IMPERIAL COLLEGE LONDON

DOCTORAL THESIS

Non-parametric machine learning for biological sequence data

Author:
Jonathan ISH-HOROWICZ

Supervisor:
Dr. Sarah FILIPPI
Prof. William Cookson

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Department of Mathematics
Imperial College London

August 8, 2022

Declaration of Authorship

I, Jonathan ISH-HOROWICZ, declare that this thesis titled, "Non-parametric machine learning for biological sequence data" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

IMPERIAL COLLEGE LONDON

Abstract

Faculty of Natural Sciences
Imperial College London

Doctor of Philosophy

Non-parametric machine learning for biological sequence data

by Jonathan ISH-HOROWICZ

In the past decade there has been a massive increase in the volume of biological sequence data, driven by massively parallel sequencing technologies. This has enabled data-driven statistical analyses using non-parametric predictive models (including those from machine learning) to complement more traditional, hypothesis-driven approaches. This thesis addresses several challenges that arise when applying non-parametric predictive models to biological sequence data.

Some of these challenges arise due to the nature of the biological system of interest. For example, in the study of the human microbiome the phylogenetic relationships between microorganisms are often ignored in statistical analyses. This thesis outlines a novel approach to modelling phylogenetic similarity using string kernels and demonstrates its utility in the two-sample test and host-trait prediction.

Other challenges arise from limitations in our understanding of the models themselves. For example, calculating variable importance (a key task in biomedical applications) is not possible for many models. This thesis describes a novel extension of an existing approach to compute importance scores for grouped variables in a Bayesian neural network. It also explores the behaviour of random forest classifiers when applied to microbial datasets, with a focus on the robustness of the biological findings under different modelling assumptions.

Acknowledgements

I have been very lucky to work with such kind and brilliant people during my PhD. Thank you to Sarah, Bill and Miriam for your insights, help and unwavering support. Thank you also to Leah for teaching me everything I know about sequencing data and to Lorin and Seth for all your help with RATE over the years. Thank you Anna, for always believing in me and encouraging me to leave the house. And most especially, thank you to Heather, Léonie, Maddy and Tara for keeping me company through these long years.

Contents

Declaration of Authorship iii

Abstract v

Acknowledgements vii

1 Introduction 1
1 The importance of sequence data in biological research . . . 1
1.1 The human microbiome . . . 2
1.2 The lung microbiome . . . 2
1.3 Collection of microbial datasets . . . 3
Taxonomic ranks . . . 3
Microbial identification and quantification via 16S rRNA and ITS2 sequencing . . . 3
2 Non-parametric predictive modelling . . . 6
2.1 Machine learning in biology . . . 6
3 Data . . . 7
4 Thesis contributions . . . 7
4.1 Chapter contributions . . . 8
4.2 Other work . . . 10

2 Mathematical Background 11
1 Supervised learning . . . 11
1.1 Gaussian process regression . . . 12
The Mean Function and Kernel . . . 12
Gaussian process posterior and predictive posterior . . . 13
Model selection in Gaussian process regression . . . 15
The Kernel Trick . . . 15
1.2 Sparse Gaussian process regression . . . 16
Background and motivation . . . 16
SGPR model description . . . 17
SGPR training via variational inference . . . 18
1.3 Deterministic neural networks . . . 20
Training the network . . . 20
1.4 Bayesian neural networks . . . 21
Benefits of prediction uncertainty . . . 21
Bayesian neural network model description . . . 21
Approximate inference for Bayesian neural networks . . . 22
Bayesian neural network training via variational inference . . . 23
Last layer Bayesian neural networks . . . 24
Last layer Bayesian regression model . . . 24
1.5 Decision tree ensembles . . . 25
2 Variable importance analyses . . . 27

2.1 Interpretability via variable importance . . . 28
Global and local variable importance . . . 28
2.2 Variable importance vs variable selection . . . 29
2.3 Types of variable importance methods . . . 30
2.4 Model-agnostic variable importance measures . . . 30
Permutation importance . . . 30
Shapley values . . . 30
2.5 Variable importance methods for neural networks . . . 31
Saliency maps . . . 31
Description of saliency-based methods used in this thesis . . . 32
Mimic models . . . 33
2.6 Variable importance for random forests . . . 33
Mean decrease Gini and Mean decrease accuracy . . . 34
Assessing statistical significance for random forest scores . . . 35
3 RATE (RelATive cEntrality) . . . 36
3.1 Calculating RATE scores . . . 36
3.2 Effect size analogues . . . 36
3.3 Variable importance using relative centrality measures . . . 37

3 Differential abundance and two-sampling testing of microbial airway communities using random forest 39
1 The role of the microbiome in respiratory disease . . . 39
2 Study aims . . . 40
3 Two-sample testing with binary classifiers . . . 41
4 Relevant work . . . 42
4.1 Random forests for microbiome . . . 42
4.2 Classifier-based two-sample testing . . . 43
5 Data . . . 43
5.1 Quantifying community composition . . . 43
5.2 Transforming taxa abundances . . . 45
5.3 Sample groups . . . 45
6 Two-sample testing using random forests . . . 46
6.1 Nested cross-validation estimates of generalisation performance 46
6.2 Results of the two-sample test . . . 47
6.3 Coverage of LeDell's confidence intervals . . . 48
6.4 Two-sample testing using LeDell confidence intervals agree with permutation tests . . . 49
6.5 DeLong's test . . . 50
Type I error rate of DeLong's test . . . 51
6.6 Effect of cross-kingdom interactions on discriminative power . 52
7 Random forest variable importance for differential abundance . . . 53
7.1 Variable rankings under different variable importance methods and transformations . . . 54
Statistical significance . . . 57
7.2 Stability of variable importance scores . . . 59
8 Discussion . . . 61

4 Grouped variable prioritisation for Bayesian neural networks 67
1 Chapter aims and contributions . . . 67
2 Computing variable importances using RATE . . . 68
2.1 Variable prioritisation vs variable selection . . . 68

2.2 Recap of the RATE calculation . . . 68
2.3 Interpretation of RATE scores . . . 70
2.4 Three-variable toy example . . . 71
3 Alternative projections for effect sizes analogues . . . 73
3.1 Computing the ESA posterior . . . 74
4 GroupRATE: variable prioritisation for grouped variables . . . 74
4.1 Calculating GroupRATE values . . . 75
4.2 Calculating GroupRATE for Bayesian neural networks . . . 75
Last layer Bayesian regression network . . . 75
Computing GroupRATE values in closed-form . . . 76
5 Group prioritisation simulations: simulated covariates . . . 78
5.1 Simulation aims . . . 78
5.2 A group-dependent covariance structure . . . 78
5.3 Phenotype model . . . 79
5.4 Strong hierarchy assumption . . . 80
5.5 Selecting main and pairwise effects . . . 81
5.6 Final simulation procedure . . . 82
5.7 AUC as an evaluation metric for group prioritisation . . . 82
5.8 Predictive performance of the Bayesian neural network . . . 83
5.9 Correcting the bias in KL-divergences from group size . . . 83
5.10 Calculating group-level importance scores . . . 84
5.11 Evaluation of group prioritisation methods . . . 86
5.12 Empirical computation times . . . 86
6 Genotype simulations . . . 89
6.1 Simulation aims . . . 89
6.2 Simulation data . . . 91
6.3 Description of models . . . 91
6.4 Final simulation setup . . . 92
6.5 Predictive performance of the four models . . . 92
6.6 Group prioritisation performance . . . 93
7 Real data applications from computer vision . . . 95
7.1 Bayesian neural network classifier . . . 95
7.2 Global variable importances for images using saliency maps . . . 97
7.3 MNIST . . . 98
7.4 Automatic diagnosis of pneumonia from chest X-rays . . . 99
Distribution shift in medical imaging datasets . . . 99
Predictive performance of a convolutional neural network . . . 101
Defining groups based on pixel correlation . . . 101
GroupRATE suggests different explanations on training and validation data . . . 102
8 Discussion . . . 102

5 Modelling phylogeny in microbial datasets using string kernels 107
1 Chapter aims and contributions . . . 107
2 Characteristics of 16S rRNA datasets . . . 108
2.1 Essential components of microbial datasets . . . 108
2.2 Phylogenetic trees . . . 109
2.3 Interpreting operational taxonomic units . . . 110
2.4 Compositionality . . . 110
3 Simulating 16S rRNA data . . . 112
3.1 Dirichlet-multinomial models of OTU abundance . . . 114

Modelling a variable number of reads per sample 114
3.2 Real datasets 115
4 Kernel two-sample testing for 16S rRNA data 115
4.1 Kernel mean embeddings 116
4.2 Maximum mean discrepancy 117
4.3 Kernels for 16S rRNA two-sample testing 118
5 Relevant work 119
6 Description of phylogenetic kernels 120
6.1 UniFrac kernel 121
6.2 String kernels 122
Relationship between positive-definite matrices and kernels . . 123
Spectrum Kernel 123
Mismatch Kernel 124
Gappy Pair Kernel 124
Computing String kernels 125
6.3 Implications of compositionality for kernel methods 126
7 Two-sample testing simulation study 127
7.1 Simulation aims 127
7.2 Controlling the phylogenetic differences between P and Q . . . 128
7.3 Phylogeny-aware clustering of OTUs 128
7.4 Simulation Setup 130
7.5 Simulation results I: Type I error and power 132
When $\varepsilon = 0$, a test with any phylogenetic kernel has well-calibrated Type I error 132
The Spectrum ($k = 20$) kernel and two UniFrac kernels have high power when $\varepsilon \geq 10^{-2}$ 132
The power of the Spectrum ($k = 20$) kernel depends on the choice of transformation when $\varepsilon = 10^{-3}$ 132
Tests using non-phylogenetic kernels have high power for all $\varepsilon > 0$, but are not sensitive to ε 132
For phylogenetic kernels differences in MMD are driven by phylogeny 135
For non-phylogenetic kernels differences in MMD are driven by the size of the permutation space $\pi_\varepsilon(\cdot)$ 135
Larger k-mer lengths increase power for String kernels 136
The behaviour of the phylogenetic kernels is stable between the two datasets 139
8 Host trait prediction using Gaussian process regression 139
8.1 Simulation aims 139
8.2 Simulation setup 139
OTU effect sizes 142
8.3 Gaussian process regression model 142
8.4 Full simulation procedure 143
8.5 Results 144
Log-marginal likelihoods 144
Log-predictive densities 144
8.6 Effect of string kernel hyperparameters 146
9 Discussion 148

6 Discussion and Conclusions **151**

Bibliography **153**

List of Figures

1.1 The central dogma of molecular biology describes how proteins are synthesised from sections of DNA (genes) via RNA. The study of sequence data (omics) is therefore a critical tool for the study of many biological systems. Figure created using BioRender.com. 2
1.2 A simplified workflow of the experimental (plot A) and computational (plot B) steps required to collect a 16S rRNA dataset. Figure created using BioRender.com. 5

2.1 Samples from Gaussian process distributions. The shaded area is the 95% credible interval. The sampled functions are plotted as lines for aesthetic reasons but are finite vectors of length 100. 14
2.2 The XOR problem is not linearly separable in its two-dimensional original space (left plot, colours indicate class membership). The feature map $\phi(x^{(1)}, x^{(2)}) = (x^{(1)^2}, x^{(2)^2}, x^{(1)}x^{(2)})$ projects into a three-dimensional space where the data are linearly separable (right plot), demonstrating the potential of kernels to capture complex dependencies in data. . 16
2.3 Given toy regression data of 20 data points (plot A), the full GP posterior (plot B) can be approximated using a sparse GP. The quality of the approximation improves when the number of inducing points increases from 3 (plot C) to 6 (plot D). 19
2.4 An example of a last layer Bayesian network. The first layer weights/biases and the final layer bias θ are point estimates, while the final layer weights w are assumed to be distributed under the prior $\pi(\cdot)$. The input variables are fed through the hidden layer to compute the hidden layer activations $(h_1, h_2, h_3)^T$. Samples of the predictions f are obtained from a linear combination of these activations with samples from the posterior of $w = (w_1, w_2, w_3)$, which is $q_\phi(w)$. This figure does not include the bias terms. 25
2.5 An example of a decision tree binary classifier constructed on a training set of 9 examples. Red dots and blue crosses denote the two classes. 27
2.6 Regression coefficients β are a projection of $y = f + \varepsilon$ onto $C(X)$, the column space of X (a plane in this 2-variable illustration). The component of the data explained by the model, $f = X\beta$, lies in this plane while the residuals ε (in red) are perpendicular to it. 37

3.1 Whittaker/rank-abundance plots (A) and the abundance (relative to the total dataset reads) and prevalence (B) for the agglomerated taxa in each kingdom of the FAME dataset. These types of plots are commonly used in ecology to visualise the rarity of organisms. The inclusion thresholds for the random forest modelling are denoted by dashed lines in plot B. 44

3.2 Mean held-out AUCs for random forest models. Two-sided 95% CIs calculated using the method of E. LeDell, Petersen, and Laan (2015) with a Bonferroni correction. Red dashed line is a mean held-out AUC of 0.5. BG: bacterial genus, FS: fungal species. 48
3.3 Empirical coverage of LeDell's confidence intervals on cross-validated AUC estimates under the null hypothesis (500 replicates). The confidence intervals are too narrow as the coverage is lower than the theoretical coverage (denoted by the red dotted line). 49
3.4 Rejection rate of DeLong's test comparing the AUCs two random forest models trained on permuted labels. The solid red line denotes the nominal significance level α and the dashed lines show its 95% binomial proportion confidence interval. 52
3.5 Spearman correlation between the variable methods using each data transformation. 56
3.6 The top four ranked variables for random forest models using different importance measures and transformations. 58
3.7 False discovery rate-adjusted p-values from Altmann's method using 1,000 permutations. Red dotted lines denote $p = 0.10$ and $p = 0.05$. . . 60
3.8 Stability of each variable importance method under dataset perturbations (removal of 10% of samples). Model: predicting Disease group from bacterial genus. 61
3.9 Stability of each variable importance method under dataset perturbations (removal of 10% of samples). Model: Predicting Disease group from fungal species. 62
3.10 Stability of each variable importance method under dataset perturbations (removal of 10% of samples). Model: Predicting Fungal disease group from fungal species. 63

4.1 A visualisation of partitioning the ESA posterior parameters $p(\tilde{\beta} \mid X, y)$ for a three-variable example. The precision matrix $\Lambda = \Omega^{-1}$ is partitioned in the same manner as Ω. Note that only $\mu_j, \omega_j, \omega_{-j}$ and Λ_{-j} (not shown) are used to calculate RATE scores for variable j. . . . 70
4.2 Visualisation of the different terms in the RATE calculation (plot B) under three ESA covariance structures (plot A). 72
4.3 An sample of the covariance structure and variable groupings used in the simulations in Section 5. The matrices are partitioned based on the group structure. 79
4.4 Mean squared error of the Bayesian neural network models across 100 replicates. A baseline mean model has an expected mean squared error of 1 (red dashed line). 84
4.5 The KL-divergence values for a group are positively correlated with the group size (plot A). This can be mitigated by dividing each KLD by the corresponding group size when calculating GroupRATE scores (plot B). 85
4.6 Variable prioritisation AUCs from 50 replicates. The red horizontal line indicates an AUC of 0.5 (the expected performance of random importance scores). 87
4.7 Group prioritisation ROC curves from 50 replicates. The black line denotes the median true positive rate across replicates and the shaded areas are the 10^{th} and 90^{th} percentiles on the empirical true positive rate. 88

4.8 Mean empirical computation times for the different methods across 100 replicates. The GroupRATE plots (A-C) have a different vertical scale to the non-GroupRATE plot (D). Computations are run in parallel using 32 threads. . . . 90
4.9 The size of each of the 1,255 Chromosome 1 genes included in the genotype simulations. . . . 91
4.10 Number of SNPs, p, for different numbers of genes G in the genotype simulations. . . . 93
4.11 Predictive mean squared error (MSE) of the four models in the genotype simulations. Red line indicates baseline performance of a model predicting mean of training labels. The Bayesian neural network posterior mean is computed using 100 Monte Carlo samples. . . . 94
4.12 Group prioritisation AUCs for the nine methods in the genotype simulations. The red dashed line denotes the expected performance of random group importances. SGP: sparse GP, BNN: Bayesian neural network, GRF: random forest with group importance scores, GLasso: GroupLasso. . . . 96
4.13 Global variable importance methods require images to be aligned for their results to be meaningful as they assign each pixel a global score. If images are aligned then a pixel has a fixed interpretation across the images (a region of the chest in plot A). If images are not aligned then the interpretation of a pixel varies from image to image (the position of cat ears is not fixed between images in plot B). . . . 98
4.14 Pixel importances for a convolutional neural network classifying odd and even digits in the MNIST dataset. The heat maps on the diagonal show the importance of each pixel (normalised to aid comparison between methods), with darker red indicating higher importance. Lower diagonal scatter plots allow pairwise comparisons of the scores of two methods, with the corresponding Spearman correlations in the upper diagonal. . . . 100
4.15 An ablation plot for a convolutional neural network classifying odd and even digits in the MNIST dataset. Pixels are shuffled in order of their importance, meaning that an accurate pixel ranking gives a steeper decrease in test accuracy. . . . 100
4.16 ROC curves for the convolutional neural network distinguishing patients with an without pneumonia from chest x-rays. Shaded areas denote the 95% confidence interval on the curves while the AUC values include 95% confidence intervals calculated using DeLong's method (DeLong et al., 1988). . . . 101
4.17 Training examples from the pneumonia (A) and control (B) groups. For GroupRATE pixels are clustered into 744 groups based on their Pearson correlation coefficient (C). The minimum correlation within a cluster is 0.7. . . . 103
4.18 GroupRATE values calculated on the training (A) and test (B) sets prioritise different groups. The top-ranked groups differ greatly between the two sets of GroupRATE values (C). . . . 104

5.1 Phylogenetic trees for the Busselton and FAME OTUs. Branches are coloured by genus, with any genus containing fewer than 50 OTUs marked as *Other*. Trees are inferred using FastTree2 (M. N. Price et al., 2010). . . . 109

5.2 The ground truth contains three clusters that are separable using (unobserved) absolute abundance (plot A). However, Clusters 2 and 3 have the same proportions of the two components and so are not separable using the measured abundance (plot B). This is a feature of compositional data and cannot be overcome using statistical methods. 113
5.3 16S rRNA commonly exhibit variable numbers of read per sample (plot A). This is emulated in the simulated datasets by modelling the number of reads per sample, N, as being drawn from a negative binomial $\mathrm{NB}(10^5, b)$ with different values of the dispersion parameter b (plot B). 115
5.4 Visualisation of the kernel mean embeddings of two distributions P and Q, where each contain the marginal densities of two indistinguishable OTUs. A characteristic kernel leads to a large MMD (plot A) but if the kernel models phylogenetic relationships it correctly finds that the distance between P and Q is small (plot B). 118
5.5 Underlying marginal distributions for 4 OTUs in a toy example with two populations P (panel A and (5.18)) and Q (panel B and (5.19)). Distributions with the same colour are identical. In a scenario where OTUs 1 and 2 and OTUs 3 and 4 are biologically equivalent a two-sample test should not reject the null hypothesis. 120
5.6 Calculating the UniFrac distance between pairs of samples. Both the weighted and unweighted UniFrac distance between Sample 1 (A) and Sample 2 (B) is 1 as the two samples do not share any branches (C). The weighted and unweighted UniFrac distances between Sample 3 (D) and Sample 4 (E) are both less than 1, as they share some branches (F), but they will not be equal to one another as the samples have different abundances. 122
5.7 Spectrum kernels for 100 most abundant OTUs in the Busselton dataset with k-mer length $k \in \{4, 12, 20\}$. Coloured bars indicate the Family of the OTU - these show that the blocks of highly similar OTUs correspond to taxonomic classifications. The value of k can be tuned to correspond to different taxonomic classifications. 124
5.8 An example of a trie for the 5-mers ATCTA, ATCTG, GTCTA, ATCGG, TTCGA and TGCGA. Any of the 5-mers can be represented by a depth-first traversal of the trie. Tries enable efficient computation of string kernels. 125
5.9 Empirical computation times for the string similarity matrix S for different hyperparameter values. Calculations were run on 8 cores using the Kebabs package for R (Palme et al., 2015). 126
5.10 A-C: Clusters of OTUs for $\varepsilon \in \{0.03, 0.01, 0.003\}$ for a subset of the FAME phylogenetic tree. Red boxes indicate clusters of OTUs and singleton clusters are not marked. D: the region of the tree in panels A-C in the context of the entire tree. 129
5.11 Distribution of OTU cluster sizes for the two datasets at different values of the phylogenetic distance threshold ε (non-singleton clusters only). 130

5.12 The difference between the two populations in the two-sample test simulation study is a permutation that restricts swaps to those within a set of clusters $\mathcal{C}_\varepsilon = \{c_1, \ldots, c_{|\mathcal{C}_\varepsilon|}\}$. Here $\alpha_i^{(c_k)}$ is the DMN concentration of the i^{th} OTU in cluster c_k. In this example the clusters c_1, c_2 and $c_{|\mathcal{C}_\varepsilon|}$ have sizes 3, 1 and 2 respectively. 130
5.13 Rate of null hypothesis rejections at a significance level of 0.1 for MMD two-sample tests with the highest-power phylogenetic kernels. The solid red line denotes the nominal significance level (0.1) and the dashed lines show its 95% binomial proportion confidence interval. Data were simulated using the phylogenetic trees and DMN concentrations of the Busselton (A) and FAME (B) datasets. Generated from 1,000 replicates of Algorithm 7.1. 133
5.14 Rate of null hypothesis rejections at a significance level of 0.1 for MMD two-sample tests with non-phylogenetic kernels. The solid red line denotes the nominal significance level (0.1) and the dashed lines show its 95% binomial proportion confidence interval. Data were simulated using the phylogenetic trees and DMN concentrations of the Busselton (A) and FAME (B) datasets. Generated from 1,000 replicates of Algorithm 7.1. 134
5.15 The ratio of the empirical MMD when $\varepsilon = 0.1$ to when $\varepsilon = 1$ across 1,000 replicates. The red line indicates equality between the MMD in the two scenarios. The top row contains kernels that exhibit desirable behaviour (phylogenetic kernels with well-selected hyperparameters) while the bottom row contains kernels which do not exhibit this behaviour (non-phylogenetic kernels). 137
5.16 Distributions of MMD values from 1,000 replicates of Algorithm 7.1. The shaded area denotes the 2.5%, 50.0%, 97.5% percentiles across the replicates. These results are for $b = 10$ but are representative of the other values tested. The CLR transformed is used for all non-UniFrac kernels. UniFrac kernels use the $\log(x + 1)$ transform. 138
5.17 Type I error rate of string kernels with different hyperparameters at a nominal significance level of 0.1. The solid red line denotes the nominal significance level (0.1) and the dashed lines show its 95% binomial proportion confidence interval. These results are for a group size of 200 using the CLR transform and $b = 3$ but are representative of all simulation scenarios tested. Generated from 1,000 replicates of Algorithm 7.1. 140
5.18 Power of string kernels with different hyperparameters at a nominal significance level of 0.1. These results are for a group size of 200 using the CLR transform and $b = 3$ but are representative of all simulation scenarios tested. Generated from 1,000 replicates of Algorithm 7.1. 141
5.19 Generating OTU effect sizes that are related to phylogeny (plot A) or are unrelated to phylogeny (plot B). Unmarked leaves denote OTUs with zero effect size in the phenotype model. 143
5.20 $\text{LML}_{\text{string}} - \text{LML}_{\text{other}}$, where LML_k is the log-marginal likelihood of a GP regression with kernel k. The red line indicates where both kernels have the same log-marginal likelihood. 145

5.21 Comparing the LML (log-marginal likelihood) of GP models trained with a String or Linear kernel differentiates between the two hypotheses. The dashed line indicates when the two kernels result in the same LML. The difference in LMLs between the two models is a Bayes factor. 146
5.22 $\text{LPD}_{\text{string}} - \text{LPD}_{\text{other}}$, where LPD_k is the test log-predictive density of a GP regression with kernel k. The red line indicates where both kernels have the same LPD on the test set. 147
5.23 Number of times different String kernel hyperparameters are selected in 1,000 replicates of the GP regression experiments. String kernel hyperparameters are selected using the log-marginal likelihoods of the resulting GP model. These plots are for $b = 10$ and $\sigma^2 = 0.3$ but are representative of the results with other values. 148

List of Tables

1.1 The main taxonomic ranks in the International Code of Zoological Nomenclature. Examples are taken from the animal kingdom in order to be more familiar to the reader. 4
1.2 The real datasets used in this thesis, their number of samples n and number of variables p. The number of variables refers to the number included in the analysis after any pre-precessing. 8

3.1 Covariate sets in the FAME dataset are the agglomerated OTUs of each kingdom. Only taxa accounting for more than 0.01% of reads or are present in at least 20% of samples are included in the random forest modelling. RF: random forest. 44
3.2 The different count transformation used in this chapter. $\tilde{x}_i^{(j)}$ are transformed counts from raw counts $x_i^{(j)}$ using $\tilde{x}_i^{(j)} = g\left(x_i^{(j)}\right)$ 46
3.3 The binary classification tasks for the random forest. 46
3.4 P-values from a permutation test on the mean validation AUC being greater than 0.5. P-values within each column are adjusted for multiple comparisons using the false discovery rate. *: P<0.1, **: P<0.05, ***: P<0.01. 50
3.5 P-values from one-sided DeLong's test comparing the discriminative power of the fungal and bacterial communities using their respective AUCs (AUC_{FS} and AUC_{BG}). CFPE is excluded here as neither the fungal nor bacterial communities are predictive of CFPE in this dataset. P-values are corrected for multiple comparisons using false discovery rate. $\Delta\text{AUC} = \text{AUC}_{\text{FS}} - \text{AUC}_{\text{BG}}$. *: P<0.1, **: P<0.05, ***: P<0.01. FS: fungal species, BG: bacterial genus 51
3.6 P-values from one-sided permutation test (500 permutations) comparing the discriminative power after adding the other kingdom as covariates to a random forest model. CFPE is excluded here as neither the fungal nor bacterial communities were predictive of CFPE. P-values are corrected for multiple comparisons using false discovery rate. ΔAUC is the change in mean held-out AUC when the second kingdom abundances are added. FS: fungal species, BG: bacterial genus 53

4.1 Comparison between quantities in the RATE and GroupRATE calculation for p variables. 75

5.1 Real lung microbiome datasets used in this chapter to simulate OTU abundances. The phylogenetic tree is also used in order to have realistic phylogenetic relationships between the OTUs. The original study groups are not used in the simulations but are included here for completeness. 115
5.2 The size of the permutation set $\pi_\varepsilon(\cdot)$ for different ε. 136

List of Abbreviations

ARD	Automatic relevance determination
AUC	Area under receiver operating characteristic curve
ASV	Amplicon sequence variant
BNN	Bayesian neural network
BX	Non-cystic fibrosis bronchiectasis
CF	Cystic fibrosis
CI	Confidence interval
CART	Classification and regression tree
CFPE	Cystic fibrosis pulmonary exacerbation
CLR	Centred log-ratio
CSLD	Chronic suppurative lung diseases
DMN	Dirichlet-multinomial
DNN	Deep neural network
DNA	Deoxyribonucleic acid
ELBO	Evidence lower bound
FDR	False discovery rate
FB	Fungal bronchitis
GLM	Generalised linear model
GP	Gaussian process
GWAS	Genome-wide association study
ITS2	Internal transcribed spacer 2
MAP	*Maximum a posteriori*
MCMC	Markov chain Monte Carlo
MMD	Maximum mean discrepancy
MDA	Mean squared error
MDG	Mean decrease Gini
MSE	Mean decrease accuracy
NB	Negative binomial
NAFD	No active fungal disease
PSD	Positive semi-definite
RBF	Radial basis function
ReLU	Rectified linear unit
RF	Random forest
RKHS	Reproducing kernel Hilbert space
rRNA	Ribosomal ribonucleic acid
ROC	Receiver operating characteristic
SNP	Single-nucleotide polymorphism
SVGP	Sparse variational Gaussian process
SGPR	Sparse Gaussian process regression
SVM	Support vector machine
OLS	Ordinary least squares
OTU	Operational taxonomic unit
WGS	Whole genome sequencing

Chapter 1

Introduction

1 The importance of sequence data in biological research

Sequences are ubiquitous in biological research. The most notable example is deoxyribonucleic acid (DNA), which contains the hereditary genetic material of almost all life on Earth. A second nucleic acid is ribonucleic acid (RNA), which is synthesised by cells according to the DNA sequence in a process called transcription. RNA carries out a number of roles, the most important of which is translation - the transfer of genetic information during protein synthesis. Proteins, which are themselves formed of amino acid sequences, perform most of the tasks essential for life. This transfer of genetic information to functional products is the central dogma of molecular biology (see Figure 1.1).

The analysis of sequence data is therefore a rich source of information with which to study biological systems via the so-called omics disciplines. Some common omics disciplines include genomics (the study of DNA), transcriptomics (RNA) and proteomics (proteins), epigenomics (chemical modifications to DNA) and microbiomics (microorganisms living inside our bodies). Sequence data provide measurements at a previously impossible resolution that have led to many breakthroughs in our understanding of human health. In genomic studies sequence data have revealed the genetic factors driving complex diseases such as Type 2 diabetes and schizophrenia, which in turn has led to the development of novel treatments (Visscher et al., 2017). Transcriptomics has revolutionised the understanding of cancer (Supplitt et al., 2021), while the study of the human microbiome has revealed its role in an ever-growing list of human diseases (Young, 2017).

Biology is currently experiencing a massive increase in the volume of collected sequencing data driven by the decreasing cost of nucleotide sequencing, which since 2007 has been falling with a halving time of less than two years. This has been achieved by utilising massively parallel processing (so-called next generation sequencing, Muir et al., 2016) and has enabled the collection of "big" datasets that have been at the heart of the vast majority of the biological breakthroughs in this period. This increase in the volume and complexity of data and accompanying statistical developments have together enabled data-driven analyses to complement more traditional, hypothesis-driven approaches (Ratti, 2015; Leonelli, 2016). One area in which such data-driven analyses are increasingly employed is in the study of the human microbiome (Moreno-Indias et al., 2021).

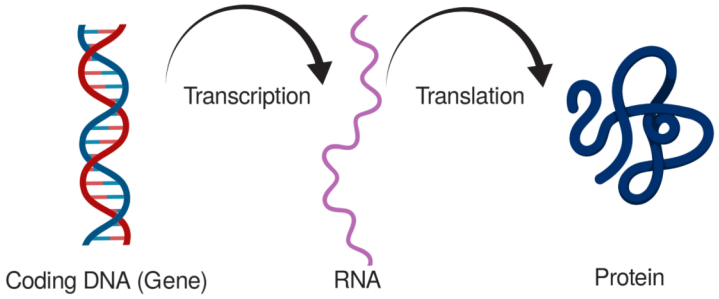


FIGURE 1.1: The central dogma of molecular biology describes how proteins are synthesised from sections of DNA (genes) via RNA. The study of sequence data (omics) is therefore a critical tool for the study of many biological systems. Figure created using BioRender.com.

1.1 The human microbiome

A large part of the work in this thesis concerns the microbiome - the microorganisms (including bacteria, fungi and viruses), their genetic material and their interactions that live in or on a host organism. The human body is itself a vast and diverse microbial ecosystem, with estimates placing the number of microbial genes at up to ten times larger than the number of human genes (Turnbaugh, Ley, et al., 2007). In humans the primary microbial habitats are the mouth, skin, nostrils, gut and urogenital tract (NIH Human Microbiome Portfolio Analysis Team, 2019). It is now accepted scientific fact that the 10-100 trillion microbial cells living in each individual have a critical effect on health.

The study of the microbiome is a relatively new field, with sequencing of microbial samples beginning in the 2000s and the subsequent launch of the Human Microbiome Project in 2008 (NIH Human Microbiome Portfolio Analysis Team, 2019). Since then, the risk and severity of diseases such as obesity (Turnbaugh, Hamady, et al., 2009), diabetes (Larsen et al., 2010) and autism (Parracho et al., 2005) have been related to characteristics of the host microbial communities. Its importance has been further illustrated by observations of distinctive gut microbial communities in disorders of the central nervous system, which had been previously assumed to be unconnected to the gut (Y. Wang and Kasper, 2014). This discovery of the "gut-brain" axis (Cryan et al., 2019) has been followed by the "gut-lung" (Budden et al., 2017) and "gut-bladder" (Worby et al., 2022) axes as the understanding of the critical and interconnected role microbial communities play in human health increases. Despite rapid progress the vast majority of questions remain unanswered, such as whether causal links exist between the characteristic microbial communities observed in disease groups or how these communities interact with host genetic and environmental factors (Giliberti et al., 2022).

1.2 The lung microbiome

A large part of the work presented in this thesis has been performed while working in the Asmarley Centre for Genomic Medicine, which is part of the National Heart and Lung Institute at Imperial College London. This group's primary interest is the genomic and microbial drivers of lung diseases and so a large part of this thesis concerns lung microbial communities.

The prevailing dogma at the launch of the Human Microbiome Project was that the lungs and airways were a sterile environment. They were therefore excluded from the list of the 18 sites sampled. This has since been shown to be incorrect and that this erroneous belief was due to difficulties in obtaining microbial samples from the lung. Hilty et al. (2010) took the first culture-independent samples from the airways of healthy patients and demonstrated that the airways contain a rich community of microbes.

Since then, a growing number of studies have investigated the complex relationships between respiratory infection, host immune response, environmental factors and the lung microbiome (O'Dwyer et al., 2016). It has now been shown that both healthy and diseased lungs host large and diverse microbial communities, making the lung microbiome a promising area of research into respiratory diseases and their treatments (Dickson et al., 2016). Several studies have reported systematic differences (including a loss of diversity) between sufferers of respiratory disease and healthy controls, with the nature of the difference being specific to the disease (Faner et al., 2017; Ding et al., 2021). There are also reported links between the lung microbial community and immune response (Paudel et al., 2020; Clark, 2020). However, it is currently unknown whether the observed dysbiosis (microbial imbalance) is a cause or consequence of disease.

1.3 Collection of microbial datasets

Taxonomic ranks

In order to quantify the composition of a microbial community it is necessary to identify the microorganisms that comprise it. This identification is performed by placing them in the taxonomic hierarchy, which describes the evolutionary relationships between all the known organisms on Earth. At the highest level (not part of the taxonomic hierarchy), all living cells are either Eukaryotes and Prokaryotes, where Eukaryotes have a membrane-bound nucleus in their cell while Prokaryotes do not. The highest level of the taxonomic hierarchy (Domain) then divides life into Archaea and Bacteria (both Prokaryotes) and Eukaryotes, which includes all animals, plants and fungi.

Table 1.1 lists the taxonomic ranks as described by the International Code of Zoological Nomenclature. The most relevant ranks for this work are genus and species. The term taxon (plural taxa) is used to refer to a taxonomic group of any rank.

Microbial identification and quantification via 16S rRNA and ITS2 sequencing

While the cost of sequencing has decreased in the last decade, whole genome sequencing of the thousands of microbes present at a given site is still extremely challenging and in many cases impossible. Next-generation sequencing relies on processing a large number of sequence fragments in parallel, which requires a high level of sequencing depth in order to re-assemble these fragments after sequencing. Re-assembly of multiple bacterial genomes therefore requires sampling sufficient biomass of each organism, which is often impractical or even impossible. This means that, despite the many advantages of whole genome sequencing (primarily better taxa identification, Ranjan et al., 2016), it is standard practice to target a small region of the microbial genome for identification and quantification. The resulting

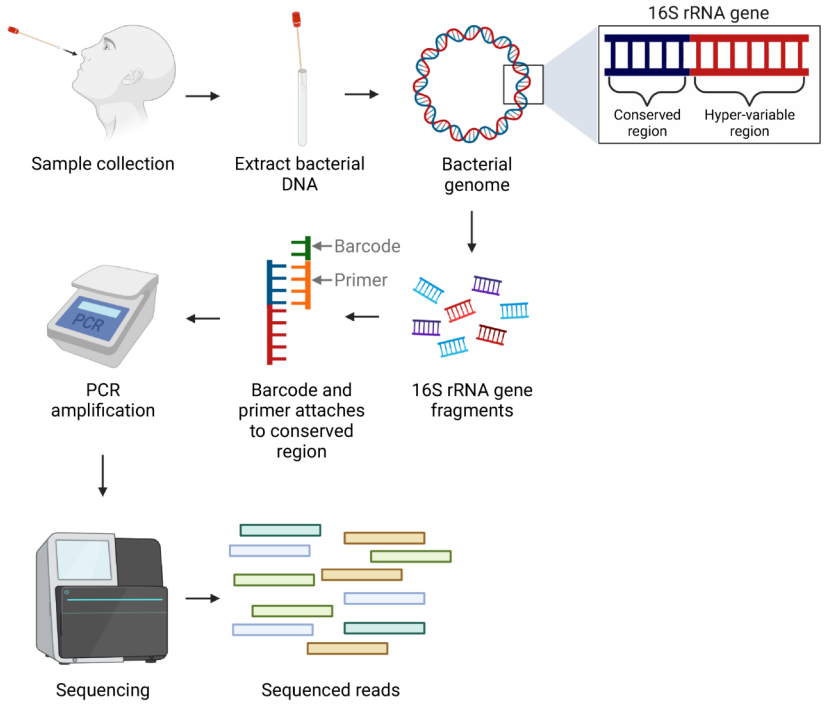
TABLE 1.1: The main taxonomic ranks in the International Code of Zoological Nomenclature. Examples are taken from the animal kingdom in order to be more familiar to the reader.

Rank	Common examples
Domain	Archaea, Bacteria, and Eukaryotes
Kingdom	Animals, Plants, Fungi, Monera (prokaryotes), ...
Phylum	Nematoda (roundworms), Arthropods, ...
Class	Mammalia, Aves (birds), Reptiles ...
Order	Primates, Rodentia, ...
Family	*Hominidae* (great apes), *Felidae* (cats), *Canidae* (dogs), ...
Genus	*Homo*, *Canis* (includes wolves and dogs), ...
Species	*Homo sapiens*, *Canis familiaris*, ...

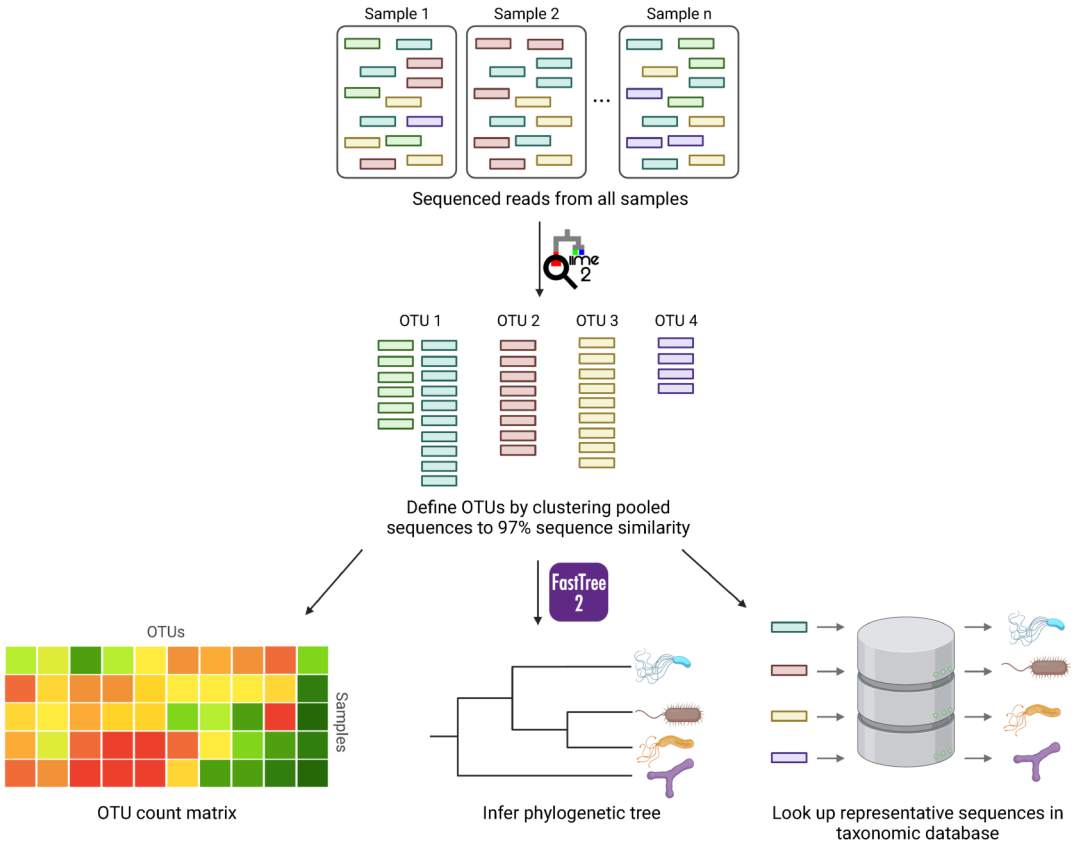
modalities are named after the targeted genomic region: 16S rRNA (ribosomal ribonucleic acid) gene sequencing for bacteria and ITS2 (internal transcriber spacer 2) sequencing for fungi. In both modalities fewer than 1,000 base pairs (genomic positions) are sequenced from every organism in a sample, meaning that the vast majority of the microbial genome is not recorded.

