes are transcribed by rna polymerase ii. *The EMBO journal*, 23(20):4051–4060.

Lee, Y.-J. and Mangasarian, O. L. (2001). Rsvm: Reduced support vector machines. In *Proceedings of the 2001 SIAM International Conference on Data Mining*, pages 1–17. SIAM.

Leon-Icaza, S. A., Zeng, M., and Rosas-Taraco, A. G. (2019). micrornas in viral acute respiratory infections: immune regulation, biomarkers, therapy, and vaccines. *ExRNA*, 1:1–7.

Lewis, B. P., Burge, C. B., and Bartel, D. P. (2005). Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microrna targets. *cell*, 120(1):15–20.

Lewis, B. P., Shih, I.-h., Jones-Rhoades, M. W., Bartel, D. P., and Burge, C. B. (2003). Prediction of mammalian microrna targets. *Cell*, 115(7):787–798.

Li, P., Shi, R., and Zhang, Q. C. (2020). icshape-pipe: A comprehensive toolkit for icshape data analysis and evaluation. *Methods*, 178:96–103.

Li, R., Fan, W., Tian, G., Zhu, H., He, L., Cai, J., Huang, Q., Cai, Q., Li, B., Bai, Y., et al. (2010). The sequence and de novo assembly of the giant panda genome. *Nature*, 463(7279):311–317.

Li, Y., Huang, C., Ding, L., Li, Z., Pan, Y., and Gao, X. (2019). Deep learning in bioinformatics: Introduction, application, and perspective in the big data era. *Methods*, 166:4–21.

Li, Y., Li, C., Xia, J., and Jin, Y. (2011). Domestication of transposable elements into microrna genes in plants. *Plos one*, 6(5):e19212.

Lim, L. P., Glasner, M. E., Yekta, S., Burge, C. B., and Bartel, D. P. (2003a). Vertebrate microrna genes. *Science*, 299(5612):1540–1540.

Lim, L. P., Lau, N. C., Weinstein, E. G., Abdelhakim, A., Yekta, S., Rhoades, M. W., Burge, C. B., and Bartel, D. P. (2003b). The micrornas of caenorhabditis elegans. *Genes & development*, 17(8):991–1008.

Lin, K.-M. and Lin, C.-J. (2003). A study on reduced support vector machines. *IEEE transactions on Neural Networks*, 14(6):1449–1459.

Liu, W. and Wang, X. (2019). Prediction of functional microrna targets by integrative modeling of microrna binding and target expression data. *Genome biology*, 20(1):18.

Loher, P. and Rigoutsos, I. (2012). Interactive exploration of rna22 microrna target predictions. *Bioinformatics*, 28(24):3322–3323.

Long, D., Lee, R., Williams, P., Chan, C. Y., Ambros, V., and Ding, Y. (2007). Potent effect of target structure on microrna function. *Nature structural & molecular biology*, 14(4):287–294.

Lorenz, R., Bernhart, S. H., Zu Siederdissen, C. H., Tafer, H., Flamm, C., Stadler, P. F., and Hofacker, I. L. (2011). Viennarna package 2.0. *Algorithms for molecular biology*, 6(1):26.

Love, M. I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, 15(12):550.

Lucks, J. B., Mortimer, S. A., Trapnell, C., Luo, S., Aviran, S., Schroth, G. P., Pachter, L., Doudna, J. A., and Arkin, A. P. (2011). Multiplexed rna structure characterization with selective 2′-hydroxyl acylation analyzed by primer extension sequencing (shape-seq). *Proceedings of the National Academy of Sciences*, 108(27):11063–11068.

Lukasik, A., Wójcikowski, M., and Zielenkiewicz, P. (2016). Tools4mirs–one place to gather all the tools for mirna analysis. *Bioinformatics*, 32(17):2722–2724.

Lund, E. and Dahlberg, J. (2006). Substrate selectivity of exportin 5 and dicer in the biogenesis of micrornas. In *Cold Spring Harbor symposia on quantitative biology*, volume 71, pages 59–66. Cold Spring Harbor Laboratory Press.

Madeira, F., Pearce, M., Tivey, A. R., Basutkar, P., Lee, J., Edbali, O., Madhusoodanan, N., Kolesnikov, A., and Lopez, R. (2022). Search and sequence analysis tools services from embl-ebi in 2022. *Nucleic acids research*, 50(W1):W276–W279.

Mahen, E. M., Watson, P. Y., Cottrell, J. W., and Fedor, M. J. (2010). mrna secondary structures fold sequentially but exchange rapidly in vivo. *PLoS biology*, 8(2):e1000307.

Mahmud, M., Kaiser, M. S., Hussain, A., and Vassanelli, S. (2018). Applications of deep learning and reinforcement learning to biological data. *IEEE transactions on neural networks and learning systems*, 29(6):2063–2079.

Mann, M., Wright, P. R., and Backofen, R. (2017). Intarna 2.0: enhanced and customizable prediction of rna–rna interactions. *Nucleic acids research*, 45(W1):W435–W439.

Maragkakis, M., Alexiou, P., Papadopoulos, G. L., Reczko, M., Dalamagas, T., Giannopoulos, G., Goumas, G., Koukis, E., Kourtis, K., Simossis, V. A., et al. (2009a). Accurate microrna target prediction correlates with protein repression levels. *BMC bioinformatics*, 10:1–10.

Maragkakis, M., Reczko, M., Simossis, V. A., Alexiou, P., Papadopoulos, G. L., Dalamagas, T., Giannopoulos, G., Goumas, G., Koukis, E., Kourtis, K., et al. (2009b). Diana-microt web server: elucidating microrna functions through target prediction. *Nucleic acids research*, 37(suppl_2):W273–W276.

Maragkakis, M., Vergoulis, T., Alexiou, P., Reczko, M., Plomaritou, K., Gousis, M., Kourtis, K., Koziris, N., Dalamagas, T., and Hatzigeorgiou, A. G. (2011). Diana-microt web server upgrade supports fly and worm mirna target prediction and bibliographic mirna to disease association. *Nucleic acids research*, 39(suppl_2):W145–W148.

Margulies, M., Egholm, M., Altman, W. E., Attiya, S., Bader, J. S., Bemben, L. A., Berka, J., Braverman, M. S., Chen, Y.-J., Chen, Z., et al. (2005). Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–380.

Martin, M. (2011). Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet. journal*, 17(1):10–12.

Mason, C. H. and Perreault Jr, W. D. (1991). Collinearity, power, and interpretation of multiple regression analysis. *Journal of marketing research*, 28(3):268–280.

McGeary, S. E., Bisaria, N., Pham, T. M., Wang, P. Y., and Bartel, D. P. (2022). Microrna 3′-compensatory pairing occurs through two binding modes, with affinity shaped by nucleotide identity and position. *Elife*, 11:e69803.

McGeary, S. E., Lin, K. S., Shi, C. Y., Pham, T. M., Bisaria, N., Kelley, G. M., and Bartel, D. P. (2019). The biochemical basis of microrna targeting efficacy. *Science*, 366(6472):eaav1741.

Mehta, M., Rissanen, J., Agrawal, R., et al. (1995). Mdl-based decision tree pruning. In *KDD*, volume 21, pages 216–221.

Microsoft (2023). *The TypeScript Handbook*. `https://www.typescriptlang.org/docs/handbook/intro.html` [Accessed: 18-09-2023].

Midi, H., Sarkar, S. K., and Rana, S. (2010). Collinearity diagnostics of binary logistic regression model. *Journal of interdisciplinary mathematics*, 13(3):253–267.

Miller, J. R., Koren, S., and Sutton, G. (2010). Assembly algorithms for next-generation sequencing data. *Genomics*, 95(6):315–327.

Miranda, K. C., Huynh, T., Tay, Y., Ang, Y.-S., Tam, W.-L., Thomson, A. M., Lim, B., and Rigoutsos, I. (2006). A pattern-based method for the identification of microrna binding sites and their corresponding heteroduplexes. *Cell*, 126(6):1203–1217.

Mitra, R., Adams, C. M., Jiang, W., Greenawalt, E., and Eischen, C. M. (2020). Pan-cancer analysis reveals cooperativity of both strands of microrna that regulate tumorigenesis and patient survival. *Nature Communications*, 11(1):968.

Morales, J., Pujar, S., Loveland, J. E., Astashyn, A., Bennett, R., Berry, A., Cox, E., Davidson, C., Ermolaeva, O., Farrell, C. M., et al. (2022). A joint ncbi and embl-ebi transcript set for clinical genomics and research. *Nature*, 604(7905):310–315.

Morris, K. V. and Mattick, J. S. (2014). The rise of regulatory rna. *Nature Reviews Genetics*, 15(6):423–437.

Morrison, K., Manzano, M., Chung, K., Schipma, M. J., Bartom, E. T., and Gottwein, E. (2019). The oncogenic kaposi's sarcoma-associated herpesvirus encodes a mimic of the tumor-suppressive mir-15/16 mirna family. *Cell reports*, 29(10):2961–2969.

Muir, P., Li, S., Lou, S., Wang, D., Spakowicz, D. J., Salichos, L., Zhang, J., Weinstock, G. M., Isaacs, F., Rozowsky, J., et al. (2016). The real cost of sequencing: scaling computation to keep pace with data generation. *Genome biology*, 17(1):53.

Müller, M. B., Hübner, M., Li, L., Tomasi, S., Ließke, V., Effinger, D., Hirschberger, S., Pogoda, K., Sperandio, M., and Kreth, S. (2022). Cell-crossing functional network driven by microrna-125a regulates endothelial permeability and monocyte trafficking in acute inflammation. *Frontiers in Immunology*, 13:826047.

Murchison, E. P. and Hannon, G. J. (2004). mirnas on the move: mirna biogenesis and the rnai machinery. *Current opinion in cell biology*, 16(3):223–229.

Nam, J.-W., Rissland, O. S., Koppstein, D., Abreu-Goodger, C., Jan, C. H., Agarwal, V., Yildirim, M. A., Rodriguez, A., and Bartel, D. P. (2014). Global analyses of the effect of different cellular contexts on microrna targeting. *Molecular cell*, 53(6):1031–1043.

Nielsen, C. B., Shomron, N., Sandberg, R., Hornstein, E., Kitzman, J., and Burge, C. B. (2007). Determinants of targeting by endogenous and exogenous microrna and sirnas. *Rna*, 13(11):1894–1910.

Noble, W. S. (2006). What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567.

O'Brien, J., Hayder, H., Zayed, Y., and Peng, C. (2018). Overview of microrna biogenesis, mechanisms of actions, and circulation. *Frontiers in endocrinology*, 9:402.

Ogbourne, S. and Antalis, T. M. (1998). Transcriptional control and the role of silencers in transcriptional regulation in eukaryotes. *Biochemical Journal*, 331(1):1–14.