What follows is a simplified description of how microbial datasets are collected. It describes the 16S rRNA gene sequencing modality but the steps are largely the same as those used in ITS2 sequencing. The first steps of this process are performed in the laboratory, starting with a patient sample (Figure 1.2(A)). The first laboratory step extracts any bacterial DNA from a patient sample. From each bacterial genome the 16S rRNA gene is isolated and amplified via polymerase chain reaction (PCR). The 16S rRNA gene is used as a target because it consists of conserved and hyper-variable regions (Clarridge III, 2004). This enables the development of targeted primers that attach to the conserved region during PCR amplification. The hyper-variable regions are sufficiently specific that they can be used for taxa identification. The primers also contain a barcode sequence that records which sample a sequence came from. Following PCR amplification the 16S rRNA gene fragments are sequenced, which produces a set of sequenced reads representing the hyper-variable regions, tagged with sample information (the barcode).

A series of computational steps then produces the final dataset by identifying the microorganisms represented by each sequenced read (Figure 1.2(B)). The sequenced reads from all samples are pooled and clustered to 97% sequence similarity using the open-source bioinformatics software QIIME (Bolyen et al., 2019). A cluster of sequences defines an operational taxonomic unit (OTU), which is assigned its most central member as its representative sequence. OTUs are the variables in a 16S rRNA or ITS2 dataset. The representative sequences are then used to (i) assign a taxonomic identification to the OTU using a reference database and (ii) infer a phylogenetic tree describing the evolutionary relationships between the OTUs (M. N. Price et al., 2010). An OTU table is also computed by counting the observed abundance of each OTU in each sample. The OTU table, phylogenetic tree and the OTU identities form the final dataset used for subsequent statistical analysis. In addition, host metadata (such as clinical measurements and demographic data) are also typically included.



(A) Laboratory steps



(B) Computational steps

FIGURE 1.2: A simplified workflow of the experimental (plot A) and computational (plot B) steps required to collect a 16S rRNA dataset. Figure created using BioRender.com.

2 Non-parametric predictive modelling

In the last decade powerful predictive models have become ubiquitous in data-rich fields such as digital advertising, fraud detection and engineering. Increasingly powerful computation and statistical advances have enabled the deployment of these models to solve previously impossible predictive tasks. These successes have led to similar approaches being applied in biological research, although the nature of biological problems and data collection mean that a naive out-of-the-box application of these models is rarely appropriate (Lopatkin and Collins, 2020).

In many cases these predictive models are non-parametric in nature, meaning that they contain fewer assumptions than their parametric alternatives and are able to increase their complexity as the volume of data increase. As parametric modelling typically requires specifying *a priori* the types of dependencies that exist within the data it can be difficult to apply them when there is a lack of prior knowledge (or any prior knowledge is difficult to express in a mathematical form). This scenario is becoming increasingly prevalent as biological datasets increase in complexity, in which case non-parametric modelling can be an attractive and flexible alternative.

A good example of the potential utility of non-parametric methods is regression. While parametric linear regression enforces a linear relationship between the parameters and response, a non-parametric regression model (such as kernel regression or regression trees) is able to capture more complex dependencies within the data without explicitly specifying them in advance. This makes them well-suited to the types of data-driven analysis that are becoming feasible with the increasing size of biological datasets (for example, public health databases such as UK Biobank (Sudlow et al., 2015)) and increased computational capabilities. The superior predictive performance of non-parametric predictive models is often cited as proof that their additional capacity is a closer representation of the underlying biological process, which is often achieved by leveraging interactions between variables. However, it should be noted that choosing a non-parametric model over a parametric one does not necessarily lead to superior predictive performance and is much more likely to result in severe over-fitting due to their additional capacity. Furthermore, the lack of interpretability of many non-parametric models means it is usually more difficult to use them for inference than an equivalent parametric model.

2.1 Machine learning in biology

One area of statistics that has particularly benefited from the increase in computational power and volume of data is supervised machine learning, which contains a variety of non-parametric predictive models that have become ubiquitous in data-rich fields over the past decade. Two high-profile examples of such fields are computer vision and natural language processing, where the vast amounts of available data have aided the explosive growth of deep learning. Deep learning models that were originally developed for non-biomedical applications are now able to surpass human expert level performance in medical image analysis (X. Liu et al., 2021), as well as solving the previously intractable problem of predicting protein structure from sequence (Jumper et al., 2021). These achievements are possible due to the ability of deep learning algorithms to automatically identify complex predictive patterns in large datasets, as opposed to the hand-crafted feature engineering that was pervasive in the era before "big" data. Other popular supervised machine learning

models include decision tree ensembles, support vector machines and kernel methods, which lack the fame of deep learning but are often favoured in fields where the dataset sizes are too small to train a deep learning model (Greener et al., 2022).

The relative lack of human involvement in feature engineering presents challenges when attempting to use these models for biological research as the patterns they detect may not be available to the user (the model is a black box) or too difficult for a human to understand. This focus on prediction over inference is often used as the dividing line between classical statistics and machine learning (Bzdok et al., 2018). However, this boundary is blurred and current research activity is serving to blur it further. For example, the intense research focus on explainable/interpretable machine learning is a reflection of the need for further improvements to many machine learning models before they can fulfil their potential utility in biology (Murdoch et al., 2019; Roscher et al., 2020). This requirement for interpretability means that linear models are still the standard tool in most biological applications, such as univariate linear regression or univariate linear mixed models in genome-wide association studies (Purcell et al., 2007; Lippert et al., 2011) or generalised linear models in differential expression analysis (Love et al., 2014; Ritchie et al., 2015).

3 Data

The collection of biological datasets consists of many stages of patient recruitment, laboratory work and pre-processing. The computational pre-processing steps are often placed in the discipline of bioinformatics, which is usually considered to be separate from the subsequent statistical analysis. This thesis focuses on these subsequent analyses and so the extensive and specialised pre-processing is outside of its scope, as is the collection of the data themselves. The real datasets used in this thesis are listed in Table 1.2. The collection, quality control and pre-processing of the FAME and Busselton datasets was performed by a skilled set of collaborators and so are not part of the contributions of this thesis. The remaining three datasets are publicly available.

Three of the six datasets concern the lung microbial community and one is human genetic sequence data. These processed datasets take the form of an $n \times p$ matrix of counts representing the abundance of a p sequences in n samples. The sequence themselves are not modelled directly in the majority of the chapters. The exception is Chapter 5, which models the similarity between microbial taxa via the similarity in their underlying DNA sequence.

4 Thesis contributions

Non-parametric statistical methods for analysing biological data are particularly useful when there are complex, non-linear relationships between the variables, which is a common feature of biological datasets. This has driven the increasing popularity of non-parametric predictive modelling (including supervised machine learning) in biomedical studies. However, the additional complexity of these models compared to parametric alternatives means that a naive application of these approaches to biological data is rarely appropriate. The work in this thesis addresses a range of challenges that arise when applying non-parametric methods to biological sequence data.

TABLE 1.2: The real datasets used in this thesis, their number of samples n and number of variables p. The number of variables refers to the number included in the analysis after any pre-precessing.

Chapter	Dataset name	n	p	Modality	Citation
3	FAME (fungal)	107	2,770 OTUs	ITS2 sequencing	Cuthbertson, Felton, et al. (2021)
3, 5	FAME (bacterial)	107	1,189 OTUs	16S rRNA gene sequencing	Ish-Horowicz, Cuthbertson, et al. (2022)
4	MNIST	60,000	324 pixels	Grayscale imaging	LeCun (1998)
4	Chest X-Ray Images	5,863	40,000 pixels	X-ray imaging	Kermany et al. (2018)
4	WTCCC	10,000	7,405 SNPs (1,255 genes)	Human genome sequencing	WTCCC et al. (2007)
5	Busselton	578	1,689 OTUs	16S rRNA gene sequencing	McBrien (2020)

4.1 Chapter contributions

The two-sample test is one of the most common statistical tasks in biomedical studies as it can be used to establish whether two disease groups are distinct or whether a treatment has had a detectable effect. The two-sample test is especially important in studies of the human microbiome as researchers seek to establish whether different disease groups have characteristic microbial communities. However, microbial datasets present specific challenges for the two-sample test as a multivariate test is required to detect community-level differences. The complexity of microbial datasets has led to random forests emerging as a popular tool for classifier-based two-sample testing.

Chapter 3 presents an investigation of the bacterial and fungal communities of two lung diseases, cystic fibrosis and non-cystic fibrosis bronchiectasis. It is based on the pre-print by Ish-Horowicz, Cuthbertson, et al. (2022) and utilises random forest modelling for classifier based-two sample testing and differential abundance analysis. While the pre-print focuses on the biological results this chapter includes a novel study of the effect of modelling decisions (the choice of data transformation and variable importance measure) on the biological conclusions. It also examines the behaviour of popular hypothesis tests and confidence interval that are commonly applied with this type of data. Its statistical contributions are:

- a study on the impact of data transformations on the results of random forest analyses of a 16S rRNA and ITS2 dataset;

- an empirical study of the Type I error behaviour of random forest-based two-sample testing; and
- a study of the behaviour of the four most popular variable importance methods for random forest-based differential abundance.

One of the benefits of a random forest-based two-sample test is the ability to calculate variable importance scores, which can be used to identify specific taxa that drive any differences between groups. The ability to compute variable importance is a key element of many biomedical statistical analyses, which precludes the use of many of the most powerful predictive models in a number of applications as they are not interpretable. One such model is a Bayesian neural network, which is a Bayesian extension of the popular deep learning models that have transformed many non-biomedical fields.

Chapter 4 presents an extension of RATE (RelAtive cEntrality, Crawford et al., 2019), a variable importance method for Gaussian process regression to Bayesian neural networks. An additional extension considers importance scores for grouped variables, which are common in biological datasets in general and sequencing datasets in particular. This chapter is largely based on the pre-print by Ish-Horowicz, Udwin, et al. (2019) but includes additional simulation results and computer vision examples. Its contributions are

- extending the RATE methodology to a last layer only Bayesian neural network architecture;
- investigating the utility of two alternative projection operators for RATE;
- extending the original RATE criterion to grouped variables (GroupRATE);
- demonstrating the ability of GroupRATE to prioritise causal groups on two simulated sequencing datasets; and
- demonstrating how GroupRATE can be applied to a Bayesian neural network classifier trained on a medical imaging dataset.

Phylogeny is an important feature of microbial datasets but it is commonly ignored when performing the two-sample test with 16S rRNA data. This is a potentially severe limitation as there is typically a large degree of degeneracy in OTU definitions (many OTUs corresponding to microorganisms that may be functionally or biologically equivalent). A standard approach to the two-sample test may therefore reject the null hypothesis on the basis of differences with no biological relevance. Chapter 5 contains a simulation study on modelling phylogeny in bacterial microbial datasets in a kernel two-sample testing procedure, where phylogeny is encoded using string kernels. The performance of string kernels is also explored in the context of host-trait prediction using Gaussian process regression. Its contributions are:

- showing that kernel-based two-sample tests with popular kernels reject the null hypothesis in the case of biologically irrelevant differences between groups;
- developing phylogeny-aware kernels based on string sequence similarity measures that do not exhibit this behaviour; and

- showing via simulation studies that a Gaussian process regression model with a string kernel can identify how microbial effects are related to 16S rRNA gene sequence similarity.

Chapter 2 contains a detailed mathematical description of the various predictive models and associated statistical methods that are utilised in Chapters 3-5, while Chapter 6 summarises the overall findings and discusses avenues for future work.

4.2 Other work

I have also contributed to other research projects during my PhD that is not included in this thesis:

- I am part of the Imperial College Covid Response Team and have been involved in two papers on Bayesian modelling of the Covid-19 pandemic in Europe and the USA (Unwin et al., 2020; Monod et al., 2021). My contributions focused on implementing the model checking and evaluation elements of the analysis pipeline. I am also a contributor and paper author for the accompanying R package `epidemia` (Scott et al., 2021).
- I am a developer of `GpABC`, a Julia package for emulating differential equation models from systems biology using Gaussian process regression. `GpABC` also provides model selection and parameter estimation methods using approximate Bayesian computation (Ish-Horowicz, Tankhilevich, et al., 2020).
- I am involved in the first study to culture and sequence airway microbiota. My contributions have focused on correlation network analysis and developing a framework to assign functional annotations to clusters of OTUs (Cuthbertson, Forslund, et al., 2022, Manuscript in prepration).

Chapter 2

Mathematical Background

This chapter describes the predictive models that are used in the subsequent chapters, which are Gaussian process regression (and its sparse approximation), Bayesian neural networks and random forests. After describing each model and its training procedure the focus moves on to the calculation of variable importance scores for each model.

1 Supervised learning

Each of the studies in this thesis heavily involve predictive modelling, which puts them in the supervised learning framework. Given a dataset $\mathcal{D}$ of n input-output pairs,

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}\,, \tag{2.1}$$
$$x_i \in \mathcal{X}\,, y_i \in \mathcal{Y}\,, \quad i = 1, \ldots, n\,, \tag{2.2}$$

supervised learning seeks to find a function $f : \mathcal{X} \mapsto \mathcal{Y}$ for input and response domains $\mathcal{X}$, $\mathcal{Y}$. Note that this dataset can be equivalently expressed as the $n \times p$ design matrix X and n-dimensional vector y, where $x_i = (x_i^{(1)}, \ldots, x_i^{(p)})$ is the i^{th} row of X and y_i is the i^{th} element of y. Throughout this thesis subscripts index samples and superscripts index variables. For the problems considered here the variables that define $\mathcal{X}$ are p-dimensional counts ($\mathcal{X} = \mathbb{Z}_{\geq 0}^{p}$) or reals ($\mathcal{X} = \mathbb{R}^p$). The

The function f is selected from a pre-specified family of functions $\mathcal{F}$ by solving

$$f = \underset{f \in \mathcal{F}}{\arg\min} \quad \mathbb{E}_{x \sim \mathcal{X}, y \sim \mathcal{Y}}[\mathcal{L}(y, f(x))]\,, \tag{2.3}$$

where $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$ is a loss function such as the mean squared error or binary cross entropy. In practice (2.3) can only be solved for the observed values, meaning that training the model requires solving

$$f = \underset{f \in \mathcal{F}}{\arg\min} \quad \frac{1}{n} \sum_{i=1}^{n} [\mathcal{L}(y_i, f(x_i))]\,, \tag{2.4}$$

which is known as empirical risk minimisation (Vapnik, 1991). Before proceeding further with model training the space of candidate models $\mathcal{F}$ must be defined. The

following sections describe the different types of non-parametric models utilised in this thesis:

- Gaussian process regression and its sparse approximation;
- Bayesian neural networks; and
- Random forests.

1.1 Gaussian process regression

Gaussian process (GP) regression is a Bayesian, non-parametric regression model first proposed in the machine learning literature by Williams and Rasmussen (1995). Its flexibility has made it a popular choice where the response variable has a complex and hard to define dependencies on the covariates. Common non-biological applications of GP regression include predicting the outputs of computer simulation codes (such as climate forecasting models (Andrianakis and Challenor, 2012)) or predicting the generalisation error of machine learning models in automated hyperparameter searches (Snoek, Larochelle, et al., 2012). Analogous biological applications include using predicting the output of mechanistic systems biology models from parameter values (Ish-Horowicz, Tankhilevich, et al., 2020).

This remainder of this section outlines GP regression using the "function-space view," as described in Williams and Rasmussen (2006), in which a Gaussian process is defined as follows:

Definition 2.1. *A* Gaussian process *is a collection of random variables, any finite number of which have a joint Gaussian distribution.*

For this reason, GPs are often characterised as a *distribution over functions*.

The Mean Function and Kernel

A GP regression model for the dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ is

$$y = f + \varepsilon \tag{2.5}$$
$$f(x) \sim \mathcal{GP}\left(m(x), k(x, x')\right) \tag{2.6}$$
$$\varepsilon \sim \mathcal{N}(0, \tau^2)\,, \tag{2.7}$$

where $m(x)$ is the mean function, $k(x, x')$ is the symmetric, positive semi-definite kernel function and τ^2 is the noise variance. Together, the mean function and kernel fully determine a GP.

Expressing the dataset as $\mathcal{D} = (X, y)$, a zero-mean GP prior over $f = (f(x_1), \ldots, f(x_n))$ is given by

$$p(f|X) = \mathcal{N}(\mu, K_{XX})\,, \tag{2.8}$$

where K_{XX} is the positive semi-definite matrix with elements $(K_{XX})_{ij} = k(x_i, x_j), i, j = 1, \ldots, n$. The choice of kernel function controls the types of functions which can be

sampled from the GP and so its selection is the most important aspect of GP modelling. The most popular kernel is the squared-exponential/radial basis function (RBF) kernel,

$$k(x, x') = \sigma_f^2 \exp\left(\frac{-\|x - x'\|^2}{2l^2}\right), \tag{2.9}$$

where $\sigma_f^2, l > 0$ are the signal variance and lengthscale hyperparameters. The effect of l on the functions sampled from (2.8) is illustrated in Figure 2.1(A) - given a pair of inputs, shorter lengthscales reduce the covariance between the corresponding function values. This results in more "wiggly" functions being sampled from the GP. It is also possible to use one lengthscale per dimension, in which case $l \in \mathbb{R}^p_{>0}$. Such kernels are named automatic relevance determination (ARD) and the RBF version is given by

$$k(x, x') = \sigma_f^2 \exp\left(\sum_{j=1}^{p} \frac{-\left(x^{(j)} - x'^{(j)}\right)^2}{2l^{(j)^2}}\right), \tag{2.10}$$

where $l = (l^{(1)}, \ldots, l^{(p)})$ is the vector containing per-dimension lengthscales. ARD kernels have built-in variable importance as those with the shortest lengthscales are more strongly associated with the response (assuming all input variables are transformed to be on the same scale). However, in the $n \ll p$ regime the additional model capacity from using an ARD kernel is likely to lead to over-fitting. All the GP models in this thesis are used in the $n \ll p$ regime and so use non-ARD kernels.

Gaussian process posterior and predictive posterior

The posterior over f is given by Bayes rule,

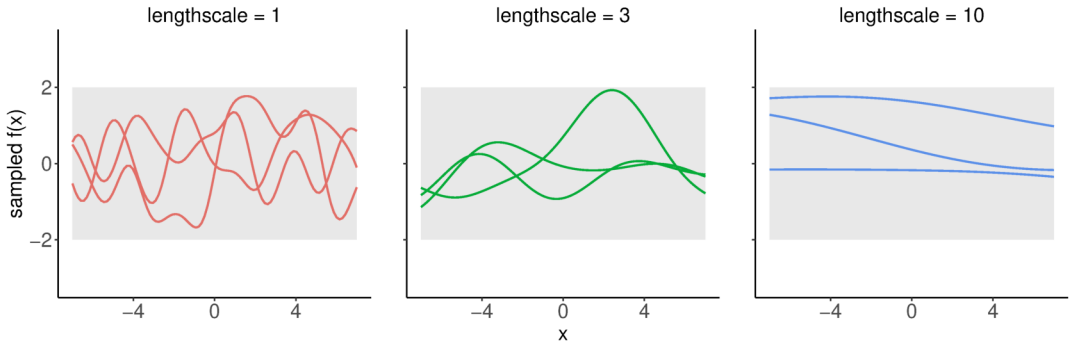
$$p(f \mid X, y) = \frac{p(y \mid f)\, p(f \mid X)}{\int p(y \mid f')\, p(f' \mid X)\, \mathrm{d}f'}, \tag{2.11}$$

with Gaussian likelihood $p(y \mid f) = \mathcal{N}(y \mid f, \tau^2 I)$ and GP prior (2.8), where I is the identity matrix. All the densities in (2.11) can be solved in closed-form using the Schur complement, which results in a Gaussian posterior with parameters

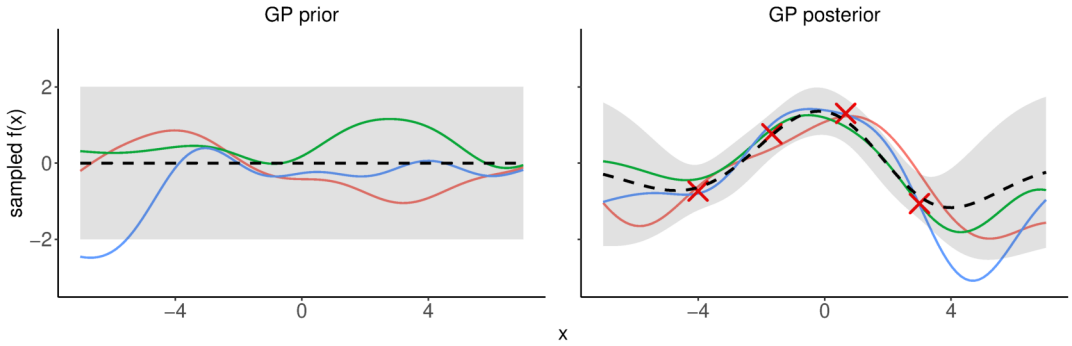
$$\mathbb{E}[f \mid X, y] = K_{XX}(K_{XX} + \tau^2 I)^{-1} y \tag{2.12}$$

$$\mathbb{V}[f \mid X, y] = K_{XX} - K_{XX}(K_{XX} + \tau^2 I)^{-1} K_{XX}. \tag{2.13}$$

The effect of observed data is therefore to reduce the variance relative to the prior, while the mean function approximately interpolates the data (see Figure 2.1(B-C)). Far from the observed data, both the mean and variance revert to the prior. For a set of unseen inputs X^*, the predictive posterior is also a Gaussian with parameters



(A) Samples from a zero-mean GP prior with RBF kernel of different lengthscales.



(B) Samples from a zero-mean GP prior with RBF kernel (left) and the corresponding posterior (right) after observing 4 data points (red crosses). The black dotted line is the mean.

FIGURE 2.1: Samples from Gaussian process distributions. The shaded area is the 95% credible interval. The sampled functions are plotted as lines for aesthetic reasons but are finite vectors of length 100.

$$\mathbb{E}[f \mid X, y] = K_{XX^*}(K_{XX} + \tau^2 I)^{-1} y \tag{2.14}$$
$$\mathbb{V}[f \mid X, y] = K_{X^*X^*} - K_{XX^*}(K_{XX} + \tau^2 I)^{-1} K_{XX^*}^T , \tag{2.15}$$

where K_{XX^*} and $K_{X^*X^*}$ are formed by pairwise evaluations of $k(\cdot, \cdot)$ between the samples in their respective subscripts.

Model selection in Gaussian process regression

In the context of GP regression, model selection consists of selecting a kernel and learning its hyperparameters. The most common approach is known as ML-II and involves optimising the log of the denominator of (2.11) (the log-marginal likelihood) using gradient-based methods. In the GP regression case the log-marginal likelihood is given by

$$\log p(y \mid X) = -\frac{1}{2} y^T (K_{XX} + \tau^2 I)^{-1} y - \frac{1}{2} \log |(K_{XX} + \tau^2 I)| - \frac{n}{2} \log 2\pi , \tag{2.16}$$

while its gradients with respect to the kernel hyperparameters are

$$\frac{\partial}{\partial \theta_j} \log p(y \mid X) = \frac{1}{2} y^T K_{XX}^{-1} \frac{\partial K_{XX}}{\partial \theta_j} K_{XX}^{-1} y - \frac{1}{2} \operatorname{trace}\left(K_{XX}^{-1} \frac{\partial K_{XX}}{\partial \theta_j} \right) , \tag{2.17}$$

where K_{XX} is a function of the hyperparameters θ (Williams and Rasmussen, 2006). The terms of (2.16) can be interpreted as a data fit term $y^T (K_{XX} + \tau^2 I)^{-1} y$ and a regularisation term $\log |(K_{XX} + \tau^2 I)|$ that penalises the complexity of the model. This makes it an appropriate optimisation objective for model selection in GP regression. Markov chain Monte Carlo (MCMC) methods can also be used to draw samples from the posterior of the kernel hyperparameters.

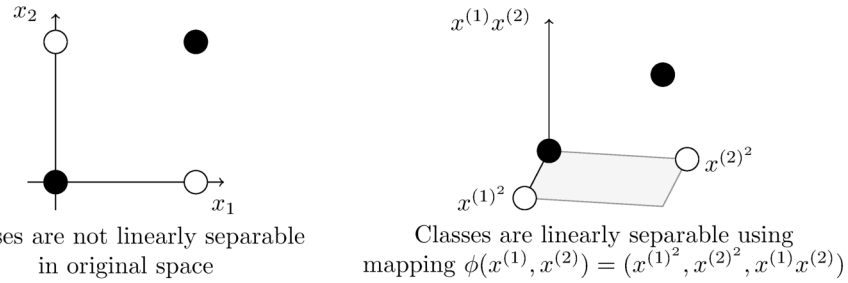
The Kernel Trick

A symmetric, positive semi-definite kernel function $k(\cdot, \cdot)$ satisfies

$$k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}} , \tag{2.18}$$

for feature map $\phi : \mathcal{X} \to \mathcal{H}$ which induces the reproducing kernel Hilbert space (RKHS) $\mathcal{H}$. Kernels therefore compute inner products in a feature space defined by $\phi(\cdot)$. The utility of such a mapping is often utilised in classification problems, a simple example of which (the XOR problem) is shown in Figure 2.2. In the XOR problem the two classes become linearly separable after applying the feature mapping $\phi(x^{(1)}, x^{(2)}) = (x^{(1)^2}, x^{(2)^2}, x^{(1)} x^{(2)})$, enabling the use of a linear model (usually a support vector machine, Ben-Hur et al., 2008).

The real power of kernels comes from the fact that (5.14) holds even when $\phi(\cdot)$ is unknown or cannot be expressed mathematically. This is known as the kernel trick, and enables the use of infinite-dimensional feature maps via the appropriate choice of kernel. Kernels are also powerful when used with structured data such as strings,



Classes are not linearly separable in original space

Classes are linearly separable using mapping $\phi(x^{(1)}, x^{(2)}) = (x^{(1)^2}, x^{(2)^2}, x^{(1)}x^{(2)})$

FIGURE 2.2: The XOR problem is not linearly separable in its two-dimensional original space (left plot, colours indicate class membership). The feature map $\phi(x^{(1)}, x^{(2)}) = (x^{(1)^2}, x^{(2)^2}, x^{(1)}x^{(2)})$ projects into a three-dimensional space where the data are linearly separable (right plot), demonstrating the potential of kernels to capture complex dependencies in data.

as they provide a simple procedure to encode complex domain knowledge via the kernel function (Schölkopf et al., 2004). This property of kernel methods is exploited in Chapter 5 to model the phylogenetic relationships between bacterial taxa.

1.2 Sparse Gaussian process regression

Background and motivation

Calculating the GP posterior parameters using (2.12)-(2.13) has $\mathcal{O}(n^3)$ running time complexity as they require the Cholesky factor of $K_{XX} + \tau^2 I$, which is an $n \times n$ matrix. Furthermore, the gradient-based optimisation of the marginal likelihood using (2.17) also requires this Cholesky factor to be recomputed at every step. This has limited the use of Gaussian process to relatively small ($n \lessapprox 10^3$) datasets.

This lack of scalability with n has motivated the development of so-called *sparse* Gaussian processes, which initially used the Nyström method to approximate K_{XX} with

$$K_{XX} \approx K_{XZ} K_{ZZ}^{-1} K_{ZX}\,, \tag{2.19}$$

where K_{ZZ} is a kernel matrix with elements $(K_{ZZ})_{ij} = k(z_i, z_j)$, $i, j = 1, \ldots, m$ and $Z = (z_1, \ldots, z_m)$ is an $m \times p$ matrix of $m < n$ rows sampled without replacement from X (Williams and Seeger, 2000). Inverting this approximation to K_{XX} only takes $\mathcal{O}(mn^2)$ time, which is a significant reduction if $m \ll n$. The intuition behind this approximation is that if there is a large amount of redundant information in the rows of X then it should be possible to construct a reasonable approximation of the full GP using only a subset of those rows.

A seminal work by Snelson and Ghahramani (2005) showed that Z need not be formed by subsampling rows from X and that it was possible to use a Z made up of pseudo-inputs (called inducing points) that were not present in the dataset but that could be selected by gradient-based optimisation of the marginal likelihood. This has since been surpassed by the variational Bayes approach of Titsias, in which the exact GP posterior is approximated by a variational distribution whose parameters include the inducing points (Titsias, 2009). Hensman et al. (2015) further extended

the variational GP model to allow stochastic optimisation, largely motivated by the desire to train scalable GPs with non-conjugate likelihoods (such as those for classification). This approximation is often called the sparse variational Gaussian process (SVGP). SVGP models have been fitted on datasets of over 5×10^6 samples (Hensman et al., 2015).

SGPR model description

This thesis utilises the sparse GP regression model of Titsias (SGPR), which is described here. Let $u \in \mathbb{R}^m$ be the function values evaluated at the m inducing points $Z \in \mathbb{R}^{m \times p}$. Fitting a sparse GP model means finding the posterior over both f and u, which is given by Bayes theorem:

$$p(f, u \mid y) = \frac{p(y \mid f, u)\, p(f, u)}{\int\int p(y \mid f', u')\, p(f', u')\, \mathrm{d}f'\, \mathrm{d}u'}\,, \tag{2.20}$$

where the likelihood simplifies to $p(y \mid f, u) = p(y \mid f) = \mathcal{N}(y \mid f, \tau^2 I)$ as it depends only on the training data. Using this likelihood (2.20) can be solved in closed form using the joint prior $p(y, f, u)$,

$$p(y, f, u) = \mathcal{N}\left(0, \begin{pmatrix} K_{XX} + \tau^2 I & K_{XX}^T & K_{XZ}^T \\ K_{XX} & K_{XX} & K_{XZ}^T \\ K_{XZ} & K_{XZ} & K_{ZZ} \end{pmatrix}\right). \tag{2.21}$$

However, this would result in an identical model to exact GP regression if the effect of u is marginalised in the denominator and so would still require $\mathcal{O}(n^3)$ time to solve. The solution to this scalability problem is to use variational inference (also known as variational Bayes) to train the SGPR model. Variational inference approximates a posterior density with a parametric family q_ϕ, where the parameters of the chosen family (ϕ, the variational parameters) are learned via an optimisation procedure. Variational inference is a popular method for approximate inference when the exact posterior density is intractable. The specifics of variational inference are described in the next section during the derivation of the SGPR optimisation objective.

In the SGPR model $p(f, u \mid y)$ is replaced with a variational approximation $q_\phi(f, u)$ that factorises as

$$q_\phi(f, u) = p(f \mid u)\, q_\phi(u)\,, \tag{2.22}$$

where $p(f \mid u)$ is found by conditioning the joint prior (2.21). The variational posterior $q_\phi(u)$ is chosen to be the multivariate Gaussian

$$q_\phi(u) = \mathcal{N}(\mu_u, L_u L_u^T)\,, \tag{2.23}$$

where the variational parameters $\phi = \{\mu_u, L_u\}$ are its mean μ_u and Cholesky factor of its covariance L_u. The motivation for selecting the factorisation (2.22) is allows conditioning (2.21) as

$$p(f \mid u) = \mathcal{N}(f \mid K_{XZ} K_{ZZ}^{-1} u, K_{XX} - K_{XZ} K_{ZZ}^{-1} K_{XZ}^T)\,, \tag{2.24}$$

which only requires inverting K_{ZZ}, an $m \times m$ matrix, as opposed to the $n \times n$ matrix inversion needed for the full GP. This is then combined with the variational posterior $q_\phi(u)$ rather than marginalising the effect of u.

SGPR training via variational inference

The SGPR model approximates $p(f, u \mid y)$ using a variational posterior $q_\phi(f, u)$, which can be formulated as solving

$$\arg\min_{\phi} \quad \mathrm{KL}\left(q_\phi(f,u) \,||\, p(f,u \mid y)\right), \tag{2.25}$$

where $\mathrm{KL}(q(x) \,||\, p(x))$ is the Kullback-Leibler (KL) divergence between two continuous densities,

$$\mathrm{KL}(q(x) \,||\, p(x)) = \int q(x) \, \log \frac{q(x)}{p(x)} \, \mathrm{d}x \,. \tag{2.26}$$

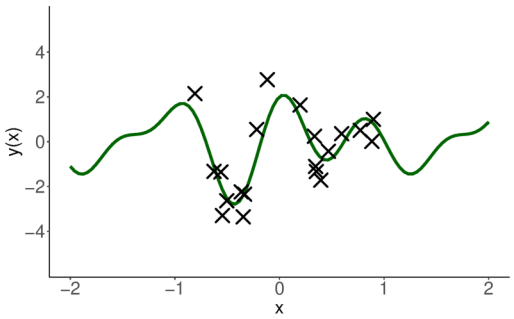
Using this definition of the KL-divergence the objective in (2.25) can be written

$$\begin{aligned}
\mathrm{KL}\left(q_\phi(f,u) \,||\, p(f,u \mid y)\right) &= \int\int q_\phi(f,u) \, \log \frac{q_\phi(f,u)}{p(f,u \mid y)} \, \mathrm{d}f \, \mathrm{d}u && (2.27)\\
&= \int\int q_\phi(f,u) \, \log \frac{q_\phi(f,u)\, p(y)}{p(y \mid f)\, p(f,u)} \, \mathrm{d}f \, \mathrm{d}u && (2.28)\\
&= \underbrace{\mathrm{KL}\left(q_\phi(f,u) \,||\, p(f,u)\right) - \mathbb{E}_{q_\phi(f,u)}\left[\log p(y \mid f)\right]}_{-\mathcal{L}_{\mathrm{SGPR}}(\phi)} + \log p(y) \,, && (2.29)
\end{aligned}$$

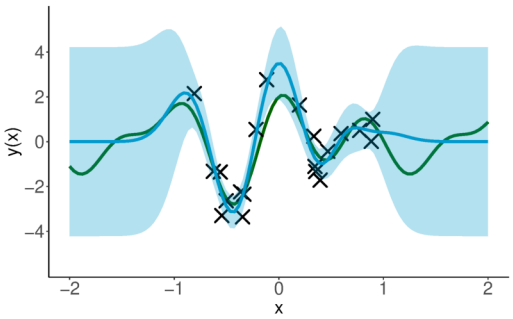
where (2.20) is used substituted for $p(f, u \mid y)$ to obtain the second line and $\mathcal{L}_{\mathrm{SGPR}}(\phi)$ is the evidence lower bound (ELBO, Blei et al., 2017). As the KL-divergence is a distance (it is non-negative) and $\log p(y)$ does not depend on ϕ, maximising $\mathcal{L}_{\mathrm{SGPR}}(\phi)$ with respect ϕ to is equivalent to solving (2.25). The SGPR training objective is therefore

$$\arg\max_{\phi} \quad \mathbb{E}_{q_\phi(f,u)}\left[\log p(y \mid f)\right] - \mathrm{KL}(q_\phi(f,u) \,||\, p(f,u)) \,, \tag{2.30}$$

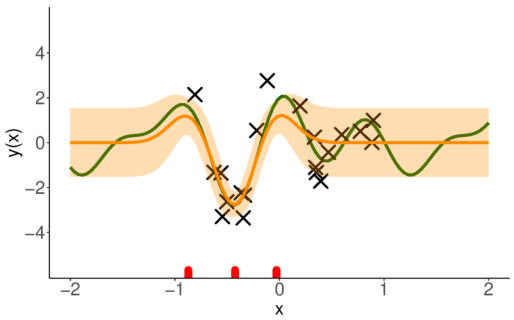
which balances a data fitting term and a regularisation term that penalises divergence of the variational posterior $q_\phi(f, u)$ from the prior $p(f, u)$. As every density in $\mathcal{L}_{\mathrm{SGPR}}(\phi)$ is multivariate Gaussian (2.30) can be solved efficiently in $\mathcal{O}(mn^2)$ time (Titsias, 2009). The training procedure jointly optimises the kernel hyperparameters, noise variance τ^2 and variational parameters μ_u and L_u. The number of inducing points, m, must be specified in advance, where larger values improve the quality of the approximation at the cost of added computational expense. An example of a SGPR fit to a univariate regression problem is shown in Figure 2.3.



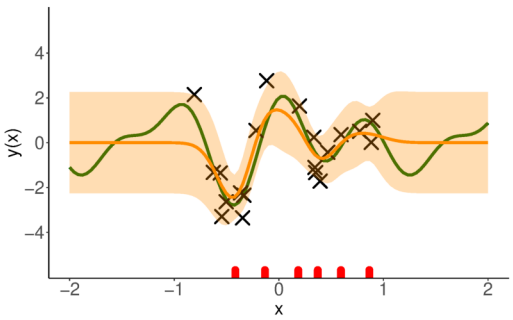
(A) Generative model (green line) and observed data (black crosses)



(B) Exact GP posterior mean (blue line) and 95% credible interval (blue shaded area)



(C) Sparse GP (3 inducing points shown as red rug) posterior mean (orange line) and 95% credible interval (orange shaded area)



(D) Sparse GP (6 inducing points shown as red rug) posterior mean (orange line) and 95% credible interval (orange shaded area)

FIGURE 2.3: Given toy regression data of 20 data points (plot A), the full GP posterior (plot B) can be approximated using a sparse GP. The quality of the approximation improves when the number of inducing points increases from 3 (plot C) to 6 (plot D).

1.3 Deterministic neural networks

Chapter 4 proposes a variable importance measure for a Bayesian neural network. Before describing Bayesian neural networks it is necessary to outline deterministic neural networks in order to understand the motivation for using its Bayesian equivalent. An L-layer feed-forward neural network for regression can be formulated as

$$y_i \sim \mathcal{N}(f(x_i;\theta),\sigma^2)\,, \quad f(x_i;\theta) = t_L(g_{L-1}(t_{L-1}\dots(t_1(x_i)))\,, \quad i = 1,\dots,n\,, \tag{2.31}$$

where $t_l\,, l = 1,\dots,L$ are element-wise affine transformations of the form $t_l(z) = w_l^T z + b_l$ with weights w_l and bias b_l and $g_l(\cdot)\,, l = 1,\dots,L$ are non-linear activation functions. The parameters of the network are $\theta = \{w_l, b_l\}_{l=1}^L$, wile the set $\{\dim(w_l)\}_{l=1}^L$ define the widths of the layers, with $\dim(w_1) = p$ (the number of input dimensions) and $\dim(w_L) = 1$ for a single-output network. The equivalent classification network is given by

$$y_i \sim \text{Bernoulli}\left(\frac{1}{1+\exp(-f(x_i))}\right)\,, \quad f(x_i) = t_L(g_{L-1}(t_{L-1}\dots(t_1(x_i)))\,, \quad i = 1,\dots,n\,, \tag{2.32}$$

where the un-normalised prediction $f(x)$ is known as a logit. According to the universal approximation theorem an exponentially wide neural network or a width-bounded deep network can approximate any continuous function to arbitrary precision (Cybenko, 1989; Z. Lu et al., 2017).

The network defined by (2.31) only contains fully-connected/dense layers, which do not account for structure inherent in the variables of x. In the case of structured data (such as images or text), specialised layers exist that retain this information. They are typically used in the early layers of a network as feature extractors with their output being passed to fully-connected layers later in the network for the final prediction. Convolutional layers achieve state of the art performance on computer vision tasks (Krizhevsky et al., 2012; Ronneberger et al., 2015) while recurrent layers do the same for sequential data such as text or time series (Graves et al., 2013). These architectures often have hundreds of layers and so are referred to as deep neural networks. The reader is directed to Goodfellow et al. (2016) for further details.

Training the network

The parameters of a neural network (θ, the weights and biases of the layers) are estimated via Maximum Likelihood using mini-batch stochastic gradient descent of a loss function (mean squared error in regression, binary cross-entropy in binary classification). The optimisation is non-convex and the existence of many local minima has led to significant research focus being placed on the optimisation procedure of neural network training. The results are a wide range of specialist gradient-based stochastic optimisation algorithms (Zeiler, 2012; G. Hinton, Nitsh Srivastava, et al., 2012; Ziegler and König, 2014; Diederik P Kingma and Ba, 2014). This is a conceptually simple calculation as the network is a composition of differentiable functions, meaning the derivates of the loss function with respect to θ can easily be computed using the chain rule. Deterministic regularisation is commonly applied by adding L1

or L2 penalties, while stochastic regularisation (in the form of dropout, which randomly sets weights to zero during training) is also used to encourage the network to learn redundant features (Nitish Srivastava et al., 2014).

1.4 Bayesian neural networks

Benefits of prediction uncertainty

While deep neural networks have revolutionised many data-rich fields, they suffer a critical limitation that hinders their adoption for critical decision making. A neural network with pointwise parameter estimates does not compute uncertainties, which are essential for high-stakes decision such as patient diagnosis. The deployment of neural networks for critical decision-making requires well-calibrated uncertainty, which cannot be obtained using pointwise parameter estimates (Leibig et al., 2017). Furthermore, the predicted probabilities of modern neural network classifiers have been shown to be highly mis-calibrated (C. Guo et al., 2017). For a well-calibrated classifier a predicted probability of 0.5 will be correct 50% of the time, but modern neural network architectures produce classifiers that are highly over-confident and unable to identify unseen samples that are far away from the training data. This has led to a recent increase in the popularity of Bayesian neural networks, where pointwise parameter estimates are replaced with posterior densities (Jospin et al., 2022).

Bayesian neural network model description

A Bayesian neural network model for regression (analogous to (2.31)) is

$$y \sim \mathcal{N}(f(x;\theta),\sigma^2)\,, \quad f(x;\theta) = t_L(g_{L-1}(t_{L-1}\ldots(t_1(x)))\,, \quad \theta \sim \pi(\theta)\,, \tag{2.33}$$

where $\pi(\theta)$ is the prior distribution over the network parameters. Training the network using Bayesian inference requires evaluating the posterior using Bayes Theorem,

$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta)\, p(\theta)}{\int p(\mathcal{D} \mid \theta')\, p(\theta')\, \mathrm{d}\theta'}\,. \tag{2.34}$$

The gold-standard for performing Bayesian inference in non-conjugate models is to use MCMC sampling, which is not feasible in the deep learning setting for a number of reasons. Firstly, existing MCMC sampling algorithms are not scalable to the dataset sizes required to train an effective deep learning model as they require the whole dataset to be processed at each sampling iteration. In addition, the integral in (2.34) is over the network parameters, which number comfortably over 10^6 in modern architectures (Ronneberger et al., 2015). Neural networks are also invariant under a number of transformations (weight permutations and sign flips, for example), meaning that the posterior contains a large number of equally likely modes. This presents an additional challenge for MCMC samplers, leading to low acceptance rates and slow convergence (Papamarkou et al., 2019).