Ohler, U., Yekta, S., Lim, L. P., Bartel, D. P., and Burge, C. B. (2004). Patterns of flanking sequence conservation and a characteristic upstream motif for microrna gene identification. *Rna*, 10(9):1309–1322.

Ong, C.-T. and Corces, V. G. (2011). Enhancer function: new insights into the regulation of tissue-specific gene expression. *Nature Reviews Genetics*, 12(4):283–293.

Ooms, J. (2014). The jsonlite package: A practical and consistent mapping between json data and r objects. *arXiv:1403.2805 [stat.CO]*.

Oracle (2023). *MySQL 8.0 Reference Manual*. `https://dev.mysql.com/doc/refman/8.0/en/` [Accessed: 18-09-2023].

Oshiro, T. M., Perez, P. S., and Baranauskas, J. A. (2012). How many trees in a random forest? In *Machine Learning and Data Mining in Pattern Recognition: 8th International Conference, MLDM 2012, Berlin, Germany, July 13-20, 2012. Proceedings 8*, pages 154–168. Springer.

Osuna, E. and Girosi, F. (1998). Reducing the run-time complexity of support vector machines. In *International Conference on Pattern Recognition (submitted)*. Citeseer.

Otwell, T. (2023). *Laravel Documentation*. `https://laravel.com/docs/10.x/readme` [Accessed: 18-09-2023].

Pal, M., Ponticelli, A. S., and Luse, D. S. (2005). The role of the transcription bubble and tfiib in promoter clearance by rna polymerase ii. *Molecular cell*, 19(1):101–110.

Paraskevopoulou, M. D., Georgakilas, G., Kostoulas, N., Vlachos, I. S., Vergoulis, T., Reczko, M., Filippidis, C., Dalamagas, T., and Hatzigeorgiou, A. G. (2013). Diana-microt web server v5. 0: service integration into mirna functional analysis workflows. *Nucleic acids research*, 41(W1):W169–W173.

Pasquinelli, A. E. (2012). Micrornas and their targets: recognition, regulation and an emerging reciprocal relationship. *Nature Reviews Genetics*, 13(4):271.

Patel, R. K. and Jain, M. (2012). Ngs qc toolkit: a toolkit for quality control of next generation sequencing data. *PloS one*, 7(2).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pennacchio, L. A., Bickmore, W., Dean, A., Nobrega, M. A., and Bejerano, G. (2013). Enhancers: five essential questions. *Nature Reviews Genetics*, 14(4):288–295.

Penone, C., Davidson, A. D., Shoemaker, K. T., Di Marco, M., Rondinini, C., Brooks, T. M., Young, B. E., Graham, C. H., and Costa, G. C. (2014). Imputation of missing data in life-history trait datasets: which approach performs the best? *Methods in Ecology and Evolution*, 5(9):961–970.

Peterson, S. M., Thompson, J. A., Ufkin, M. L., Sathyanarayana, P., Liaw, L., and Congdon, C. B. (2014). Common features of microrna target prediction tools. *Frontiers in Genetics*, 5.

Pettersson, E., Lundeberg, J., and Ahmadian, A. (2009). Generations of sequencing technologies. *Genomics*, 93(2):105–111.

Pham, K. T. M., Inoue, Y., Van Vu, B., Nguyen, H. H., Nakayashiki, T., Ikeda, K.-i.,

and Nakayashiki, H. (2015). Moset1 (histone h3k4 methyltransferase in magnaporthe oryzae) regulates global gene expression during infection-related morphogenesis. *PLoS genetics*, 11(7):e1005385.

Pimentel, H., Bray, N. L., Puente, S., Melsted, P., and Pachter, L. (2017). Differential analysis of rna-seq incorporating quantification uncertainty. *Nature methods*, 14(7):687.

Podralska, M., Ciesielska, S., Kluiver, J., van den Berg, A., Dzikiewicz-Krawczyk, A., and Slezak-Prochazka, I. (2020). Non-coding rnas in cancer radiosensitivity: Micrornas and lncrnas as regulators of radiation-induced signaling pathways. *Cancers*, 12(6):1662.

Pollard, K. S., Hubisz, M. J., Rosenbloom, K. R., and Siepel, A. (2010). Detection of nonneutral substitution rates on mammalian phylogenies. *Genome research*, 20(1):110–121.

Poyatos, R., Sus, O., Badiella, L., Mencuccini, M., and Martínez-Vilalta, J. (2018). Gap-filling a spatially explicit plant trait database: comparing imputation methods and different levels of environmental information. *Biogeosciences*, 15(9):2601–2617.

Pratt, A. J. and MacRae, I. J. (2009). The rna-induced silencing complex: a versatile gene-silencing machine. *Journal of Biological Chemistry*, 284(27):17897–17901.

Preston-Werner, T. (2009). *Semantic Versioning 2.0.0*. `https://semver.org/spec/v2.0.0.html` [Accessed: 30-07-2023].

Puigdevall, P. and Castelo, R. (2018). Genomicscores: seamless access to genomewide position-specific scores from r and bioconductor. *Bioinformatics*, 34(18):3208–3210.

Qi, Y. (2012). Random forest for bioinformatics. *Ensemble machine learning: Methods and applications*, pages 307–323.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Raden, M., Ali, S. M., Alkhnbashi, O. S., Busch, A., Costa, F., Davis, J. A., Eggenhofer, F., Gelhausen, R., Georg, J., Heyne, S., et al. (2018). Freiburg rna tools: a central online resource for rna-focused research and teaching. *Nucleic acids research*, 46(W1):W25–W29.

Rainer, J., Gatto, L., and Weichenberger, C. X. (2019). ensembldb: an r package to create and use ensembl-based annotation resources. *Bioinformatics*, 35(17):3151–3153.

Ramachandran, G. t. and Sasisekharan, V. (1968). Conformation of polypeptides and proteins. *Advances in protein chemistry*, 23:283–437.

Reczko, M., Maragkakis, M., Alexiou, P., Grosse, I., and Hatzigeorgiou, A. G. (2012). Functional microrna targets in protein coding sequences. *Bioinformatics*, 28(6):771–776.

Reese, W. (2008). Nginx: the high-performance web server and reverse proxy. *Linux Journal*, 2008(173):2.

Rehmsmeier, M., Steffen, P., Höchsmann, M., and Giegerich, R. (2004). Fast and effective prediction of microrna/target duplexes. *Rna*, 10(10):1507–1517.

Reis-Filho, J. S. (2009). Next-generation sequencing. *Breast cancer research*, 11(3):S12.

Rhoades, M. W., Reinhart, B. J., Lim, L. P., Burge, C. B., Bartel, B., and Bartel, D. P. (2002). Prediction of plant microrna targets. *cell*, 110(4):513–520.

Riffo-Campos, Á. L., Riquelme, I., and Brebi-Mieville, P. (2016). Tools for sequence-based mirna target prediction: what to choose? *International journal of molecular sciences*, 17(12):1987.

Riolo, G., Cantara, S., Marzocchi, C., and Ricci, C. (2020). mirna targets: from prediction tools to experimental validation. *Methods and protocols*, 4(1):1.

Rodríguez-Molina, J. B., West, S., and Passmore, L. A. (2023). Knowing when to stop: Transcription termination on protein-coding genes by eukaryotic rnapii. *Molecular cell*.

Sachs, M. C. (2017). plotroc: a tool for plotting roc curves. *Journal of statistical software*, 79.

Safavian, S. R. and Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229.

Sanger, F., Nicklen, S., and Coulson, A. R. (1977). Dna sequencing with chain-terminating inhibitors. *Proceedings of the national academy of sciences*, 74(12):5463–5467.

Schadt, E. E., Turner, S., and Kasarskis, A. (2010). A window into third-generation sequencing. *Human molecular genetics*, 19(R2):R227–R240.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.

Schnall-Levin, M., Rissland, O. S., Johnston, W. K., Perrimon, N., Bartel, D. P., and Berger, B. (2011). Unusually effective microrna targeting within repeat-rich coding regions of mammalian mrnas. *Genome research*, 21(9):1395–1403.

Schönemann, L., Kühn, U., Martin, G., Schäfer, P., Gruber, A. R., Keller, W., Zavolan, M., and Wahle, E. (2014). Reconstitution of cpsf active in polyadenylation: recognition of the polyadenylation signal by wdr33. *Genes & development*, 28(21):2381–2393.

Schwanhäusser, B., Busse, D., Li, N., Dittmar, G., Schuchhardt, J., Wolf, J., Chen, W., and Selbach, M. (2011). Global quantification of mammalian gene expression control. *Nature*, 473(7347):337.

Schwartz, B., Zaitsev, P., and Tkachenko, V. (2012). *High performance MySQL: optimization, backups, and replication*. " O'Reilly Media, Inc.".

Sclafani, J., Tirrell, T. F., and Franko, O. I. (2013). Mobile tablet use among academic physicians and trainees. *Journal of medical systems*, 37(1):9903.

Seok, H., Ham, J., Jang, E.-S., and Chi, S. W. (2016). Microrna target recognition: insights from transcriptome-wide non-canonical interactions. *Molecules and cells*, 39(5):375.

Sethupathy, P., Megraw, M., and Hatzigeorgiou, A. G. (2006). A guide through present computational approaches for the identification of mammalian microrna targets. *Nature methods*, 3(11):881–886.

Shabalina, S. A., Ogurtsov, A. Y., and Spiridonov, N. A. (2006). A periodic pattern of mrna secondary structure created by the genetic code. *Nucleic acids research*, 34(8):2428–2437.

Siepel, A., Bejerano, G., Pedersen, J. S., Hinrichs, A. S., Hou, M., Rosenbloom, K., Clawson, H., Spieth, J., Hillier, L. W., Richards, S., et al. (2005). Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome research*, 15(8):1034–1050.

Sindi, H. A., Russomanno, G., Satta, S., Abdul-Salam, V. B., Jo, K. B., Qazi-Chaudhry, B., Ainscough, A. J., Szulcek, R., Jan Bogaard, H., Morgan, C. C., et al. (2020). Therapeutic potential of klf2-induced exosomal micrornas in pulmonary hypertension. *Nature communications*, 11(1):1185.

Slatko, B. E., Gardner, A. F., and Ausubel, F. M. (2018). Overview of next-generation sequencing technologies. *Current protocols in molecular biology*, 122(1):e59.

Smith, C., Heyne, S., Richter, A. S., Will, S., and Backofen, R. (2010). Freiburg rna tools: a web server integrating intarna, exparna and locarna. *Nucleic acids research*, 38(suppl_2):W373–W377.

Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, 14:199–222.

Soneson, C., Love, M. I., and Robinson, M. D. (2015). Differential analyses for rna-seq: transcript-level estimates improve gene-level inferences. *F1000Research*, 4.

Speiser, J. L., Miller, M. E., Tooze, J., and Ip, E. (2019). A comparison of random forest variable selection methods for classification prediction modeling. *Expert systems with applications*, 134:93–101.