Approximate inference for Bayesian neural networks

These obstacles have led researchers and practitioners to focus almost entirely on variational inference to train Bayesian neural networks by approximating the intractable true posterior (G. E. Hinton and Van Camp, 1993; Barber and Bishop, 1998; Graves, 2011). This more scalable approach replaces the high-dimensional sampling required to compute (2.34) with an optimisation problem. Similarly to the SGPR model, the posterior $p(\theta|\mathcal{D})$ is approximated using $q_\phi(\theta)$. The network is trained by solving

$$\hat{\phi} = \arg\min_{\phi} \quad \mathrm{KL}(q_\phi(\theta) \,||\, p(\theta|\mathcal{D}))\,, \tag{2.35}$$

which selects the member of the family that is closest to the true posterior. The objective in (2.35) contains the unknown true posterior but is equivalent to the ELBO,

$$\mathcal{L}_{\mathrm{BNN}}(\phi) = \mathbb{E}_{q_\phi(\theta)} \log\, p(y \mid \theta) - \mathrm{KL}(q_\phi(\theta) \,\|\, \pi(\theta))\,, \tag{2.36}$$

where $\pi(\theta)$ is the prior of the network parameters. The equivalence between (2.35) and (2.36) is shown in the next subsection and follows a similar logic as is used for the SGPR ELBO derivation.

Once again, the ELBO is made up of a reconstruction error term and a term that provides regularisation by keeping $q_\phi(\theta)$ close to the prior. The Bayesian neural network ELBO can be optimised using the same stochastic mini-batch algorithms that are used to solve (2.35) using Monte Carlo estimates of the reconstruction term (Diederik P Kingma and Welling, 2013; Rezende et al., 2014; Durk P Kingma et al., 2015; Blundell et al., 2015).

Despite the popularity of variational inference for Bayesian neural networks relatively few works consider the accuracy of the variational approximation (Yao et al., 2018; Huggins et al., 2020), instead focusing on empirical evaluation of uncertainty calibration (Filos et al., 2019; Ovadia et al., 2019). In the majority of cases practitioners use Gaussian mean-field variational inference, where the variational posterior $q(\theta)$ is a Gaussian that factorises fully over the parameters. This is a major limitation, although it has been shown that it is less restrictive in deep networks than in shallow ones (Farquhar et al., 2020). More importantly, maximising (2.36) favours a solution that underestimates posterior variances, meaning that they are often unsuitable for critical decision-making as such networks produce over-confident predictions. Significant research effort has been placed on finding richer variational families for neural networks, including Gaussians with rank-one-plus-diagonal covariance (Rezende et al., 2014; Mishkin et al., 2018), mixture distributions (Graves, 2016), "boosted" mixtures (F. Guo et al., 2016; A. C. Miller et al., 2017), matrix-variate Gaussian posteriors (Louizos and Welling, 2016) and normalising flows (Louizos and Welling, 2017). It is standard practice in Bayesian deep learning to use a standard normal prior, but recent work has achieved sparsity in the network weights using horseshoe priors (Ghosh et al., 2019).

It has been claimed that the stochastic regularisation technique dropout is equivalent to variational Bayesian inference in a deep Gaussian process model (Gal and Ghahramani, 2016), although this has been contested by the observation that the posterior induced by dropout does not concentrate as the number of data increase

(Osband, 2016). More recent work has shown that the quality of the approximation is very poor even in extremely simple cases, and worse than the mean-field approximation (Folgoc et al., 2021).

An alternative line of research seeks to develop MCMC samplers that can operate on mini-batches, named stochastic gradient Markov chain Monte Carlo (after the stochastic optimisers that have helped fuel the rise of deep learning, Nemeth and Fearnhead, 2021). Other approaches to estimate neural network uncertainty utilise an ensemble of networks, with each member trained using a different parameter initialisation (so-called deep ensembles, Lakshminarayanan et al., 2017). While the original deep ensembles have no Bayesian interpretation, subsequent work modified the training procedure such that the resulting ensemble is equivalent to samples from a Gaussian process posterior (B. He et al., 2020).

Bayesian neural network training via variational inference

Fitting a Bayesian neural network using variational Bayes requires solving

$$\hat{\phi} = \arg\min_{\phi} \quad \mathrm{KL}(q_\phi(\theta) \,||\, p(\theta \mid y)) \,, \tag{2.37}$$

to find the member of the parametric family $q_\phi(\theta)$ that is closest to the true posterior. This is the same procedure as was used to derive the SGPR ELBO in the previous section, but here the variational posterior is over the network parameters. Following the same steps as in the SGPR ELBO derivation,

$$\begin{aligned}
\mathrm{KL}(q_\phi(\theta) \,||\, p(\theta \mid y)) &= \int q_\phi(\theta) \log \frac{q_\phi(\theta)}{p(\theta \mid y)} \,\mathrm{d}\theta && (2.38)\\
&= \int q_\phi(\theta) \log \frac{q_\phi(\theta)\, p(y)}{p(y \mid \theta)\, p(\theta)} \,\mathrm{d}\theta && (2.39)\\
&= \mathrm{KL}(q_\phi(\theta) \,||\, p(\theta)) - \mathbb{E}_{q_\phi(\theta)} \left[\log p(y \mid \theta)\right] + \log p(y) \,, && (2.40)
\end{aligned}$$

where the ELBO here is given by

$$\mathcal{L}_{\mathrm{BNN}}(\phi) = -\mathrm{KL}(q_\phi(\theta) \,||\, p(\theta)) + \mathbb{E}_{q_\phi(\theta)} \left[\log p(y \mid \theta)\right] \,. \tag{2.41}$$

Once again, the KL-divergence term in (2.40) is non-negative and $\log p(y)$ is independent of ϕ, meaning that (2.35) is equivalent to

$$\arg\max_{\phi} \quad \mathbb{E}_{q_\phi(\theta)} \left[\log p(y \mid \theta)\right] - \mathrm{KL}(q_\phi(\theta) \,||\, p(\theta)) \,. \tag{2.42}$$

When training Bayesian neural networks using variational inference it is common to introduce an additional hyperparameter and solve

$$\arg\max_{\phi} \quad \mathbb{E}_{q_\phi(\theta)} \left[\log p(y \mid \theta)\right] - \beta \,\mathrm{KL}(q_\phi(\theta) \,||\, p(\theta)) \,, \tag{2.43}$$

where $\beta > 0$ weights the relative importance of the negative log-likelihood (data fitting) and the prior (Higgins et al., 2016)

Last layer Bayesian neural networks

A Bayesian neural network trained using mean-field variational inference has twice as many parameters as the equivalent deterministic neural network as each parameter is replaced by its mean and variance. There are computational difficulties in learning a posterior distribution with such a high dimensionality, which has led to the application of *last layer Bayesian* neural networks in areas such as large-scale regression (Lázaro-Gredilla and Figueiras-Vidal, 2010; Watson et al., 2021), the multi-armed bandit problem (Riquelme et al., 2018; Weber et al., 2018) and Bayesian optimisation (Snoek, Rippel, et al., 2015).

Treating only the final layer as Bayesian fits into the general understanding of neural networks where inner layers are feature extractors and the final layers compute predictions based on those features (Notley and Magdon-Ismail, 2018). Restricting Bayesian treatment to the parameters of the final layers (those which compute predictions) therefore decouples the task of representation learning and uncertainty quantification. Recent studies have shown that last layer Bayesian networks are sufficient to capture relevant uncertainty, detect out-of-distribution samples and address overconfidence problems (Zeng et al., 2018; Brosse et al., 2020; Kristiadi et al., 2020). This suggests that uncertainty in the learned representations is often less useful than uncertainty for the prediction layers and therefore not worth the additional computational and statistical challenges that accompany a Bayesian treatment of the entire network.

Last layer Bayesian regression model

The last layer Bayesian model that is used in Chapter 4 assumes

$$y_i \sim \mathcal{N}(\mu(x), \sigma^2(x_i)), \quad i = 1, \ldots, n, \tag{2.44}$$

where $\mu(\cdot)$ and $\sigma^2(\cdot)$ are the two outputs of a single neural network, given by

$$\begin{pmatrix} \mu(\cdot) \\ \sigma^2(\cdot) \end{pmatrix} = w\, h_\theta(\cdot) + b, \quad \begin{pmatrix} w \\ b \end{pmatrix} \sim p(\tilde{\theta}), \tag{2.45}$$

where the predicted mean and variance that parametrise (2.44) are linear combination of the penultimate layer activations $h_\theta(\cdot)$ and the weights and biases of the final layer $\{w, b\} = \tilde{\theta}$. The final layer parameters $\tilde{\theta}$ are random variables while the inner layer parameters, denoted by θ, are point estimates. The model defined by (2.44)-(2.45) is essentially Bayesian linear regression with neural network features $h_\theta(\cdot)$. An simple example architecture is shown in Figure 2.4.

The loss function is the ELBO (2.36), but is written here with the inner and final layer parameters separated:

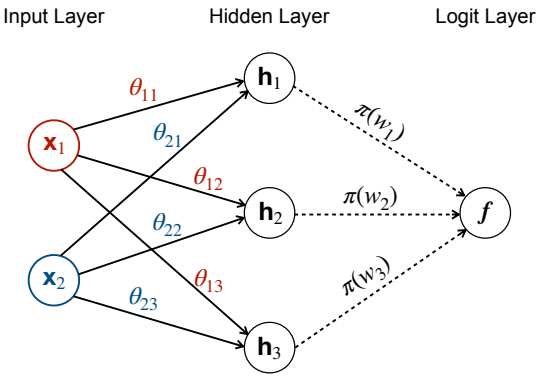


FIGURE 2.4: An example of a last layer Bayesian network. The first layer weights/biases and the final layer bias θ are point estimates, while the final layer weights w are assumed to be distributed under the prior $\pi(\cdot)$. The input variables are fed through the hidden layer to compute the hidden layer activations $(h_1, h_2, h_3)^T$. Samples of the predictions f are obtained from a linear combination of these activations with samples from the posterior of $w = (w_1, w_2, w_3)$, which is $q_\phi(w)$. This figure does not include the bias terms.

$$\arg\max_{\phi} \quad \mathbb{E}_{q_\phi(\tilde{\theta})} \left[\log p(y \mid x, \tilde{\theta}, \theta) \right] - \beta \, \mathrm{KL} \left[q_\phi(\tilde{\theta}) \,||\, p(\tilde{\theta}) \right] . \tag{2.46}$$

Note that the regularisation term only depends on $\tilde{\theta}$ and not on any pointwise parameters from the inner layers.

Samples from the predictive posterior are drawn as follows:

$$\begin{aligned}
\begin{pmatrix} \widehat{w} \\ \widehat{b} \end{pmatrix} &\sim q_\phi(\tilde{\theta}) && \text{sample final layer parameters} \\
\begin{pmatrix} \mu(x) \\ \sigma(x) \end{pmatrix} &= \widehat{w}\, h_\theta(x) + \widehat{b} && \text{mean/variance from neural network} \\
\hat{y} &\sim \mathcal{N}(\mu(x), \sigma^2(x)) && \text{sample prediction from normal distribution}
\end{aligned}$$

A similar last layer Bayesian neural network for classification is described in Chapter 4 (Section 7).

1.5 Decision tree ensembles

Decision trees are a popular supervised learning method that learn simple splitting rules from training data. These splits define a recursive partitioning of the input space, where each region is assigned a single response value. Decision trees therefore compute piecewise constant approximations. An example of a classification tree is shown in Figure 2.5(A), with the corresponding partitioning of the input space shown in Figure 2.5(B).

A single decision tree is easily interpretable but is also severely prone to over-fitting, especially as their depth increases. This can be addressed by pruning a fully grown

tree, but a more popular option is to use an ensemble of decision trees (each of which are weak learners), whose average prediction may be extremely accurate (a strong learner). The two most popular decision tree ensembles models are random forest (Breiman et al., 1984) and gradient boosting machines (Friedman, 2001). In a random forest each tree is trained on a bootstrap sample (bootstrap aggregating, or *bagging*), while in a gradient boosting machine the trees are trained sequentially, with each subsequent tree correcting the mistakes of its predecessor (boosting). Chapter 3 focuses on random forests.

A random forest consisting of T decision trees makes predictions according to

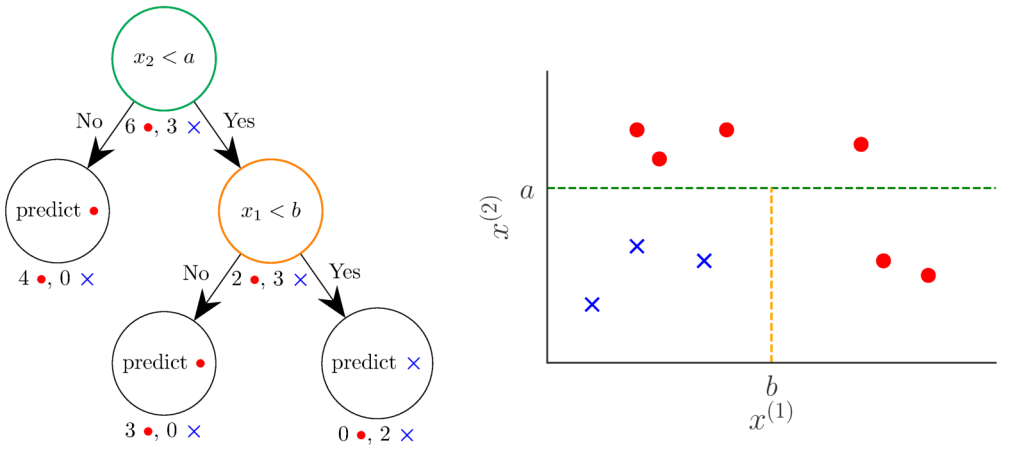
$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} f_t(\tilde{X}_t; \theta_t)\,, \tag{2.47}$$

where each $\tilde{X}_t \in \mathbb{R}^{n\times p}\,, t = 1, \ldots, T$ is a design matrix constructed by bootstrapping the rows of X and $f_t(\cdot, \theta_t)$ is a decision tree defining a piecewise-constant partition of the input space with split points θ_t. This partitioning corresponds to a tree structure where the root node of the t^{th} tree contains all the samples in $\tilde{X}_t$ and the leaf nodes are the average label (mean in regression and mode in classification) of their samples (see the example in Figure 2.5). For each tree in the ensemble, the samples not included in $\tilde{X}_t$ are called the out-of-bag (OOB) samples.

There are a number of greedy algorithms that construct decision trees, the most popular of which is Classification and Regression Trees (CART, Breiman et al., 1984). CART performs a greedy search of variable splits to partition the input space in increasingly "pure" regions (or more commonly, decreasingly impure regions). In CART impurity is measured for a split point using the variance of the labels (for regression) or the Gini impurity (classification). CART performs a greedy search for variable splits that have the largest impurity decrease. This can be seen from Figure 2.5(B) - the selected split points maximise the impurity gain as they define a partitioning of the input space where each region contains only a single class.

Tree construction algorithms contain a number of hyperparameters that are selected using cross-validation, such as the maximum tree depth and the number of candidate variables tested for each split (often called `mtry`). Increasing the number of trees will always increase the performance of a random forest, but the computational cost increases linearly with T while the performance benefit tends to plateau once it increases above a dataset dependent-value. Increasing the number of trees also increases the stability of variable importance scores (Huazhen Wang et al., 2016).

Random forests are one of the most popular models in bioinformatics as they are non-linear, non-parametric, are well-suited to the $n << p$ regime and are able to model complex correlation structures between variables (Ishwaran, Kogalur, Gorodeski, et al., 2010; Qi, 2012). This flexibility means they often exhibit superior predictive performance relative to linear models for biological datasets (Fernández-Delgado et al., 2014; Couronné et al., 2018). Furthermore, they offer a degree of interpretability as it is possible to evaluate variable importance, although the interpretation of these importance scores is more difficult than for linear models. The primary biomedical applications of random forest have been in genomics (X. Chen and Ishwaran, 2012) and survival analysis (Ishwaran, Kogalur, X. Chen, et al., 2011; Hong Wang and G. Li, 2017). Other example applications of random forests include the identification of microbial species associated with Crohn's disease (Tedjo et al., 2016), viral sequences associated with Type 1 diabetes (G. Zhao et al., 2017) and important transcription



(A) The decision tree expressed as a set of splitting rules.

(B) The corresponding piecewise-constant partitioning of the input space

FIGURE 2.5: An example of a decision tree binary classifier constructed on a training set of 9 examples. Red dots and blue crosses denote the two classes.

factors for different mouse cell types (Consortium et al., 2018). The characteristics of random forest variable importance scores are discussed in detail in the next section.

2 Variable importance analyses

Given a trained model it is common to perform some type of variable importance analysis to identify a subset of variables (the dimensions of $\mathcal{X}$) that are associated with the response. For example, the variables may be biologically or clinically relevant quantities such as age, genetic sequence or other biomarkers, while the response is a phenotype (an observable characteristic). This requirement naturally follows from the research aims of biological research projects, which are to learn about the underlying process that generate data. The predictive model is assumed to be an abstraction of this process that can be used to investigate mechanisms that drive the response. Common examples of variable importance analysis include genome-wide association studies (regression coefficients summarise the association between a position on the genome and phenotype), biomarker discovery (Y.-H. Yun et al., 2016; Leclercq et al., 2019) and popular generalised linear model-based differential expression analysis tools such as Limma and DeSeq2 (Love et al., 2014; Ritchie et al., 2015).

The variable importance analysis can be *post-hoc* (as in the case of permute-and-predict methods) or be part of the training procedure (regression coefficients in linear models). The interpretation of these variable importance scores depends on both the type of model and the choice of variable importance method.

Models with readily available importance scores are often considered interpretable. The standard example of an interpretable model is the generalised linear model, which computes regression coefficients as part of the model fitting process. These regression coefficients (effect sizes) quantify the dependence of the transformed response on each variable and summarise the global behaviour of the model. There also exists a long history of statistical research on the interpretation of regression

coefficients and their asymptotic properties, which enables null hypothesis significance testing and robust confidence intervals. At the other end of the spectrum, kernels and neural networks operate as black boxes.

2.1 Interpretability via variable importance

As neural networks have become increasingly popular the fact that they operate as black boxes has become an important focus of machine learning research. This lack of interpretability has been the major challenge as researchers seek to apply these hugely successful models to high-stakes applications in healthcare and the life sciences. This work has coalesced into a formal sub-field of artificial intelligence known as "interpretable AI". There exist multiple definitions for what interpretable models are/should be, with some definitions focusing on how well a human can understand a model's prediction(s) (T. Miller, 2019), while others prioritise how well a human can anticipate them (Been Kim et al., 2016). However, there is a general consensus around the goal of interpretability as extracting human-understandable information on relationships from a trained model despite the ongoing lack of a formal framework (Doshi-Velez and Kim, 2017; Gilpin et al., 2018; Lipton, 2018).

One possible avenue to interpretability is variable importance (Murdoch et al., 2019). This route is particularly relevant in the life sciences for the reasons outlined in the previous section. The desire to utilise more complex machine learning models has spawned a significant volume of work on computing *post-hoc* interpretations of deep neural networks by inspecting their gradients or activations (Alqaraawi et al., 2020). Variable importance for decision tree ensembles have also been the subject of ongoing research for a number of years, largely motivated by biological applications (Strobl, Boulesteix, Zeileis, et al., 2007; Strobl, Boulesteix, Kneib, et al., 2008; Altmann et al., 2010; Nembrini et al., 2018; Ishwaran and M. Lu, 2019; Degenhardt et al., 2019).

Global and local variable importance

One common distinction in the interpretable machine learning literature is between global and local importance scores. Local importance scores explain a single instance/example while global scores aim to explain on the level of an entire dataset. The most famous local method is LIME (local interpretable model-agnostic explanations, Ribeiro et al., 2016), which assumes that the prediction function of a black box model is linear in the vicinity of a given example. Examples of global importance scores include effect sizes in linear models or mean decrease accuracy and mean decrease Gini scores for tree ensembles (described in Section 2.6).

The choice of global vs local importance is determined by the application and aims of an analysis. Local importance is more useful in situations where the variables already have an established and well-understood meaning, such as pixels in an image. However, when the variables themselves are relatively poorly understood - which is often the case in biological studies - a global importance score is usually more appropriate. The variable importance analyses in this thesis are global methods for this reason.

For an example of the different utilities of global and local scores consider a case-control genome-wide association study. The study aim is not to explain why specific individuals have the disease of interest, but rather to identify which variables (positions on the genome) are associated with an increased disease risk across all

study participants. A global importance score is therefore required, which is provided by the regression coefficient of a linear model. Now consider a model deployed in a clinical setting to predict the risk of a well-understood disease. If such a model is trained on a set of genetic markers, biomarkers and environmental features known to be relevant to the disease in question, then the problem of explaining a predicted risk for a given patient is the more relevant (local interpretability). Given a model prediction for a specific patient it would be of high clinical relevance to explain which variables were driving the prediction in order to tailor treatment options to the patient in question. This scenario is the long-term goal of precision medicine, where treatments are tailored to individual genetics, environment and lifestyle (Ginsburg and Phillips, 2018). These two examples demonstrate the different contexts in which global and local interpretability can be applied in a biomedical context. It is also worth noting that the genetic features included in the model of the well-understood disease would most likely have been established by global importance analyses in previous genome-wide association studies.

In some settings global importance scores can be computed from a set of local scores using a simple aggregation method. For example, permute-and-predict scores are the mean score over the set of permuted examples. However, not all local scores can be combined in this way - the authors of LIME, for example, explicitly warn against combining LIME's local importances as each linear model used to assign importance to an example is only valid locally.

2.2 Variable importance vs variable selection

Variable selection is a closely-related statistical procedure to variable importance, which is also commonly referred to as feature selection in the literature (Saeys et al., 2007; Jović et al., 2015; Remeseiro and Bolon-Canedo, 2019). Variable selection methods are generally model-agnostic and produce a final model trained on a subset of the available predictors, with that subset assumed to be the most associated with the response. This is typically done in an iterative fashion, where the initial model is trained on all available variables with variables iteratively eliminated (backward elimination), or where the initial model contains no variables (forward selection, Heinze et al., 2018). The stopping criterion for these iterations can be based on significance testing, Akaike/Bayesian information criterion or predictive performance on held-out data, while the criterion for selecting which variable to add/remove can also be formulated in a similar way, or by using variable importance scores. The popular recursive feature elimination algorithm combines backward elimination with an elimination criterion based on variable importance scores (Guyon et al., 2002; Svetnik et al., 2004; Gregorutti et al., 2017).

The studies in this thesis do not explicitly investigate variable selection but there are significant connections between the two tasks, as any set of variable importance scores can be used for variable selection given a threshold score below which variables are excluded. This threshold is often chosen heuristically in random forest modelling (Genuer et al., 2010). Variable selection can also be included in some models via an L1 penalty, as is the case in Lasso and ElasticNet (Tibshirani, 1996; Zou and Hastie, 2005).

2.3 Types of variable importance methods

The rest of this section described the mathematical basis for the variable importance methods used in this thesis. The majority of these methods are *post-hoc*, meaning that they operate on the trained model. The only exception is the impurity importance for random forest models which only depends on the structure of the trees. Given a trained model each method computes a set of per-variable scores $s = (s^{(1)}, \ldots, s^{(p)})$, where a larger value of $s^{(j)}$ implies a higher level of association between variable j and the response.

2.4 Model-agnostic variable importance measures

Permutation importance

The most conceptually simple variable importance scoring method is the permutation importance, which was first described by (Breiman et al., 1984) in the context of random forests. Given the prediction function of a trained model $f(X)$, dataset $\mathcal{D} = (X, y)$ and scoring function $L(y, f(X))$ it assigns scores using

$$s_j = L(y, f(\tilde{X}_j)) - L(y, f(X)), \tag{2.48}$$

where $\tilde{X}_j$ is formed by permuting the j^{th} column of X. Permutation importance therefore scores each variable according to the decrease in the performance metric defined by L when that variable is permuted.

A recent review strongly advised against using permutation importance to interpret black box models when there are dependencies between variables (as is almost always the case), finding that they give highly misleading results (Hooker et al., 2021). This stems from the fact that permuting a variable will only reduce predictive performance by an amount proportional to its true importance if there are no other variables in the dataset that also contain redundant predictive signal. Furthermore, permutation importance is also biased by the fact that permuting features is liable to produce unrealistic (or even impossible) data instances (Molnar, 2020). For example, permuting a feature is likely to generate examples that are far away from any of the training or test data (especially in high dimensions), while impossible instances can occur when features are highly correlated. The issue of highly correlated features is part of the motivation for the grouped variable importance approach presented in Chapter 4.

Shapley values

Shapley values are a game-theoretic concept that are becoming increasingly popular in the interpretability literature. In the game theory setting a set of players (variables) generate a payout (model prediction) and Shapley values calculate the optimal share of the payout that should go to each player by considering their relative contributions. Shapley values therefore provide a natural way of explaining black box predictions. Shapley values compute global importance scores.

The Shapley value for the j-th variable is,

$$s_j = \sum_{S \subseteq \{1,\dots,p\} \setminus j} \frac{|S|!(p-|S|-1)!}{p!} \left(v(S \cup \{j\}) - v(S)\right) , \tag{2.49}$$

where S is a subset of the features in the model and $v(\cdot)$ is the value function given by

$$v(S) = \int f(x^{(1)}, \dots, x^{(p)}) \, d\mathbb{P}_{\{x^{(k)}: k \notin S\}} - \mathbb{E}_X[f(X)] , \tag{2.50}$$

where $\mathbb{E}_X[f(X)]$ is the model prediction on the observed samples and $x^{(j)} = (x_1^{(j)}, \dots, x_n^{(j)})$ is the j^{th} column of X. If no variables are excluded then $S = \emptyset$, in which case the Shapley value of the j-th variable is fully determined by $v(\{j\})$, which is

$$v(\{j\}) = \int f(x^{(1)}, \dots, x^{(p)}) \, d\mathbb{P}_{\{x^{(k)}: k \neq j\}} - \mathbb{E}_X[f(X)] . \tag{2.51}$$

This illustrates that Shapley values are the contribution of variable j to the observed prediction, as only variable j is not marginalised in the first term. This is prohibitively expensive to estimate for any medium-sized model so a Monte Carlo estimator is used in practice (Štrumbelj and Kononenko, 2014)

2.5 Variable importance methods for neural networks

As discussed in previous sections, neural networks have become ubiquitous in many fields due to their impressive predictive performance, which they achieve by leveraging interactions between variables to construct complex predictive features. However, they operate as black boxes, meaning that it is difficult or impossible to evaluate variable importance. This lack of interpretability has hindered the adoption of deep learning even as these models match the diagnostic performance of human experts in areas such as medial imaging-based diagnoses (Esteva et al., 2017; Ting et al., 2017).

Saliency maps

Saliency maps are one of the most popular methods for interpreting neural networks and are most commonly applied in computer vision. Saliency maps are an active area of interpretability research and so are included in Chapter 4. In computer vision applications saliency maps are used to explain a network's mis-classified examples in order to shed light on a model's behaviour. All saliency methods assign importance using the gradient of the network's prediction function with respect to a single input and so are local importance methods. However, they can be agglomerated over a set of examples to produce a global score.

Despite their popularity, several major criticisms of saliency maps have been highlighted, such as invariance under randomisation of network parameters or permutations of class labels (Adebayo et al., 2018) and high levels of instability under adversarial attack (Ghorbani et al., 2019). Saliency maps provide very limited insight on how a model will behave for unseen samples, instead only providing information on where the network is looking for a given image (Rudin, 2019; Alqaraawi et al.,

2020). It has also been reported that saliency maps computed for deep neural networks trained on medical images do not correspond to relevant regions according to human experts, calling into question the safety and efficacy of other results that report human-level prediction (Saporta et al., 2021; Arun et al., 2021). Despite this, a plethora of new saliency methods are proposed each year with a relatively small emphasis placed on systematic evaluation of existing methods (Kummerer et al., 2018).

Description of saliency-based methods used in this thesis

Given an input $x \in \mathbb{R}^p$, the first and most simple saliency-based method for deep neural networks attributes a vector of scores $s \in \mathbb{R}^p$ equal to the absolute value of gradient of the model output $f(x)$ with respect to the input,

$$s_{\text{vanilla-grad}} = \left| \frac{\partial f(x)}{\partial x} \right| , \tag{2.52}$$

where f are usually logits (un-normalised pre-activation final layer outputs) in classification problems (Simonyan et al., 2014). However, f can be chosen to be (e.g.) the pre-activations of a hidden layer in order to inspect the features learned by the network. Assigning importance using (2.52) is often termed the Vanilla gradients method. A simple extension of (2.52) is the so-called gradient$\times$input method, where the gradients are weighted by the values of inputs,

$$s_{\text{grad-input}} = \left| \frac{\partial f(x)}{\partial x} \right| \odot x , \tag{2.53}$$

where $\odot$ denotes the Hadamard (element-wise) product. This is designed to avoid an unfortunate feature of (2.52), where it attributes artificially high importance to variables with very small values. This gradient$\times$input method is equivalent to another popular attribution method called Layerwise Relevance Propagation (Bach et al., 2015) for networks with ReLU (rectified linear unit) activations for their hidden layers (Kindermans et al., 2016; Shrikumar et al., 2017). A second undesirable property of the importance scores computed using (2.52) is that they are prone to saturation. For ReLU activations (where $g(x) = \max(0, x)$), if an input decreases below zero its gradient is zero. Such an input will be assigned zero importance using both (2.52) and (2.53). Integrated gradients seeks to address this issue by integrating gradients along a path from a baseline input $\bar{x}$ to the input itself,

$$s_{\text{int-grad}} = (x - \bar{x}) \int_0^1 \frac{\partial f(x + \alpha(x - \bar{x}))}{\partial x} \, \mathrm{d}\alpha , \tag{2.54}$$

where $\alpha \in [0, 1]$ is a scalar that linearly interpolates the path from x to the baseline input, which is usually set to be all zeros (Sundararajan et al., 2017). Another undesirable property of scoring variables using (2.52) is the sensitivity to noise, resulting in saliency maps that appear visually noisy to humans. The Smoothed gradients method (Smilkov et al., 2017) addresses this by adding Gaussian noise to the gradients and taking the mean over S Monte Carlo samples,

$$s_{\text{smooth-grad}} = \frac{1}{S}\sum_{k=1}^{S}\left|\frac{\partial f(x+\varepsilon_k)}{\partial x}\right|, \quad \varepsilon_k \sim \mathcal{N}(0,\sigma^2), \quad k = 1,\ldots,S. \tag{2.55}$$

Guided back-propagation (Springenberg et al., 2015) was developed for convolutional neural networks with ReLU activations and uses a modified form of the back-propagation algorithm used for network training to attribute importance. In Guided back-propagation only positive activations are back-propagated through the network,

$$s_{\text{guided-bp}} = \prod_{l=L}^{1}\left|\frac{\partial f_l(x)}{\partial f_{l-1}(x)}\right| \mathbb{1}\left(f_l(x) > 0\right), \tag{2.56}$$

where $l \in \{L, \ldots, 1\}$ indexes the layers in reverse order, $f_l(x)$ is the output of the l-th layer and $\mathbb{1}(\cdot)$ is the indicator function.

Mimic models

An alternative approach to achieving global model interpretability is to train an interpretable model (typically a decision tree ensemble) on the predicted class probabilities of the neural network, then use the variable importance scores of this mimic model as a surrogate for those of the neural network. These ideas originated in the field of model distillation, where the predictive power of a trained deep network is transferred to a much smaller network for computational reasons (e.g. to run on handheld devices, (Ba and Caruana, 2014; G. Hinton, Vinyals, et al., 2015)). The smaller model is unable to achieve sufficiently strong predictive performance when trained directly on the data but is able to effectively mimic the larger network, which has been trained directly on the data. Decision tree ensembles are popular choices for mimic modelling as they are non-parametric and non-linear, which are useful properties when the response is the decision function of a neural network (Che et al., 2016; Q. Zhang et al., 2019). Decision trees and decision rule lists are also popular choice for the interpretable mimic but suffer from a lack of capacity to model the full complexity of neural network predictions (Davoodi and Moradi, 2018).

Given a dataset $\mathcal{D} = (X, y)$ and a trained model with prediction function $f(x)$, a mimic model is a regression model trained on a new dataset $\tilde{\mathcal{D}} = (X, f)$, where f is the vector of predictions that correspond to X.

2.6 Variable importance for random forests

Individual decision trees are rarely applied to complex biological datasets as they are weak learners and random forests are a much more popular choice for the reasons discussed in Section 1.5. However, this increase in model capacity is offset by an equivalent loss in interpretability as the ensemble usually contains over 100 trees. While less interpretable than decision trees, interpretability is still available for random forests via variable importance methods, which is one of the major reasons for their enduring popularity in bioinformatics.

Mean decrease Gini and Mean decrease accuracy

The two most common global variable importance methods for random forest are mean decrease Gini (MDG) and mean decrease accuracy (MDA), both of which were proposed in the original CART book (Breiman et al., 1984). MDA is simply the model agnostic permute-and-predict method described in Section 2.4 applied with the out-of-bag samples for each tree in the forest. Recall that the CART algorithm greedily selects the split point that maximises the impurity gain, where the impurity at a given split point is quantified in the CART algorithm using one of

$$\mathcal{I} = \begin{cases} 1 - \sum_{i=1}^{k} p_k^2 & \text{Gini impurity (classification)} \\ \dfrac{1}{n-1} \sum_{i=1}^{n} (y_i - \bar{y})^2 & \text{variance (regression)} \end{cases} \tag{2.57}$$

where n is the number of samples, p_k is the frequency of class k, y_i is the predicted (continuous) value for the i^{th} sample and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ is the mean prediction. All the quantities in (2.57) are defined for a single node in the tree. MDG gets it name from the CART algorithm, which uses Gini impurity to evaluate splitting rules in classification trees. While the Gini impurity is not defined for continuous labels the name MDG has been adopted as a general term for impurity-based importance measures even when the impurity measure is not necessarily measured using Gini (for example, in regression or in classification trees using alternative impurity measures).

The idea behind MDG is that important variables are those which are responsible for large decreases in impurity in the tree. The importance of variable j is calculated using

$$\mathrm{s}_{\text{imp}}^{(j)} = \frac{1}{|\mathcal{V}_j|} \sum_{v \in \mathcal{V}_j} \Delta\mathcal{I}_v \,, \tag{2.58}$$

where $\mathcal{V}_j$ is the set of nodes in the forest at which variable j is used to split and $\Delta\mathcal{I}_v$ is the corresponding impurity decrease from node v to its two children. However, variable importance scores calculated using (2.58) are well-known to be biased towards discrete variables with a larger number of categories or continuous variables on a larger scale (Strobl, Boulesteix, Zeileis, et al., 2007). MDA has therefore become the more popular option for calculating random forest variable importance scores in biomedical applications (Archer and Kimes, 2008; Nicodemus et al., 2010; Szymczak et al., 2016; Gregorutti et al., 2017; Ishwaran and M. Lu, 2019). This is despite its much larger computational cost, especially for high-dimensional data. MDG importance scores can be more stable under data perturbations (removal of 10% of the samples) than MDA scores in differential expression simulations (Calle and Urrea, 2011). However, the opposite result can be observed when there is high levels of correlation between variables (Nicodemus, 2011). The desirable properties of MDG scores motivated the development a series of works to de-bias MDG scores by decomposing the impurity decrease in (2.58) as

$$\Delta\mathcal{I}_v = \Delta\mathcal{I}_v^{(s)} + \Delta\mathcal{I}_v^{(b)} \,, \tag{2.59}$$

where $\Delta\mathcal{I}_v^{(s)}$ is the impurity decrease due to the true importance of a variable and

$\Delta\mathcal{I}_v^{(b)}$ is the impurity decrease due to its structure (e.g. its scale or number of categories, Sandri and Zuccolotto, 2008). The most computationally efficient method is that of Nembrini et al. (2018), which trains the random forest on an augmented dataset of $2p$ variables, where the set of variables $\mathbb{O} = \{1, \ldots, p\}$ are those observed in the datasets and the remainder $\mathbb{P} = \{p+1, \ldots, 2p\}$ are permuted versions of each member of $\mathbb{O}$. The reasoning behind this approach is each variable in $\mathbb{O}$ will have the same $\Delta\mathcal{I}_v^{(b)}$ as its counterpart in $\mathbb{P}$, but the variables in $\mathbb{P}$ will have $\Delta\mathcal{I}_v^{(s)} = 0$.

Trees are constructed by sampling variables from $\mathbb{O} \cup \mathcal{P}$ with variable importance scores calculated using

$$\mathrm{s}^{(j)}_{\text{db-imp}} = \frac{1}{|\mathcal{V}_j|} \sum_{v \in \mathcal{V}_j} \Delta\mathcal{I}_v - \sum_{v \in \mathcal{V}_{j'}} \Delta\mathcal{I}_v \,, \tag{2.60}$$

where j' is the member of $\mathbb{P}$ that corresponds to j ($j' = j + p$). Therefore the importance of the perturbed variables, for which $\Delta\mathcal{I}_v^{(s)} = 0$, is subtracted from the importance score, leaving an estimate of $\Delta\mathcal{I}_v^{(s)}$.

Assessing statistical significance for random forest scores

While useful for ranking variables, raw variable importance scores from a random forest lack a clear interpretation. For example, given a set of scores it is not always clear which scores correspond to a real association as there is no scale by which to judge association strength. In addition, these scores are rarely exactly zero for uninformative variables due to the stochastic nature of tree construction. Motivated by association testing, Altmann et al. (2010) developed a permutation-based approach for computing the statistical significance of MDA scores. The method computes "null importances" by permuting the response vector and uses these to approximate the null distribution. The p-value of the MDA score is then computed using the positively biased estimator $1/(1+a)$, where a is the number of null importances that are smaller than the observed MDA score. This permutation scheme does not scale well to datasets with large numbers of variables.

A second approach by Janitza et al. (2018) is specifically designed for high-dimensional data and uses any negative or zero importances to approximate the null distribution, removing the need for expensive permutation computations. As negative importances values result from unbiased estimation of a random variable with zero expectation (the importance of a non-associated variable) they are mirrored and combined with the observed negative importances to approximate the null importance distribution. However, this procedure requires a large number of negative scores for a reasonable approximation and so the method of Altmann et al. is preferred for datasets with few variables (the smallest dataset included in the original paper by Janitza et al. contains 2,000 variables).

Both these methods were originally developed for MDA scores as the de-biased MDG scores of Nembrini et al. (2018) had not been published, but the authors showed that both methods are appropriate for assessing the statistical significance of de-biased MDG scores.

3 RATE (RelATive cEntrality)

As outlined in Chapter 1 (Section 4), Chapter 4 contains extensions to the variable prioritisation method called RATE (RelATive cEntrality, Crawford et al., 2019). RATE was developed for GP regression modelling of biomedical datasets, particularly genome-wide association studies. Given a trained GP regression model, RATE calculates *post-hoc* variable importance scores which can be used to rank variables while accounting for their interactions. This section outlines the required mathematical background for the extensions described in Chapter 4.

3.1 Calculating RATE scores

Consider a regression dataset $\mathcal{D} = \{(x_i, y_i)_{i=1}^n\}$, where $x_i = (x_i^{(1)}, \ldots, x_i^{(p)}) \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$. In some equations the notation $X = (x_1, \ldots, x_n) \in \mathbb{R}^{n \times p}$, $y = (y_1, \ldots, y_n) \in \mathbb{R}^n$ is also used. Using these data the GP regression model

$$y = f + \varepsilon, \qquad f \sim \mathcal{GP}(m(x), k(x, x')), \quad \varepsilon \sim \mathcal{N}(0, \tau^2 I)\,, \tag{2.61}$$

is fitted, where $f \in \mathbb{R}^n$ is a vector of latent function values and τ^2 is the noise variance. As this is a Bayesian model the fitting procedure computes the posterior distribution $p(f \mid X, y)$, as described in Section 1.1.

Given $p(f \mid X, y)$, RATE values are calculated using a two-step process:

1. Compute the posterior $p(\tilde{\beta} \mid X, y)$ over effect sizes analogues $\tilde{\beta} \in \mathbb{R}^p$ using $p(f \mid X, y)$ and a projection $\tilde{\beta} = \text{Proj}(X, f)$.
2. Calculate KL-divergences using

$$\text{KLD}_j := \text{KL}\left(p(\widetilde{\beta}_{-j}) \,||\, p(\widetilde{\beta}_{-j} \mid \widetilde{\beta}_j = 0)\right) \quad j = 1, \ldots, p\,,$$

   where the subscript $-j$ denotes indexing $\tilde{\beta}$ with $\{1, \ldots, p\} \setminus j$, such that $\tilde{\beta}$ can be partitioned as $(\tilde{\beta}_j, \tilde{\beta}_{-j})$.

The final RATE scores are given by

$$\gamma_j = \frac{\text{KLD}_j}{\sum_k \text{KLD}_k}\,, \tag{2.62}$$

which restricts them to [0,1] (as each $\text{KLD}_j > 0$) for ease of interpretation.