Srivastava, P. K., Moturu, T. R., Pandey, P., Baldwin, I. T., and Pandey, S. P. (2014). A comparison of performance of plant mirna target prediction tools and the characterization of features for genome-wide target prediction. *BMC genomics*, 15(1):348.

Stark, A., Brennecke, J., Bushati, N., Russell, R. B., and Cohen, S. M. (2005). Animal micrornas confer robustness to gene expression and have a significant impact on 3′ utr evolution. *Cell*, 123(6):1133–1146.

Stathakis, D. (2009). How many hidden layers and nodes? *International Journal of Remote Sensing*, 30(8):2133–2147.

Sun, L., Xu, K., Huang, W., Yang, Y. T., Li, P., Tang, L., Xiong, T., and Zhang, Q. C. (2021). Predicting dynamic cellular protein–rna interactions by deep learning using in vivo rna structures. *Cell research*, 31(5):495–516.

Tafer, H., Ameres, S. L., Obernosterer, G., Gebeshuber, C. A., Schroeder, R., Martinez, J., and Hofacker, I. L. (2008). The impact of target site accessibility on the design of effective sirnas. *Nature biotechnology*, 26(5):578–583.

Tamim, S., Vo, D. T., Uren, P. J., Qiao, M., Bindewald, E., Kasprzak, W. K., Shapiro, B. A., Nakaya, H. I., Burns, S. C., Araujo, P. R., et al. (2014). Genomic analyses reveal broad impact of mir-137 on genes associated with malignant transformation and neuronal differentiation in glioblastoma cells. *PloS one*, 9(1):e85591.

Tang, Y., Zhang, Y.-Q., Chawla, N. V., and Krasser, S. (2008). Svms modeling for highly imbalanced classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(1):281–288.

Tastsoglou, S., Alexiou, A., Karagkouni, D., Skoufos, G., Zacharopoulou, E., and Hatzigeorgiou, A. G. (2023). Diana-microt 2023: including predicted targets of virally encoded mirnas. *Nucleic Acids Research*, page gkad283.

Tavallaee, G., Lively, S., Rockel, J. S., Ali, S. A., Im, M., Sarda, C., Mitchell, G. M., Rossomacha, E., Nakamura, S., Potla, P., et al. (2022). Contribution of microrna-27b-3p to synovial fibrotic responses in knee osteoarthritis. *Arthritis & Rheumatology*, 74(12):1928–1942.

Teixeira, A. L., Dias, F., Gomes, M., Fernandes, M., and Medeiros, R. (2014). Circulating biomarkers in renal cell carcinoma: the link between micrornas and extracellular vesicles, where are we now? *Journal of kidney cancer and VHL*, 1(8):84.

Thalmann, M., Souza, A. S., and Oberauer, K. (2019). How does chunking help working memory? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 45(1):37.

The Bioconductor Dev Team (2014). *BSgenome.Hsapiens.NCBI.GRCh38: Full genome sequences for Homo sapiens (GRCh38)*. R package version 1.3.1000.

Thomas, A. J., Petridis, M., Walters, S. D., Gheytassi, S. M., and Morgan, R. E. (2017). Two hidden layers are usually better than one. In *Engineering Applications*

of Neural Networks: 18th International Conference, EANN 2017, Athens, Greece, August 25–27, 2017, Proceedings, pages 279–290. Springer.

Ui-Tei, K., Naito, Y., Nishi, K., Juni, A., and Saigo, K. (2008). Thermodynamic stability and watson–crick base pairing in the seed duplex are major determinants of the efficiency of the sirna-based off-target effect. *Nucleic acids research*, 36(22):7100–7109.

Van Buuren, S. and Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, 45:1–67.

Van Buuren, S. and Oudshoorn, K. (1999). *Flexible multivariate imputation by MICE*. Leiden: TNO.

Van Rossum, G. and Drake Jr, F. L. (1995). *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam.

Venables, W. N. and Ripley, B. D. (2013). *Modern applied statistics with S-PLUS*. Springer Science & Business Media.

W3C (2023). *CSS Snapshot*. `https://www.w3.org/TR/CSS/` [Accessed: 11-05-2024].

W3C (2024). *HTML Living Standard*. `https://html.spec.whatwg.org/` [Accessed: 11-05-2024].

Wagner, G. P., Kin, K., and Lynch, V. J. (2012). Measurement of mrna abundance using rna-seq data: Rpkm measure is inconsistent among samples. *Theory in biosciences*, 131:281–285.

Wang, S., Xu, J., Guo, Y., Cai, Y., Ren, X., Zhu, W., Geng, M., Meng, L., Jiang, C., and Lu, S. (2021). Microrna-497 reduction and increase of its family member microrna-424 lead to dysregulation of multiple inflammation related genes in synovial fibroblasts with rheumatoid arthritis. *Frontiers in Immunology*, 12:619392.

Wang, X. (2008). mirdb: a microrna target prediction and functional annotation database with a wiki interface. *Rna*, 14(6):1012–1017.

Wang, X. (2016). Improving microrna target prediction by modeling with unambiguously identified microrna-target pairs from clip-ligation studies. *Bioinformatics*, 32(9):1316–1322.

Wang, X. and El Naqa, I. M. (2008). Prediction of both conserved and nonconserved microrna targets in animals. *Bioinformatics*, 24(3):325–332.

Wang, X. and Wang, X. (2006). Systematic identification of microrna functions by combining target prediction and expression profiling. *Nucleic acids research*, 34(5):1646–1652.

Wang, Z., Gerstein, M., and Snyder, M. (2009). Rna-seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, 10(1):57.

Watson, J. D. and Crick, F. H. (1953a). Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–738.

Watson, J. D. and Crick, F. H. (1953b). The structure of dna. In *Cold Spring Harbor symposia on quantitative biology*, volume 18, pages 123–131. Cold Spring Harbor Laboratory Press.

Wes McKinney (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61.

Wheeler, D. L., Barrett, T., Benson, D. A., Bryant, S. H., Canese, K., Chetvernin, V., Church, D. M., DiCuccio, M., Edgar, R., Federhen, S., et al. (2007). Database resources of the national center for biotechnology information. *Nucleic acids research*, 35(suppl_1):D5–D12.

Wong, N. and Wang, X. (2015). mirdb: an online resource for microrna target prediction and functional annotations. *Nucleic acids research*, 43(D1):D146–D152.

Wright, P. R., Georg, J., Mann, M., Sorescu, D. A., Richter, A. S., Lott, S., Kleinkauf, R., Hess, W. R., and Backofen, R. (2014). Coprarna and intarna: predicting small rna targets, networks and interaction domains. *Nucleic acids research*, 42(W1):W119–W123.

Wright, P. R., Richter, A. S., Papenfort, K., Mann, M., Vogel, J., Hess, W. R., Backofen, R., and Georg, J. (2013). Comparative genomics boosts target prediction for bacterial small rnas. *Proceedings of the National Academy of Sciences*, 110(37):E3487–E3496.

Wu, T.-F., Lin, C.-J., and Weng, R. (2003). Probability estimates for multi-class classification by pairwise coupling. *Advances in Neural Information Processing Systems*, 16.

Xu, W., San Lucas, A., Wang, Z., and Liu, Y. (2014). Identifying microrna targets in different gene regions. *BMC bioinformatics*, 15:1–11.

Xu, Y. and Goodacre, R. (2018). On splitting training and validation set: a comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. *Journal of analysis and testing*, 2(3):249–262.

Yang, J.-S., Phillips, M. D., Betel, D., Mu, P., Ventura, A., Siepel, A. C., Chen, K. C., and Lai, E. C. (2011). Widespread regulatory activity of vertebrate microrna* species. *Rna*, 17(2):312–326.

Ying, X. (2019). An overview of overfitting and its solutions. In *Journal of physics: Conference series*, volume 1168, page 022022. IOP Publishing.

You, E. (2023). *Vue.js Introduction*. `https://vuejs.org/guide/introduction.html` [Accessed: 18-09-2023].

You, E. (2024). *Vite Guide*. `https://vitejs.dev/guide/` [Accessed: 11-05-2024].

Yusupova, G. Z., Yusupov, M. M., Cate, J., and Noller, H. F. (2001). The path of messenger rna through the ribosome. *Cell*, 106(2):233–241.

Zeitlin, S., Parent, A., Silverstein, S., and Efstratiadis, A. (1987). Pre-mrna splicing and the nuclear matrix. *Molecular and Cellular Biology*, 7(1):111–120.

Zhang, B., Wang, Q., and Pan, X. (2007). Micrornas and their regulatory roles in animals and plants. *Journal of cellular physiology*, 210(2):279–289.

Zhou, X., Duan, X., Qian, J., and Li, F. (2009). Abundant conserved microrna target sites in the 5′-untranslated region and coding sequence. *Genetica*, 137:159–164.

Zhu, A., Ibrahim, J. G., and Love, M. I. (2019). Heavy-tailed prior distributions for sequence count data: removing the noise and preserving large differences. *Bioinformatics*, 35(12):2084–2092.

Zhuang, Z., Yu, P., Xie, N., Wu, Y., Liu, H., Zhang, M., Tao, Y., Wang, W., Yin, H., Zou, B., et al. (2020). Microrna-204-5p is a tumor suppressor and potential therapeutic target in head and neck squamous cell carcinoma. *Theranostics*, 10(3):1433.

Appendix A

Additional Datasets

The following datasets were used in Chapter 5 to increase the number of transfections available to ML.

Table A.1: Additional datasets used to support ML

Internal ID	02-wan-SW982
Accession	PRJNA669842
Species	*Homo sapiens*
Data Type	RNA-seq
Procedure	2 miRNA transfections
Cell Line	SW982
Biological Replicates	4
Sequence Type	Paired-end
Source	Wang et al. (2021)

Internal ID	03-lmu-HUVEC
Accession	PRJNA784113
Species	*Homo sapiens*
Data Type	RNA-seq
Procedure	1 miRNA transfection
Cell Line	HUVEC
Biological Replicates	5
Sequence Type	Paired-end
Source	Müller et al. (2022)

Internal ID	04-bev-22Rv1
Accession	PRJNA674323
Species	*Homo sapiens*
Data Type	RNA-seq
Procedure	1 miRNA transfection
Cell Line	22Rv1
Biological Replicates	3
Sequence Type	Single-end
Source	Beveridge et al. (2021)

Internal ID	05-ham-LX-2
Accession	PRJNA665381
Species	*Homo sapiens*
Data Type	RNA-seq
Procedure	1 miRNA transfection
Cell Line	LX-2
Biological Replicates	2
Sequence Type	Paired-end
Source	Hamada-Tsutsumi et al. (2020)

Internal ID	06-kim-MDA-MB-231
Accession	PRJNA625036
Species	*Homo sapiens*
Data Type	RNA-seq
Procedure	1 miRNA transfection
Cell Line	MDA-MB-231
Biological Replicates	2
Sequence Type	Single-end
Source	Kim et al. (2020)

Internal ID	07-hms-HUVEC
Accession	PRJNA439194
Species	*Homo sapiens*
Data Type	RNA-seq
Procedure	1 miRNA transfection
Cell Line	HUVEC
Biological Replicates	2
Sequence Type	Paired-end
Source	Data accessible at NCBI GEO database (Edgar et al., 2002), accession GSE112059

Internal ID	08-mor-HEK293T
Accession	PRJNA528188
Species	*Homo sapiens*
Data Type	RNA-seq
Procedure	2 miRNA transfections
Cell Line	HEK-293-T
Biological Replicates	3
Sequence Type	Single-end
Source	Morrison et al. (2019)

Internal ID	09-tam-U251, 10-tam-U343
Accession	PRJNA231155
Species	*Homo sapiens*
Data Type	RNA-seq
Procedure	1, 1 miRNA transfection
Cell Line	U251, U343
Biological Replicates	2
Sequence Type	Single-end
Source	Tamim et al. (2014)

Internal ID	11-nam-IMR90, 12-nam-Huh7, 13-nam-HEK293, 14-nam-HeLa
Accession	PRJNA229375
Species	*Homo sapiens*
Data Type	RNA-seq
Procedure	1, 2, 2, 2 miRNA transfections
Cell Line	IMR-90, Huh-7, HEK-293, HeLa
Biological Replicates	2
Sequence Type	Single-end
Source	Nam et al. (2014)

Internal ID	15-hu-293T
Accession	PRJNA362193
Species	*Homo sapiens*
Data Type	RNA-seq
Procedure	1 miRNA transfection
Cell Line	HEK-293-T
Biological Replicates	3
Sequence Type	Single-end
Source	Hu et al. (2018)

Internal ID	16-icl-HPAEC
Accession	PRJNA531359
Species	*Homo sapiens*
Data Type	RNA-seq
Procedure	2 miRNA transfections
Cell Line	HPAEC
Biological Replicates	2
Sequence Type	Paired-end
Source	Sindi et al. (2020)

Internal ID	17-guo-HeLa
Accession	PRJNA129385
Species	*Homo sapiens*
Data Type	RNA-seq
Procedure	2 miRNA transfections
Cell Line	HeLa
Biological Replicates	3
Sequence Type	Single-end
Source	Guo et al. (2010)

Internal ID	18-mit-A549
Accession	PRJNA597999
Species	*Homo sapiens*
Data Type	RNA-seq
Procedure	2 miRNA transfections
Cell Line	A549
Biological Replicates	3
Sequence Type	Paired-end
Source	Guo et al. (2010)

Internal ID	18-mit-A549
Accession	PRJNA597999
Species	*Homo sapiens*
Data Type	RNA-seq
Procedure	2 miRNA transfections
Cell Line	A549
Biological Replicates	3
Sequence Type	Paired-end
Source	Mitra et al. (2020)

Internal ID	19-tav-FLS
Accession	PRJNA639874
Species	*Homo sapiens*
Data Type	RNA-seq
Procedure	1 miRNA transfection
Cell Line	FLS
Biological Replicates	3
Sequence Type	Paired-end
Source	Tavallaee et al. (2022)

Internal ID	20-sob-MSC
Accession	PRJNA834773
Species	*Homo sapiens*
Data Type	RNA-seq
Procedure	2 miRNA transfections
Cell Line	MSC
Biological Replicates	3
Sequence Type	Paired-end
Source	Data accessible at NCBI GEO database (Edgar et al., 2002), accession GSE202135

Internal ID	21-zhu-UM-SCC-1
Accession	PRJNA528871
Species	*Homo sapiens*
Data Type	RNA-seq
Procedure	1 miRNA transfection
Cell Line	UM-SCC-1
Biological Replicates	3
Sequence Type	Paired-end
Source	Zhuang et al. (2020)

Internal ID	22-sxh-PBMC
Accession	PRJNA607802
Species	*Homo sapiens*
Data Type	RNA-seq
Procedure	1 miRNA transfection
Cell Line	PBMC
Biological Replicates	3
Sequence Type	Paired-end
Source	Data accessible at NCBI GEO database (Edgar et al., 2002), accession GSE145652

Appendix B

Individual Benchmark Results

The following results were generated in Section 5.3.3 to compare the prediction accuracy of miRsight against TargetScan, MirTarget and DIANA-microT.

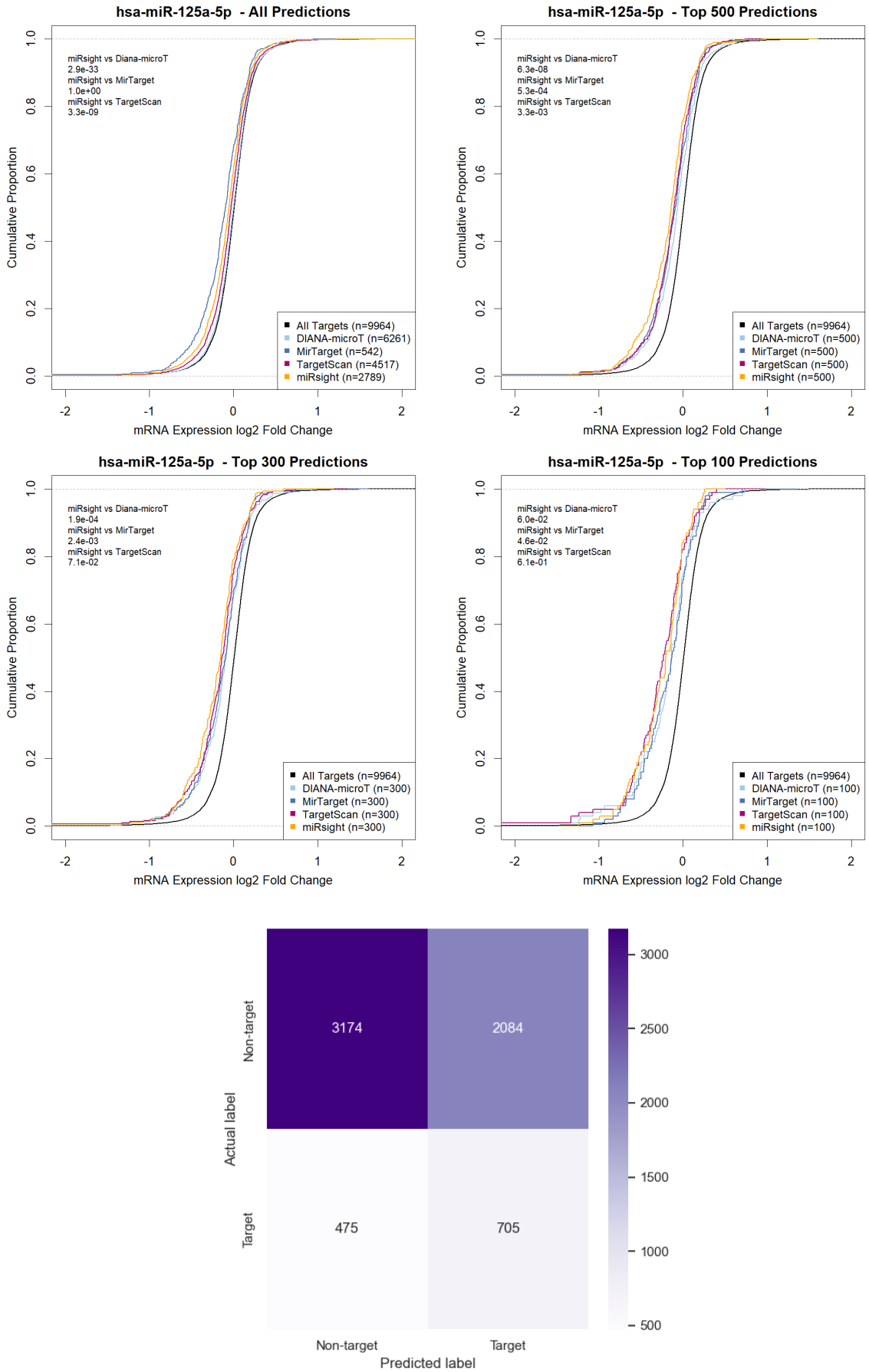


Figure B.1: Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-125a-5p. (Top left) All predictions. (Top right) Top 500 predictions. (Middle left) Top 300 predictions. (Middle right) Top 100 predictions. (Bottom) miRsight confusion matrix.