3.2 Effect size analogues

The first step of RATE is to calculate $p(\tilde{\beta} \mid X, y)$ using $p(f \mid X, y)$ and $\tilde{\beta} = \text{Proj}(X, f)$. Effect size analogues summarise the marginal effect of each variable and are motivated by ordinary least squares (OLS) regression, where the regression coefficients (effect sizes) β are projections of the response vector y onto the column space of X

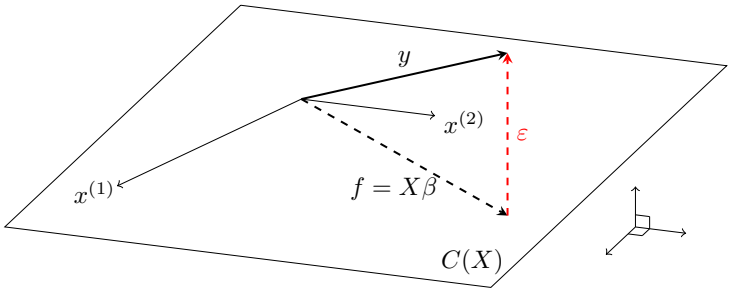


FIGURE 2.6: Regression coefficients β are a projection of $y = f + \varepsilon$ onto $C(X)$, the column space of X (a plane in this 2-variable illustration). The component of the data explained by the model, $f = X\beta$, lies in this plane while the residuals ε (in red) are perpendicular to it.

(see Figure 2.6). Effect size analogues provide an analogous summary of the variables in a non-linear model (in this case a GP) by projecting f onto X. In OLS regression the closed-form solution for β is

$$\beta = \text{Proj}(X, y) = (X^T X)^{-1} X^T y\,, \tag{2.63}$$

which motivates the projection used in the original RATE paper:

$$\tilde{\beta} = \text{Proj}(X, f) = (X^T X)^{-1} X^T f\,, \tag{2.64}$$

which is referred to from hereon in as the Pseudoinverse projection due to the fact that $(X^T X)^{-1} X^T = X^\dagger$ is the Moore-Penrose pseudoinverse of X. As (2.64) is a linear operation and $p(f \mid X, y)$ is multivariate Gaussian for the GP model this ensures $p(\tilde{\beta} \mid X, y)$ is also multivariate Gaussian. In other cases (such as a non-linear projection operator or a non-Gaussian posterior over f) it is possible to transform samples from $p(f \mid X, y)$ to obtain samples from $p(\tilde{\beta} \mid X, y)$, although this is more computationally expensive. Both the original RATE paper and the extensions in this thesis only consider models with multivariate Gaussian $p(f \mid X, y)$ and linear projections.

3.3 Variable importance using relative centrality measures

Having computed $p(\tilde{\beta} \mid X, y)$, the next step of the RATE calculation requires solving

$$\text{KLD}_j := \text{KL}\left(p(\widetilde{\beta}_{-j}) \,||\, p(\widetilde{\beta}_{-j} \mid \widetilde{\beta}_j = 0)\right)\,, \quad j = 1, \ldots, p\,, \tag{2.65}$$

to compute the importance for each variable. When $p(\tilde{\beta} \mid X, y) = \mathcal{N}(\mu, \Omega)$ is multivariate Gaussian this can be solved in closed-form using an appropriate partitioning of the posterior mean μ, covariance Ω and precision $\Lambda = \Omega^{-1}$,

$$\mu = \begin{pmatrix} \mu_j \\ \mu_{-j} \end{pmatrix}, \qquad \Omega = \begin{pmatrix} \omega_j & \omega_{-j}^T \\ \omega_{-j} & \Omega_{-j} \end{pmatrix}, \qquad \Lambda = \begin{pmatrix} \lambda_j & \lambda_{-j}^T \\ \lambda_{-j} & \Lambda_{-j} \end{pmatrix}, \tag{2.66}$$

where $\mu_j, \omega_j, \lambda_j \in \mathbb{R}$, $\mu_{-j}, \omega_{-j}, \lambda_{-j} \in \mathbb{R}^{p-1}$ and $\Omega_{-j}, \Lambda_{-j} \in \mathbb{R}^{(p-1)\times(p-1)}$. The Kullback-Leibler divergence from the multivariate Gaussian density $\mathcal{N}_0(\mu_0, \Sigma_0)$ to

$\mathcal{N}_1(\mu_1, \Sigma_1)$ is

$$\mathrm{KL}(\mathcal{N}_0 \,||\, \mathcal{N}_1) = \frac{1}{2}\left[\mathrm{trace}(\Sigma_1^{-1}\Sigma_0) + (\mu_1 - \mu_0)^T\Sigma_1^{-1}(\mu_1 - \mu_0) - k + \log\frac{|\Sigma_1|}{|\Sigma_0|}\right], \tag{2.67}$$

where k is the number of dimensions in the distribution. For the calculation of KLD_j the two densities of interest are

$$\mathcal{N}_0 := p(\tilde{\beta}_{-j}) = \mathcal{N}(\mu_{-j}, \Omega_{-j}) \tag{2.68}$$

$$\mathcal{N}_1 := p(\widetilde{\beta}_{-j} \mid \widetilde{\beta}_j = 0) = \mathcal{N}(\mu_{-j} - \omega_{-j}\omega_j^{-1}\mu_j,\, \Lambda_{-j}^{-1}), \tag{2.69}$$

where the covariance of $p(\widetilde{\beta}_{-j} \mid \widetilde{\beta}_j = 0)$ is equal to Λ_{-j}^{-1}. Crawford et al. (2019) note that the trace and log-determinant terms in (2.67) do not vary much across different values of j when $\mathcal{N}_0$ and $\mathcal{N}_1$ are given by (2.68)-(2.69), meaning that the order of the variables is determined entirely by the quadratic term $(\mu_1 - \mu_0)^T\Sigma_1^{-1}(\mu_1 - \mu_0)$. Substituting the appropriate quantities from (2.68)-(2.69) into (2.67) and ignoring terms that are constant for different values of j gives

$$\mathrm{KLD}_j \approx \frac{1}{2}(\omega_{-j}\omega_j^{-1}\mu_j)^T\Lambda_{-j}(\omega_{-j}\omega_j^{-1}\mu_j), \quad j = 1, \ldots, p, \tag{2.70}$$

which is used to calculate the KL-divergences in the RATE calculation.

Chapter 3

Differential abundance and two-sampling testing of microbial airway communities using random forest

As researchers seek to characterise the microbial communities of different disease groups the two-sample test is an important statistical procedure. A multivariate, non-parametric test is especially useful for such applications as a microbial community is inherently multivariate and may not meet parametric assumptions. Performing such a test with a random forest classifier is increasingly popular as it consistently exhibits strong predictive performance predicting host traits from microbial community composition. An additional benefit of this approach is that it allows a subsequent variable importance analysis to identify taxa that drive a difference between the two groups (differential abundance). However, many of the properties of the random forest model are currently unexplored even as its use becomes standard in microbiome modelling pipelines. This chapter presents an empirical study of the behaviour of random forest classifiers for the two-sample test and differential abundance analysis in the context of lung disease.

1 The role of the microbiome in respiratory disease

Chronic suppurative lung diseases (CSLD) are a group of respiratory diseases whose symptoms include chronic coughing, excess sputum build-up and recurrent pulmonary infections (McCallum and Binks, 2017). These recurrent infections are the primary cause of patient mortality. The two CSLDs investigated in this study are:

- cystic fibrosis (CF): an inherited genetic disorder in which both copies of the cystic fibrosis transmembrane conductance regulator gene contain mutations; and
- non-cystic fibrosis bronchiectasis (BX): a permanent dilation of the airways due to recurrent (and often severe) infection.

Despite the genetic differences between CF and BX both conditions are characterised by the typical set of CSLD symptoms, which includes a decreased ability to clear the airways of mucus. As airways are constantly exposed to bacteria and fungi in

the environment this places patients at increased risk of infection. For this reason, bacterial and fungal infections have been identified as a key factor in both CF and BX disease progression, with many of of the same organisms being associated with poor clinical outcomes in both groups (Amin et al., 2010; Chotirmall et al., 2010; Zemanick and Hoffman, 2016; Maselli et al., 2017; Máiz et al., 2018). It is unknown if, despite these similarities, microbial communities can successfully differentiate between these two diseases.

A previous study by Cuthbertson, Felton, et al. (2021) compared the fungal communities of CF and BX patients, identifying differences in fungal diversity between the two groups. These samples have since been subjected to 16S rRNA sequencing to quantify their bacterial community composition, enabling an investigation of the role of cross-kingdom interactions in these two diseases. There is growing evidence of important bacteria-fungi interactions that affect health (Deveau et al., 2018; Santus et al., 2021). In the context of CF, a study by Soret et al. (2020) explored the cross-kingdom relationship between the bacterial and fungal communities in the context of CF pulmonary exacerbations (CFPE), reporting the presence of clinically relevant cross-kingdom interactions. Characteristic cross-kingdom interactions have also been highlighted in patients experiencing BX exacerbations (Mac Aogáin et al., 2021).

2 Study aims

This Chapter is based on a pre-print by Ish-Horowicz, Cuthbertson, et al. (2022) on the analysis of the bacterial and fungal communities in the lungs of patients with CF or BX. The original article is a collaborative work exploring interactions between the bacterial and fungal communities in the FAME dataset (Fungal Airway MicrobiomE, Cuthbertson, Felton, et al., 2021), and so its focus is on the biological findings.

The primary biological aims of the original paper are:

1. to establish if either the biological or fungal communities are distinct between:
    - the CF and BX groups;
    - patients with and without fungal infections (within the CF group); and
    - patients currently experiencing symptom exacerbations (within the CF group).
2. to identify differentially abundant taxa between any distinct groups; and
3. to investigate the role of cross-kingdom interactions in any differences detected in Aim 1.

In the original pre-print Aims 1 and 2 are achieved using a random forest classifier-based two-sample test (described in Section 3). A subsequent variable importance analysis achieves Aim 3. When using random forests there are several modelling choices that can affect the resulting conclusions. These include the type of transformation applied to the taxa count tables and the choice of variable importance method. The contribution of this chapter is a series of empirical studies on the sensitivity of the biological conclusions to these choices.

There is a lack of studies on the empirical behaviour of random forests when applied to microbial datasets, despite their increasing popularity in this setting. Existing studies are either limited in scope or do not include microbial datasets (Ranganathan and Borges, 2011; Degenhardt et al., 2019; M. Zhang and W. Shi, 2019; Tolosana-Delgado et al., 2019). The results presented in this chapter address this gap in the literature by exploring the behaviour of several important stages of the random forest analysis pipeline (data transformation, model evaluation and variable importance analysis) in the context of microbial datasets.

These studies find that, for the FAME dataset:

- a two-sample test using the popular LeDell confidence intervals (E. LeDell, Petersen, and Laan, 2015) for cross-validated AUCs lead to the same conclusions as a more computationally expensive permutation test;
- however, such a test using these confidence intervals has an inflated Type I error rate;
- DeLong's test (DeLong et al., 1988) for comparing paired receiver operating characteristic (ROC) curves also has an inflated Type I error rate; and
- the biological conclusions from two-sample tests and differential abundance analyses are largely robust to the choice of data transformation when using random forests.

Overall, the results of these studies illustrate the danger of over-interpreting the results of a single analysis strategy. Robust random forest-based analyses of microbial datasets requires assessing the stability of findings as well as their consistency across different modelling choices.

3 Two-sample testing with binary classifiers

An important problem in biology is to determine whether two groups of samples are drawn from distinct distributions (the two-sample test). Given two sets of samples $X = \{x_{1_i}\}_{i=1}^{n_1}$ and $X_2 = \{x_{2_i}\}_{i=1}^{n_2}$, where each set contains p-dimensional vectors and

$$X_1 \sim P\,, \quad X_2 \sim Q\,, \tag{3.1}$$

the two-sample (hypothesis) test is

$$H_0 : P = Q\,, \quad H_1 : P \neq Q\,, \tag{3.2}$$

where H_0 and H_1 are the null and alternative hypotheses. The nature of microbiome data motivates a non-parametric test as the appropriate parametric distribution of microbiome counts is often difficult to specify. A multivariate test is also desirable given that the variables in X_1 and X_2 represent a complex ecological community of interacting organisms.

Such a non-parametric, multivariate test can be performed by reformulating the two-sample test in terms of supervised learning. This has been a particularly popular

approach in genomics and neuroscience, which share many statistical difficulties with microbial datasets (Rosenblatt et al., 2021). In the re-formulated problem a classifier is trained on the dataset $\mathcal{D} = (X, y)$, where

$$X = (X_1, X_2)^T, \quad y = (y_i)_{i=1}^{n_1+n_2}. \tag{3.3}$$

The design matrix X is the row-wise concatenation of X_1 and X_2 and $y = (y_i)_{i=1}^{n_1+n_2}$, $y_i = \mathbb{1}(i > n_1)$ indicates the group membership of the i^{th} sample.

Given some scoring rule (e.g. accuracy or area under curve, AUC) the two-sample test is performed by establishing whether a classifier has better-than-random performance on unseen data. The intuition is that if $P \neq Q$ then a classifier will be able to identify which distribution unseen sample are drawn from. Furthermore, the larger the difference between P and Q, the easier the classification task. The predictive performance of the model is therefore a summary statistic for the difference between P and Q.

This approach is popular in the biostatistics literature but is usually applied outside of an explicit hypothesis testing framework (Komiyama et al., 2016; Rossi et al., 2019; Xicota et al., 2019). That is to say, the generalisation performance of a trained classifier is evaluated using a scoring metric (most commonly the AUC) and if this value is judged to be larger than 0.5 (equivalent to random performance) then the model is considered to have better-than-random performance, with larger AUCs indicating better models (Mandrekar, 2010). These results are usually interpreted in terms of "good" and "bad" models, where the quality of a model refers to its predictive performance. However, if the predictive performance of a model is considered as a function of the difficulty of separating the two classes then a binary classification problem can equivalently be interpreted in terms of the size of difference between the groups (with a larger difference corresponding to an easier problem).

The decision on whether the AUC is larger than 0.5 is usually made using confidence intervals on the AUC rather than by explicitly computing a p-value. If p-value is required it can be obtained using a permutation test on the group labels y. However, this is a computationally expensive procedure for large datasets as it requires retraining the model with each set of permuted labels.

One of the benefits of the classifier-based approach is that the test inherits the variable importance measures of the classifier. In the microbiome setting this means that it is possible to perform a *post-hoc* differential abundance analysis to identify the microbial drivers of any detected difference between the two groups. Two-sample tests with random forests are therefore an attractive option for microbiome studies, although their behaviour has not been studied in detail despite the rapidly increasing prevalence of such analyses.

4 Relevant work

4.1 Random forests for microbiome

As outlined in Chapter 2 (Section 1.5), random forests are popular models for biological data analysis. Included in this statement is their increasing popularity for the data-driven analysis of microbiome data (Ssekagiri et al., 2017; Roguet et al., 2018; Corrigan et al., 2018; Thompson et al., 2019; Nagpal et al., 2020; Das et al., 2021). This

popularity arises from the fact that they often exhibit the strongest predictive performance on host-trait prediction tasks in microbial studies due to their ability for non-parametric, non-linear modelling (Statnikov et al., 2013; Zhou and Gallins, 2019; Topçuoğlu et al., 2020). These properties are especially useful in microbiome studies, in which the covariates represent a dynamic and interacting ecological system of microbes. Random forests are therefore often preferred over more interpretable linear models, with a systematic review finding that random forest was the most popular machine learning model for differential abundance testing in microbiome studies and the third most popular for microbiome analysis overall (Bardenhorst et al., 2021).

4.2 Classifier-based two-sample testing

Due to the prevalence of "informal" classifier-based two-sample testing (using a predictive model to assess the degree of difference between two groups) in the biomedical literature there have been recent efforts to provide theoretical guarantees for such tests with general classifiers (Gagnon-Bartsch and Shem-Tov, 2016; H. Cai et al., 2020; Rosenblatt et al., 2021; I. Kim et al., 2021). These works investigate the asymptotic behaviour of a two sample-test using an unspecified classifier with accuracy as its test statistic. Another paper by Hediger et al. (2022) considers two-sample tests using random forest classifiers specifically (Hediger et al., 2022). It includes comparisons to an alternative non-parametric two-sample test using kernels (such as those investigated in Chapter 5), reporting that random forest-based tests have higher power when the difference between two populations is driven by their marginal distributions (Hediger et al., 2022).

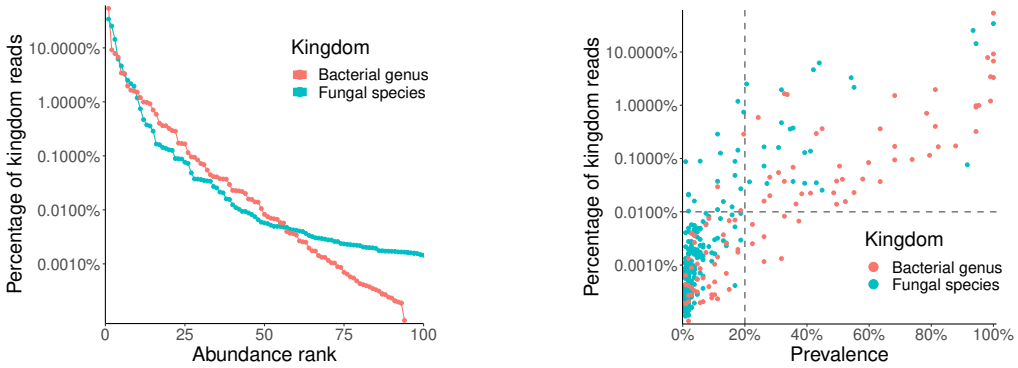
5 Data

5.1 Quantifying community composition

The sputum samples from the 107 individuals included in Cuthbertson, Felton, et al. (2021) were subjected to 16S rRNA (ribosomal ribonucleic acid) gene sequencing to quantify their bacterial community composition, as described in Chapter 1 (Section 1.3). The fungal taxa were previously quantified using ITS2 (internal transcriber spacer 2) sequencing in the original study. A detailed description of the data collection and pre-processing can be found in Ish-Horowicz, Cuthbertson, et al. (2022). This data collection is not part of the contributions of this work as it was performed by collaborators.

For all analyses the 1,189 bacterial OTUs are agglomerated to the genus level, as has been done in similar studies (L. Chen et al., 2020; Qian et al., 2020; Cekikj et al., 2022). This is motivated by recent findings that indicate that 16S rRNA sequencing does not has sufficient resolution to identify beyond genus-level (Jeong et al., 2021). Agglomeration to genus level is also used in a number of previous studies applying random forests to microbial datasets, where agglomeration aids in the interpretation of differential abundance results and improves predictive performance by combining a large number of rare but indistinguishable taxa (Jasner et al., 2021).

The fungal community is quantified sequenced using ITS2, the analogous sequencing modality for the fungal community. This resulted in 2,770 OTUs. There has been



(A) Rank-abundance (Whittaker) plot

(B) Abundance (relative to the total kingdom reads) and prevalence

FIGURE 3.1: Whittaker/rank-abundance plots (A) and the abundance (relative to the total dataset reads) and prevalence (B) for the agglomerated taxa in each kingdom of the FAME dataset. These types of plots are commonly used in ecology to visualise the rarity of organisms. The inclusion thresholds for the random forest modelling are denoted by dashed lines in plot B.

less work on the resolution to which it can accurately identify fungi as it is a less mature modality than 16S rRNA. However, in this dataset the Whittaker plot in Figure 3.1(A) suggests that the species level is an appropriate rank at which to agglomerate the fungal taxa. This is not a rigorous method and further studies are required to establish the resolution of ITS2 sequencing (Nilsson et al., 2019). The fungal and bacterial agglomerated taxa are the two sets of covariates for this study (see Table 3.1).

Figure 3.1(B) shows that the two kingdoms have distinct relative-abundance and prevalence distributions, with the fungal community containing a larger number of very rare (low-prevalence) taxa. It has been reported that the removal of rare taxa can mitigate problems due to technical variability between labs without affecting the predictive power and variable selection properties of random forest (Cao et al., 2021). Taxa that account for fewer than 0.01% of reads for their respective kingdom or are present in fewer than 20% of samples are therefore excluded. The final datasets contain over 97% of the total reads for each kingdom.

TABLE 3.1: Covariate sets in the FAME dataset are the agglomerated OTUs of each kingdom. Only taxa accounting for more than 0.01% of reads or are present in at least 20% of samples are included in the random forest modelling. RF: random forest.

	Sequencing modality	Number of taxa (total)	Number of taxa (inc. in RF modelling)
Bacterial genus	16S rRNA	98	44
Fungal species	ITS2	239	19

5.2 Transforming taxa abundances

The raw covariates in this dataset take the form of $n \times p$ count matrices, for n samples and p taxa. However, it is common practice to transform count data prior to fitting predictive models, but such transformations of 16S rRNA counts have been reported to have a large effect on the results of other statistical analyses. The choice of normalisation has been found to critically affect correlation estimates calculated from 16S rRNA counts (Badri et al., 2018), while other evaluations in the context of differential abundance analysis have reported that the best choice of normalisation strategy varies between datasets and analysis methods (Weiss et al., 2017; H. Lin and S. D. Peddada, 2020). The transformation strategies included in these experiments are described in Table 3.2.

Count data are ubiquitous in biology, particularly in ecology, and log-transformations ($\log(x+1)$ due to the presence of zeros) are widely used when the counts are the response. The transformed quantity is considered to be closer to normally distributed than to the raw counts, which are positive by definition and often positively skewed with few large values. Transformation using a $\log(x+1)$ improves the held-out AUC of classification models for a number of 16S rRNA datasets (Jasner et al., 2021).

Another common transformation in microbiome studies is to use relative abundances, where each sample is normalised using the sum of its reads. The relative abundance is a popular method as it corrects for variation in number of reads in each sample, which can be substantial due to technical factors in the measurement process. The term relative abundance has also been used to refer to a division by all the reads in the dataset (H. Lin and S. D. Peddada, 2020), but here *relative abundance* refers to normalising each sample by the sum of its reads, while *sum normalisation* refers to normalising counts by the total number of reads in the dataset.

The idea that microbial count data contain only relative information is the key feature of compositional data analysis. The principles of compositional data and its impact on statistical analysis are discussed in more detail in Chapter 5. The introduction of compositional data principles to biological sequence data analysis has been relatively recent (Gloor et al., 2017; Quinn, Erb, et al., 2018), meaning that studies on the effects of applying random forests to compositional data come mainly from the geosciences literature, where compositional data are ubiquitous (Ranganathan and Borges, 2011; Tolosana-Delgado et al., 2019; M. Zhang and W. Shi, 2019). The effect of the most commonly-applied compositional transform, centred log-ratio (CLR, Aitchison, 1982), is explored here.

Random forests are theoretically invariant under monotonic transformations (Breiman et al., 1984). However, the type of transformation often impacts the predictive performance in practice due to numerical reasons (Kimmel and Oliver, 2006; Galili and Meilijson, 2016). Furthermore, the CLR and Relative abundance transformations are not monotonic. It is therefore important to establish how the results of random-forest based analyses depend on the choice of transformation as it is common practice to present results using only a single transformation.

5.3 Sample groups

In addition to the primary group definitions (CF and BX), two additional sub-groupings were investigated from amongst the CF samples (Table 3.3). The first of these groups denotes the fungal disease status of the CF patients as either fungal bronchitis (FB,

TABLE 3.2: The different count transformation used in this chapter. $\tilde{x}_i^{(j)}$ are transformed counts from raw counts $x_i^{(j)}$ using $\tilde{x}_i^{(j)} = g\left(x_i^{(j)}\right)$

Transformation	Description	$g\left(x_i^{(j)}\right)$	Monotonic?
log1p	log-transform with pseudo-count	$\tilde{x}_i^{(j)} = \log(x_i^{(j)} + 1)$	Monotonically increasing
Relative abundance	Normalise using number reads in the sample	$\tilde{x}_i^{(j)} = \frac{x_i^{(j)}}{\sum_j x_i^{(j)}}$	No
Sum normalisation	Normalise using number reads in the dataset	$\tilde{x}_i^{(j)} = \frac{x_i^{(j)}}{\sum_{i,j} x_i^{(j)}}$	Monotonically increasing
Centred log-ratio (CLR)	Transforms compositional data to Euclidean space	$\tilde{x}_i^{(j)} = \log \frac{x_i^{(j)}}{\prod_j x_i^{(j)}}$	No

clinical diagnosis of fungal lung disease), or having no active fungal disease (NAFD). The second set of groups denotes whether a patient was experiencing a CF pulmonary exacerbation (CFPE, defined as a clear deterioration in CF symptoms) at the time the of sampling.

TABLE 3.3: The binary classification tasks for the random forest.

Group name	Description	Classes	Sample size
Disease	If a patient has CF or BX	CF, BX	83 CF, 24 BX (107 total)
Fungal disease	If patient has fungal bronchitis (FB) or no active fungal disease (NAFD)	FB, NAFD	20 FB, 39 NAFD (59 total)
CFPE	If CF patient is experiencing CFPE when sampled collected	Yes, No	36 Yes, 47 No (83 total)

6 Two-sample testing using random forests

6.1 Nested cross-validation estimates of generalisation performance

Classifier-based two-sample tests require an estimate of the generalisation performance of a classifier. The datasets in this chapter have insufficient samples to reserve some for estimating the generalisation error as all samples are required to train the model. A nested cross-validation scheme is therefore used to estimate the out-of-sample error. The outer loop performs 5-fold cross-validation, with the inner loop also performing 5-fold cross-validation to select hyperparameters. The hyperparameters search considers ten random different combinations of `mtry` (the number

of variables sampled at each split point), the splitting rule and minimum samples per leaf. Each model contains 1,000 trees.

The out-of-sample error is estimated using the mean AUC on the held-out (validation) samples in each iteration of K-fold nested cross-validation,

$$\widehat{\text{AUC}}_{\text{heldout}} = \frac{1}{k}\sum_{k=1}^{K}\widehat{\text{AUC}_k}\,, \tag{3.4}$$

where $\widehat{\text{AUC}_k}$ is the AUC estimate on the held-out samples in outer fold k. Significance is assessed using a confidence interval on the mean validation AUC calculated using the method of E. LeDell, Petersen, and Laan (2015). If a confidence interval with width $1-\alpha$ excludes 0.5 then H_0 is rejected at a significance threshold α. The one-sided test is used in practice as worse-than-random performance (an AUC less than 0.5) should be treated the same as random performance - both indicate no detectable difference between the groups.

6.2 Results of the two-sample test

As there are three sets of group definitions (see Table 3.3) and three sets of possible covariates (bacterial genus, fungal species and both bacterial genus and fungal species) there are nine random forest models in total. Their mean held-out AUCs are shown in Figure 3.2 with two-sided 95% confidence intervals calculated using the method of E. LeDell, Petersen, and Laan (2015). The intervals are corrected for multiple comparisons using the Bonferroni method (Bonferroni, 1936), so their width is $1-\alpha/9$. The correction is not applied across the different transformations as the aim is to compare the results that would be obtained if a single transformation were used.

Using these 95% two-sided intervals corresponds to a test with a 10% significance threshold. The first conclusion from these results is that random forest models find statistically significant differences between the CF and BX groups using all three sets of covariates (Figure 3.2, left panel). This conclusion would be reached using any of the four data transformations. For fungal disease (centre panel), H_0 is only rejected when the fungal species abundances are used as covariates (using any transformation other than sum normalisation), with no difference detected based on bacterial genus abundance (under any transformation). When both kingdoms are included in the model to predict fungal disease status the test result depends on the choice of transformation, which calls into question the robustness of the result. Finally, no set of covariates under any transformation result in H_0 being rejected for CFPE (right panel).

These results show that both the bacterial and fungal communities are able to discriminate between the CF and BX groups in this dataset. However, they do not establish which of the two communities has more discriminative power. In addition, this confidence interval approach does not suggest that including both kingdoms improves the ability of the model to discriminate between CF and BX. There is also insufficient evidence to conclude that the bacterial community composition can discriminate FB from NAFD. The opposite finding would be evidence of cross-kingdom dependencies (possibly involving the fungal pathogens driving an FB diagnosis), but these results suggest a degree of independence between the two communities. Finally, none of the models can discriminate between patients with and without

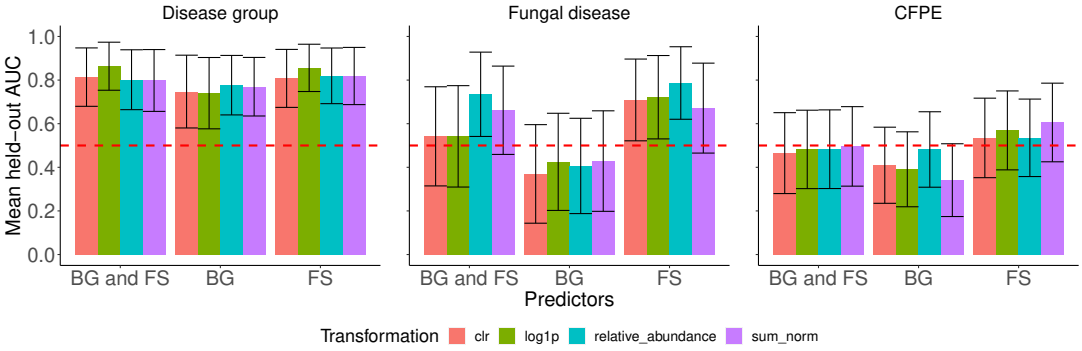


FIGURE 3.2: Mean held-out AUCs for random forest models. Two-sided 95% CIs calculated using the method of E. LeDell, Petersen, and Laan (2015) with a Bonferroni correction. Red dashed line is a mean held-out AUC of 0.5. BG: bacterial genus, FS: fungal species.

CFPE. This is most likely because cross-sectional data is inappropriate for studying the drivers of CFPE, which is by definition a temporary state defined relative to the baseline symptoms of an individual. Longitudinal data are therefore required in order to quantify both the intra- and inter-patient variation and effectively model CFPE.

6.3 Coverage of LeDell's confidence intervals

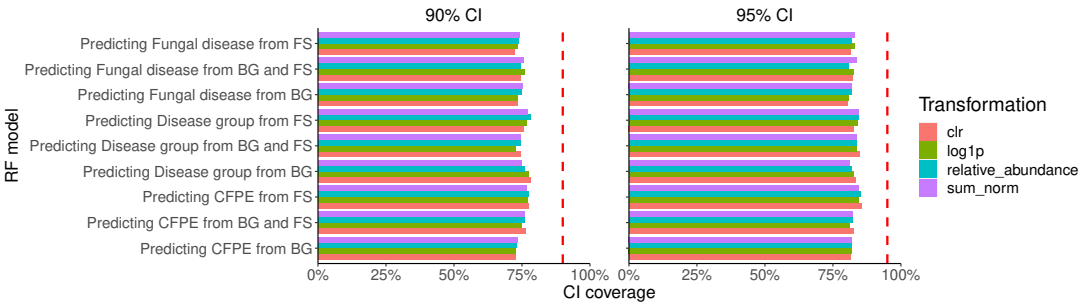
LeDell's confidence intervals are routinely applied to a range of dataset types and sizes in biomedical studies (B. Shi et al., 2018; Toivonen et al., 2019; Roimi et al., 2020; Fu et al., 2020). Given a set of AUCs evaluated on the held-out samples at each fold of K-fold cross validation, $\{\widehat{\text{AUC}}_k\}_{k=1}^K$, LeDell's method calculates an asymptomatically normal confidence interval on their mean using influence curves (E. LeDell, Petersen, and Laan, 2015). Their popularity arises from their computational efficiency (the time and memory required to calculate them is negligible relative to model training) and the easy accessibility of the accompanying package, `cvAUC`, (E. LeDell, Petersen, Laan, and M. E. LeDell, 2022). To the best of my knowledge there are no empirical evaluations of these popular confidence intervals on real datasets in the literature.

A confidence interval of width $1-\alpha$ on the mean validation AUC should exclude 0.5 with at a rate of α under the null hypothesis (no difference between the groups). The empirical coverage under the null hypothesis can be established using a label randomisation test with the following setup. A random forest model is trained using nested cross-validation with dataset $\mathcal{D}_1 = (X, \tilde{y})$, where X is a table of transformed OTU counts and $\tilde{y}$ is a permuted version of the observed y. The LeDell confidence intervals are recorded and the procedure is repeated for P distinct permutations of y.

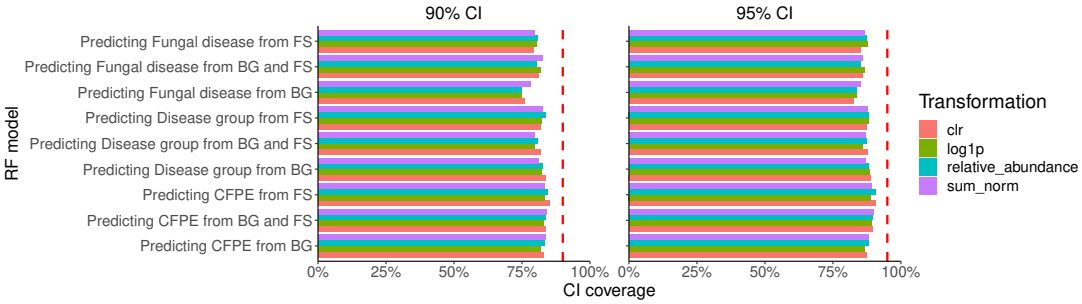
The empirical coverage of an interval with width $1-\alpha$ is

$$\mathcal{C}_\alpha = \frac{1}{P}\sum_{l=1}^{P} \mathbb{1}(0.5 \in [a_l, b_l])\,, \tag{3.5}$$

where $[a_l, b_l]$ is the interval calculated at permutation l. For a one-sided interval $a_l = -\infty$. The empirical coverage should be approximately equal to α if the interval is



(A) Coverage of the two-sided interval



(B) Coverage of the one-sided interval

FIGURE 3.3: Empirical coverage of LeDell's confidence intervals on cross-validated AUC estimates under the null hypothesis (500 replicates). The confidence intervals are too narrow as the coverage is lower than the theoretical coverage (denoted by the red dotted line).

well-calibrated, however, Figure 3.3 shows that the empirical coverage is lower than α for both the two-sided (plot A) and one-sided intervals (plot B). The confidence intervals are therefore too narrow as the coverage is lower than the theoretical value (the red dashed lines in Figure 3.3). This is observed for all four transformations.

These narrow confidence intervals imply an inflated Type I error in the corresponding two-sample test. While the mis-calibration in the empirical coverage is a concern, it is preferable that the coverage is too low than too large as this would suggest a high likelihood of false positive results. In addition, the one-sided intervals (which are more useful in practice) have empirical coverage closer to the desired theoretical coverage. However, without a more detailed simulation study under the alternative hypothesis (a difference between the two groups) it is not possible to make concrete statements about the Type II error behaviour of LeDell confidence intervals for this type of data.

6.4 Two-sample testing using LeDell confidence intervals agree with permutation tests

Another way of evaluating LeDell intervals in this setting is to compare them to the results of an equivalent permutation test. The result of such a test (with 500 permutations) are displayed in Table 3.4. They are consistent with the confidence interval-based test results in Figure 3.2 but are much more computationally expensive (the entire nested cross-validation procedure must be repeated for each permutation). This shows the utility of the confidence intervals proposed by E. LeDell, Petersen,

and Laan (2015) despite the fact that they are slightly narrow in the case of the null hypothesis.

TABLE 3.4: P-values from a permutation test on the mean validation AUC being greater than 0.5. P-values within each column are adjusted for multiple comparisons using the false discovery rate. *: P<0.1, **: P<0.05, ***: P<0.01.

Random forest model	CLR	log1p	Relative abundance	Sum norm.
Predicting Disease group from BG	0.00***	0.01***	0.00***	0.00***
Predicting Disease group from FS	0.00***	0.01***	0.00***	0.00***
Predicting Disease group from BG and FS	0.00***	0.01***	0.00***	0.00***
Predicting Fungal disease from BG	0.91	0.78	0.56	0.76
Predicting Fungal disease from FS	0.00***	0.01***	0.00***	0.00***
Predicting Fungal disease from BG and FS	0.29	0.13	0.00***	0.10**
Predicting CFPE from BG	0.91	0.78	0.46	0.91
Predicting CFPE from FS	0.91	0.78	0.37	0.63
Predicting CFPE from BG and FS	0.91	0.85	0.56	0.91

6.5 DeLong's test

The question of whether the bacterial or fungal communities are more distinct between the CF and BX groups can also be answered by comparing the ROC curves of the relevant models in a hypothesis testing framework. However, this is not a two-sample test but rather DeLong's test (DeLong et al., 1988). This is a non-parametric test between two paired ROC curves, where "paired" refers to the fact that the two models under comparison share the same labels. The test is derived using the insight that the AUC can be interpreted as the probability that the score of a randomly-selected item from the positive class has a higher score than a randomly-selected item from the negative class. This implies that the AUC is a Mann-Whitney U/Wilcoxon rank sum test on the predicted probabilities of the positive and negative groups. DeLong et al. were the first to note this equivalence and use this result to derive an asymptotically normal distribution for the AUC.

DeLong's test is commonly used to compare the discriminative power of disjoint sets of clinical variables as a proxy for comparing the strength of their associations with the response (Y. Huang, 2016; Garg et al., 2021). The two-sided version of the test is

$$H_0 : \text{AUC}_1 = \text{AUC}_2\,, \quad H_1 : \text{AUC}_1 \neq \text{AUC}_2\,, \tag{3.6}$$

where AUC_1 and AUC_2 are the AUCs of the two models. However, the one-sided test

$$H_0 : \text{AUC}_1 < \text{AUC}_2\,, \quad H_1 : \text{AUC}_1 > \text{AUC}_2\,, \tag{3.7}$$

is often more useful as it can be used to compare the discriminative power of two sets of covariates. One drawback of DeLong's test is that it has low power when

the two models share covariates (Demler et al., 2012), which precludes comparisons involving the models trained on both fungal and bacterial abundances. This comparison is made using a permutation test in the following subsection.

P-values from both directions of two-sided DeLong's tests are shown in Table 3.5. They indicate that there is not sufficient evidence to conclude that there a significant difference in discriminative power between the two kingdoms for CF and BX. They also show that the fungal abundances are significantly more discriminative of fungal disease status than the bacterial community (which is expected from both common sense and the confidence intervals on the mean validation AUCs in Figure 3.2). These results have the additional robustness of being independent of the choice of transformation.

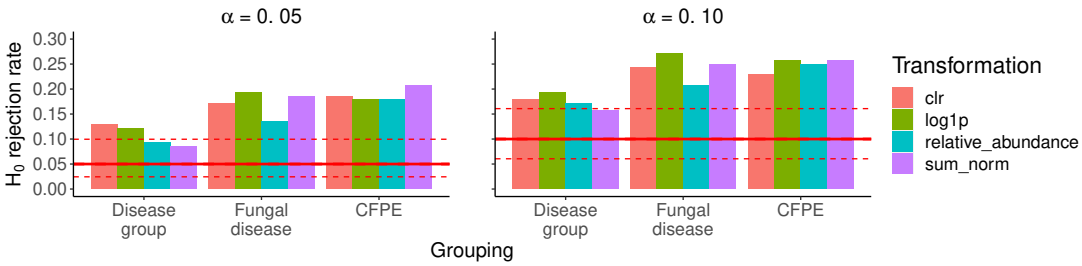
TABLE 3.5: P-values from one-sided DeLong's test comparing the discriminative power of the fungal and bacterial communities using their respective AUCs (AUC_{FS} and AUC_{BG}). CFPE is excluded here as neither the fungal nor bacterial communities are predictive of CFPE in this dataset. P-values are corrected for multiple comparisons using false discovery rate. $\Delta AUC = AUC_{FS} - AUC_{BG}$. *: P<0.1, **: P<0.05, ***: P<0.01. FS: fungal species, BG: bacterial genus

Labels	Transformation	ΔAUC	$H_1 : AUC_{FS} < AUC_{BG}$	$H_1 : AUC_{FS} > AUC_{BG}$
Disease group	CLR	0.06	0.96	0.6
	log1p	0.11	1.00	0.21
	Rel. abund.	0.03	0.85	0.86
	Sum norm.	0.05	1.00	0.41
Fungal disease	CLR	0.36	1.00	0.00***
	log1p	0.30	1.00	0.01***
	Rel. abund.	0.42	1.00	0.00***
	Sum norm.	0.25	0.99	0.01***

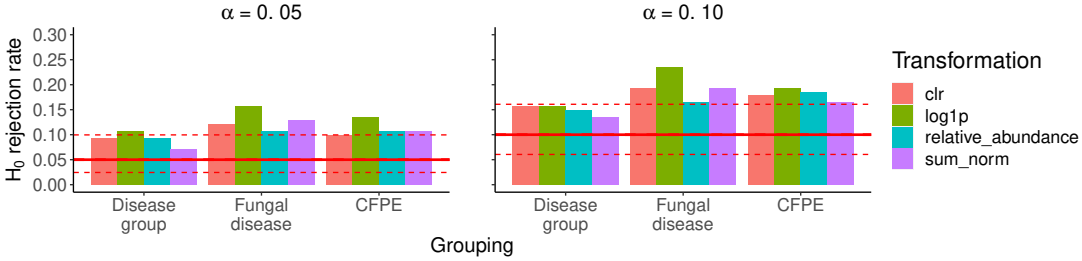
Type I error rate of DeLong's test

The empirical behaviour of DeLong's test under the null hypothesis can be tested by training random forest models on permuted labels using the same procedure as was used to investigate the coverage of LeDell's confidence intervals in the previous section. For each group definition (Disease group, fungal disease and CFPE) a random forest model is trained using nested cross-validation with the dataset $\mathcal{D}_1 = (X_{FS}, \tilde{y})$, where X_{FS} are the fungal species abundances and $\tilde{y}$ is a permuted version of the observed y. A second model is then trained on $\mathcal{D}_2 = (X_{BG}, \tilde{y})$ containing the bacterial genus abundances X_{BG}. The mean validation AUCs are then compared using two- and one-sided DeLong's tests. As both models are trained on randomised labels (and hence neither contain predictive signal) the test should be rejected at a rate of α.