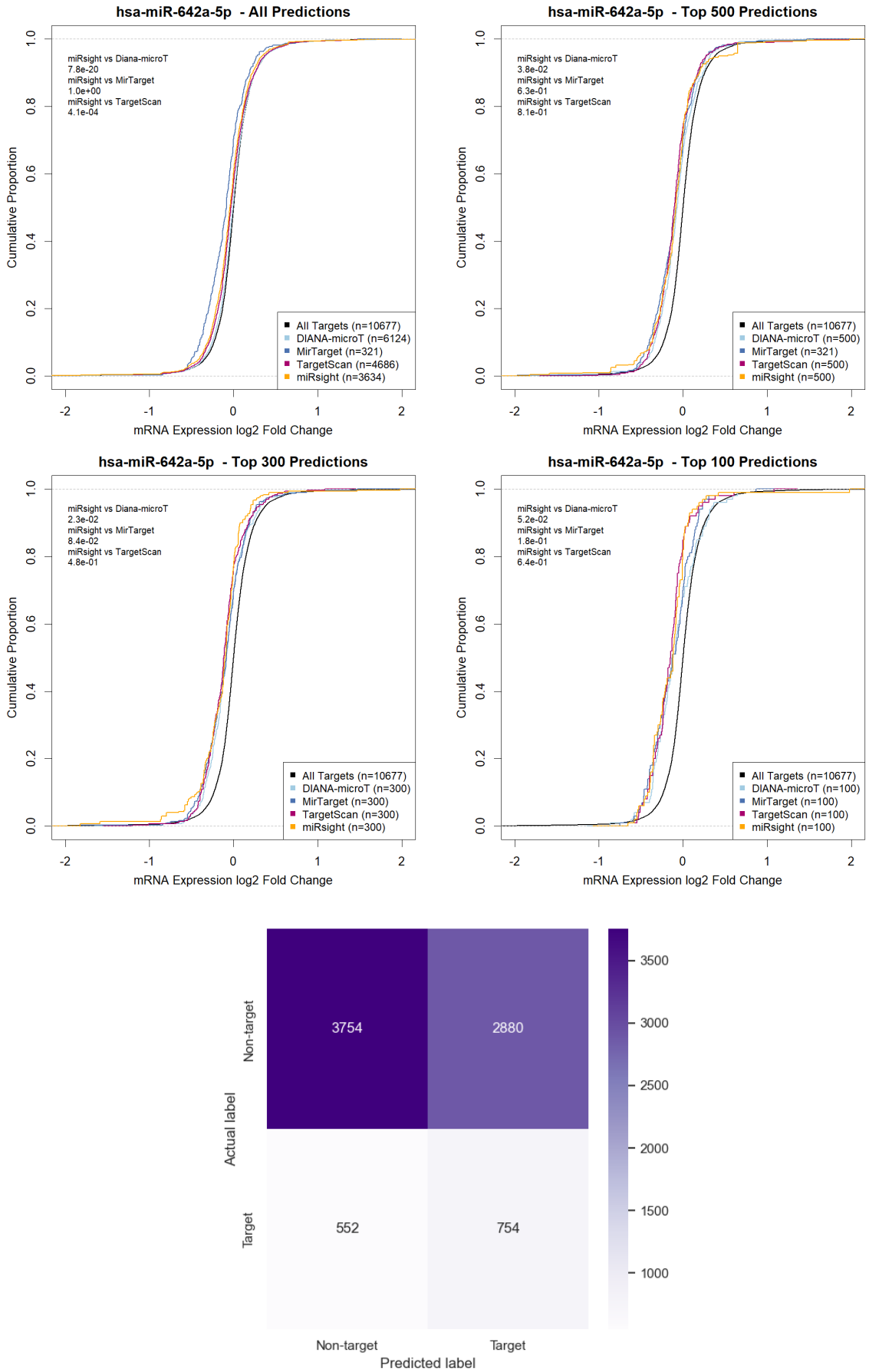


Figure B.2: Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-642a-5p. Note that MirTarget does not produce the required 500 predictions in the top 500 category. (Top left) All predictions. (Top right) Top 500 predictions. (Middle left) Top 300 predictions. (Middle right) Top 100 predictions. (Bottom) miRsight confusion matrix.

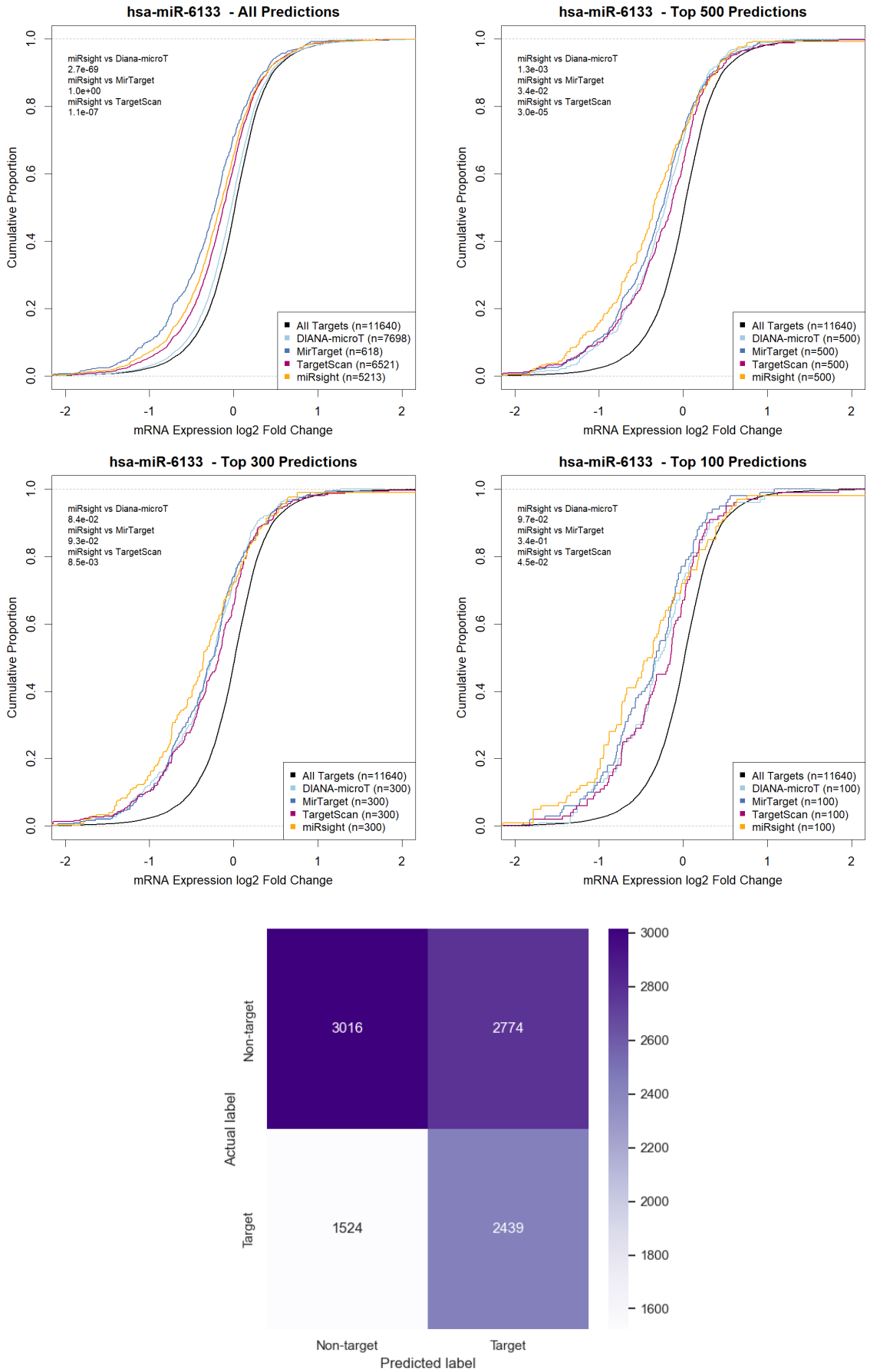


Figure B.3: Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-6133. (Top left) All predictions. (Top right) Top 500 predictions. (Middle left) Top 300 predictions. (Middle right) Top 100 predictions. (Bottom) miRsight confusion matrix.

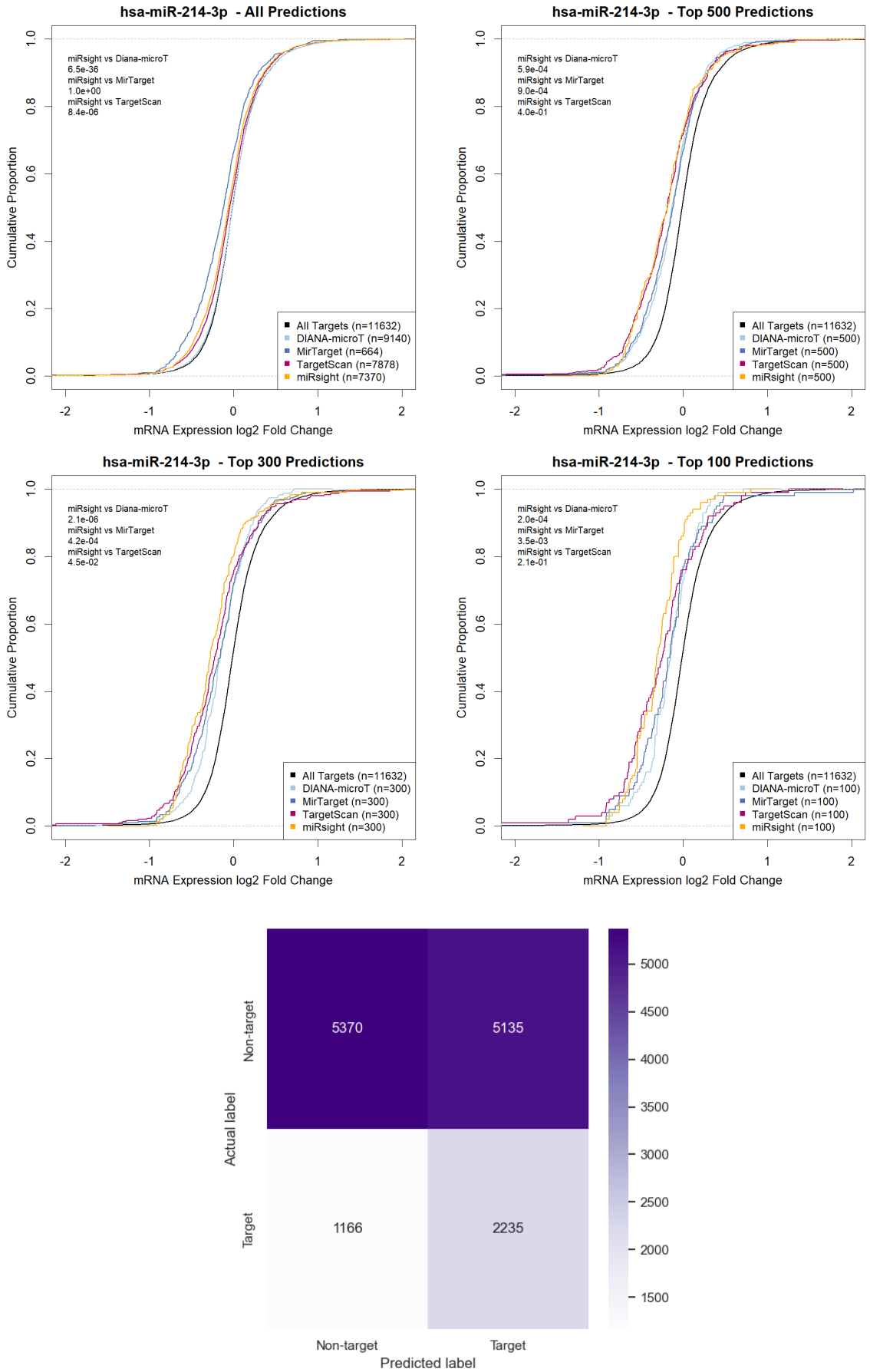


Figure B.4: Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-214-3p. (Top left) All predictions. (Top right) Top 500 predictions. (Middle left) Top 300 predictions. (Middle right) Top 100 predictions. (Bottom) miRsight confusion matrix.

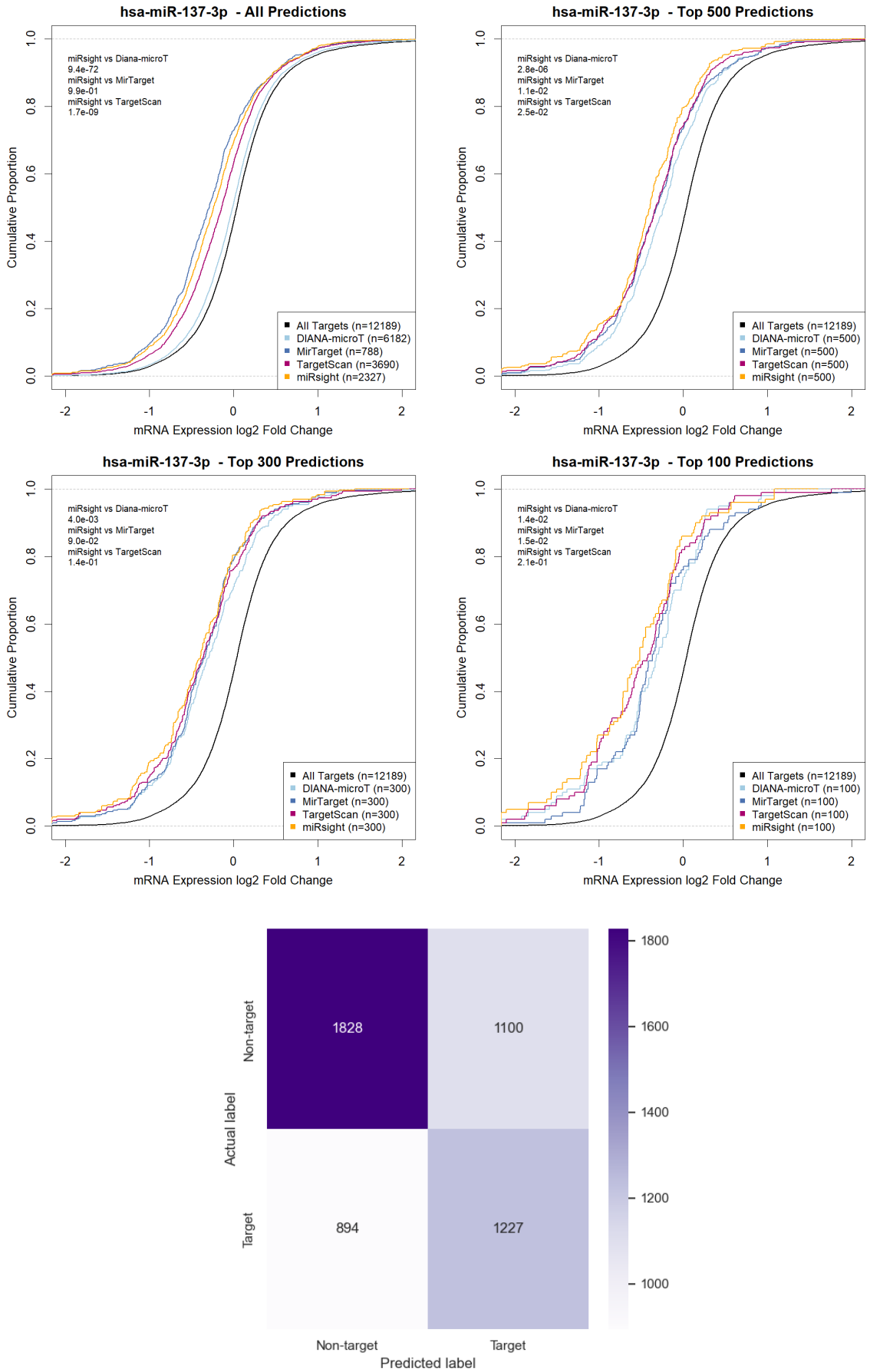


Figure B.5: Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-137-3p. (Top left) All predictions. (Top right) Top 500 predictions. (Middle left) Top 300 predictions. (Middle right) Top 100 predictions. (Bottom) miRsight confusion matrix.

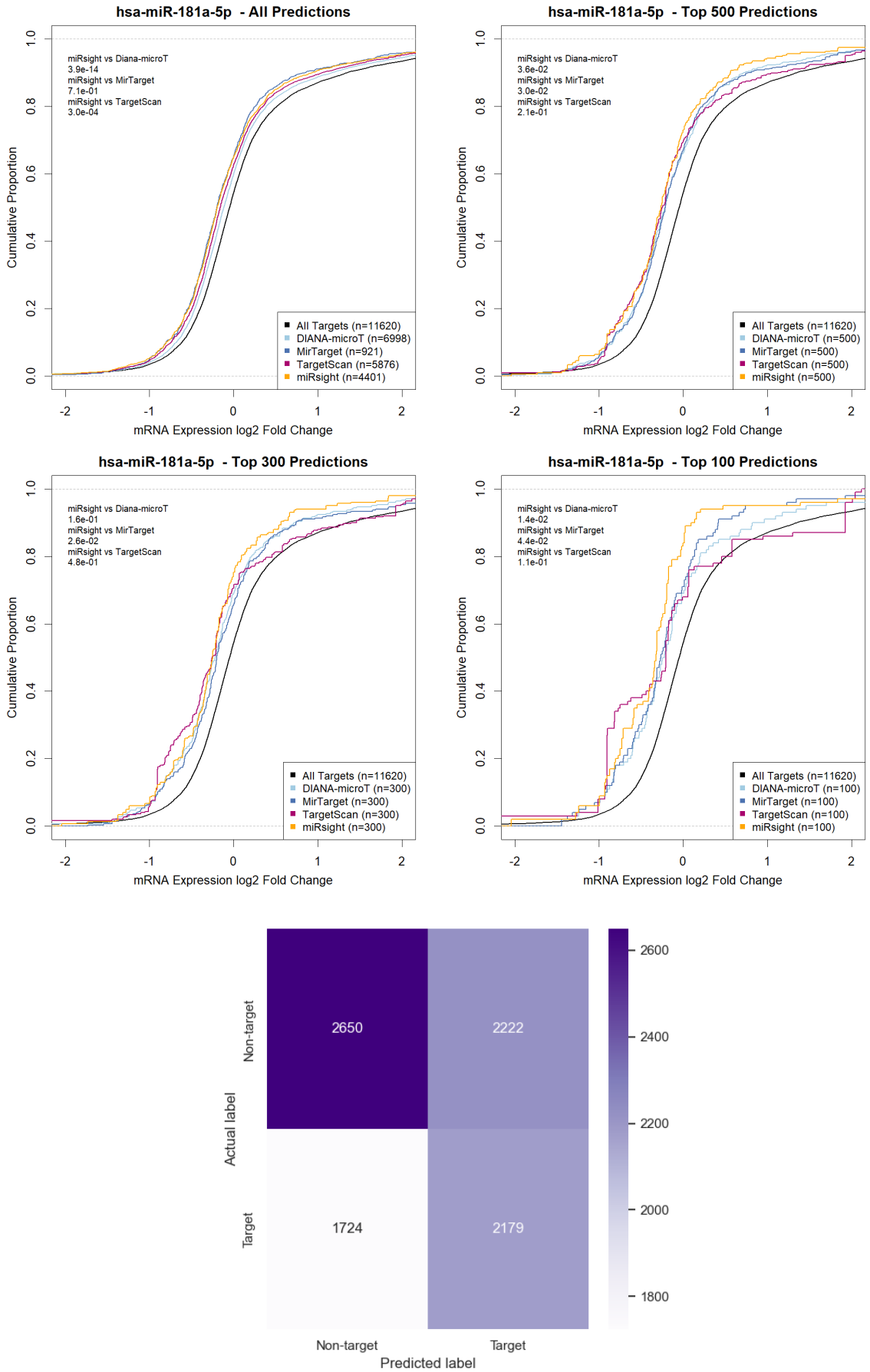


Figure B.6: Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-181a-5p. (Top left) All predictions. (Top right) Top 500 predictions. (Middle left) Top 300 predictions. (Middle right) Top 100 predictions. (Bottom) miRsight confusion matrix.

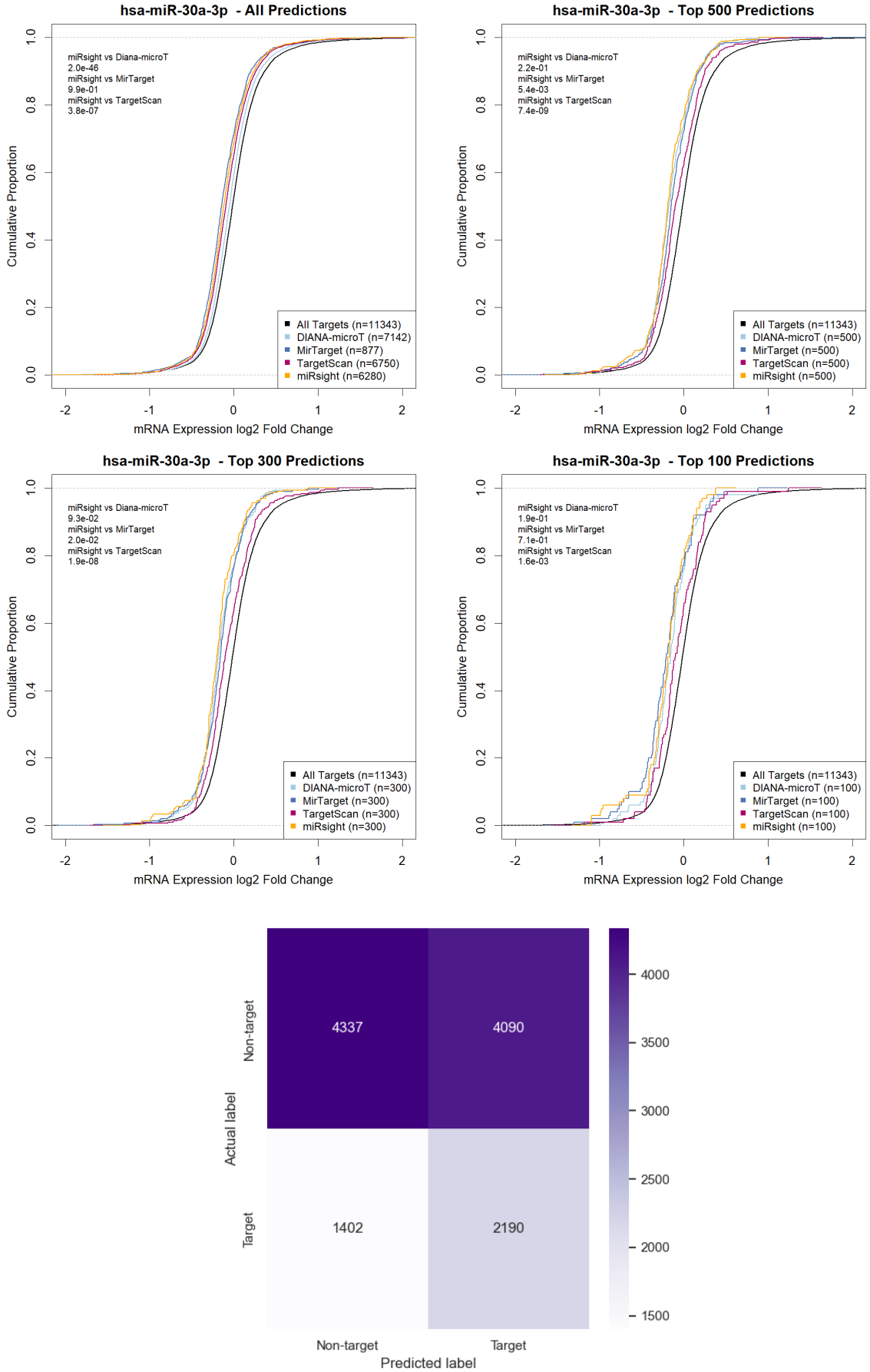


Figure B.7: Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-30a-3p. (Top left) All predictions. (Top right) Top 500 predictions. (Middle left) Top 300 predictions. (Middle right) Top 100 predictions. (Bottom) miRsight confusion matrix.