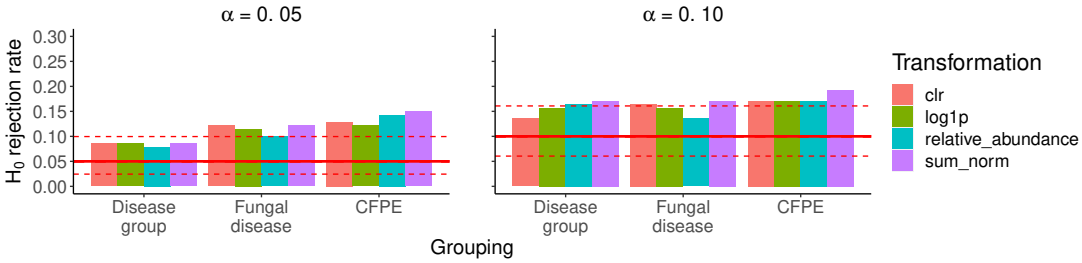
Figure 3.4 shows that the rejection rate of over 500 replicates is larger than the nominal significance threshold α for $\alpha \in \{0.10, 0.05\}$, indicating an inflated likelihood of Type I error. This inflated Type I error rate is present for all the data transformations but varies across the different group definitions. Tests involving random forest models trained to predict Disease group have a Type I error rate that is closest to α as this is the group with the largest sample size ($n = 107$), while the tests with models



(A) Two-sided test: $H_0 : \text{AUC}_1 = \text{AUC}_2$, $H_1 : \text{AUC}_1 \neq \text{AUC}_2$



(B) One sided test: $H_0 : \text{AUC}_1 < \text{AUC}_2$, $H_1 : \text{AUC}_1 > \text{AUC}_2$



(C) One sided test: $H_0 : \text{AUC}_2 < \text{AUC}_1$, $H_1 : \text{AUC}_2 > \text{AUC}_1$

FIGURE 3.4: Rejection rate of DeLong's test comparing the AUCs two random forest models trained on permuted labels. The solid red line denotes the nominal significance level α and the dashed lines show its 95% binomial proportion confidence interval.

predicting fungal disease status ($n = 87$) or CFPE ($n = 59$) have higher Type I error rates. However, the difference in Type I error rate across the three group definitions cannot be explained solely by differing sample sizes as H_0 is rejected at the same rate for Fungal disease ($n = 59$) and CFPE ($n = 87$).

While this is a concern, this is mitigated by the fact that the more useful one-sided test (Figure 3.4(B-C)) rejects H_0 at a rate that is closer to α than the two-sided test (Figure 3.4(A)).

6.6 Effect of cross-kingdom interactions on discriminative power

When DeLong's test is not appropriate - it loses power when the two models under comparison share covariates - a permutation test is a preferable option, where the null hypothesis is that adding the abundances of the second kingdom does not increase discriminative power. Starting with a random forest model trained on the abundance of a single kingdom (either bacterial genera or fungal species) such a test can be used to assess the whether the community composition of both kingdoms

is more discriminative than a single kingdom alone. Such an approach provides information on the role of cross-kingdom interactions in the two groups.

The P-values from a 500-permutation one-sided test are shown in Table 3.6. These show the change in mean held-out AUC when the second kingdom is added to the random forest model as covariates where the null hypothesis is that adding the second kingdom does not improve discriminative power for CF/BX. The P-values show there is insufficient evidence to reject H_0 at a significance level of 10%.

TABLE 3.6: P-values from one-sided permutation test (500 permutations) comparing the discriminative power after adding the other kingdom as covariates to a random forest model. CFPE is excluded here as neither the fungal nor bacterial communities were predictive of CFPE. P-values are corrected for multiple comparisons using false discovery rate. ΔAUC is the change in mean held-out AUC when the second kingdom abundances are added. FS: fungal species, BG: bacterial genus

Labels	Model 1 abundances	Model 2 abundances	Transformation	ΔAUC	P-value
Disease group	BG and FS	BG	CLR	0.07	1.0
			log1p	0.07	0.9
			Rel. abund.	0.07	1.0
			Sum. norm	-0.01	0.6
		FS	CLR	0.03	1.0
			log1p	-0.01	0.9
			Rel. abund.	0.08	1.0
			Sum. norm	-0.01	0.6
Fungal disease	BG and FS	BG	CLR	0.26	1.0
			log1p	0.18	0.9
			Rel. abund.	0.32	1.0
			Sum. norm	0.21	1.0
		FS	CLR	-0.18	0.4
			log1p	-0.12	0.8
			Rel. abund.	-0.03	1.0
			Sum. norm	-0.11	0.6

7 Random forest variable importance for differential abundance

One of the benefits of random forests is their ability to compute variable importance scores, which in microbiome studies is typically framed in terms of differential abundance if the model is a binary classifier. A recent review recognised that different differential abundance tools produce varying results on a single dataset and that a consensus analysis is required to ensure robust biological interpretations (Nearing et al., 2022). Such consensus analyses benefit from including tools with varying assumptions and so including random forests to complement bespoke tools (which generally focus on univariate or parametric approaches) is a useful and increasingly popular approach (Bardenhorst et al., 2021).

Using random forest models for differential abundance analysis requires additional modelling choices by the practitioner in addition to those made when training the model itself. The most important of these is the choice of variable importance methods. This study compares four possible variable importance scoring methods for random forests:

- mean decrease accuracy (MDA or permutation importance);
- mean decrease Gini (MDG or impurity importance);
- de-biased MDG (Nembrini et al., 2018); and
- Shapley values.

These are described in detail in Chapter 2 (Section 2.6) but the most relevant points are re-stated here. MDA scores are calculated by permuting the out-of-bag samples and recording the corresponding decrease in accuracy. MDG scores are the mean decrease in impurity across all the nodes in the forest where a given variable is used as a splitting point. The naive MDG has been found to be biased in certain situations (it artificially inflates the importance of continuous variables on a large scale or categorial variables with many categories), which motivates the bias-corrected scores of Nembrini et al. (2018). These de-biased scores are the difference between the observed MDG score and the MDG score of a permuted version of that variable (thereby removing the contribution to the importance that is solely due to a variable's structure). Finally, Shapley values are a model-agnostic approach derived from game theoretic principles that attributes each variable an importance score related to its contribution to a predicted value.

The following models from the previous section are carried forward for variable importance analysis:

1. predicting CF/BX from bacterial genus;
2. predicting CF/BX from fungal species; and
3. predicting FB/NAFD from fungal species.

These three models are selected as they detected statistically significant differences between their respective groups. For clarity they will be referred to by these numbers for the remainder of this section. The random forest model predicting CF/BX using both bacterial and fungal abundances is not included despite the fact that its confidence interval excluded 0.5 in Figure 3.2. This is because its variable importance results largely mirror those of the two individual models (Models 1 and 2) that have been included.

7.1 Variable rankings under different variable importance methods and transformations

Clearly the choice of variable importance method affects the results of a random forest differential abundance analysis. However, variable importance scores for random forests are usually used to rank methods due to the lack of a clear mathematical

meaning of the scores, meaning that consistency between the rankings is more of interest than consistency in the actual scores themselves (Behnamian et al., 2017).

As Shapley values produce signed scores (they have an effect direction) their absolute value is used in these analyses for fair comparison with other (undirected) scores. While these other scores can take negative values, a negative MDG or MDA score indicates extremely low importance rather than importance in the direction of the negative class.

For Model 1 (Figure 3.5(A)), the MDA and MDG scores show consistently high level of agreement under all four transformations (Spearman's $\rho > 0.67$). For the log1p, and relative abundance transformations the MDA, MDG and Corrected MDG scores agree more closely to one another ($\rho > 0.53$) than with Shapley values ($\rho < 0.46$), but when the CLR transform is used there is more consistent agreement across the four methods ($0.46 \leq \rho \leq 0.64$).

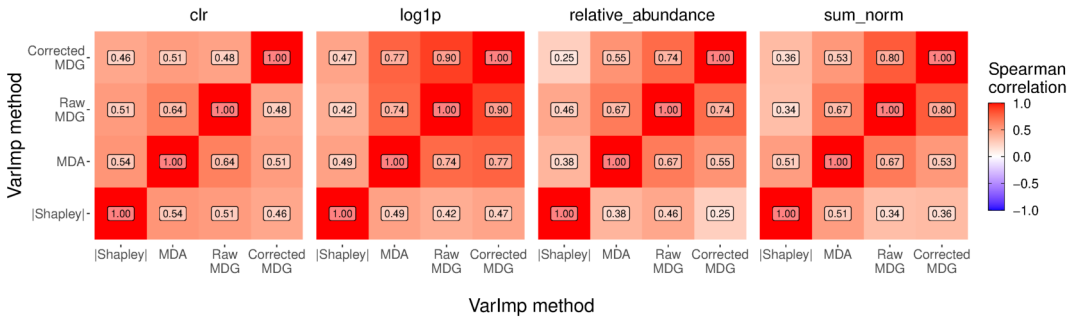
For Model 2 (Figure 3.5(B)) there is more overall agreement between the rankings than is observed in Model 1, but there are still differences under the four transformations. The log1p and sum normalisation transformations give strong agreement between all four methods ($\rho > 0.72$) but there is less overall agreement under the CLR or relative abundance transformation ($\rho > 0.51$).

Model 3 (Figure 3.5(C)), on the other hand, shows the strongest overall agreement between the models when the relative abundance is used ($\rho > 0.54$) and less agreement for sum normalisation ($\rho > 0.25$). The degree of agreement between the variable rankings therefore varies across the three models and four transformations, which suggests that is difficult to know *a priori* which variable importance method is the most appropriate for a given task as it is highly data-dependent.

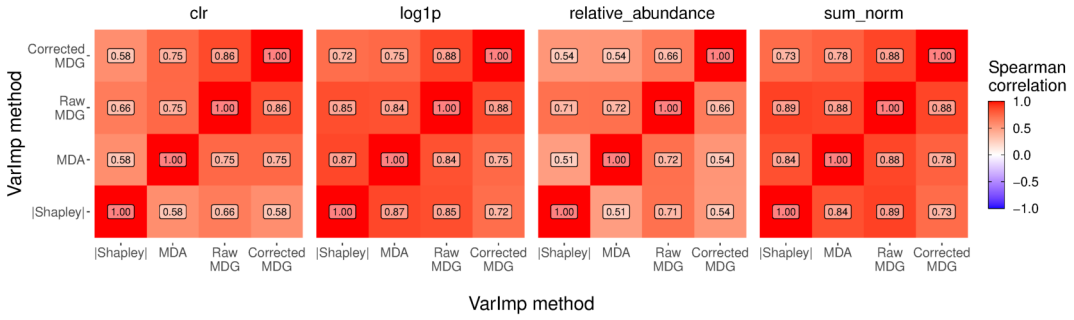
While the overall agreement of the variable rankings between methods is of interest, it is the top-ranked variables that are the focus of a random forest variable importance analysis. These top-ranked variables are investigated further and possibly considered for follow-up studies. Figure 3.6 compares the consistency of the top four ranked taxa (i) using the different variable importance methods using and (ii) under the different transformations.

For Model 1 (Figure 3.6(A)), the bacterial genus *Pseudomonas* is in the top-2 ranked taxa according to Corrected MDG, MDG and MDA scores when using CLR, log1p or the relative abundance transformations. However, it does not appear in the top four taxa when using sum normalisation or Shapley values. A second genus, *Neisseria* is also consistently observed in the top-four taxa for Corrected MDG, MDG and MDA scores (under all four transformations) but is not in the top four for Shapley values. These two taxa are therefore more likely to represent true associations than others that are highly ranked by only a few of the variable importance methods under certain transformations (e.g. *Lactobacillus*, *Tannerella* and *Treponema*). The associations for *Pseudomonas* and *Neisseria* are not a causal statements but these results suggest that their association is not due modelling artefacts and so are more likely to be present in the data.

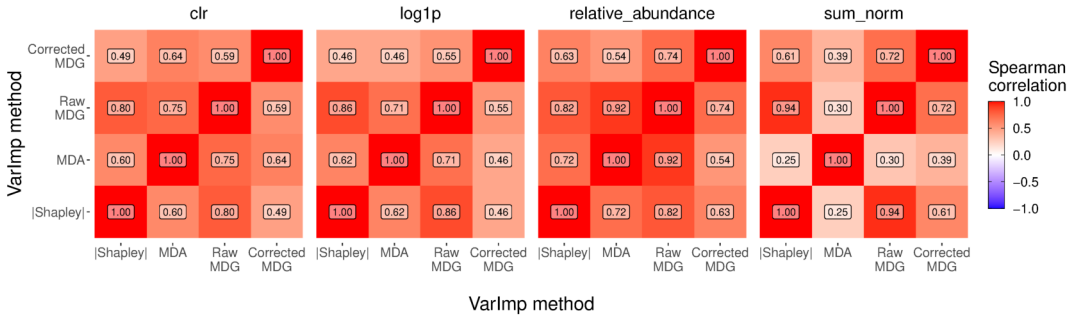
For Model 2 (Figure 3.6(B)), *Penicillium psychrosexualis*, *Malassezia restricta* and *Penicillium thomii* are the top-ranked taxa irrespective of the type of variable importance method or transformation used. *Candida parapsilosis* also appears amongst the top-ranked taxa for three of the four variable importance methods irrespective of the choice of transformation.



(A) Predicting Disease group from bacterial genus



(B) Predicting Disease group from fungal species



(C) Predicting Fungal disease status from fungal species

FIGURE 3.5: Spearman correlation between the variable methods using each data transformation.

For Model 3 (Figure 3.6(C)) there is a clear consensus that *Exophiala dermatitidis*, *Aspergillus fumigatus*, *Scedosporium boydii* and *Candida albicans* are the most important taxa driving fungal disease in CF patients. The first three of these are those identified by Cuthbertson, Felton, et al. (2021) in their analysis (which did not use random forest) while *Candida albicans* is another well-known opportunistic pathogen in immunocompromised patients (Pendleton et al., 2017).

There is varying stability in the rankings across these three models, which suggests that the main factors in random forest variable importance stability are data-dependent. This is consistent with previous findings, which suggest that it is difficult to know *a priori* which variable importance method will produce the best results (Huazhen Wang et al., 2016). Even between these three models (which are trained on different portions of the same dataset) there is substantial variations in stability across the models, with models trained on fungal species abundance exhibiting the most stability across variable importance methods and transformations. This is most likely due to the fact that is has the fewest variables ($p = 19$, compared to $p = 44$ bacterial genera).

Statistical significance

Assessing statistical significance is a key part of a differential abundance analysis. Recall that statistical significance for random forest variable importance scores is computed using null importances (importance scores for models with permuted labels). For a given variable, the corresponding p-value is given by

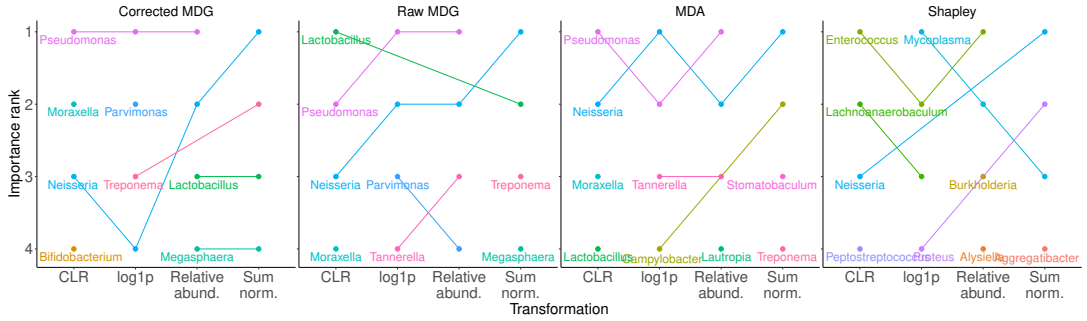
$$P = \frac{1 + b}{1 + M}, \tag{3.8}$$

where M is the number of permutations and b is the number of permutations in which the permuted score is larger than the observed score. There are two methods for calculating the null importance scores (see Chapter 2, Section 2.6 for details):

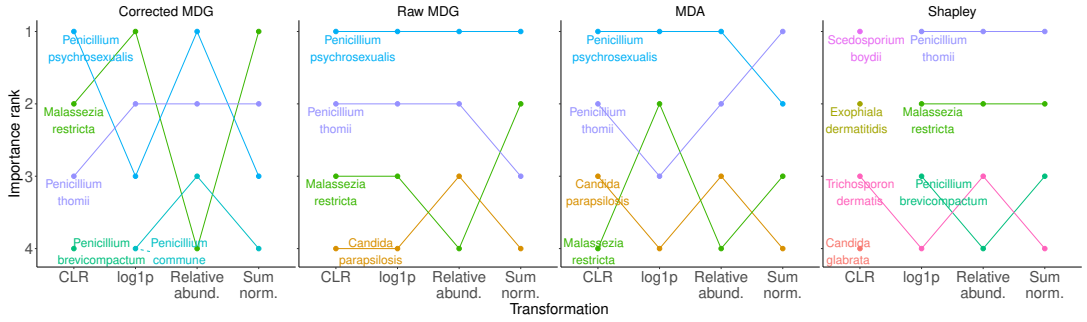
- the approach of Altmann et al. (2010), which calculates the null importances using these permutations; and
- the approach of Janitza et al. (2018), which approximates the null importances using the observed negative importances.

The method of Janitza et al. (2018) relies on a large number of negative or zero importances and so is specifically-designed for large-p datasets ($p \gtrapprox 10^3$), in which case the computational cost of Altmann et al. (2010) becomes prohibitively expensive. The dataset sizes in this study are therefore suitable for the method of Altmann et al. (2010), which can be calculated for MDA or Corrected (de-biased) MDG scores. This enables a comparison between the p-values according to the two variable importance methods as well as an investigation of how the data transformation affects statistical significance.

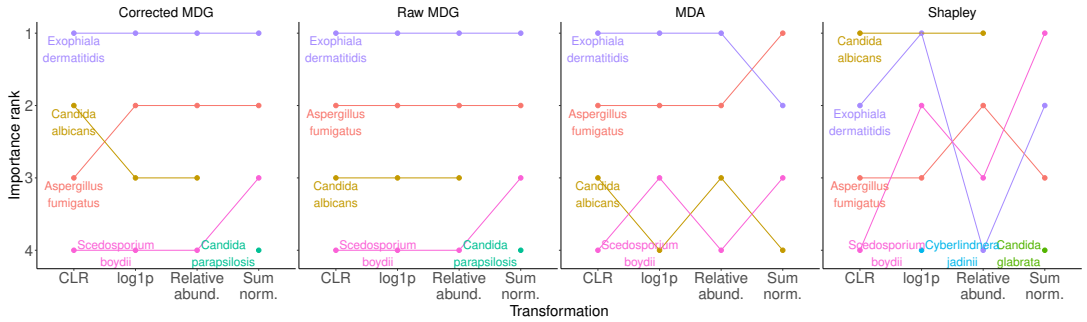
Figure 3.7 shows the agreement between the statistical significance of MDA and Corrected MDG scores according to the method of Altmann et al. (2010) using 1,000 permutations. For all three models there is good agreement between the two sets of p-values. For Model 1 (Figure 3.7(A)) there are no statistically significant hits for



(A) Predicting Disease group from bacterial genus



(B) Predicting Disease group from fungal species



(C) Predicting fungal disease status from fungal species

FIGURE 3.6: The top four ranked variables for random forest models using different importance measures and transformations.

either variable importance method at 10% significance. This is true for all four transformations. For Model 2 (Figure 3.7(B)) there is also general agreement between the p-values but the statistically significant hits depend on both the variable importance method and the choice of transformation. For example, *Malassezia restricta* is significantly associated with Disease group (P<0.05) when using either variable importance method and the log1p transformation. However, it only significant at $\alpha = 0.05$ when using MDA importance with the Relative abundance transformation, and not significant for either variable importance method when using the CLR transform.

This comparison illustrates one of the well-known pitfalls of p-values, which is their reliance on arbitrary significances thresholds (Halsey, 2019).

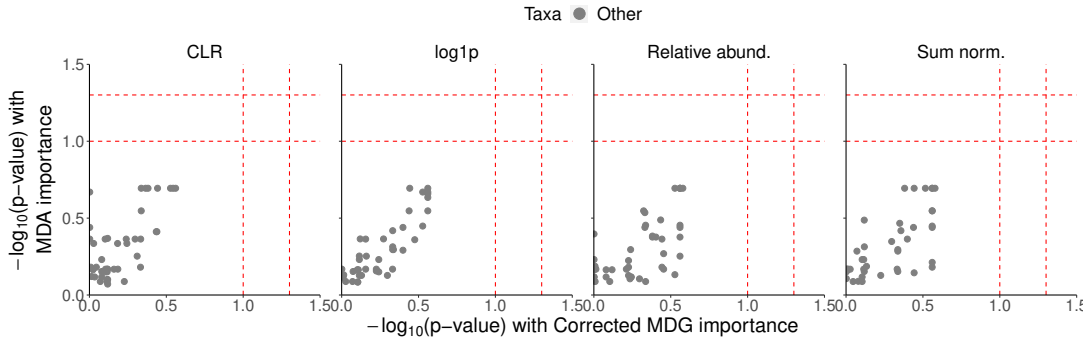
7.2 Stability of variable importance scores

The stability of importance scores under data perturbations is an important property in differential abundance analysis - if the scores (or their rankings for a random forest-based analysis) are unstable under small changes to the dataset then this calls into question the reliability of the results and robustness of the conclusions. Such instability suggests that the rankings are driven by a small subset of samples, which makes drawing global conclusions inappropriate. Variable ranking stability has been identified as an important requirement for random forest variable importance methods (Calle and Urrea, 2011; Nicodemus, 2011; Huazhen Wang et al., 2016). Previous studies have investigated this stability in the context of data perturbations (removing 10% of the samples) and varying degrees of variable collinearity.

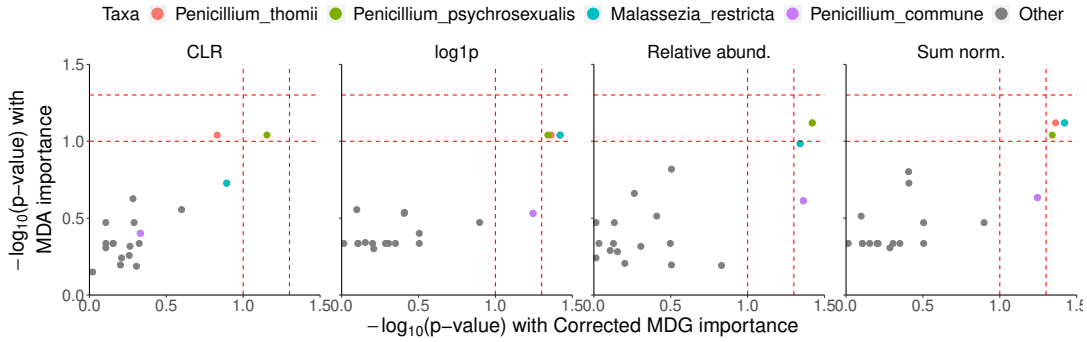
This section repeats the procedure of Calle and Urrea (2011) and Nicodemus (2011) in which a random forest model is trained on a perturbed dataset of size $0.9n$ formed by sampling examples without replacement from the full dataset. The variable ranking in the perturbed dataset is then compared to the ranking in the original dataset for each combination of variable importance method and data transformation (Figures 3.8-3.10). If the original variable ranking were replicated in every perturbed dataset then these plots would show a red diagonal line. The less stable a ranking method is, the more points are observed in the upper-left and lower-right quadrants (this corresponds to variables that are highly-ranked in the full dataset having low ranks in the perturbed dataset and vice versa).

For Model 1 (Figure 3.8) the Shapley rankings show the lowest stability (this is also observed in the other two models). The other variable importance methods show more stability, but the degree of stability depends on the transformation, with transformations that involve sample-wise normalisation (CLR and Relative abundance) having less stable rankings. The most stable rankings are from MDA or MDG when the Sum normalisation is used, where the rankings of the top 3 variables is largely preserved in the perturbed datasets.

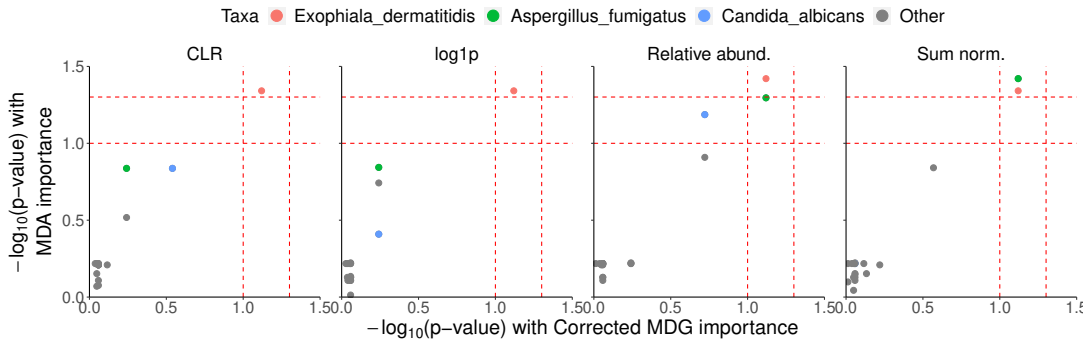
For Model 2 (Figure 3.9) there is more ranking stability than Model 1 (not including Shapley values, which again produce highly unstable rankings). For example, the top-ranked variable according to MDG is almost always in the top 4 using any of the three non-Shapley variable importance methods (for any transformation). This indicates that the ranking for Model 2 are less likely to be driven by a small number of samples than in Model 1. For this Model the ranking stability is more dependent on the choice of variable importance method than data transformation and each variable importance method has a "block" of top-ranked variables whose high ranks



(A) Predicting Disease group from bacterial genus (no statistically significant hits)



(B) Predicting Disease group from fungal species



(C) Predicting Fungal disease status from fungal species

FIGURE 3.7: False discovery rate-adjusted p-values from Altmann's method using 1,000 permutations. Red dotted lines denote $p = 0.10$ and $p = 0.05$.

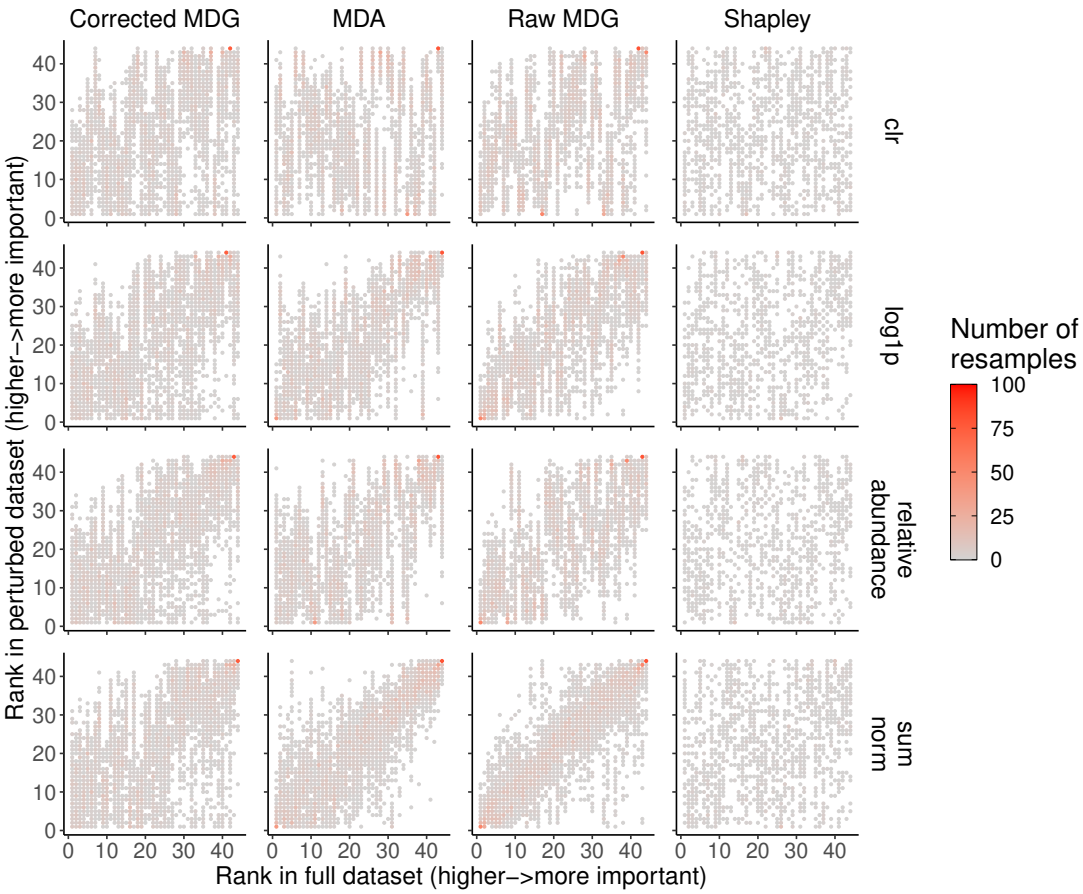


FIGURE 3.8: Stability of each variable importance method under dataset perturbations (removal of 10% of samples). Model: predicting Disease group from bacterial genus.

(the most relevant part of the ranking) are largely preserved across the perturbed datasets. Overall, MDG scores produce the most stable ranking, followed by MDA.

The rankings for Model 3 (Figure 3.10) are also stable. However, for Corrected MDG the stability is only observed for the top-ranked variables, with a large amount on instability amongst the remaining (presumably unassociated) variables. Similarly to Model 2, MDG scores produce the most stable ranking, followed by MDA.

8 Discussion

This chapter presented an analysis of the bacterial and fungal communities of CF and BX patients using a random forest-based two-sample test and differential abundance analysis. As well as studying the differences between the CF and BX groups, two additional binary classification tasks were also investigated within the CF group: (i) predicting the presence of clinically diagnosed fungal disease or (ii) the presence of symptom exacerbations (CFPE). Differences between the fungal and bacterial communities of the CF and BX groups were identified, while only the fungal community was discriminative of fungal disease status within the CF group. Neither the bacterial nor fungal communities were discriminative of CFPE.

The findings that will be of most biological interest are that there was no evidence of systematic difference in the bacterial community of patients with and without fungal

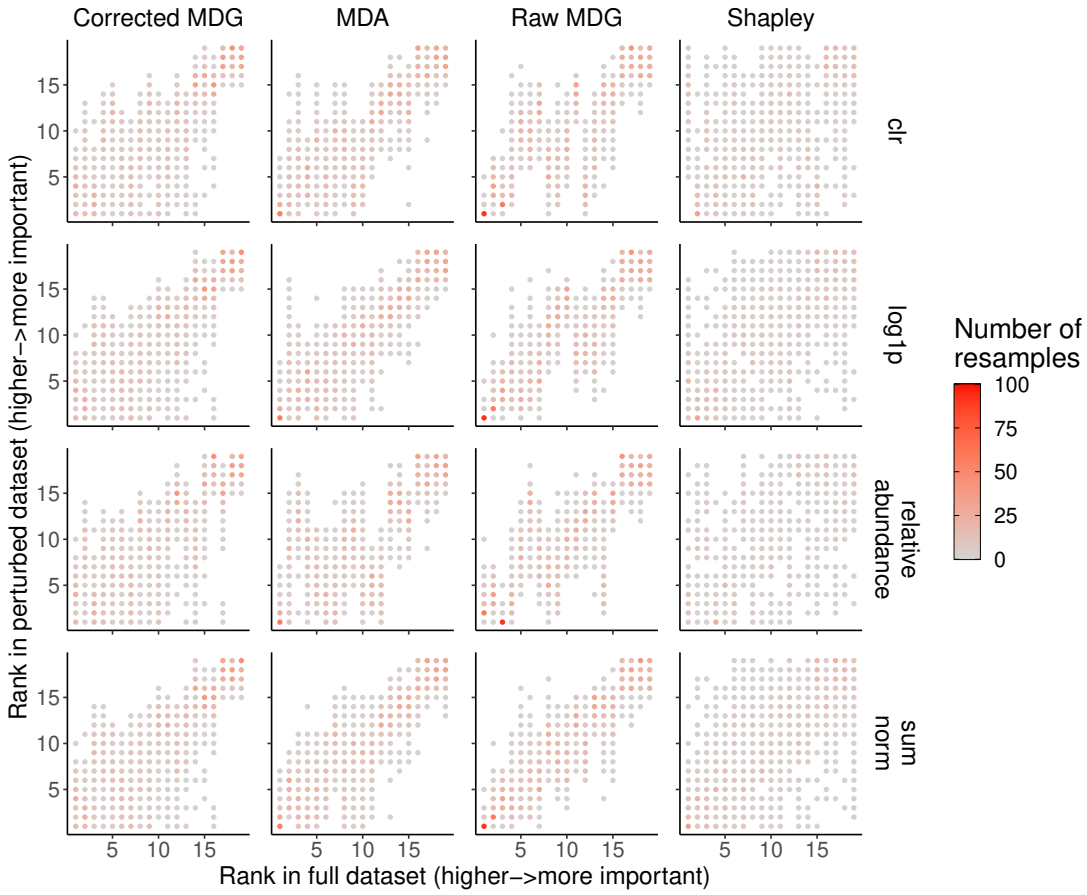


FIGURE 3.9: Stability of each variable importance method under dataset perturbations (removal of 10% of samples). Model: Predicting Disease group from fungal species.

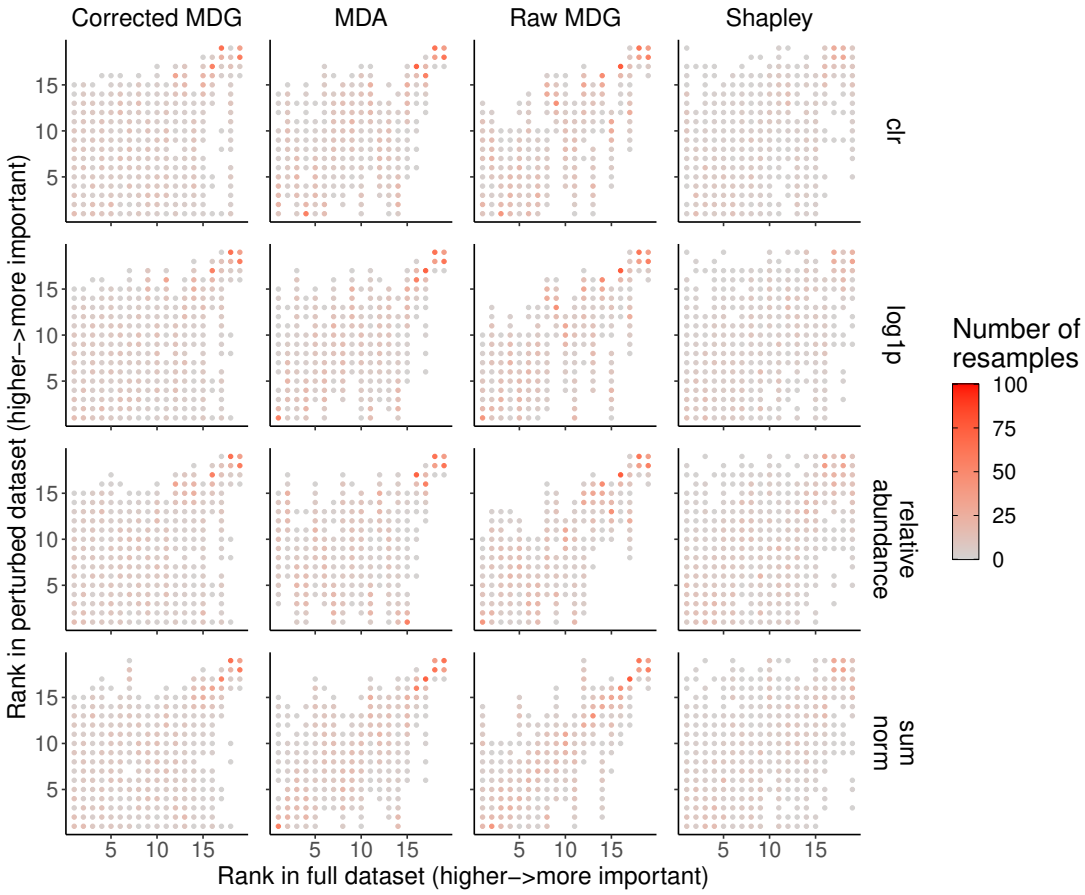


FIGURE 3.10: Stability of each variable importance method under dataset perturbations (removal of 10% of samples). Model: Predicting Fungal disease group from fungal species.

lung infection. This is somewhat surprising as cross-kingdom interactions have been reported as playing an important role in bronchiectasis exacerbations (Mac Aogáin et al., 2021) and asthma (C. Huang et al., 2020). These studies utilise network analysis to examine the cross-kingdom interactions, suggesting that this should be the next analysis step applied to these data. Other future analysis to complement these random forest results should also include more sophisticated approaches to data integration, as more sophisticated methods than the column-wise concatenation used here may be able to establish the role of cross-kingdom interactions.

In addition to the biological findings empirical studies also investigated the behaviour of LeDell's confidence intervals on the mean cross-validated AUC (E. LeDell, Petersen, and Laan, 2015), DeLong's test for paired ROC curves (DeLong et al., 1988), the effect of data transformations and choice of variable importance method. The main statistical results are that hypothesis tests using DeLong's method and LeDell's confidence intervals both have an inflated Type I error rate when used for these three binary classification tasks. These two methods are widely applied in biomedical studies in applications that are often far from the original setting in which methods were developed. For example, the confidence intervals of LeDell et al. are validated in simulations using a Lasso model with datasets of various sizes. The smallest n/p ratio in their simulations is 2.5, in which case the 95% confidence interval has an empirical coverage of 87.8% (E. LeDell, Petersen, and Laan, 2015). This n/p ratio is large by the standard of biomedical datasets (many of which have $n \ll p$) and so it is important for researchers to perform such sanity checks to understand how applicable a method is to their dataset.

That is not to say that these methods are not extremely useful (or that they should not be used), but rather that a simple permutation test is able to establish whether the empirical coverage of a confidence interval (or Type I error of a test) is unacceptably wide for a given dataset and model. In this setting the effect is not so large that it brings into question the validity of the conclusions and a permutation test leads to the same conclusions as are obtained as LeDell's confidence intervals (with much greater computational cost). While running such checks may be prohibitively expensive for some datasets, in the microbiome setting datasets are still sufficiently small that running permutations is very feasible on a high-performance computing cluster.

Overall, these results show that random-based two-sample tests and subsequent variable importance analyses are largely robust to the choice of data transformation and variable importance method. However, Shapley values should be avoided as they are very unstable under small perturbations to the dataset (removal of 10% of samples). The remaining variable importance methods exhibited stable rankings, especially amongst the top-ranked taxa.

The main conclusion that should be drawn from the differential abundance results is that the stability robustness analyses described here is a very informative practice when performing random forest variable importance analysis. This has been suggested previously but this advice is often not followed (Calle and Urrea, 2011; Nicodemus, 2011). In the absence of asymptotic guarantees on the behaviour of random forest variable importance scores (as are available for linear models) this type of empirical study offers the best available option to guard against presenting associations that are artefacts of modelling choices. While many studies seek to establish the "best" random forest variable importance approach using simulation studies, the complex and wide-ranging nature of biological datasets means that it is unlikely

that a given simulation setup can successfully emulate all possible settings. The variable importance methods tested here showed significant variation in behaviour across the three classification tasks in this chapter, despite that fact that all three are derived from the same dataset. For example, the agreement between taxa rankings differs between tasks, while the transformation that results in the most stable rankings is also different across the three tasks. This suggests that recommendations about the "best" variable importance method are inappropriate as the factors that affect the performance of a variable importance method are currently poorly understood. Increasing our understanding of these factors is an area where simulation studies are most useful, but applications of these models should include additional analysis steps to avoid over-interpreting any results.

A simulation study to compare the power of the different variable importance methods is the logical extension of these results. However, such a study will need to be designed carefully to realistically capture the characteristics of microbial datasets. A similar stimulation study for the two-sample test is required to assess the power of the classifier two-sample test in this setting. Such a study could also consider different predictive models as well as alternative metrics for assessing the random forest model. The AUC is only one of many metrics that is used to quantify predictive performance in the machine learning literature. Accuracy, precision-recall curve and Matthews correlation coefficient are the most popular alternatives in biomedical applications of machine learning (Hicks et al., 2022). The choice of metric is likely to have a significant effect on the behaviour of the two-sample test. For example, the precision-recall curve is better suited than ROC curves to problems with extreme class imbalances and there is existing work on calculating confidence intervals (Boyd et al., 2013).

Chapter 4

Grouped variable prioritisation for Bayesian neural networks

As black box models in general (and neural networks in particular) have become ubiquitous in data-rich fields there has been an increasing research focus on developing *post-hoc* variable importance methods as a route to interpreting their predictions. Such methods aim to identify the source of the superior predictive performance exhibited by neural networks on many complex problems, of which there are many in biology. This chapter describes an extension to RelATive cEntrality (RATE, Crawford et al., 2019), a variable prioritisation method for Bayesian non-parametric models, to the Bayesian neural network setting. A second extension considers grouped variables, which are another common feature of biological datasets.