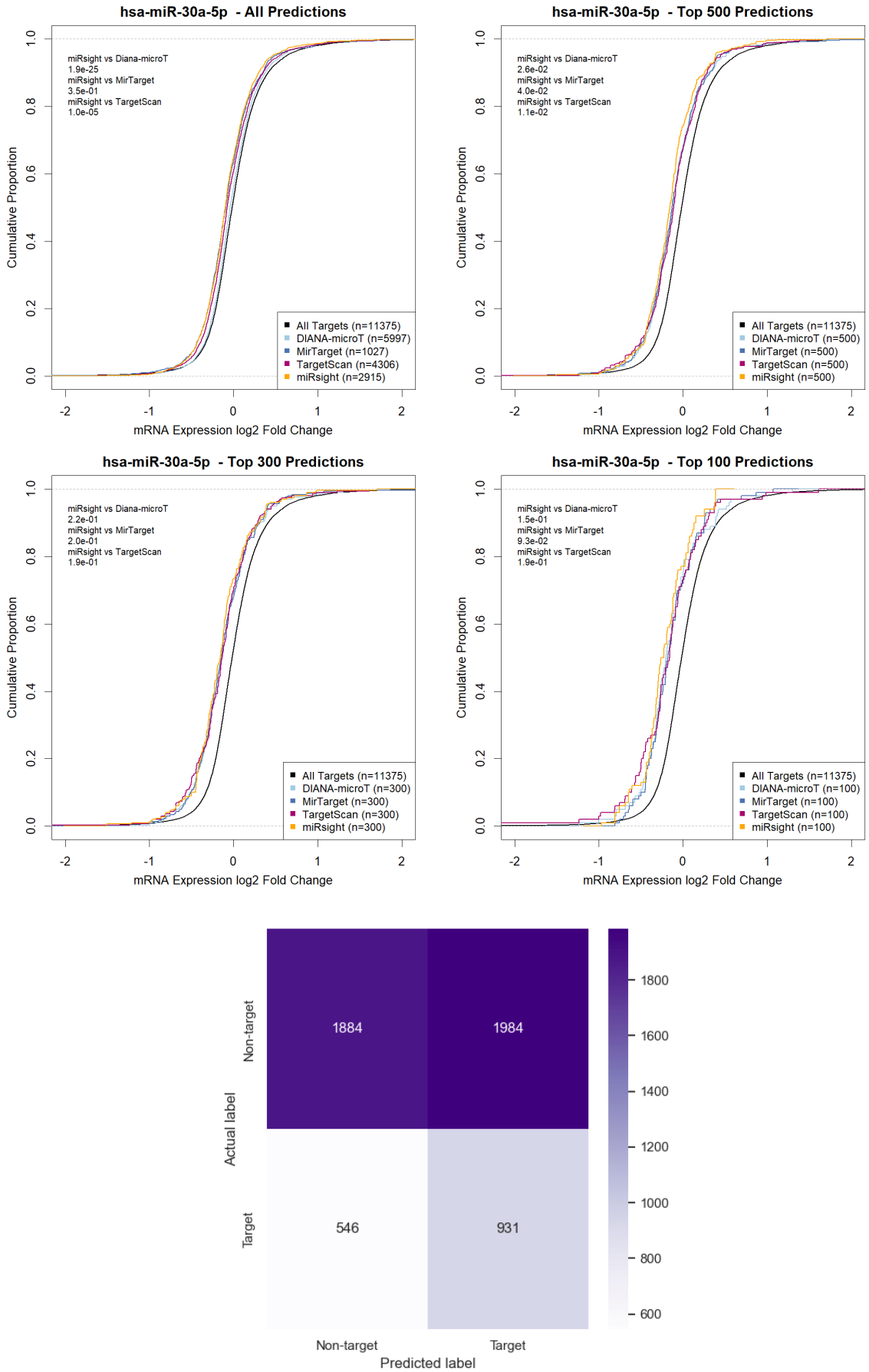


Figure B.8: Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-30a-5p. (Top left) All predictions. (Top right) Top 500 predictions. (Middle left) Top 300 predictions. (Middle right) Top 100 predictions. (Bottom) miRsight confusion matrix.

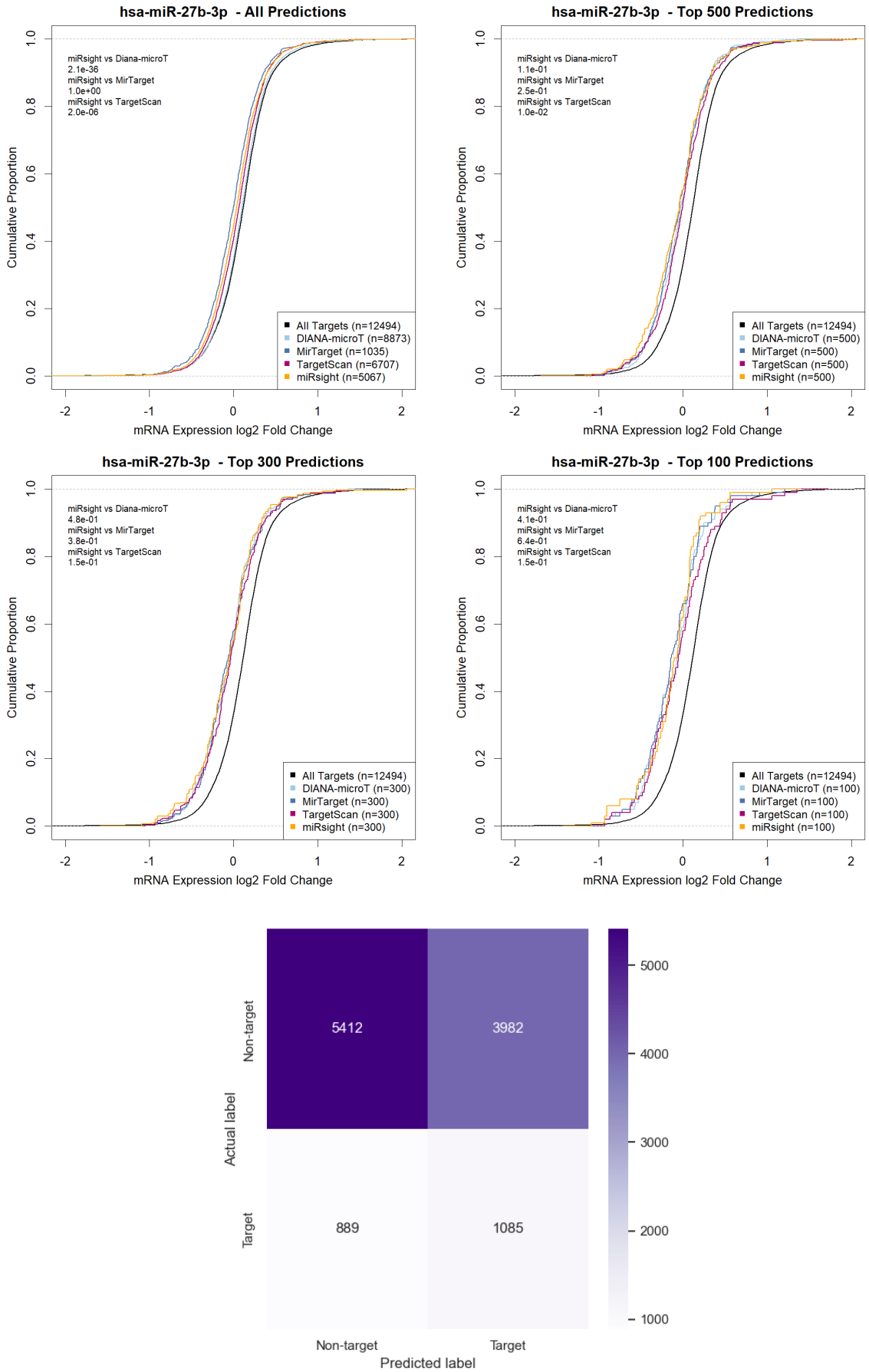


Figure B.9: Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-27b-3p. (Top left) All predictions. (Top right) Top 500 predictions. (Middle left) Top 300 predictions. (Middle right) Top 100 predictions. (Bottom) miRsight confusion matrix.

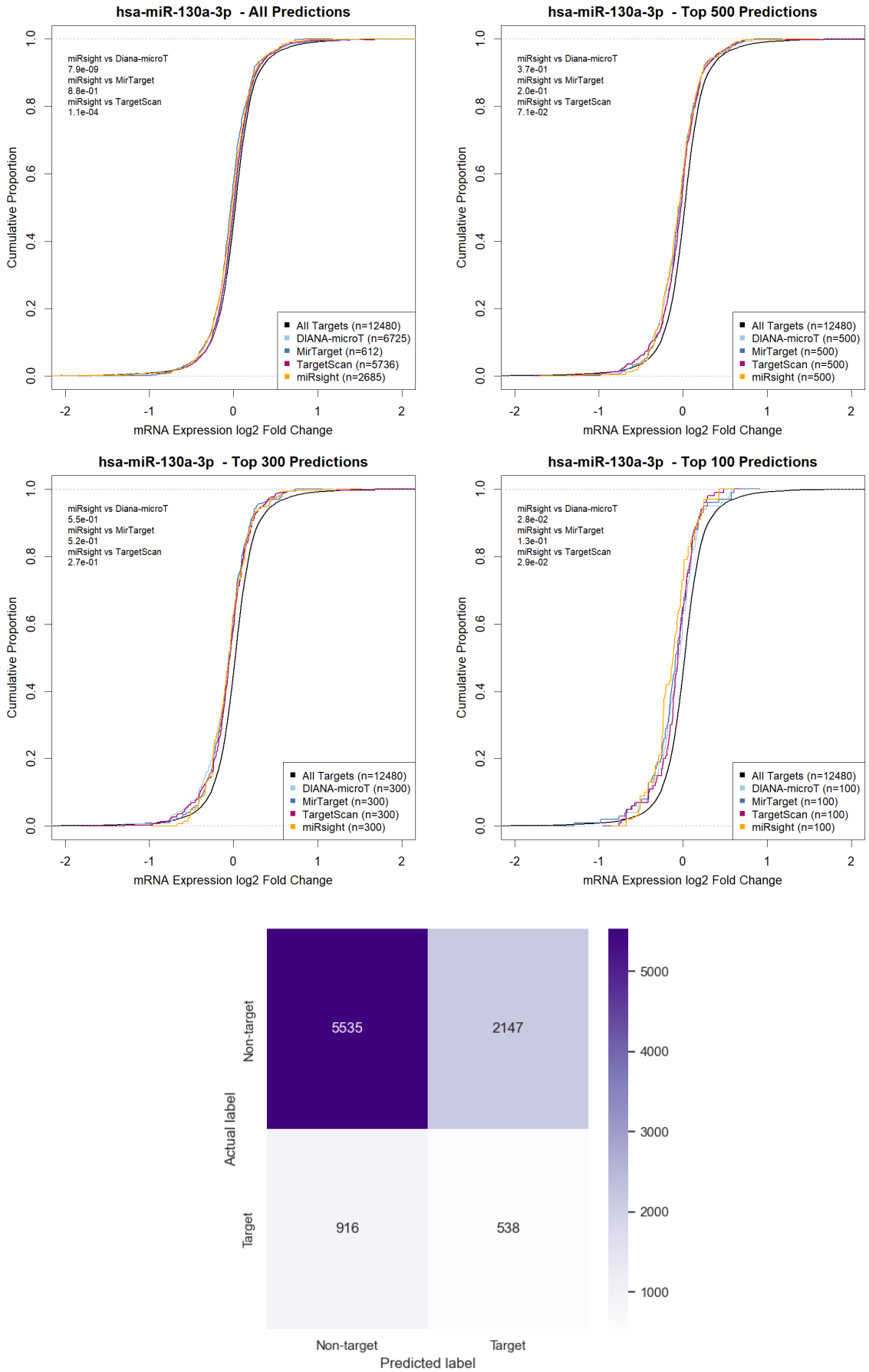


Figure B.10: Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-130a-3p. (Top left) All predictions. (Top right) Top 500 predictions. (Middle left) Top 300 predictions. (Middle right) Top 100 predictions. (Bottom) miRsight confusion matrix.

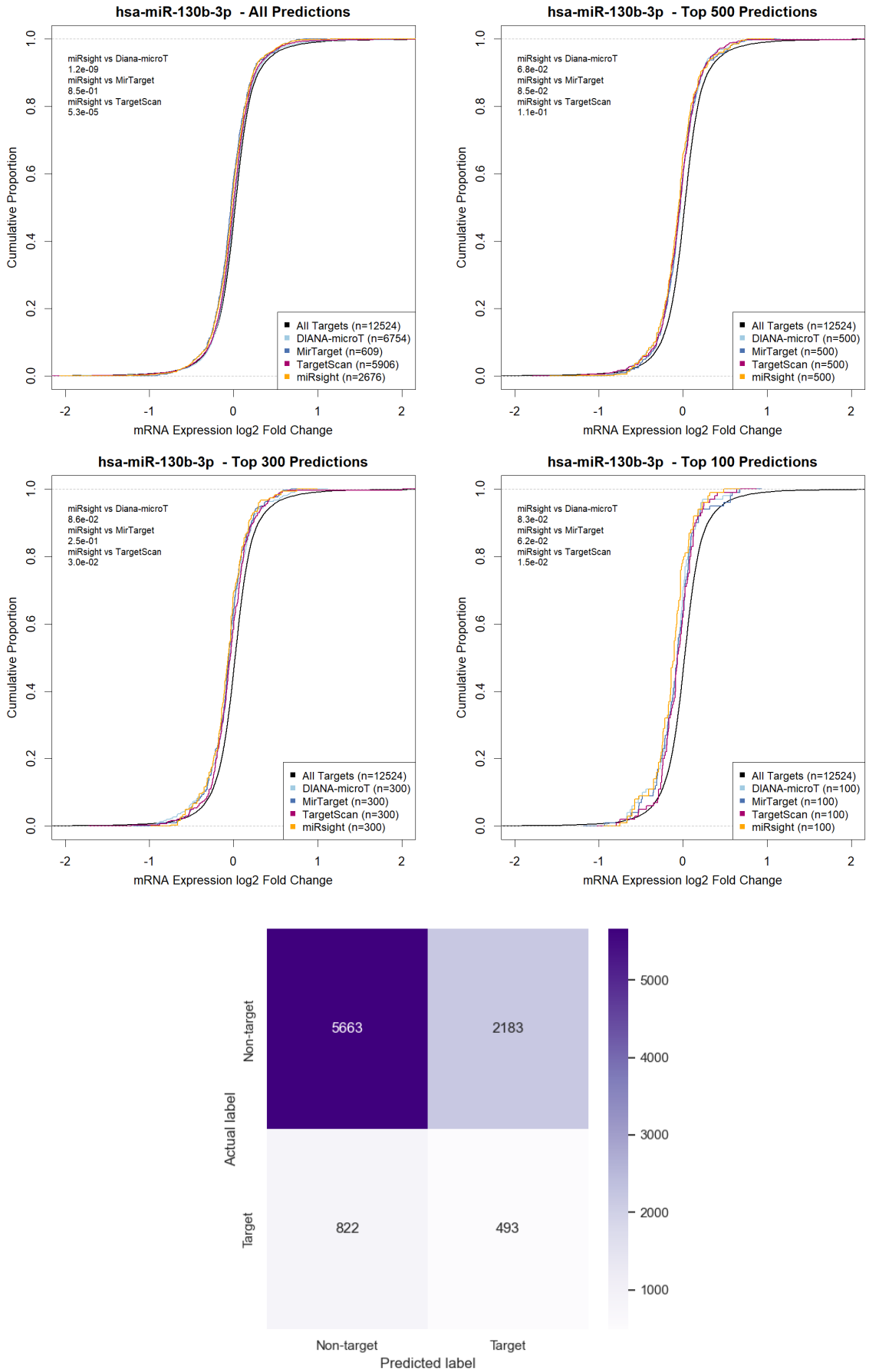


Figure B.11: Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-130b-3p. (Top left) All predictions. (Top right) Top 500 predictions. (Middle left) Top 300 predictions. (Middle right) Top 100 predictions. (Bottom) miRsight confusion matrix.

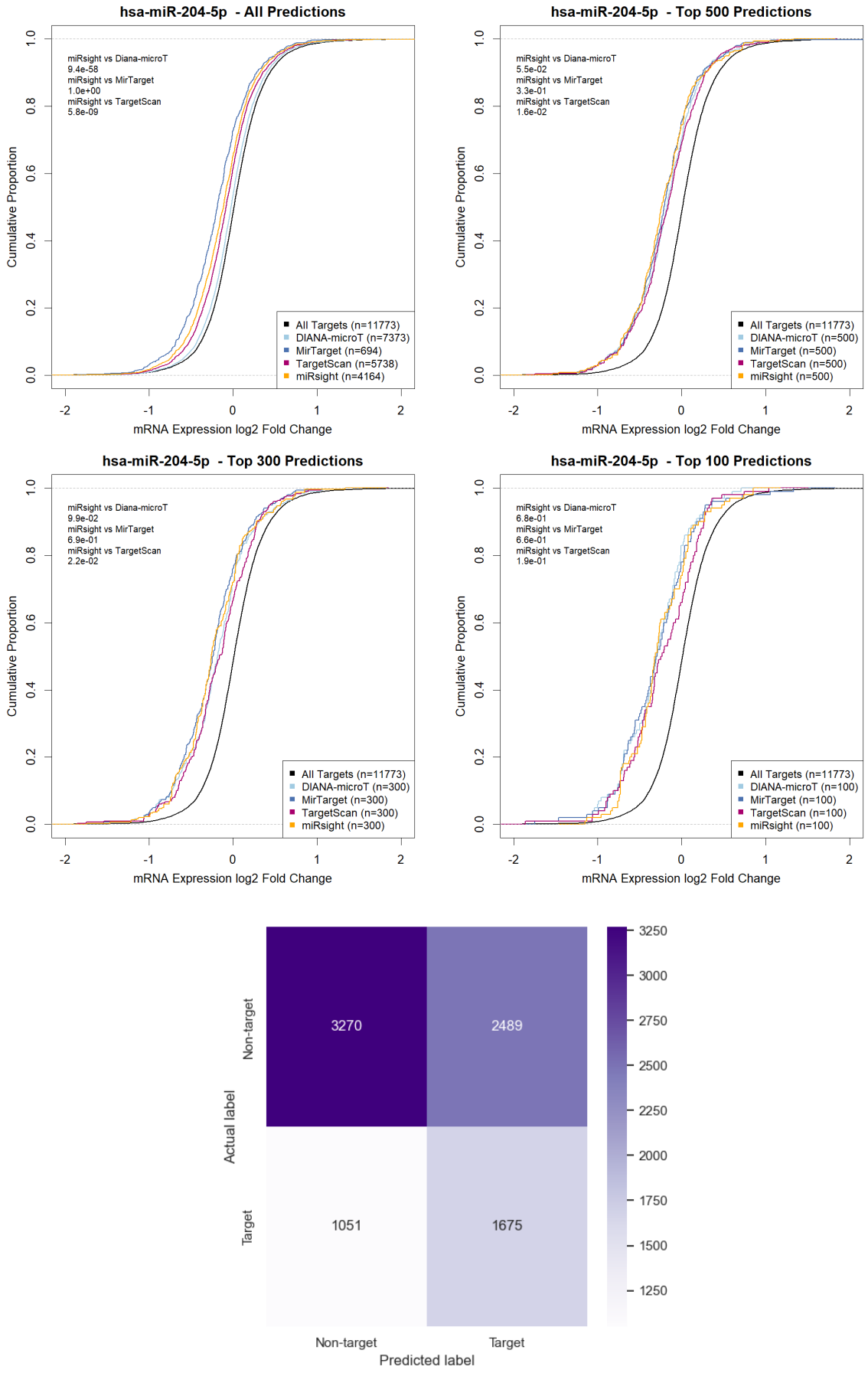


Figure B.12: Benchmark comparison of miRsight predictions against TargetScan, MirTarget and DIANA-microT for miR-204-5p. (Top left) All predictions. (Top right) Top 500 predictions. (Middle left) Top 300 predictions. (Middle right) Top 100 predictions. (Bottom) miRsight confusion matrix.

Appendix C

Unused Datasets

The following datasets were used for research in some capacity, but are not integrated in the results of any final deliverable.

Table C.1: Dataset EX-guo-U20S summary

Internal ID	EX-guo-U20S
Accession	PRJNA223608
Species	*Homo sapiens*
Data Type	RNA-seq
Procedure	2 miRNA transfections
Cell Line	U20S
Biological Replicates	1
Sequence Type	Single-end
Source	Guo et al. (2014a)