1 Chapter aims and contributions

This Chapter presents several related extensions to RelATive cEntrality (RATE, Crawford et al., 2019), a *post-hoc* variable prioritisation method for Bayesian, non-parametric supervised learning models described in Chapter 2 (Section 3). Some of the methodological extensions relating to Bayesian neural networks are also described in the pre-print by Ish-Horowicz, Udwin, et al. (2019). These extensions are:

- extending the RATE methodology to a last layer Bayesian neural network architecture;
- investigating the utility of two alternative projection operators for RATE;
- extending the original RATE criterion to grouped variables (GroupRATE);
- demonstrating the ability of GroupRATE to prioritise causal groups on two simulated sequencing datasets; and
- demonstrating how GroupRATE can be applied to a Bayesian neural network classifier trained on a medical imaging dataset.

Variable importance analysis is an essential feature of most biostatistical projects, thereby motivating the original RATE paper by Crawford et al. (2019). The contributions of this chapter increase the utility of RATE by addressing other requirements of such biological research projects. For example, the extension to neural networks allows RATE to be used with structured datasets such as images and text (which

neural networks are particularly well-suited to). In addition, grouped variable importance has not yet been studied for neural networks, despite several analogous works for other supervised models (Yuan and Y. Lin, 2006; Simon et al., 2013; Gregorutti et al., 2015; Wehenkel et al., 2018).

The structure of this chapter is as follows. Section 2 re-states the salient points from the description of the RATE methodology in Chapter 2 (Section 3) and provides useful intuition on how it prioritises variables using a toy example.

The methodological contributions begin in Section 3, which describes the new projections used to calculate effect size analogues. Section 4 describes GroupRATE and how it is calculated for last layer Bayesian neural network architectures.

The results begin in Section 5, which shows how GroupRATE can be used for *post-hoc* interpretation of a Bayesian neural network in a biomedical setting and compares its performance to several other group-level importance methods. This is done using simulated covariates and a simulated response. Section 6 describes a similar set of simulations but using real genotype data as covariates with a simulated response. Section 7 demonstrates how GroupRATE can be used to inspect a computer vision model for common biases.

2 Computing variable importances using RATE

2.1 Variable prioritisation vs variable selection

RATE performs variable prioritisation - it computes a ranking of variables that reflects their importance in the model. One limitation of RATE is that it does not provide a clear threshold below which a variable can be considered as non-causal, meaning there is no clear rule by which to decide whether or not a variable is associated with the response. This is in contrast to variable selection methods such as linear models with L1 regularisation (Lasso and ElasticNet), which explicitly exclude variables from the model by setting their coefficients to zero, resulting in a sparse set of effect sizes.

In the applications for which RATE has been developed (e.g. genetic association testing) the aim of statistical analysis is often to generate candidate variables for follow-up studies (Hormozdiari et al., 2015). Since there are limited resources to perform such studies only the most highly-ranked variables are likely to be considered for follow-up. Provided these highly-ranked variables are true associations then the fact that non-causal variables (those not associated with the response) are not explicitly excluded is not a severe limitation.

2.2 Recap of the RATE calculation

RATE is designed for use in a supervised learning setting where a dataset $\mathcal{D} = (X, y)$ consisting of an $n \times p$ design matrix X and n-dimensional vector of labels y has been used to train a Bayesian non-parametric predictive model. This training procedure calculates $p(f \mid X, y)$, the posterior distribution over $f = (f(x_1), \ldots, f(x_n))$, the predicted function values at each of the observed data. Each data point, x_i, $i = 1, \ldots, n$, is the p-dimensional vector $x_i = (x_i^{(1)}, \ldots, x_i^{(p)})$ corresponding to the i^{th} row of X.

Recall from Chapter 2 (Section 3) that RATE values are calculated for a trained model in two steps:

1. using a multivariate Gaussian $p(f \mid X, y)$ and a linear projection $\text{Proj}(X, f)$, calculate $p(\tilde{\beta} \mid X, y)$ where $\tilde{\beta} \in \mathbb{R}^p$ are per-variable effect size analogues (ESAs); then
2. calculate Kullback-Leibler (KL) divergences using

$$\text{KLD}_j = \text{KL}\left(p(\tilde{\beta}_{-j}) \,||\, p(\tilde{\beta}_{-j} \mid \tilde{\beta}_j = 0)\right), \quad j = 1, \ldots, p,$$

where $\tilde{\beta}_{-j} = (\tilde{\beta})_k$, $k \in \{1, \ldots, p\} \setminus j$.

RATE scores are the normalised KL-divergence values

$$\gamma_j = \frac{\text{KLD}_j}{\sum_{k=1}^{p} \text{KLD}_k}, \quad j = 1, \ldots, p, \tag{4.1}$$

which satisfy $\gamma_j \in [0, 1]$ as each KLD_j is positive. As $p(f \mid X, y)$ is exactly multivariate Gaussian and $\text{Proj}(X, f)$ defines a linear transformation, $p(\tilde{\beta}, X, y)$ is also multivariate Gaussian. The result is that each KLD_j can be computed in closed-form.

Following Crawford et al. (2019), the full KL-divergence in step 2 is approximated by the quadratic term as the remaining terms are approximately constant across variables and so have no effect on the resulting ranking. This term can be written as

$$\text{KLD}_j \approx \frac{1}{2}(\omega_{-j}\omega_j^{-1}\mu_j)^T \Lambda_{-j}(\omega_{-j}\omega_j^{-1}\mu_j), \tag{4.2}$$

where ω_j, ω_{-j}, Λ_{-j} and μ_j are taken from a partitioning of the ESA posterior $p(\tilde{\beta} \mid X, y) = \mathcal{N}(\mu, \Omega)$,

$$\mu = \begin{pmatrix} \mu_j \\ \mu_{-j} \end{pmatrix}, \qquad \Omega = \begin{pmatrix} \omega_j & \omega_{-j}^T \\ \omega_{-j} & \Omega_{-j} \end{pmatrix}, \qquad \Lambda = \begin{pmatrix} \lambda_j & \lambda_{-j}^T \\ \lambda_{-j} & \Lambda_{-j} \end{pmatrix},$$

for ESA posterior precision $\Lambda = \Omega^{-1}$. For a p-dimensional vector v, v_{-j} is a $(p-1)$-dimensional vector with elements v_k, $k \in \{1, \ldots, p\} \setminus j$. Similarly, for a $p \times p$ matrix M, M_{-j} is the $(p-1) \times (p-1)$ matrix with elements M_{kl}, $k, l \in \{1, \ldots, p\} \setminus j$. Therefore the importance of variable j is calculated using:

- $\mu_j \in \mathbb{R}$, the ESA posterior mean of variable j (its marginal effect);
- $\omega_j = \Omega_{jj} \in \mathbb{R}$, the ESA posterior variance of variable j;
- $\omega_{-j} \in \mathbb{R}^{(p-1)\times 1}$, the ESA posterior covariance between variable j and the other $p-1$ variables; and
- $\Lambda_{-j} \in \mathbb{R}^{(p-1)\times(p-1)}$, the ESA posterior precision of the other $j-1$ variables.

The partitioning of μ and Ω in the three-variable case is illustrated in Figure 4.1. The next section decomposes the calculation in (4.2) to show that RATE calculates

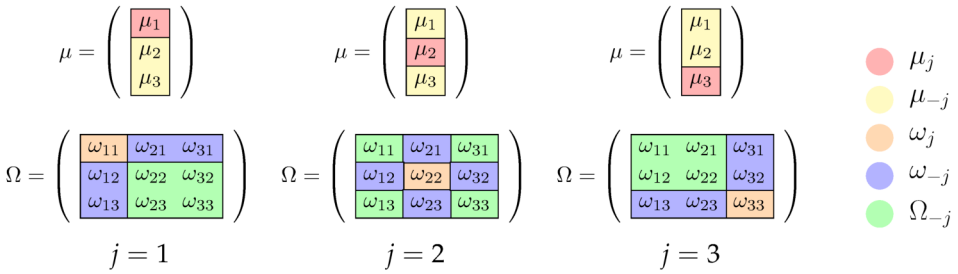


FIGURE 4.1: A visualisation of partitioning the ESA posterior parameters $p(\tilde{\beta} \mid X, y)$ for a three-variable example. The precision matrix $\Lambda = \Omega^{-1}$ is partitioned in the same manner as Ω. Note that only μ_j, ω_j, ω_{-j} and Λ_{-j} (not shown) are used to calculate RATE scores for variable j.

the importance of a variable by weighting its marginal effect with the strength of its dependencies with other variables, thereby incorporating the strength of its interactions.

2.3 Interpretation of RATE scores

Before proceeding to the extensions it is useful to gain some intuition on how the RATE calculation assigns importance. As μ_j and ω_j are scalars, (4.2) can equivalently be expressed as

$$(\omega_{-j}\omega_j^{-1}\mu_j)^T \Lambda_{-j}(\omega_{-j}\omega_j^{-1}\mu_j) = \mu_j^2 \frac{\omega_{-j}^T \Lambda_{-j}\omega_{-j}}{\omega_j^2} \tag{4.3}$$

$$= \mu_j^2\, \eta_j\,, \tag{4.4}$$

where $\eta_j = \frac{\omega_{-j}^T \Lambda_{-j}\omega_{-j}}{\omega_j^2}$ is referred to here as the *covariance-precision* term. RATE therefore assigns each variable an importance score that is proportional to the magnitude of its marginal effect multiplied by η_j, which depends only on the covariance/precision structure of the ESA posterior. If the quadratic form $\omega_{-j}^T \Lambda_{-j}\omega_{-j}$ is expanded then η_j becomes

$$\eta_j = \sum_{k,l \in \mathcal{J}} \frac{\omega_{jk}}{\omega_j} \Lambda_{kl} \frac{\omega_{jl}}{\omega_j}\,, \tag{4.5}$$

where $\mathcal{J}$ is the set $\{1, \ldots, p\} \setminus j$. For variable j, the terms in this double-summation are zero if

- the effects of variables k and l are conditionally linearly independent of one another, given the effects of all other variables ($\Lambda_{kl} = 0$), or
- either one of variables k and l are linearly independent of variable j (ω_{jk} or ω_{jl} are zero).

If $\Lambda_{kl} \neq 0$ then the corresponding summation term in (4.5) is non-zero, and is larger when the effects of variables k and l have large ESA posterior covariance with variable j. The value of η_j is therefore larger for variables whose effects show greater dependency on the effects of other variables.

2.4 Three-variable toy example

This section demonstrates how RATE balances the relative sizes of the marginal effects and the *centrality* of each variable in the ESA covariance graph to produce a ranking using a simple example. Here, centrality refers to the centrality of a variable in the graph defined by Σ. Variables whose effects have large, positive covariance with other variables have higher centrality while variables whose effects are independent of or negatively correlated with other effects have low centrality.

Figure 4.2(A) shows three ESA posterior covariance structures:

(A) $\text{cov}(\tilde{\beta}_1, \tilde{\beta}_2) = 0.7$, $\text{cov}(\tilde{\beta}_1, \tilde{\beta}_3) = 0.1$ and $\text{cov}(\tilde{\beta}_2, \tilde{\beta}_3) = 0.2$;

(B) $\text{cov}(\tilde{\beta}_1, \tilde{\beta}_2) = 0.7$, $\text{cov}(\tilde{\beta}_1, \tilde{\beta}_3) = 0.6$ and $\text{cov}(\tilde{\beta}_2, \tilde{\beta}_3) = 0.1$; and

(C) $\text{cov}(\tilde{\beta}_1, \tilde{\beta}_2) = 0.2$, $\text{cov}(\tilde{\beta}_1, \tilde{\beta}_3) = 0.6$ and $\text{cov}(\tilde{\beta}_2, \tilde{\beta}_3) = -0.4$;

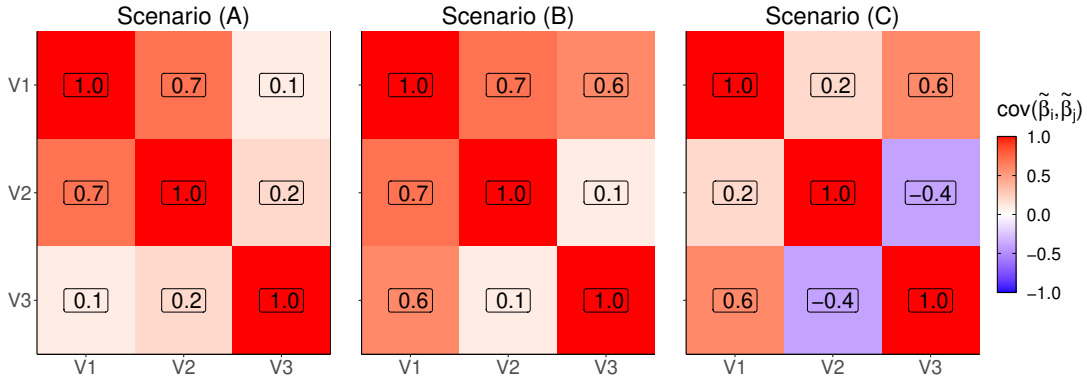
All three covariance structures have ones on the diagonal in these illustrative examples. The RATE values and the terms that comprise them are shown in Figure 4.2(B) for different values of μ_1, where $\mu = (\mu_1, 1, 0.1)$. The first row of plots shows the RATE values, the second shows the square of the marginal effect, the third shows η_j, the covariance-precision term (which is independent of μ_1) and the fourth shows the product $\mu_j^2 \eta_j$.

The third row of Figure 4.2(B) shows that each scenario produces a different ordering of the variables based on their η_j values (reflecting different degrees of centrality in the ESA covariance). These η_j values are balanced by the marginal effects of the variables, where $\mu_2 = 1$ represents a large marginal effect and $\mu_3 = 0.1$ represents a small marginal effect.

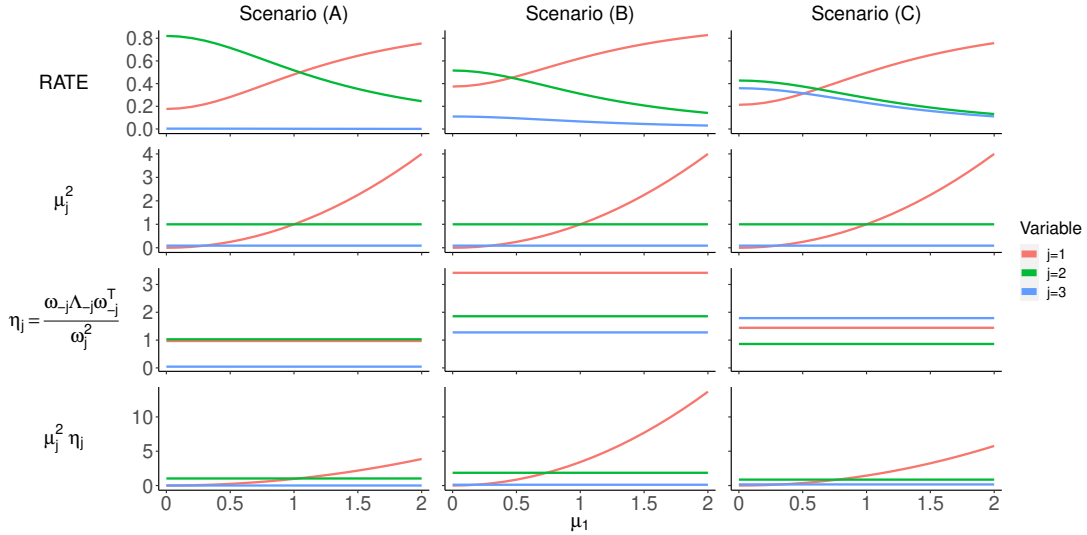
In Scenario A the effects of variables 1 and 2 are the most central as $\omega_{12} \gg \omega_{13}, \omega_{23}$, which implies that $\eta_1 \approx \eta_2$ and $\eta_1, \eta_2 \gg \eta_3$. As variable 3 also has the smallest marginal effect the order of the RATE values is determined by the relative sizes of μ_1 and μ_2.

In Scenario B $\omega_{12}, \omega_{13} \gg \omega_{23}$ and so $\eta_1 > \eta_2 > \eta_3$. Variable 3 therefore always has the smallest RATE value as its marginal effect is also the smallest. Similarly to Scenario A, the relative sizes of the RATE values for variables 1 and 2 is determined by the relative sizes of μ_1 and μ_2. However, as $\eta_1 > \eta_2$ the marginal effect of variable 1 does not need to be as large as μ_2 for its RATE value to be larger.

In Scenario C $\omega_{13} > \omega_{12} > \omega_{23}$, which means that $\eta_3 > \eta_1 > \eta_2$. The RATE values of variables 2 and 3 are approximately balanced as $\mu_2 > \mu_3$ by a similar factor as $\eta_3 > \eta_2$. This means that variable 1 has the smallest RATE value for small values of μ_1, but as μ_1 increases variable 1 becomes the highest-ranked variable.



(A) ESA covariance structures.



(B) The behaviour of each term in the RATE calculation values under the three covariance structures in (A). The value of μ_1 is varied, where $\mu = (\mu_1, 1, 0.1)$.

FIGURE 4.2: Visualisation of the different terms in the RATE calculation (plot B) under three ESA covariance structures (plot A).

3 Alternative projections for effect sizes analogues

While the previous section considered the behaviour of RATE values given $p(\tilde{\beta} \mid X, y)$, in practice this must be computed from the fitted posterior $p(f \mid X, y)$ using a projection. The projection provides a summary of each variable's marginal effect and so its choice can have a significant effect on the resulting RATE values. The original RATE paper by Crawford et al. (2019) only uses a single projection,

$$\tilde{\beta}_{\text{pinv}} = \text{Proj}(X, f) = (X^T X)^{-1} X^T f = X^{\dagger} f\,, \tag{4.6}$$

where $X^{\dagger}$ is the pseudoinverse of X. This is motivated by the fact that the Maximum Likelihood estimate of the ordinary least squares coefficients is $X^{\dagger} y$. This *Pseudoinverse* projection is therefore a linear summary of the dependence of the model prediction f on each variable in the dataset.

One of the aims of this chapter is to investigate the behaviour of some alternative projections. Recall that these projections must be linear in order to maintain the multivariate normality of $p(\tilde{\beta} \mid X, y)$ that permits the corresponding closed-form KL-divergence calculation using (4.2).

A well-known limitation of the pseudoinverse in the context of ordinary least squares is that it becomes unstable when $X^T X$ is rank-deficient (it has rank less than p), which is guaranteed when $n < p$ but can also occur when $n > p$ if the columns of X are not linearly independent. Either scenario can be addressed using L2 regularisation, which leads to a ridge regression model (Hoerl and Kennard, 1970). This naturally motivates an analogous Ridge projection,

$$\tilde{\beta}_{\text{ridge}} = \text{Proj}(X, f) = (X^T X + \lambda I)^{-1} X^T f\,, \tag{4.7}$$

where $\lambda > 0$ is a hyperparameter controlling the L2 regularisation strength whose value is typically selected using cross-validation. The final projection investigated here is the Covariance projection,

$$\tilde{\beta}_{\text{cov}} = \begin{pmatrix} \text{cov}(x^{(1)}, f) \\ \vdots \\ \text{cov}(x^{(p)}, f) \end{pmatrix}, \tag{4.8}$$

where $x^{(j)} = (x_1^{(j)}, \ldots, x_n^{(j)})$ is the value of variable j across the n samples (the j^{th} column of X). As (4.8) is the concatenation of the sample covariance of each variable with f, the Covariance projection is distinct from the multivariate Pseudoinverse and Ridge projections in that it is univariate. The equivalent matrix multiplication-based calculation of (4.8) is

$$\tilde{\beta}_{\text{cov}} = \frac{1}{n-1} X^T C f\,, \tag{4.9}$$

where $C = I - \frac{1}{n} 1_n 1_n^T$ is the centring matrix and 1_n is an n-dimensional vector of ones.

3.1 Computing the ESA posterior

If an appropriate predictive model (for example, a GP regressor) has already been fit on the dataset (X, y) then $p(f|X, y) = \mathcal{N}(\mu_f, \Omega_f)$ is available. The ESA posterior density $p(\tilde{\beta}|X, y) = \mathcal{N}(\mu, \Omega)$ is then obtained using the linear transformation

$$\mu = L\mu_f\,, \quad \Omega = L^T\Omega_f L\,, \tag{4.10}$$

where $L \in \mathbb{R}^{p\times n}$ is the linear operator that defines the chosen projection. The operators for the three projections are

$$L_{\text{pinv}} = (X^TX)^{-1}X^T \tag{4.11}$$

$$L_{\text{ridge}} = (X^TX + \lambda I)^{-1}X^T \tag{4.12}$$

$$L_{\text{cov}} = \frac{1}{n-1}X^TC\,. \tag{4.13}$$

For each of these three projections the ESA posterior mean is straightforward to understand as a linear summary of the main effect of that variable. However, the importance of a variable according to RATE also incorporates the relative centrality of each variable in the ESA posterior covariance Ω. Using the relation $\Omega = L^T\Omega_f L$ the covariance between the ESA of variables i and j is given by

$$\text{cov}(\tilde{\beta}_i, \tilde{\beta}_j) = \Omega_{ij} = \sum_{k=1}^{n}\sum_{l=1}^{n} L_{ki}\,(\Omega_f)_{kl}L_{lj}\,. \tag{4.14}$$

The terms in this summation are large if variables i and j both have large linear effects (with the same sign) in a large number of samples. The two samples must also have a large covariance in $p(f \mid X, y)$. On the other hand, summation terms are close to zero if (i) the two samples have close to zero covariance ($(\Omega_f)_{kl} = 0$) or if (ii) the variables do not have large covariances (of the same sign) in any pairs of samples.

4 GroupRATE: variable prioritisation for grouped variables

In many biological applications variables fall naturally into groups. For example, single-nucleotide polymorphisms (SNPs) form genes, while microbiota form taxa and medical images contain groups of pixels corresponding to anatomically relevant features. For this reason it is often of interest to prioritise variables at the group level. This has motivated the development of a novel extension to RATE, called GroupRATE.

Grouping variables can also dramatically reduce the number of objects to be compared by reducing the resolution at which a system is studied. This leads to a reduction in both the computational cost and statistical difficulty (for example, by reducing the number of tests when working in a hypothesis testing framework). Furthermore, groups may be a more natural resolution at which to interpret the system of interest. This is the case in brain magnetic resonance imaging (MRI) scanning,

where individual voxels have extremely limited meaning but can be grouped into brain regions that are far more relevant and interpretable (Wehenkel et al., 2018). Grouping variables before calculating importance can also give statistical benefits when variables exhibit a high degree of collinearity within a group. This is the case in medical imaging, where there is high spatial correlation between pixels/voxels as well as in genetic studies, where linkage disequilibrium can cause SNPs on a gene to be highly collinear.

4.1 Calculating GroupRATE values

RATE is calculated for a single variable using (4.3). If the variables now form a set G of groups $\mathcal{G} = \{g_1, \ldots, g_G\}$ with sizes $|g_1|, \ldots, |g_G|$, then (4.3) becomes

$$\mathrm{KLD}_g \approx \frac{1}{2}(\omega_{-g}\,\omega_g^{-1}\mu_g)^T \Lambda_{-g}(\omega_{-g}\,\omega_g^{-1}\mu_g) \quad \forall g \in \mathcal{G}\,, \tag{4.15}$$

where g is a $|g|$-dimensional index set of group members.

Given $p(\tilde{\beta} \mid X, y) = \mathcal{N}(\mu, \Sigma)$ (which is still computed on a per-variable basis), solving KLD_g for group g requires an analogous partitioning of μ and Σ as is performed in the RATE case. The main difference is that RATE requires removing a single row (and column for matrices) corresponding to the j^{th} variable, while GroupRATE removes the $|g|$ rows corresponding to all the variables in group g. The visualisation in Figure 4.1 still applies but the quantities now have the dimensions listed in Table 4.1.

TABLE 4.1: Comparison between quantities in the RATE and GroupRATE calculation for p variables.

RATE quantity	GroupRATE equivalent	Description
$j \in \{1, \ldots, p\}$	$g \subset \{1, \ldots, p\}$	variable index/group indices
$\mu_j \in \mathbb{R}$	$\mu_g \in \mathbb{R}^{\lvert g\rvert}$	ESA posterior mean of variable(s) of interest
$\mu_{-j} \in \mathbb{R}^{p-1}$	$\mu_{-g} \in \mathbb{R}^{p-\lvert g\rvert}$	ESA posterior mean of other variables
$\omega_j \in \mathbb{R}$	$\omega_g \in \mathbb{R}^{\lvert g\rvert \times \lvert g\rvert}$	ESA (co)variance for variable(s) of interest
$\omega_{-j} \in \mathbb{R}^{(p-1)}$	$\omega_{-g} \in \mathbb{R}^{(p-\lvert g\rvert)\times \lvert g\rvert}$	ESA covariance between variables of interest and the rest
$\Lambda_{-j} \in \mathbb{R}^{(p-1)\times(p-1)}$	$\Lambda_{-g} \in \mathbb{R}^{(p-\lvert g\rvert)\times(p-\lvert g\rvert)}$	ESA precision of other variables

4.2 Calculating GroupRATE for Bayesian neural networks

Last layer Bayesian regression network

Previous work on RATE focused on the interpretation of Gaussian process (GP) regression models (Crawford et al., 2019). One of the primary contributions of

this chapter is its extension to Bayesian neural networks. Chapter 2 (Section 1.4) described last layer Bayesian neural networks trained using variational inference, where variational posteriors are placed over the parameters of the final layer. The inner layers, which act as feature extractors, use point estimates as their parameters. A regression last layer only Bayesian neural network can be written as

$$y_i \sim \mathcal{N}(f(x_i), \sigma^2(x_i)), \quad i = 1, , \ldots, n, \tag{4.16}$$

where $f(\cdot)$ and $\sigma^2(\cdot)$ are the two outputs of a single neural network, given by

$$\begin{pmatrix} f(x) \\ \sigma^2(x) \end{pmatrix} = w\, h_\theta(x) + b, \quad \begin{pmatrix} w \\ b \end{pmatrix} \sim p(\tilde{\theta}), \tag{4.17}$$

where the random variables $\{w, b\} = \tilde{\theta}$ are the weights and biases of the final layer and $p(\tilde{\theta})$ is their prior. The inner-layer activations $h_\theta(x)$ depend deterministically on the inner-layer parameters θ, which are point estimates. This model is essentially a Bayesian linear regression with neural network features, where $h_\theta(x)$ are these extracted features. Training the network via variational inference (described in Chapter 2, Section 1.4) returns estimates of both $\tilde{\theta}$ and θ, which allows samples to be drawn from the predictive posterior as follows:

$$\begin{aligned}
\begin{pmatrix} \widehat{w} \\ \widehat{b} \end{pmatrix} &\sim q_\phi(\tilde{\theta}) && \text{sample final layer parameters} \\
\begin{pmatrix} f(x) \\ \sigma(x) \end{pmatrix} &= \widehat{w}\, h_\theta(x) + \widehat{b} && \text{mean/variance from neural network} \\
\hat{y} &\sim \mathcal{N}(f(x), \sigma^2(x)) && \text{sample prediction from normal distribution}
\end{aligned}$$

Calculating (Group)RATE values using the closed-form expressions in Section 2 requires access to a multivariate Gaussian $p(f \mid X, y)$, which are then transformed to effect size analogues using one of the linear projections. For GP regression this is simply the posterior distribution, which is multivariate Gaussian by definition. For this last layer Bayesian neural network $p(f \mid X, y)$ is also multivariate Gaussian when $q_\phi(\tilde{\theta})$ is Gaussian, as it is a linear transformation of $\tilde{\theta} \sim q_\phi(\tilde{\theta})$ using a deterministic set of features $h_\theta(x)$. The predicted noise variance $\sigma^2(x)$ is also multivariate Gaussian for the same reason and so could be the target in the GroupRATE calculation. As the aim is to associate variables with the mean of the response (and not the variance) $f(x)$ is used as the GroupRATE target throughout.

Computing GroupRATE values in closed-form

The first step of calculating GroupRATE values for a Bayesian neural network using a batch of n examples $X \in \mathbb{R}^{n \times p}$ is to compute the final layer activations $H = (h_\theta(x_1), \ldots, h_\theta(x_n))^T \in \mathbb{R}^{n \times K}$, where K is the size of the penultimate layer. As H depends deterministically on the inner layer parameters θ (which are point estimates) they are not random variables and so do not affect the shape of $p(f \mid X, y)$.

Last layer Bayesian neural networks make a prediction by multiplying the extracted features H by the final layer parameters $\tilde{\theta} = \{w, b\} \sim q_\phi(\tilde{\theta})$. This linear operation computes a set of means $f = (f(x_1), \ldots, f(x_n))^T$ and variances $(\sigma^2(x_1), \ldots, \sigma^2(x_n))^T$ that parametrise a univariate Gaussian, from which a predicted label is then sampled. For the GroupRATE calculation only the part of this calculation that computes f is required.

Written out explicitly, the linear operation performed by hte final layer of the work on a batch of inputs is

$$\underbrace{\begin{pmatrix} — & h_\theta(x_1) & — \\ & \vdots & \\ — & h_\theta(x_n) & — \end{pmatrix}}_{H} \underbrace{\begin{pmatrix} | & | \\ w_f & w_{\sigma^2} \\ | & | \end{pmatrix}}_{w} + \begin{pmatrix} b_f & b_{\sigma^2} \\ \vdots & \vdots \\ b_f & b_{\sigma^2} \end{pmatrix} = \begin{pmatrix} \underbrace{f(x_1)}_{f} & \sigma^2(x_1) \\ \vdots & \vdots \\ f(x_n) & \sigma^2(x_n) \end{pmatrix}.$$

The part corresponding to f can be therefore be written as

$$f = H\, w_f + (b_f, \ldots, b_f)^T, \tag{4.18}$$

which means that the posterior density of $p(f \mid X, y)$ can be obtained via a linear transformation of the variational parameters ϕ. As the variational posterior $q_\phi(\tilde{\theta})$ is mean-field Gaussian over the elements of w and b, the full set of variational parameters only contains a mean and variance for each element of w and b. It is given by

$$q_\phi(\tilde{\theta}) = \prod_{k=1}^{K} \left[\mathcal{N}\left(m_{w_{f_k}}, v_{w_{f_k}}\right) \mathcal{N}\left(m_{w_{\sigma_k^2}}, v_{w_{\sigma_k^2}}\right) \right] \mathcal{N}\left(m_{b_f}, v_{b_f}\right) \mathcal{N}\left(m_{b_{\sigma^2}}, v_{b_{\sigma^2}}\right), \tag{4.19}$$

which is a $2K + 2$ dimensional diagonal Gaussian over the elements of w and b.

The variational posterior over the elements of w_f and b_f (the elements of w and b required to compute f) is obtained by simply selecting the appropriate means and variances from this diagonal Gaussian to give,

$$q_\phi(w_f, b_f) = \mathcal{N}(m, \text{diag}(v)), \tag{4.20}$$

where $\text{diag}(v)$ denotes a matrix with v on its diagonal and zeros elsewhere and

$$m = (m_{w_{f_1}}, \ldots, m_{w_{f_K}}, m_{b_f})^T \tag{4.21}$$

$$v = (v_{w_{f_1}}, \ldots, v_{w_{f_K}}, v_{b_f})^T. \tag{4.22}$$

A linear transformation of (4.20) using (4.18) gives the posterior density $p(f \mid X, y) = \mathcal{N}(\mu_f, \Sigma_f)$, where

$$\mu_f = H\, m_{w_f} + (m_{b_f})_{i=1}^{n} \tag{4.23}$$
$$\Omega_f = H\, \mathrm{diag}(v_{w_f})\, H^T + (v_{b_f})_{i=1}^{n}\,. \tag{4.24}$$

Given a linear projection operator L, the ESA posterior parameters can then be calculated using (4.15).

$$\mu = L\mu_f\,, \quad \Omega = L^T \Omega_f L\,, \tag{4.25}$$

from which the KL-divergences can be calculated for each group using (4.15).

5 Group prioritisation simulations: simulated covariates

5.1 Simulation aims

This section presents a simulation study that emulates a scenario in which a last layer Bayesian neural network has been trained to predict a continuous response from grouped variables. The simulation procedure is

1. given a dataset $\mathcal{D} = (X, y)$, where the columns of X have a grouped structure, train a Bayesian neural network,
2. calculate *post-hoc* importances for each group,
3. evaluate the variable importance scores using the AUC (area under curve).

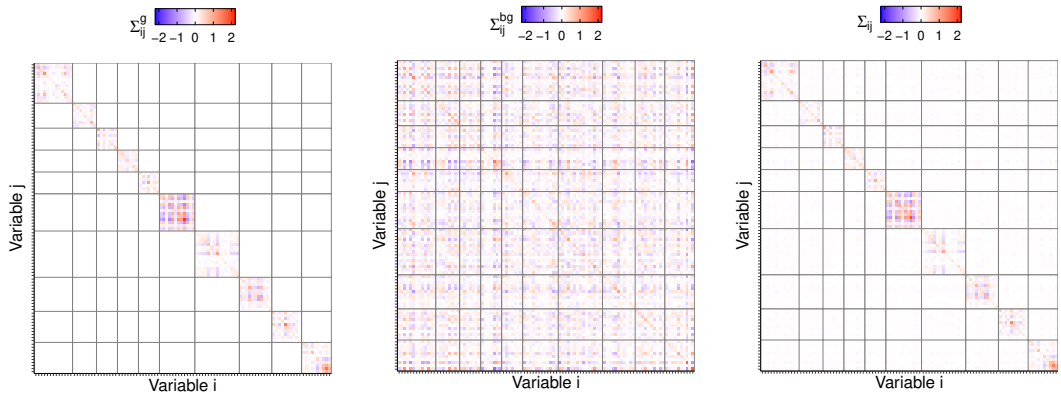
The aim is to evaluate how well GroupRATE is able to identify variables that are associated with the response. In this simulation setup both the covariates and a continuous response are simulated. The following section (Section 6) uses real covariates from human genotype data.

5.2 A group-dependent covariance structure

Covariates are simulated under a log-normal distribution as it is commonly used to model a range of biological data, such as gene expression (Torrenté et al., 2020) and single-cell RNA Seq (Luecken and Theis, 2019). Here, the design matrix $X \in \mathbb{R}^{n\times p}$ is sampled from a zero-mean log-normal distribution,

$$\log X \sim \mathcal{N}(0, \Sigma_X)\,, \tag{4.26}$$

where $\Sigma_X = 0.9\Sigma_X^{\mathcal{G}} + 0.1\Sigma_X^{\mathrm{bg}}$ for a group-dependent covariance $\Sigma_X^{\mathcal{G}}$ and background covariance Σ_X^{bg}. In each replicate a background covariance Σ_X^{bg} is sampled according to $\Sigma_X^{bg} \sim \mathcal{W}^{-1}(I_p, p+3)$, which is an inverse Wishart distribution with an identity scale matrix and $p+3$ degrees of freedom. The inverse Wishart is a distribution over positive-definite matrices and is the conjugate prior to a multivariate Gaussian covariance, which makes it a natural choice to sample covariance matrices. The



(A) Σ^{g}_{X} (B) Σ^{bg}_{X} (C) $\Sigma_X = 0.9\Sigma^{g}_{X} + 0.1\Sigma^{bg}_{X}$

FIGURE 4.3: An sample of the covariance structure and variable groupings used in the simulations in Section 5. The matrices are partitioned based on the group structure.

number of degrees of freedom in the inverse Wishart controls the concentration of the density around the scale matrix, with larger values increasing the concentration.

The structure of $\Sigma^{\mathcal{G}}_{X}$ is block-diagonal, with a block for each group in $\mathcal{G} = \{g_1, \ldots, g_G\}$ and zeroes elsewhere,

$$\Sigma^{\mathcal{G}}_{X} = \begin{pmatrix} \Sigma^{g_1}_{X} & & & \\ & \Sigma^{g_2}_{X} & & \\ & & \ddots & \\ & & & \Sigma^{g_G}_{X} \end{pmatrix}, \tag{4.27}$$

where

$$\Sigma^{g}_{X} \sim \mathcal{W}^{-1}(I_{|g|}, |g| + 3), \quad \forall g \in \mathcal{G}. \tag{4.28}$$

The group structure is an important part of these simulations. Each simulation contains $p = 10G$ variables whose sizes are distributed according to

$$|g|_1, \ldots, |g|_G \sim \text{Multinomial}\left(p, \frac{1}{p}\right), \tag{4.29}$$

which enforces $\sum_{g \in \mathcal{G}} |g| = p$ but allows for different sizes of groups. A single sample of Σ_X is shown in Figure 4.3 with $G = 10$ and $p = 100$. This construction of Σ_X ensures there is group-dependent structure in the covariance while also containing non-trivial covariances between other variables.

5.3 Phenotype model

The final element of the simulated dataset is the response. The response should depend non-linearly on the covariates as the ability of neural networks to model such non-linear behaviour is the primary motivation for their adoption in these settings. A fictitious continuous phenotype $y \in \mathbb{R}^n$ is generated from the simulated X under the following model:

$$y = \underbrace{\tilde{X}\beta}_{\text{main effects}} + \underbrace{\tilde{W}\Theta}_{\text{pairwise effects}} + \underbrace{\varepsilon}_{\text{environmental effects}}, \qquad \varepsilon \sim \mathcal{N}(0, I) \tag{4.30}$$

where $\tilde{X}$ is a matrix containing the columns of X corresponding to variables with main effects, $\tilde{W}$ contains the products of variables involved in pairwise interactions, β, Θ are the corresponding effect size vectors with elements drawn from a standard normal and I is the identity matrix. The construction of $\tilde{X}$ and $\tilde{W}$ from X is described in the next section.

The variance of y is equal to 1 and the contributions of each set of effects controlled such that

$$\text{var}(\tilde{X}\beta) + \text{var}(\tilde{W}\Theta) = H^2 \qquad \text{broad-sense heritability} \tag{4.31}$$

$$\text{var}(\tilde{X}\beta) = h^2 \qquad \text{narrow-sense heritability} \tag{4.32}$$

$$\text{var}(\tilde{W}\Theta) = H^2 - h^2 \qquad \text{variance due to interactions} \tag{4.33}$$

$$\text{var}(\varepsilon) = 1 - H^2 \qquad \text{variance due to environment} \tag{4.34}$$

Narrow- and broad-sense heritability are terms used in genetics to describe proportions of phenotypic variance. While this simulation setup is not explicitly in the genetics setting the same terms are used here for consistency with the simulations in Section 6, which do use genetic data.

The narrow-sense heritability of a trait, h^2, is the proportion of phenotypic variance explained by additive genetic effects while broad-sense heritability, H^2, is the proportion explained by all genetic effects (Tenesa and Haley, 2013). The difference $H^2 - h^2$ is therefore the proportion of phenotypic variance explained by non-additive effects, which are assumed to consist solely of pairwise interactions. The remainder of phenotypic variance ($1 - H^2$) is explained by environmental effects.

Simulations are run under two different scenarios:

Scenario A: $h^2 = 0.6$ and $H^2 - h^2 = 0.2$; and

Scenario B: $h^2 = 0.4$ and $H^2 - h^2 = 0.4$.

Scenario A simulates a phenotype that is controlled mostly by main effects, with a smaller share of the phenotypic variance controlled by pairwise interactions. In Scenario B main and pairwise effects account for an equal share of phenotypic variance.

5.4 Strong hierarchy assumption

The pairwise interactions in the model are generated under a *strong hierarchy* assumption, where interaction effects are restricted to occur between variables with a main (linear) effect. This is one of two standard genetic modelling assumptions along with *weak hierarchy*, which allows for interactions between pairs where only one variable has a main effect. Strong hierarchy is the more common of the two assumptions (T. T. Wu et al., 2009; Bien et al., 2013; M. Lim and Hastie, 2015).

Bien et al. (2013) outline two main arguments in favour of strong hierarchy assumptions. They consider the following interaction model,

$$y = (\beta_0)_{i=1}^n + \sum_j \beta_j x^{(j)} + \sum_{j,k,j\neq k} \Theta_{jk}\, x^{(j)} \odot x^{(k)} + \varepsilon\,, \quad \varepsilon \sim \mathcal{N}(0, I)\,, \tag{4.35}$$

where $x^{(j)}, x^{(k)} \in \mathbb{R}^n$ are columns j,k of $\mathcal{X}$, β_0 is the intercept, β_j are main effects, Θ_{jk} are interaction effects and $\odot$ denotes the element-wise product. The strong and weak hierarchy assumptions can be formulated as

$$\Theta_{jk} \neq 0 \implies \beta_j, \beta_k \neq 0 \qquad \text{strong hierarchy} \tag{4.36}$$

$$\Theta_{jk} \neq 0 \implies \beta_j \neq 0 \text{ or } \beta_k \neq 0\,. \qquad \text{weak hierarchy} \tag{4.37}$$

The first argument in favour of strong hierarchy considers (4.35) under strong and weak hierarchy:

$$y = \beta_0 + \beta_1 x^{(1)} + \beta_2 x^{(2)} + \Theta_{12} x^{(1)} x^{(2)} + \ldots \qquad \text{strong hierarchy} \tag{4.38}$$

$$y = \beta_0 + (\beta_1 + \Theta_{12} x^{(2)}) x^{(1)} + \ldots\,. \qquad \text{weak hierarchy} \tag{4.39}$$

In (4.38) $x^{(1)}$ will have an effect on y irrespective of the value of $x^{(2)}$. However, in (4.39) $x^{(1)}$ only affects y when $x^{(2)} \neq 0$. While this may be true in specific scenarios it is unlikely to be true in general. Furthermore, if $x^{(2)}$ undergoes an affine transformation $ax^{(2)} + b, b \neq 0$, then the model (4.39) satisfies strong hierarchy. Given that variables commonly undergo such transformations during statistical analysis pipelines (e.g. normalisation/whitening), this suggests that a strong hierarchy assumption is a sensible default.

The second argument is that a strong hierarchy assumption increases statistical power as the size of search space when identifying pairwise interactions is greatly reduced. Furthermore, interactions involving two variables with main effects are likely to be both easier to identify and of greater interest than pairs where only one variable has a main effect. This is especially true in genetics applications where there are a large number of candidate variables of interest and limited resources for follow-up experiments. Models that prioritise interactions between candidate variables with main effects are therefore preferable to those which identify hard to validate interactions.

5.5 Selecting main and pairwise effects

In each replicate a set of variables with main and pairwise effects are selected based on their grouping structure. These variables then form the columns of $\tilde{X}$ and $\tilde{W}$ in the response model (4.30). Given a set of G groups $\mathcal{G}$, a subset with size $\frac{G}{10}$ are sampled without replacement to be causal (associated with the response), denoted by $\tilde{\mathcal{G}}$. The set of variables corresponding to the groups in $\tilde{\mathcal{G}}$,

$$\tilde{\mathcal{S}} = \{j \in g : g \in \tilde{\mathcal{G}}\}\,, \tag{4.40}$$

are the candidates to be associated with the response. One variable is then selected

from each member of $\tilde{\mathcal{G}}$ to have a main effect - these variables form the columns of $\tilde{X}$ and are denoted by $\tilde{\mathcal{S}}_{\text{main}} \subset \{1, \ldots, p\}$, where $|\tilde{\mathcal{S}}_{\text{main}}| = \frac{G}{10}$.

Under the strong hierarchy assumption the variables involved pairwise interactions must both have a main effect. Let $\mathcal{P}$ denote the set of candidates for pairwise interactions, which are the unordered pairs

$$\mathcal{P} = \{\{i, j\} : i \in \tilde{\mathcal{S}}_{\text{main}}, j \in \tilde{\mathcal{S}}_{\text{main}}, i \neq j\}, \tag{4.41}$$

from which a set of of variable pairs are sampled without replacement and placed in the set of interacting variables, $\tilde{\mathcal{P}}$, where

$$\tilde{\mathcal{P}} \subset \mathcal{P}, \quad |\tilde{\mathcal{P}}| = \frac{G}{20}. \tag{4.42}$$

The columns of $\tilde{W}$ are the products of each pair in $\tilde{\mathcal{P}}$,

$$x^{(i)} \odot x^{(j)} \; \forall \, \{i, j\} \in \tilde{\mathcal{P}}, \tag{4.43}$$

where $\odot$ again denotes the element-wise product.

5.6 Final simulation procedure

The final simulation procedure is as follows. In each replicate Σ_X, the group structure and X are sampled. Then the set of causal groups, causal variables and the corresponding effect sizes are sampled and used to calculate the response. A four-layer, last layer Bayesian neural network is then trained on this simulated dataset by maximising the evidence lower bound using the Adam optimiser with a learning rate of 10^{-3} for a maximum of 300 epochs (Diederik P Kingma and Ba, 2014).

Training uses 80% of the samples while the remaining 20% are held-out as testing data. In addition, 10% of the training set is used as validation data to monitor the behaviour of the loss function - if the validation loss does not decreases for 30 epochs then training is terminated (early stopping). The weight of the KL-divergence regularisation term in the evidence lower bound (often denoted using β, Higgins et al., 2016) is set to 0.3 throughout and a standard normal prior is used for all variational parameters. Rectified linear unit activations are used for hidden layers, each of which contains eight units. The output layer contains two units and uses an identity activation. This is repeated for 100 replicates.

Note that no hyperparameter optimisation is performed on the Bayesian neural network as the aim here is not to optimise generalisation performance. In a real application it is assumed that an extensive hyperparameter search and cross-validation would already been performed to obtain a final model. The task here is then to interpret this model via a *post-hoc* analysis.

5.7 AUC as an evaluation metric for group prioritisation

As noted in the previous chapter, the AUC can be interpreted as the probability that the score of a randomly-selected positive item has a higher score than a randomly-selected negative label (Fawcett, 2006) and so is equivalent to a Mann-Whitney U/Wilcoxon

rank sum test on the scores of the positive and negative groups (Calders and Jaroszewicz, 2007). This interpretation of AUC highlights its scale-invariant property as it depends only on the ordering of the scores.

The AUC is one of the most popular metrics in machine learning for evaluating classifiers, in which case the labels correspond to the class labels and the scores are outputs from a classifier. In this setting, however, the labels denote whether groups appear in the model and the scores are group importances. The AUC is therefore an appropriate metric for variable prioritisation as it will be equal to 1 if and only if all the causal variables have higher scores than non-causal variables (i.e. a perfect ranking).

5.8 Predictive performance of the Bayesian neural network

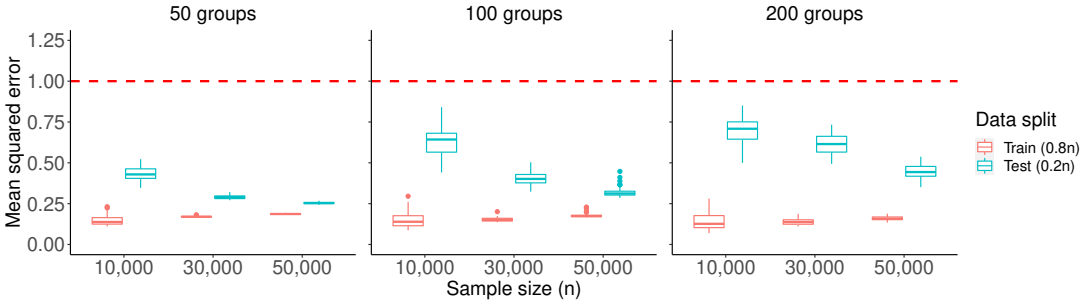
Model checking is an important part of the analysis pipeline even if predictive performance is not the primary focus of the analysis. Figure 4.4 shows the predictive mean squared error of the models across the 100 replicates. The mean squared error is determined by a combination of two factors: (i) the curse of dimensionality and (ii) the proportion of variance due to additive effects (h^2). The curse of dimensionality means that the problem becomes more difficult as the ratio n/p decreases, while a larger h^2 value also results in an easier regression task. While the Bayesian neural network is able to effectively model the non-linear portion of the response, this requires more samples than the equivalent linear signal. The regression task for smaller h^2 is therefore more difficult for a fixed sample size.

These factors explain the decreasing test mean squared error as n increases, which is to be expected. It also explains why the the test mean squared error is larger when more groups are used (for a fixed sample size), as well as why the test mean squared error is lower in Figure 4.4(A) than Figure 4.4(B). The regression task is at its most difficult in the right-hand plot of Figure 4.4(B), which is when n/p and h^2 are both set to their smallest values. This leads to some catastrophic over-fitting in some replicates, as the test mean squared error is greater than 1 (the expected mean squared error of a baseline model that predicts the mean of the training set). These replicates are excluded from the variable importance analysis in the next section.

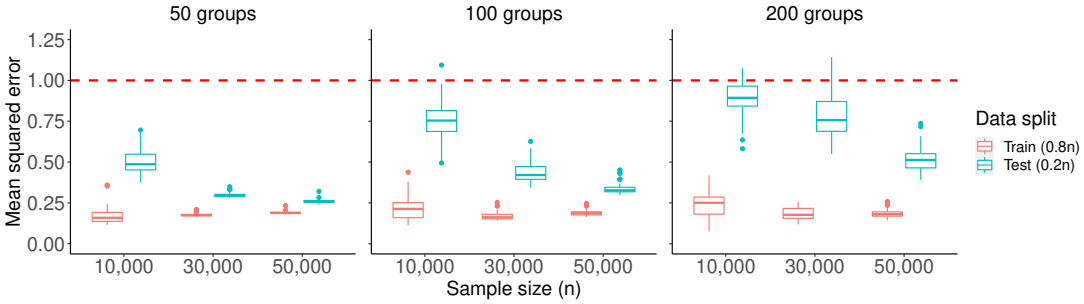
The training mean squared error is approximately constant across all the plots in Figure 4.4 as in each case the model has the capacity to memorise the training data. However, for more difficult tasks (smaller h^2 or n/p) these learned features are more likely to lead to over-fitting, resulting in a large test mean squared error.

5.9 Correcting the bias in KL-divergences from group size

The groups in this simulation have a range of sizes which may introduce a bias in GroupRATE scores that are not present in the original RATE calculation. Figure 4.5(A) shows that the KL-divergence values for a group is positively correlated with the group size, especially for the Covariance projection. This correlation decreases towards zero as n increases, but this bias can be mitigated by dividing the KL-divergences by the group size (Figure 4.5(B)). This motivates calculating GroupRATE scores using



(A) $h^2 = 0.6$, $H^2 - h^2 = 0.2$



(B) $h^2 = 0.4$, $H^2 - h^2 = 0.4$

FIGURE 4.4: Mean squared error of the Bayesian neural network models across 100 replicates. A baseline mean model has an expected mean squared error of 1 (red dashed line).

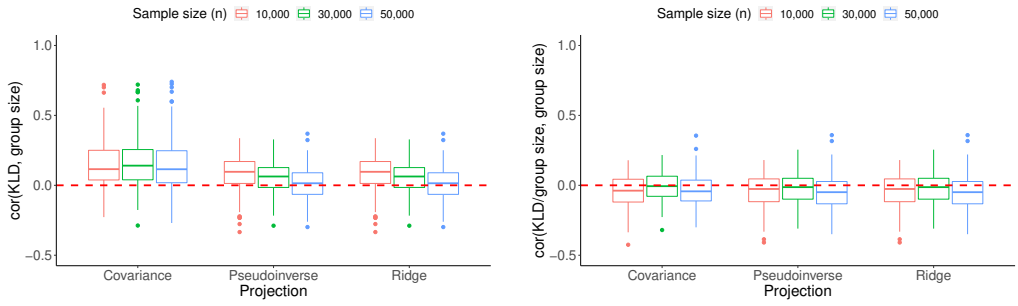
$$\gamma_j = \frac{\text{KLD}_j \frac{1}{|g_j|}}{\sum_k \text{KLD}_k \frac{1}{|g_k|}}, \tag{4.44}$$

where $|g_k|$ is the size of group k, which down-weights the GroupRATE scores of larger groups. This simple correction results in the median correlation between the KL-divergences and group size being zero.

5.10 Calculating group-level importance scores

The main area of interest for this study is the evaluation of GroupRATE's *post-hoc* grouped variable importance scores. In addition to GroupRATE, some alternative group-level importances are included here for comparison purposes. Using the trained models group importance scores are calculated with one of nine methods:

- GroupRATE with either of Covariance, Ridge or Pseudoinverse projection;
- vanilla gradients;
- gradient×input;
- integrated gradients;
- guided back-propagation;
- smoothed gradients; and



(A) (B)

FIGURE 4.5: The KL-divergence values for a group are positively correlated with the group size (plot A). This can be mitigated by dividing each KLD by the corresponding group size when calculating GroupRATE scores (plot B).

- a random forest mimic model with mean decrease Gini variable importance.

All of these methods are described in Chapter 2 (Section 2.5). All saliency methods (vanilla gradients, gradient×input, integrated gradients, guided back-propagation and smoothed gradients) compute local (per-example) scores using the gradient of the network output with respect for an example. Here, these local scores are agglomerated to global scores by taking the mean score over a set of n examples. The group-level score $s^{(g)}$ is then the mean score over the variables in a group,

$$s^{(g)} = \frac{1}{|g|n} \sum_{j \in g} \sum_{i=1}^{n} s_i^{(j)}, \quad g \in \mathcal{G}, \tag{4.45}$$

where $s_i^{(j)}$ is the local score of variable j in example i. For example, if using the vanilla gradients then then $s_i^{(j)}$ is the absolute value of the gradient,

$$s_i^{(j)} = \left| \frac{\partial f(x)}{\partial x^{(j)}} \Big|_{x=x_i} \right|, \tag{4.46}$$

evaluated at $x = x_i$. The other saliency methods are defined as in Chapter 2 (Section 2.5).

The random forest mimic model, which is a regression model trained on the original X but with the predicted probabilities of the fitted neural network as labels, computes global scores but on a per-variable basis and so its group-level scores are given by

$$s^{(g)} = \frac{1}{|g|} \sum_{j \in g} s^{(j)}, \tag{4.47}$$

where $s^{(j)}$ is the mean-decrease Gini importance of variable j according to the mimic model. Using the mean to agglomerate variable-level importances to groups has been investigated by Wehenkel et al. (2018) in the context of 3D brain imaging data and random forest importance scores. Thier simulation studies found that the mean resulted in the best variable selection performance and so that is the approach used here.

5.11 Evaluation of group prioritisation methods

Figure 4.6 shows the group prioritisation AUCs for the different methods over 50 replicates. These AUCs show almost all the methods have high power when identifying the causal groups, with AUC values greater than 0.9 for the largest sample sizes. The only exception is the random forest mimic AUC which has a median value closer to 0.85 in the easiest regression tasks (larger n/p and h^2) which decreases towards 0.6 for the hardest tasks. The other methods also exhibit decreasing AUCs as the number of groups increases, with the AUCs corresponding to integrated gradients and gradient×input also performing relatively poorly compared to the other methods. The GroupRATE AUCs are competitive with the other best-performing methods (guided back-propagation and smoothed gradients) but there are small differences between the projections. Using the Ridge and Pseudoinverse projection both lead to larger AUCs than the Covariance projection.

While these AUC values are informative it is also important to check the corresponding ROC curves. The most important part of the curve is in the low false positive rate (high sensitivity) region, as this corresponds to the highest-ranked groups. These curves are shown in Figure 4.7 and they show that smoothed gradients or guided back-propagation have high group prioritisation power in the high sensitivity region for all the dataset sizes. GroupRATE (Pseudoinverse and Ridge projections), vanilla gradients, gradient×input and smoothed gradients exhibit this behaviour in the larger samples, while the random forest mimic never has high power in the high sensitivity region.

5.12 Empirical computation times

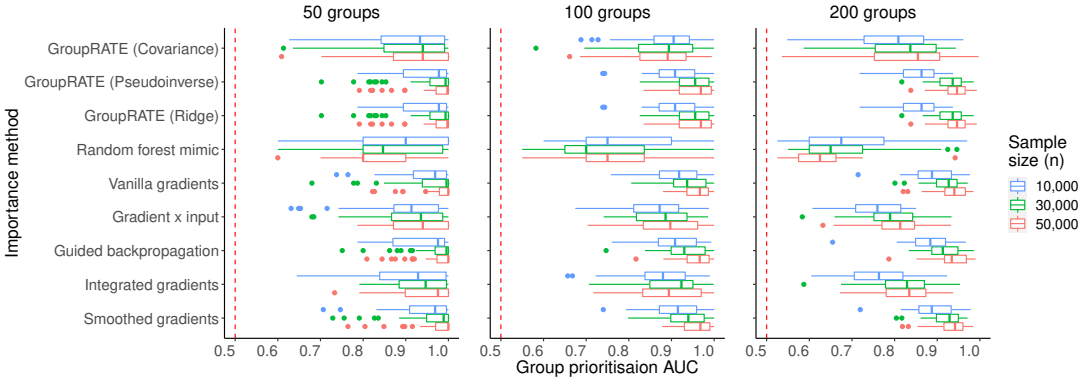
One area in which the different methods studied in this simulation setup differ is computational cost. As these are all *post-hoc* methods this discussion does not include the cost of training the Bayesian neural network as this assumed to be fixed across the methods.

The nine variable importance methods can be divided into three groups based on how they compute group scores. The first three are GroupRATE with different projections, which all require the calculating the $n \times p$ linear operator L as the first step. The Ridge and Pseudoinverse projections both require the singular value decomposition of X, which has $\mathcal{O}(n^2 p)$ running time. The Covariance projection only involves a $\mathcal{O}(n^2 p)$ matrix multiplication, which has the same asymptotic running time as the decomposition but is cheaper. This makes the Covariance projection the computationally cheapest of the three projections. Once L has been computed it is used to calculate the posterior parameters of $p(\tilde{\beta} \mid X, y)$ via an additional $\mathcal{O}(pn^2 + p^2 n)$ matrix multiplication. The final step is solving (4.2) for each of the G groups, which requires G independent solutions of a linear system, each of which are $\mathcal{O}(p^3)$. The running time complexity of the entire GroupRATE calculation is therefore

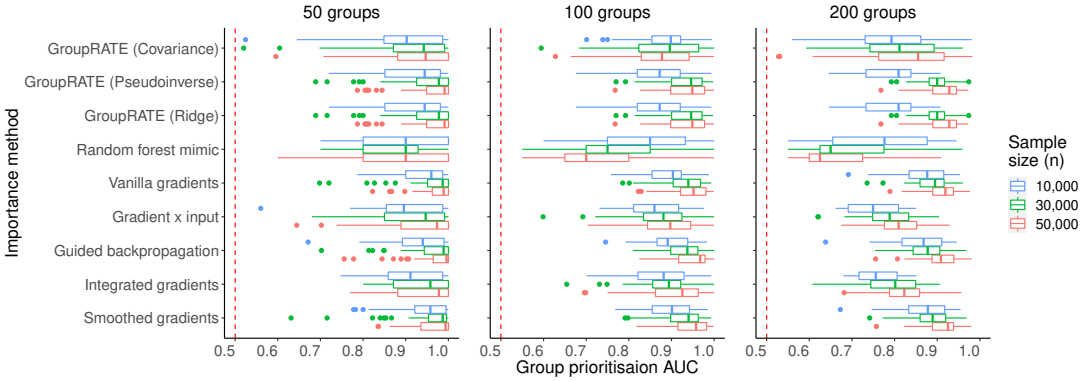
$$\mathcal{O}(pn^2 + p^2 n + G\, p^3)\,, \tag{4.48}$$

which is dominated by the ESA posterior calculation for $n \gg p$ datasets and by the solution of the KL-divergences for $p \ll n$ datasets.

The empirical computation times in these simulations with each projection are shown

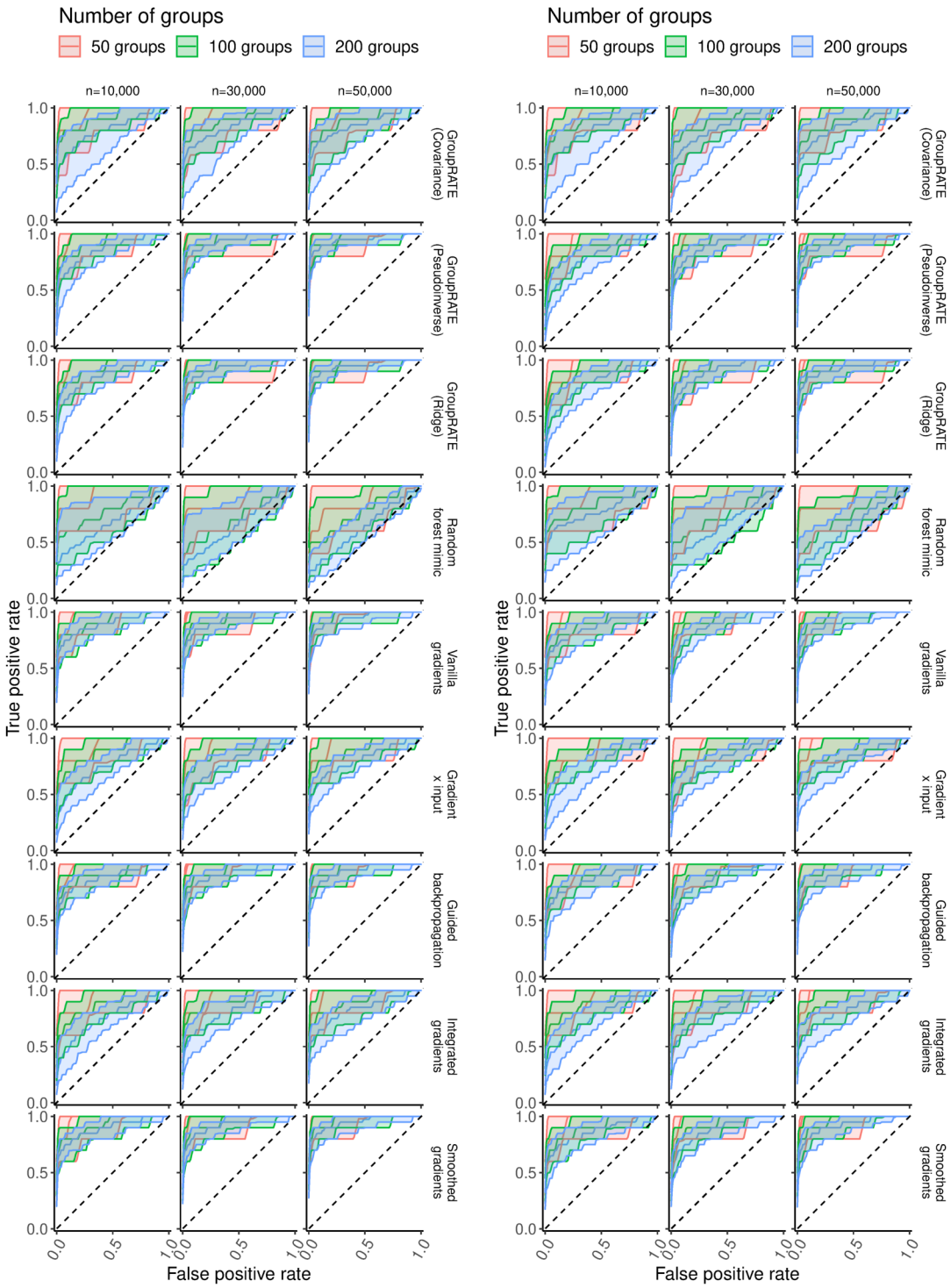


(A) $h^2 = 0.6$, $H^2 - h^2 = 0.2$



(B) $h^2 = 0.4$, $H^2 - h^2 = 0.4$

FIGURE 4.6: Variable prioritisation AUCs from 50 replicates. The red horizontal line indicates an AUC of 0.5 (the expected performance of random importance scores).



(A) $h^2 = 0.6$, $H^2 - h^2 = 0.2$ (B) $h^2 = 0.4$, $H^2 - h^2 = 0.4$

FIGURE 4.7: Group prioritisation ROC curves from 50 replicates. The black line denotes the median true positive rate across replicates and the shaded areas are the 10^{th} and 90^{th} percentiles on the empirical true positive rate.

in Figure 4.8(A-C). These timings are split into the time required to calculate the parameters of $p(\tilde{\beta} \mid X, y)$ and the subsequent KL-divergence interactions. The Covariance projection is the fastest of the three, as expected. However, the difference is in the order of minutes and so is negligible in practice for datasets of these sizes.

The second type of methods are those based on gradient evaluations (saliency maps). Here, these are implemented in TensorFlow and so the gradient evaluations are computed efficiently using automatic differentiation. However, within the saliency methods there are those requiring a single gradient evaluation (vanilla gradients, gradient$\times$input and guided back-propagation) and those that use repeated evaluations to smooth the gradients (integrated gradients and smoothed gradients).

The random forest mimic model is distinct from the other two types of method as it requires training an entire additional model. This necessitates a hyperparameter search and cross-validation, which while easy to parallelise is computationally expensive.

The empirical computation times of the saliency-based methods and random forest mimic are show in Figure 4.8(D). Integrated gradients, the random forest mimic and smoothed gradients have by far the longest running times, which are an order of magnitude longer than the other methods due to the factors outlined in the previous paragraphs (repeated gradient evaluations or a cross-validation procedure). While these methods are not particularly fast, none of these running times are sufficiently long to preclude their inclusion in an analysis for datasets of these sizes.

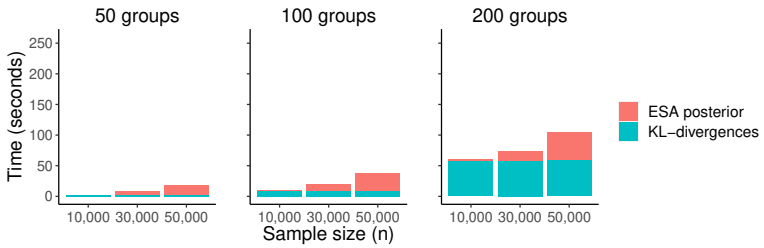
6 Genotype simulations

6.1 Simulation aims

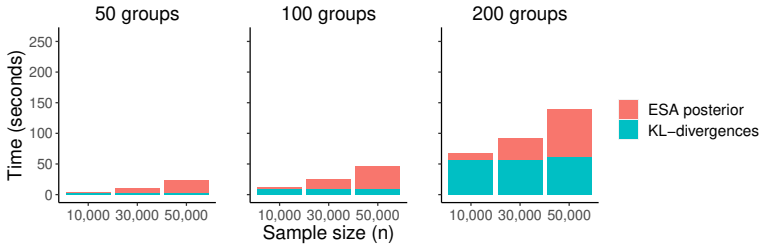
The previous set of simulations considered a scenario in which a Bayesian neural network has already been trained and needs to be interpreted via a *post-hoc* analysis, which is an increasingly common scenario as researchers seek to apply neural networks to novel problems in biology. This next set of simulations asks a related but different question - how does a Bayesian neural network interpreted with GroupRATE compare to two alternative predictive models for which group-level importances can be computed using existing methods? These two alternative models - GroupLasso and random forest with grouped importance scores - are trained directly on the observed data and so are not mimic models.

A second difference between this simulation and the previous one is the nature of the covariates. The previous simulation simulated log-normal covariates as well as a fictitious continuous response. Here, the covariates are real human genotypes that are used to simulate a fictitious continuous phenotype. This setup mirrors the genotype simulations in the original RATE paper by Crawford et al. (2019).

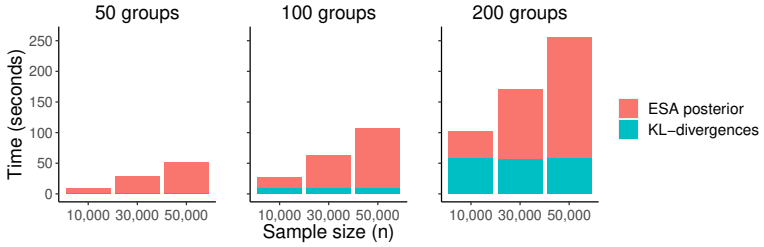
One drawback of this setup is that the maximum sample size is $n = 10,000$, which is fewer samples than are typically required to train a state-of-the-art neural network. Again mirroring Crawford et al. (2019), a GP regression model is an attractive alternative, but the $\mathcal{O}(n^3)$ required to fit such a model is too restrictive. For this reason the more scalable sparse GP model (described in Chapter 2, Section 1.2) is included alongside the Bayesian neural network. The group importance scores of the sparse GP are also computed using GroupRATE.



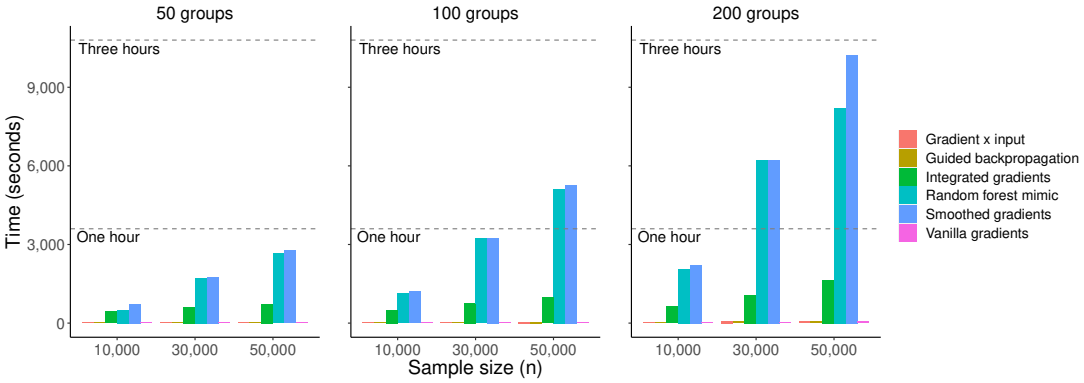
(A) GroupRATE, Covariance projection



(B) GroupRATE, Pseudoinverse projection



(C) GroupRATE, Ridge projection



(D) Non-GroupRATE methods

FIGURE 4.8: Mean empirical computation times for the different methods across 100 replicates. The GroupRATE plots (A-C) have a different vertical scale to the non-GroupRATE plot (D). Computations are run in parallel using 32 threads.

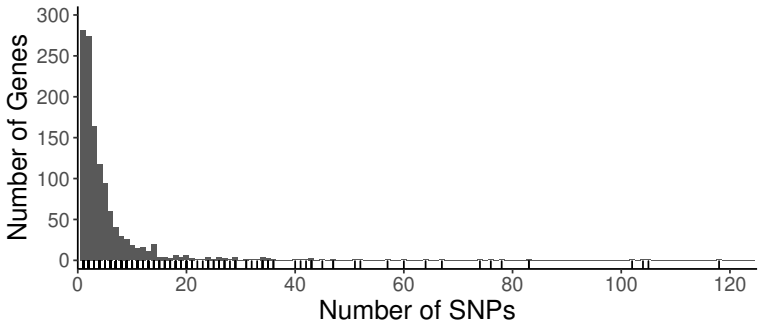


FIGURE 4.9: The size of each of the 1,255 Chromosome 1 genes included in the genotype simulations.

6.2 Simulation data

This simulation follows Crawford et al. (2019) in that it uses real genotype data, from which a fictitious continuous phenotype is simulated under a non-linear model. The real covariates are human genotype data for Chromosome 1 from the Wellcome Trust Case Control Consortium (WTCCC, WTCCC et al., 2007). As the variables represent single-nucleotide-polymorphisms (SNPs), their real group structure (genes) are used for the group definitions.

The full dataset contains the genotype information of $10,000$ individuals of European ancestry at 36,348 SNPs. Using Build 37 (hg19) of the Human genome, 16,504 of these SNPs are mapped to 1,391 genes, with the rest excluded from the simulations. 8,042 low-variance SNPs are also excluded (those with variance less than 0.2 across the $10,000$ individuals) and an additional 962 SNPs that were highly collinear with at least one other SNP (defined as a Pearson correlation greater than 0.95) are also excluded. The remaining 7,405 SNPs (mapped to 1,255 genes) are used as the covariates for these simulations. The number of SNPs for each of the included genes are shown in Figure 4.9.

6.3 Description of models

The Bayesian neural network used here is identical to the one from the previous set of simulations (see Section 5). The random forest model is trained using an identical procedure to the mimic model used in Section 5, but it is trained directly on the fictitious phenotype and not the predictions of the Bayesian neural network.

The sparse GP regression model is

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')), \tag{4.49}$$

where $m(x)$ is a mean function and $k(x, x')$ is a radial basis function kernel. The GP models are fitted on 80% of the samples via optimisation of the evidence lower bound, with the median heuristic (Flaxman et al., 2016) used as a starting guess for the single lengthscale. The locations of 1,000 inducing points are also learned during this optimisation with their starting values initialised at a random subsample of the training set. The optimisation is performed using the limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm (L-BFGS, D. Liu and Nocedal, 1989).

The other comparison model is GroupLasso, which is a popular extension to to the standard Lasso for grouped variables (Yuan and Y. Lin, 2006). It uses the Lasso loss function with an additional group-level penalty,

$$\mathcal{L}(\beta; X, y) = -\log p(y|X, \beta) + \lambda_1 \|\beta\|_1 + \lambda_G \sum_{g \in \mathcal{G}} \sqrt{G}\, \|\beta\|_2 \,, \tag{4.50}$$

where λ_1 and λ_G are hyperparameters controlling the strength of variable- and group-level regularisation (Moe, 2022). The optimisation of (4.50) results in a set of per-variable model coefficients $\beta \in \mathbb{R}^p$, meaning that the group variable prioritisation score needs to be computed. For consistency with the random forest mimic model the mean of the absolute variable coefficients,

$$s^{(g)} = \frac{1}{|g|} \sum_{j \in g} |\beta_j| \,, \tag{4.51}$$

is used to agglomerate the variable scores to the group level. The GroupLasso and random forest model hyperparameters are selected using k-fold cross-validation with $k = 5$. Note that the sparsity in sparse GPs refers to sparsity in the samples, which is distinct from GroupLasso that enforces sparsity in the features.

6.4 Final simulation setup

While the previous section used the terms *variables* and *groups*, this section uses *SNPs* and *genes* due to the nature of the data. In each replicate of this simulation a set of G genes, where $G \in \{100, 300, 1,000\}$, are sampled without replacement from the full set of 1,255 genes. The resulting number of SNPs (p) is shown in Figure 4.10. Causal genes, causal SNPs and corresponding effect sizes are then sampled using an identical procedure as used in the previous simulations (see Section 5.3).

A set of samples of size n, where $n \in \{2,000\,, 4,000\,, \ldots\,, 10,000\}$, are sampled without replacement from the observed genotypes. A fictitious continuous phenotype is simulated using the causal SNPs and their effect sizes using (4.30), which includes both main and pairwise effects. Recall that the proportion of additive variance in the phenotype is h^2 and the variance due to pairwise interaction is $H^2 - h^2$. Each of the four models (last layer Bayesian neural network, sparse GP regression, GroupLasso and random forest) is fit to the tarining set (80% of samples) and their predictive performance on the training and test sets (20% of samples) are recoded. Finally, the grouped variable importance for each model is computed. For GroupLasso and random forest computing the group scores has a negligible computational cost relative to model training. The group scores for the Bayesian neural network and sparse GP are computed using GroupRATE with each of the three projections. This is repeated for 100 replicates.

6.5 Predictive performance of the four models

GP regression models are typically evaluated using log-marginal likelihoods (for measuring goodness-of-fit on the training data) and predictive log-density (for measuring predictive performance on held-out data). These will be the metrics used

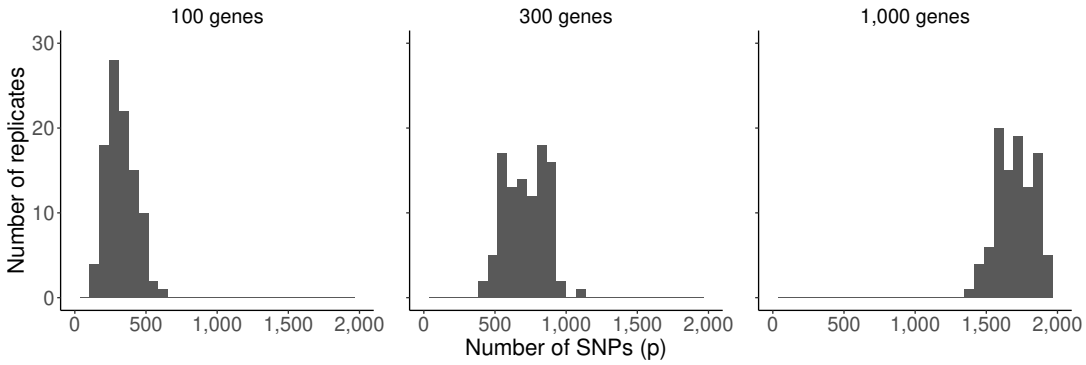


FIGURE 4.10: Number of SNPs, p, for different numbers of genes G in the genotype simulations.

in Chapter 5, where all the predictive models under consideration are GPs. However, in order to make like-for-like comparisons between the four models the mean squared error between the true labels and the predictive posterior mean is used here.

Figure 4.11 shows the predictive mean squared error for the Bayesian neural network (plot A), sparse GP (plot B), GroupLasso (plot C) and random forest (plot D). Recall that the phenotype is simulated under two scenarios: (i) where $h^2 = 0.6$, $H^2 - h^2 = 0.2$ and (ii) $h^2 = H^2 - h^2 = 0.4$.

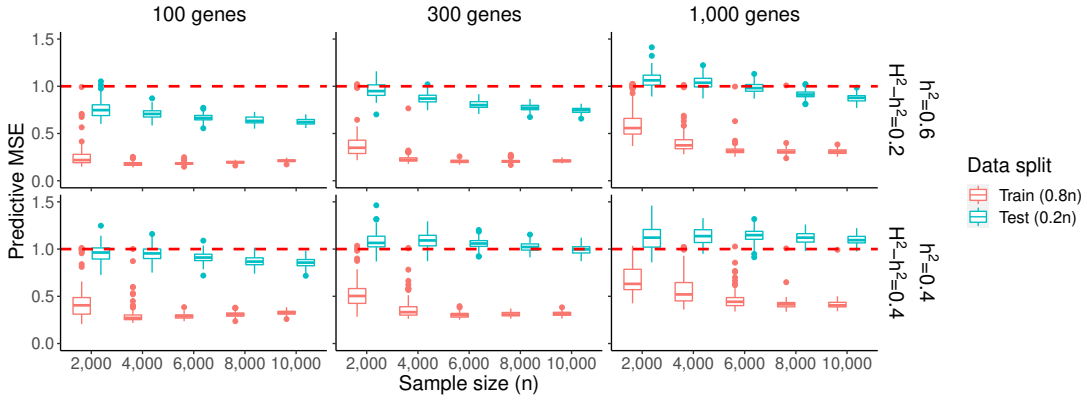
The Bayesian neural network is only able to capture predictive signal for the easiest tasks, which are those where both n/p and h^2 are at their largest values. This vindicates the inclusion of the sparse GP in this section, as the Bayesian neural network requires a much larger sample size for such high-dimensional data. The sparse GP fits the training and test data well when $G = 100$ or $G = 300$ (it is able to capture predictive signal). When $G = 1,000$ and $h^2 = 0.4$ the sparse GP essentially performs as an intercept-only model and captures no predictive signal. This also occurs in the majority of replicates when $h^2 = 0.6$. This is because the required number of inducing points increases with the dimensionality of the problem, but is fixed at 1,000 for all values of G.

GroupLasso has similarly low test mean squared error when $G \in \{100, 300\}$, but unlike the sparse GP its test mean squared error is unchanged when $G = 1,000$. This shows the power of the sparsity assumption that is unique to GroupLasso in this context. Even though the GroupLasso is unable to model non-linear dependencies between genotype and phenotype there is sufficient linear signal for GroupLasso to achieve reasonable predictive performance.

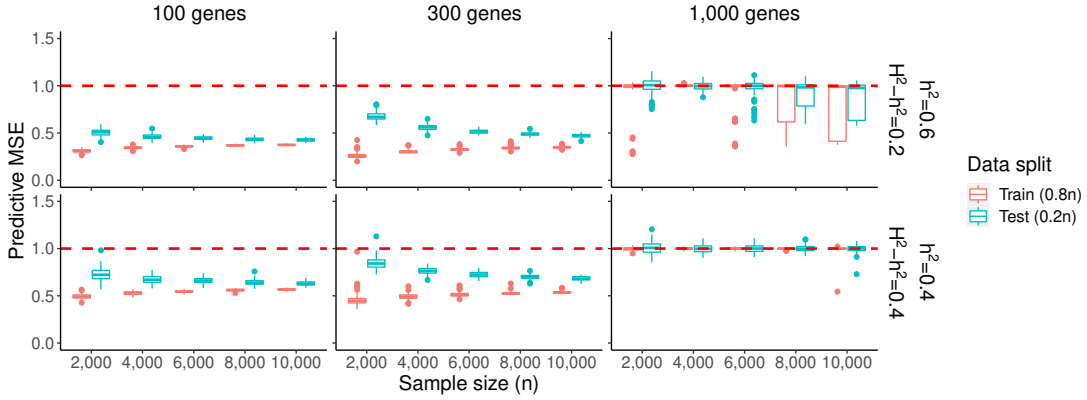
The random forest is able to capture at least some predictive signal in every setting (its test mean squared error are less than one), illustrating why it is such a popular model in bioinformatics applications. However, as the number of genes increases the amount of captured signal becomes small - this is because the cross-validation procedure used to select the hyperparameters is not increasing in size even as the regression task becomes more difficult.

6.6 Group prioritisation performance

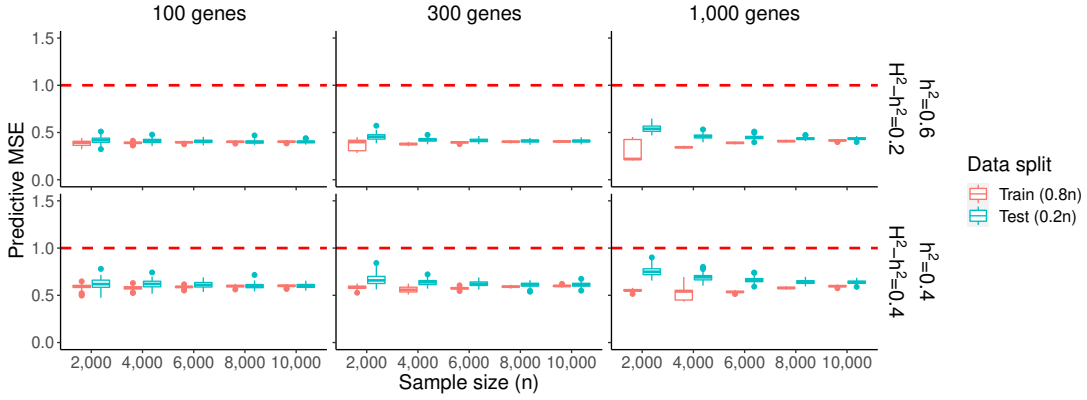
These four models offer eight different methods for group prioritisation, as both the Bayesian neural network and the sparse GP can be subjected to a *post-hoc* GroupRATE



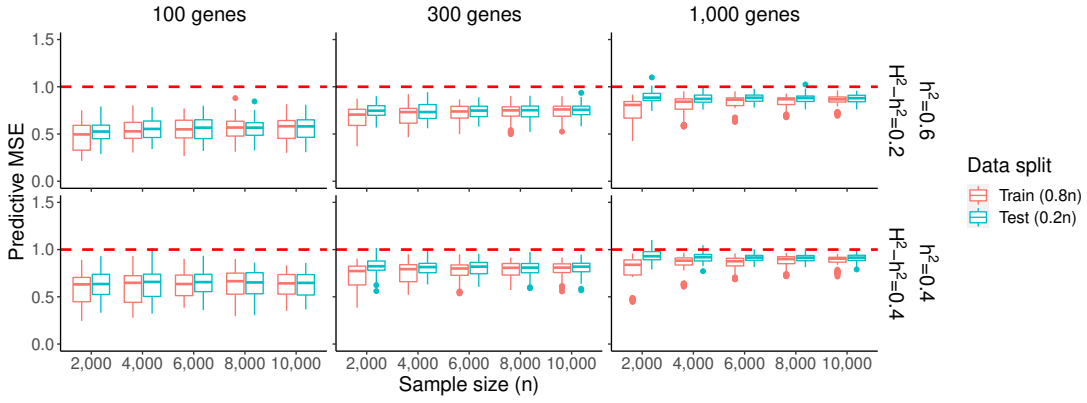
(A) Bayesian neural network



(B) Sparse GP regression



(C) GroupLasso



(D) Random forest

FIGURE 4.11: Predictive mean squared error (MSE) of the four models in the genotype simulations. Red line indicates baseline performance of a model predicting mean of training labels. The Bayesian neural network posterior mean is computed using 100 Monte Carlo samples.

analysis using one of the three projections. These group prioritisation AUCs are shown in in Figure 4.12.

GroupLasso is the best-performing method and has the largest group prioritisation AUCs across all the scenarios. This is somewhat expected due to the strong hierarchy assumption embedded into the phenotype model. This means that any group containing a SNP with a pairwise effect also has a main effect, which can be easily detected by GroupLasso.

Using GroupRATE with the sparse GP is competitive with GroupLasso in many settings, showing similarly high AUCs when $G \in \{100, 300\}$. The sparse GP is also clearly preferable to the Bayesian neural network when $G \in \{100, 300\}$ as a model for group prioritisation. When $G = 1,000$ the Bayesian neural network slightly outperforms the sparse GP, which has learnt no predictive signal. However, its AUC values are still far smaller than GroupLasso. Of the three projections, the Covariance projection has the lowest AUCs for both the Bayesian neural network and the sparse GP.

The grouped-variable importance scores of the random forest consistently has an AUC close to 0.8 across all different settings in Figure 4.12. This fits with random forest's reputation as a robust method that offers performance out-of-the-box with relatively little tuning. It is likely that a more extensive hyperparameter search would improve these AUCs, but as shown by the empirical timings in the previous section this becomes very computationally demanding.

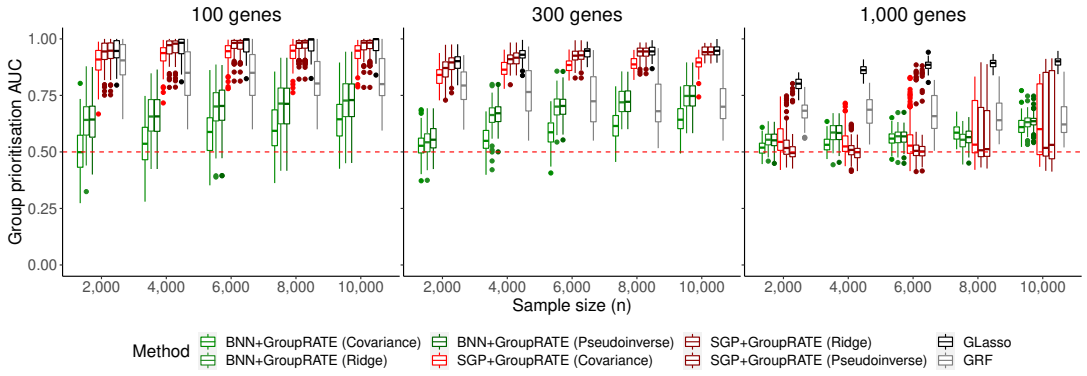
7 Real data applications from computer vision

Following these two simulation studies, this section now demonstrates how GroupRATE can be used to interpret a last layer Bayesian neural network trained on real data. However, the evaluation of real data results is challenging as ground truth labels indicating which groups are causal are not available. This section therefore utilises an alternative approach based on medical imaging data to demonstrate how grouped variable importance is a useful analysis in practical biomedical applications.

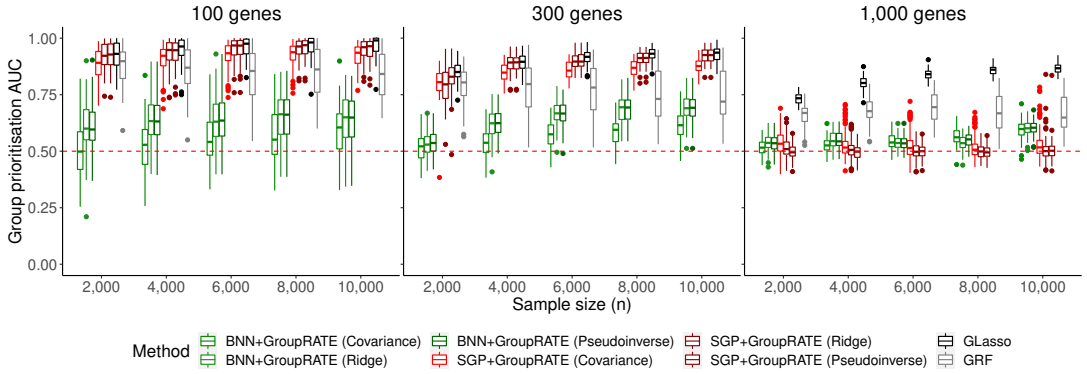
Medical images are an important type of biomedical data. Furthermore, the fact that many of the highest profile successes of deep learning have occurred in computer vision has led medical imaging researchers to adopt deep learning at a faster rate than many other biomedical fields (A. S. Lundervold and A. Lundervold, 2019).

7.1 Bayesian neural network classifier

The two simulation studies in this chapter both focused on regression tasks, but classification is far more common in computer vision and medical image-based diagnosis. Fortunately the last layer Bayesian networks can be extended to classification in a straightforward manner. In the regression case a neural network computes a mean and variance to parametrise a univariate Gaussian, from which the predicted label is sampled. For classification the network computes an un-normalised probability that parametrises a Bernoulli distribution,



(A) $h^2 = 0.6, H^2 - h^2 = 0.2$



(B) $h^2 = 0.4, H^2 - h^2 = 0.4$

FIGURE 4.12: Group prioritisation AUCs for the nine methods in the genotype simulations. The red dashed line denotes the expected performance of random group importances. SGP: sparse GP, BNN: Bayesian neural network, GRF: random forest with group importance scores, GLasso: GroupLasso.

$$y_i \sim \text{Bernoulli}\left(\frac{f(x_i)}{1+\exp(-f(x_i))}\right), \quad i = 1, \ldots, n, \tag{4.52}$$

where the un-normalised probability $f(x)$ is known as a logit. Similarly to the regression case $f(x)$ is given by

$$f(x) = w^T h_\theta(x) + b, \quad \begin{pmatrix} w \\ b \end{pmatrix} \sim q_\phi(\tilde{\theta}), \tag{4.53}$$

where $q_\phi(\tilde{\theta})$ is the mean-field Gaussian variational posterior placed over the final layer parameters $\tilde{\theta} = \{w, b\}$ and $h_\theta(x)$ is the activation of the penultimate layer that depends deterministically on the inner layer parameters θ. The features contained in $h_\theta(x)$ are learned using three convolutional layers (with 32, 32 and 16 filters) followed by three dense layers (all with size 16). The training procedure is identical to the one used for the regression networks in the previous sections.

While the response is no longer multivariate Gaussian, the choice of $q_\phi(\tilde{\theta})$ (a diagonal Gaussian) ensures that the distribution over $f(x)$ for distinct inputs is multivariate Gaussian. This means it can be targeted with (Group)RATE to calculate variable (group) importance scores.

7.2 Global variable importances for images using saliency maps

Computer vision has produced a number of local interpretability methods based on the gradient of the network output with respect to an input (see Chapter 2 Section 2.5). Throughout this chapter these local scores have been agglomerated to global scores using

$$s^{(j)} = \frac{1}{n}\sum_{i=1}^{n} s_i^{(j)}, \tag{4.54}$$

and to the group-level using

$$s^{(g)} = \frac{1}{|g|}\sum_{j \in g} s^{(j)}. \tag{4.55}$$

An implicit assumption for this agglomeration to produce meaningful results is that variables have a fixed meaning across the entire dataset. This important requirement is satisfied when the variables represent biological variables such as the expression level of a gene, but is only satisfied for image data when the images are aligned. This is illustrated in Figure 4.13. In the aligned images (plot A) the alignment of the images means that a single pixel has an approximately fixed meaning across the images (in this case a particular region of the body). However, many computer vision datasets contain unaligned images (plot B). In this example the pixels that correspond to a single feature (e.g. ears) are different from image to image. RATE and GroupRATE also require images to be aligned for the same reason as they compute a single score per pixel/group of pixels.

(A) Aligned medical images (Kermany et al., 2018)

(B) Unaligned images of cats (W. Zhang et al., 2008)

FIGURE 4.13: Global variable importance methods require images to be aligned for their results to be meaningful as they assign each pixel a global score. If images are aligned then a pixel has a fixed interpretation across the images (a region of the chest in plot A). If images are not aligned then the interpretation of a pixel varies from image to image (the position of cat ears is not fixed between images in plot B).

7.3 MNIST

The first computer vision example is the popular MNIST dataset of hand-written figures (LeCun, 1998), which contains 60,000 training images and 10,000 test images of digits 0-9. Each digit contains an approximately equal number of instances in both the training and test sets. The original images contain 28×28 pixels, but here 5 pixels of white space are cropped from each side. The resulting images contain 18×18 pixels ($p = 324$).

A binary classification task can be constructed from the MNIST data by considering odd digits as the negative class and even digits as the positive class. Figure 4.14 shows global variable importance scores for a convolutional neural network fitting the description in Section 7.1. In this simple example each pixel is in its own group, meaning that this is a RATE calculation. As in Section 5, the pixel scores are calculated using of the following methods:

- RATE with a Covariance projection;
- vanilla gradients;
- gradient$\times$input;
- integrated gradients; and
- a random forest mimic model with mean decrease Gini variable importance.

All the saliency methods are agglomerated as described in Section 7.2.

The heat maps on the diagonal of Figure 4.14 show the importance of each pixel, with redder pixels being more important. Colour bars are not included as comparing the actual values each method produces is not meaningful. The agreement between pixels rankings is of primary interest and these are shown by the scatter plots on the lower diagonal and the Spearman correlation values on the upper diagonal. The heat maps for the saliency map methods show that they are very noisy - this is a commonly observed feature of saliency maps for local importance that is also observed in this global importance setting. This makes it difficult to recognise any common features of digits in their heat maps. The random forest mimic model is at the other end of the spectrum - it places zero importance on almost all pixels with high importance placed on a small subset. The RATE scores provide a good balance

between these two extremes as they place non-zero importance on many pixel but also place very low importance on many pixels. The three saliency methods show strong agreement in their rankings (Spearman's $\rho > 0.75$). RATE shows some agreement with the saliency methods in terms of the overall pixel rankings (Spearman's $\rho > 0.3$), but it can be seen from the scatter plots that there is a higher degree of agreement when only the top-ranked pixels are considered.

While the RATE scores suggest a plausible set of important pixels under visual inspection, a quantitative assessment of these findings is also required. As this is a real dataset the ground truth importances of each pixels is not available. Ablation plots can act as a proxy to this unavailable ground truth by quantifying how important each pixel is to the model when it makes an out-of-sample prediction. An ablation plot shows the prediction accuracy as an increasingly large set of pixels in the test images are shuffled, which removes any existing dependencies between those pixels and the labels. As progressively larger subsets of pixels are shuffled the test accuracy will decrease fastest when the most relevant pixels are shuffled first, which enables a quantitative comparison of the different variable importance methods (Samek et al., 2016).

Figure 4.15 displays the ablation plot for each of the variable importance methods plus a random baseline. The RATE values lead to the steepest initial decline in test accuracy. Using vanilla gradients leads to a steeper drop in test accuracy for pixels ranked 20 to 100, but RATE then "overtakes" it. RATE is the first method to lead the network to exhibit random test performance.

7.4 Automatic diagnosis of pneumonia from chest X-rays

Distribution shift in medical imaging datasets

Binary classification tasks based on MNIST are useful for illustrative purposes but are far more straightforward than many of the problems that motivate biomedical research. Many more difficult tasks can be found in medical imaging, which is a good example of an area in which there is a large gap between the potential and realised impact of machine learning. This is evident from the massive growth in medical imaging papers related to deep learning in the past decade and the lack of actual deployment of these systems in a clinical setting (Leiner et al., 2021).

There are many reasons for this lack of clinical uptake, but one of the most pressing is the lack of generalisability of many deep learning models. Any machine learning model exhibits poor generalisation performance when it over-fits the training data, but the lack of generalisation problem is especially relevant in the medical imaging settings due to the nature of medical imaging dataset. Medical images exhibit strong "batch" effects due to the choice of imaging device and the medical centre at which the data are collected, resulting in systematic *distribution shift* between training and test data (Shad et al., 2021).

This characteristic of imaging data is exacerbated by recent work showing that convolutional neural networks learn features based on these spurious artefacts in the training set rather than learning clinically relevant signal (DeGrave et al., 2021) and a related finding that *post-hoc* local explanations do not correspond to regions identified by human experts (Saporta et al., 2021; Arun et al., 2021). Furthermore, this serious drawback cannot be fully mitigated via an external validation set as these confounding factors may be sufficiently present in an external dataset for the model

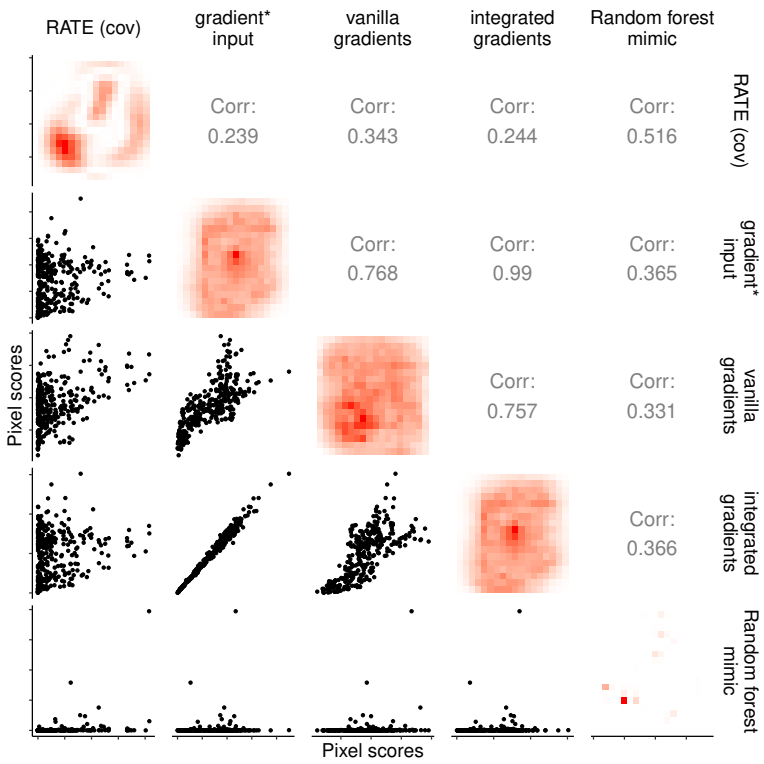


FIGURE 4.14: Pixel importances for a convolutional neural network classifying odd and even digits in the MNIST dataset. The heat maps on the diagonal show the importance of each pixel (normalised to aid comparison between methods), with darker red indicating higher importance. Lower diagonal scatter plots allow pairwise comparisons of the scores of two methods, with the corresponding Spearman correlations in the upper diagonal.

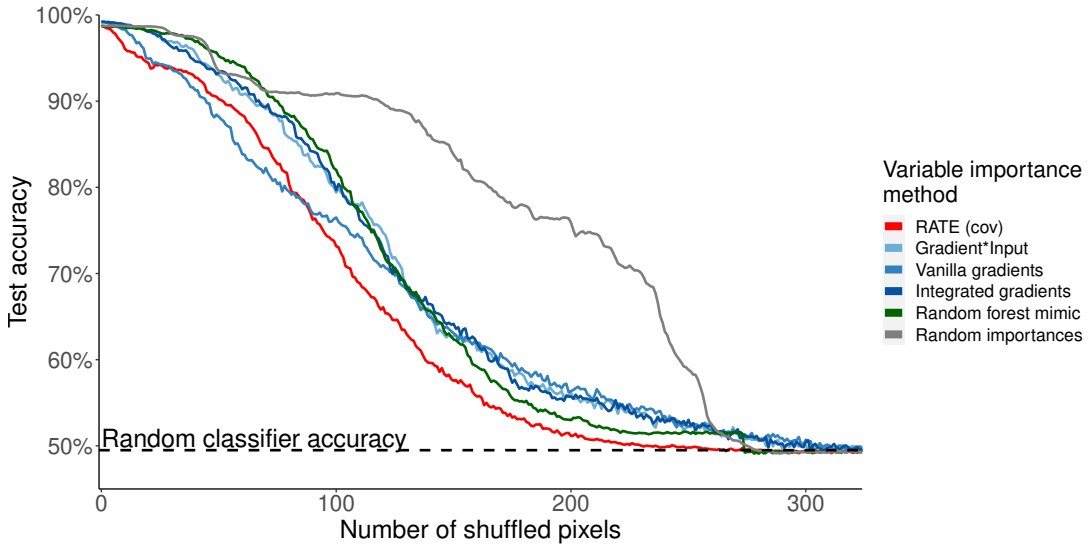


FIGURE 4.15: An ablation plot for a convolutional neural network classifying odd and even digits in the MNIST dataset. Pixels are shuffled in order of their importance, meaning that an accurate pixel ranking gives a steeper decrease in test accuracy.

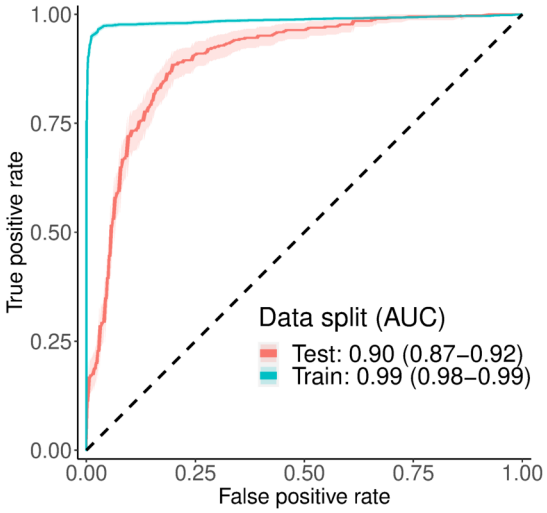


FIGURE 4.16: ROC curves for the convolutional neural network distinguishing patients with an without pneumonia from chest x-rays. Shaded areas denote the 95% confidence interval on the curves while the AUC values include 95% confidence intervals calculated using DeLong's method (DeLong et al., 1988).

to achieve reasonable predictive performance. The fact that the model is not making clinically meaningful predictions therefore passes unnoticed, with potentially disastrous consequences.

There is therefore an increasing trend to augment validation using an external test set with a saliency-style analysis when evaluating medical imaging classifiers (DeGrave et al., 2021; Shad et al., 2021). One possible approach that can aid in this evaluation is to inspect global pixel importance scores calculated on the training and test sets. If the two sets of importances are drastically different then this provides additional information on both over-fitting and distribution shift to complement estimates of generalisation performance using external validation sets. The following result demonstrates how GroupRATE can be applied in this setting.

Predictive performance of a convolutional neural network

Figure 4.16 shows the ROC curves of a last layer Bayesian convolutional neural network (as described in Section 7.1) trained on 5,232 X-rays of children, 3,883 of which had pneumonia (either bacterial of viral) and 1,349 showing healthy lungs (Kermany et al., 2018). The test set contains 234 healthy images and 390 pneumonia images. These are the same training-test splits used in the original paper. The original images were downsampled to 200×200 pixels ($p = 40,000$) in order to improve the alignment across images and reduce the computational burden. The ROC curves in Figure 4.16 indicate that the model has good generalisation performance (the test AUC 95% confidence interval is 0.87-0.92), however as has been outlined in the above paragraphs this is not sufficient to merit a clinical deployment.

Defining groups based on pixel correlation

A subsample of the training images are shown in Figure 4.18(A-B). Interpretation of medical images is typically performed using regions of neighbouring pixels as these

provide a more natural level for interpretation and often correspond to anatomical features. These regions are often computed automatically with *post-hoc* domain expert curation (Wehenkel et al., 2018). Here, this scenario is emulated by clustering the matrix of correlation distances with elements

$$d_{ij} = 1 - \text{cor}(x^{(i)}, x^{(j)}), \quad i, j = 1, \ldots, p, \tag{4.56}$$

where $x^{(i)}, x^{(j)} \in [0,1]^p$ are normalised pixel intensities and $\text{cor}(\cdot,\cdot)$ computes the Pearson correlation. The set of groups $\mathcal{G} = \{g_1, \ldots, g_{|\mathcal{G}|}\}$ are defined such that the minimum Pearson between members of a group is 0.7, which results in 744 groups (see Figure 4.17(C)).

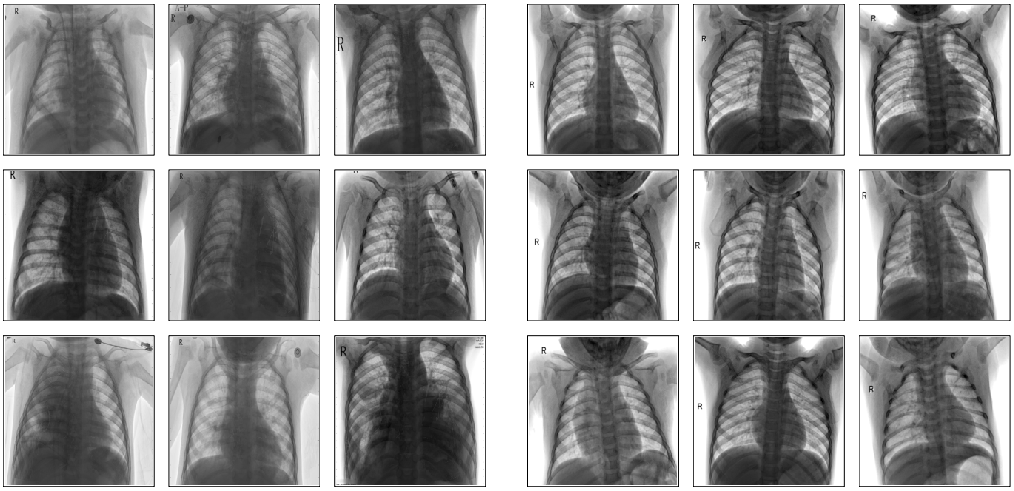
GroupRATE suggests different explanations on training and validation data

The GroupRATE values for this set of groups are shown in Figure 4.18 for the training (plot A) and test sets (plot B). The two simulation studies presented results with GroupRATE calculated on the training data only. These values are calculated using the Covariance projection. The agreement between them is shown in Figure 4.18(C), which show that the greatest degree of difference between the two sets of importances is at the top of the rankings. This is clearly a concern as it suggests that the features learned in training are not the same as those that are driving the impressive predictive performance. In practice these importance scores could be used to search for biases in the training data that are driving the model predictions or be passed to domain experts to establish which of the two rankings (if either) is closer to clinically relevant regions of the lung.

8 Discussion

This chapter proposed GroupRATE - a novel extension to the RATE method to the setting of grouped variables - and showed how it can be used to compute grouped-variable importance scores for a last layer Bayesian neural network. The ability of GroupRATE to identify causal groups was investigated using two sets of simulation studies. The first set of simulations compared GroupRATE with alternative *post-hoc* methods for computing grouped variable importance for a Bayesian neural network. The second simulation used human genotype data with the aim of identifying causal genes associated with a continuous phenotype. This second simulation setup did not assume that a Bayesian neural network was the most appropriate model and so included three alternative models (sparse GP regression, GroupLasso and random forest) for which grouped variable importances can be calculated. Finally, GroupRATE was used to identify potential data biases in a Bayesian convolutional neural network trained to diagnose pneumonia from x-ray images.

In the first set of simulations GroupRATE was able to effectively identify causal groups using all three projections, but the Ridge and Pseudoinverse projections resulted in better performance (larger AUCs), especially as the number of groups increased. However, other methods based on saliency maps (vanilla gradients and smooth gradients) showed similarly good performance, both in terms of their group prioritisation AUC and their true positive rate in the high-sensitivity region of the ROC curve. Methods based on training mimic models and other saliency maps (such

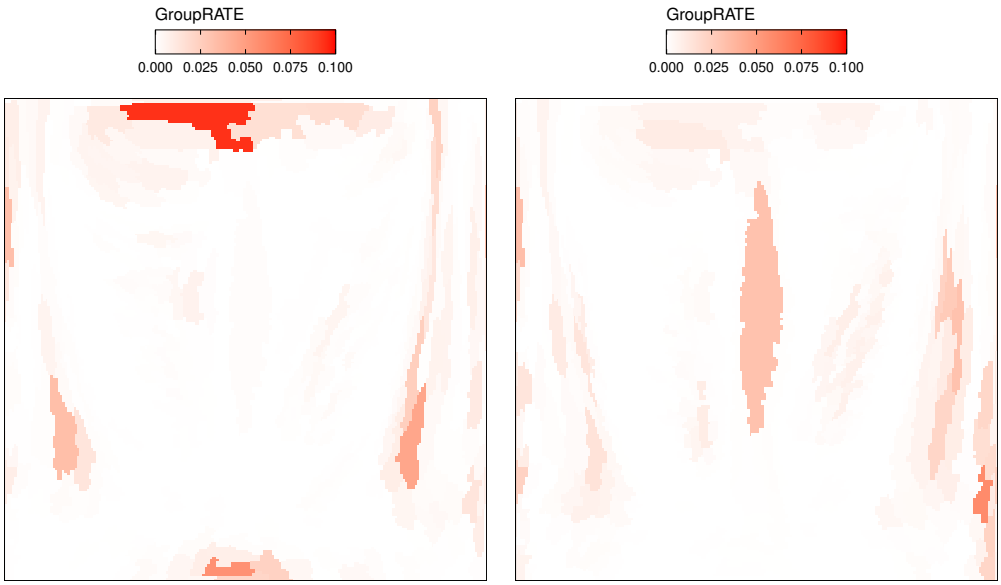


(A) Pneumonia group

(B) Control group

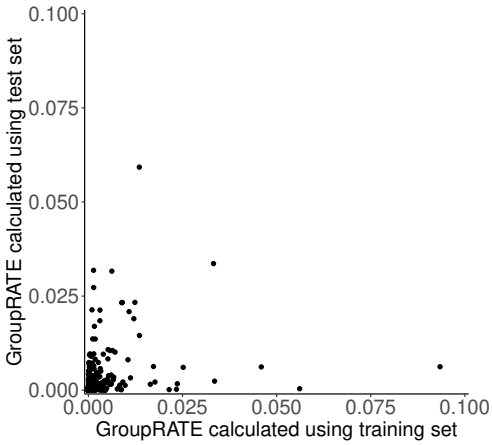
(C) Pixel groups used for the GroupRATE calculation.

FIGURE 4.17: Training examples from the pneumonia (A) and control (B) groups. For GroupRATE pixels are clustered into 744 groups based on their Pearson correlation coefficient (C). The minimum correlation within a cluster is 0.7.



(A) Calculated on training set

(B) Calculated on test set



(C) Spearman's $\rho = 0.62$.

FIGURE 4.18: GroupRATE values calculated on the training (A) and test (B) sets prioritise different groups. The top-ranked groups differ greatly between the two sets of GroupRATE values (C).

as gradient×input) were less able to prioritise causal groups. These results showed that, given a Bayesian neural network, GroupRATE is a useful tool for *post-hoc* interpretation via grouped variable importance. However, the fact that several of the saliency-based methods also showed strong group prioritisation performance suggests that they may be well-suited to global importance. Saliency maps are usually applied for local importance analysis, where they are commonly criticised in the literature for being noisy and invariant under data or model parameter randomisation (Adebayo et al., 2018), as well as other shortcomings (described in Chapter 2, Section 2.5). To the best of my knowledge there are no systematic evaluations of saliency maps for global importance analysis, but the strong performance of agglomerated saliency maps in these simulations suggest this may be a promising avenue for research.

The first set of simulations assumed that the decision has already been made to use a last layer Bayesian neural network as the predictive model. This is becoming an increasingly common scenario as neural networks become more popular for datasets of this size ($n \gtrapprox 10^5$). However, the second set of simulations showed that in other settings (namely, a genetics problem with $n \leq 10^4$) both the sparse GP regression and GroupLasso models had better predictive performance than the Bayesian neural network. This is because the sample size is small relative to the datasets where neural networks typically exhibit strong predictive performance, but also because of the discrete nature of the covariates. In the second simulation the covariates are genotypes (encoded as 0,1,2) and this discretisation of the input space means it is more difficult for the neural network to interpolate the training data. This relatively poor predictive performance was mirrored by low AUCs in the group prioritisation task when GroupRATE was applied to the Bayesian neural network. However, the group prioritisation AUCs were larger when GroupRATE was used with the sparse GP regression model and were competitive with the best-performing model.

GroupLasso was the best model for group prioritisation in the second set of simulations despite the fact that it assumes a linear relationship between the SNPs and the phenotype. This can be explained by considering the effect of the strong hierarchy assumption that was built into the simulations. Under strong hierarchy any variable with an interaction effect also has a main effect, which GroupLasso will be able to detect. This means that the GroupLasso group prioritisation performance is likely to always be strong in such settings, even if the predictive performance of the model is weaker than non-linear alternatives. This demonstrates the different requirements of prediction and variable/group selection - strong predictive performance requires assigning the correct weight to a variable, while variable selection only requires assigning a non-zero weight to a causal variable.

Like all simulation studies, the two studies described in this chapter had several weaknesses. In both studies the density of causal groups (genes) and the density of causal variables (SNPs) are both fixed. A natural extension is therefore to investigate the behaviour of the different methods when the density of the causal groups changes. Another limitation of the genotype simulations is that the confounding role of population structure (relatedness between individuals) is not explicitly considered via the inclusion of the principal components (A. L. Price et al., 2006). While the effect of population structure is relatively minor in the WTCCC genotype data as all patients are of European ancestry (WTCCC et al., 2007), this discriminatory inclusion criterion has serious ethical implications (Peterson et al., 2019).

While traditional differential expression/abundance analyses focus on detecting the

changes in mean values between two groups it is becoming increasingly recognised that relevant biological changes can lead to a change in expression variation (Ran and Daye, 2017; Jong et al., 2019). This suggests a possible avenue of future work in which GroupRATE is targeted to explain the noise variance output $\sigma^2(\cdot)$ of a Bayesian neural network that has been trained to predict gene expression levels.

Another major limitation of these types of automated simulation study with a large number of replicates is that it is difficult to automate the real process of fitting non-parametric predictive models in a large number of replicates. For Bayesian neural networks this is particularly difficult as their training procedure is highly sensitive to parameter initialisations and optimiser hyperparameters, which necessitates a large amount of hand-tuning by the practitioner. While this can be mitigated by an automated hyperparameter search the computational cost quickly becomes infeasible. The training procedure is especially sensitive when the prediction task is more difficult (due to a smaller ratio n/p or a large number of non-linear effects). The genotype simulation setup therefore favours approaches such as GroupLasso with simpler (convex) training procedures as these are both more robust and easier to automate.

As GroupRATE was the main focus of this chapter the impact of the last layer Bayesian network construction was not considered in detail. However, there are many aspects of the model construction that could have a large impact on the both the predictive power of the model as well as the performance of the group prioritisation methods. For example, a standard normal prior was used throughout this chapter as is commonplace for networks trained with mean-field variational inference.

This study was motivated by the increasing popularity of non-parametric predictive modelling (particularly neural networks) in the biomedical literature and addressed their major drawback in such settings - their lack of interpretability. As long as such models continue to be applied in areas where interpretability is a requirement *post-hoc* methods such as Group(RATE) will be required. However, it is an open question as to whether attempting to interpret neural networks is the most appropriate approach if interpretability is a central requirement, despite their impressive predictive power (in the right setting) (Rudin, 2019). This is part of the motivation of the genotype simulations, which compared GroupRATE's interpretations of the Bayesian neural network with a range of alternative models and found that an interpretable, linear approach (GroupLasso) was the best option. Rudin (2019) suggest that a better modelling approach is to place interpretability at the heart of model building from the beginning of a project. Fortunately, Bayesian neural networks offer this possibility via Bayesian variable selection, which can be used to induce sparsity in the inputs via sparsity inducing priors (Bergen et al., 2020) or from known biological annotations (Demetci et al., 2021). However, this approach is extremely challenging for problems with very little *a priori* knowledge and so *post-hoc* interpretation methods are likely to remain popular for the foreseeable future.

Chapter 5

Modelling phylogeny in microbial datasets using string kernels

One of the defining characteristics of 16S rRNA (ribosomal ribonucleic acid) datasets is the phylogenetic relationships that exist between taxa. However, this important aspect is commonly ignored when performing analyses such as two-sample testing and supervised learning. This chapter describes how kernel methods can be used to model these phylogenetic relationships in both these settings in a unified framework. This is done by utilising the power of string kernels, which were originally developed for protein classification and natural language processing tasks. This study is the first to demonstrate their utility for modelling phylogeny in 16S rRNA datasets.

1 Chapter aims and contributions

Microbial datasets, such as those collected via 16S rRNA gene sequencing, are driving our rapidly increasing understanding of the role of the microbiome in human health. The variables in a 16S rRNA dataset represent separate organisms (taxa), which are related to one another via historical evolutionary relationships (phylogeny). These phylogenetic relationships distinguish 16S rRNA datasets from those generated using other sequencing modalities and so phylogeny-aware statistical tools are required to analyse them.

This chapter presents a simulation-based investigation of string kernels (a kernel function that operates on pairs of strings) applied to two important statistical problems in microbial datasets: (i) the kernel two-sample test and (ii) host-trait prediction using Gaussian process (GP) regression. Its contributions are

- the first application of string kernels to model phylogeny in 16S rRNA datasets;
- demonstrating that string kernels induce a more appropriate kernel two-sample test than kernels that only model taxa abundance; and
- showing that a Bayesian hypothesis test involving GP regression models with a string kernel can identify how the effects of taxa on host phenotype are distributed across the phylogenetic tree.

Existing kernel-based approaches for two-sample testing seek to associate bacterial community composition with an external variable via mixed effects models rather

than performing a true kernel two-sample test (N. Zhao et al., 2015 and its extensions). The single exception by Banerjee et al. (2019) applies a kernel two-sample test but neglects the importance of phylogenetic relationships. When existing studies do include phylogeny, they only consider the UniFrac kernel (see Section 6) rather than the string kernels used here. Furthermore, the use of kernel-based methods for host-trait prediction has so far excluded Gaussian process regression. The contributions of this chapter therefore address these gaps in the literature by presenting a detailed study of the role of phylogeny in the kernel two-sample test and Gaussian process regression for the first time. In addition, its findings are also relevant to existing kernel-based approaches for microbial data as they reveal important characteristics of the reproducing kernel Hilbert spaces on which they are based.

The chapter is structured as follows. Section 2 outlines the elements that comprise a microbial dataset and how they are collected. Section 3 describes common approaches to simulating microbial datasets and outlines the approach utilised in these simulations. Section 4 describes the kernel two-sample test before Section 5 discusses relevant studies from the literature. Section 6 describes the phylogenetic kernels used in this chapter (including string kernels). The simulation studies begin in Section 7, which presents a study demonstrating the utility of string kernels for a phylogenetically-aware two-sample test. Section 8 presents an additional simulation study in the context of host-trait prediction using GP regression.

2 Characteristics of 16S rRNA datasets

2.1 Essential components of microbial datasets

Before proceeding with the analysis this section will briefly restate the most relevant points from the description of 16S rRNA datasets in Chapter 1 (Section 1.3), with a focus on the characteristics of microbial datasets that motivate this study.

A combination of financial cost and the difficulty in obtaining sufficient biomass means it is often impractical or even impossible to perform whole genome sequencing of the organisms that comprise microbial communities. The 16S rRNA gene region is part of the bacterial genome that contains both conserved regions (which are useful for designing primers to amplify the sequence) and variable regions (that are used to identify taxa). Its sequence is therefore used to both identify organisms and quantify their abundance in a sample, as well as to infer the evolutionary relationships that exist between the different organisms.

A processed 16S rRNA gene sequencing dataset consists of three elements:

- a count matrix $X \in \mathbb{Z}_{\geq 0}^{n \times p}$, where $\mathbb{Z}_{\geq 0} = \{0, 1, 2, \ldots\}$ are the non-negative integers, containing n samples and p operational taxonomic units (OTUs);
- a set of host phenotypes (observable traits); and
- a phylogenetic tree describing the evolutionary relationships between the p OTUs.

Recall that each OTU represents a set of organisms with a mutual 16S rRNA sequence similarity above 97%. The ij^{th} element of X contains the number of sequences assigned to OTU j in sample i.

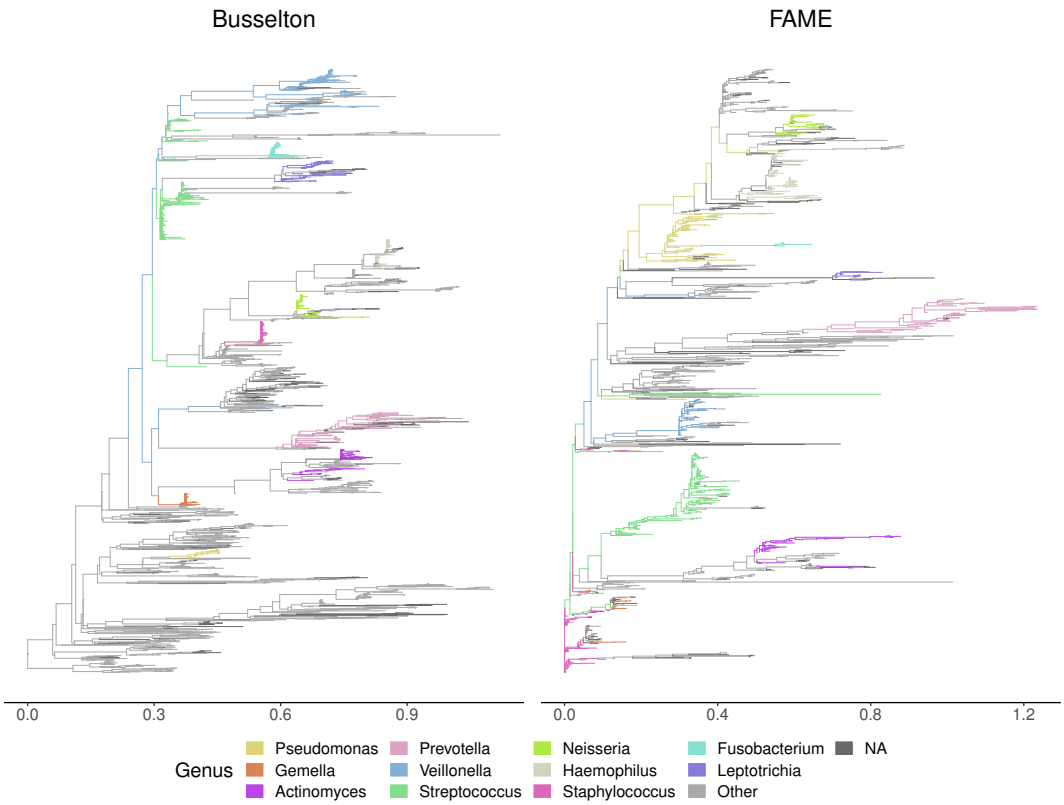


FIGURE 5.1: Phylogenetic trees for the Busselton and FAME OTUs. Branches are coloured by genus, with any genus containing fewer than 50 OTUs marked as *Other*. Trees are inferred using FastTree2 (M. N. Price et al., 2010).

2.2 Phylogenetic trees

Phylogenetic trees encode the evolutionary relationships between the OTUs in a single dataset. They are connected, acyclic graphs $G = \{V, E\}$, whose nodes $V = \{V_{\text{leaf}} \cup V_{\text{internal}}\}$ are the union of two disjoint sets: (i) leaf nodes representing observed taxa and (ii) internal nodes representing imputed common ancestors (M. N. Price et al., 2010). The edges E represent evolutionary distances inferred from the representative sequences of each OTU, as the differences between sequences are the result of mutations over time. Phylogenetic trees are typically rooted and directed, where the root node is the most recent common ancestor of all OTUs. The phylogenetic trees for the two datasets utilised in this chapter (Busselton and FAME, described in Section 3.2) are displayed in Figure 5.1.

For an internal node $v \in V_{\text{internal}}$ the subset of nodes $U_v \subset V$ descended from v is the clade induced by v. All the nodes in U_v have v as their most recent common ancestor. Clades are an important concept in evolutionary biology as they define a set of taxa that are more closely related to one another than to taxa outside of their clade. Each of the seven major taxonomic ranks (see Chapter 1, Section 1.3) defines a set of disjoint clades $\mathcal{C} = \{U_1, \ldots, U_{|\mathcal{C}|}\}$. These clades are marked at the genus level in Figure 5.1.

2.3 Interpreting operational taxonomic units

OTUs are the variables in 16S rRNA datasets so it is clearly important to be able to interpret what they mean and understand and differences between them when running statistical analyses. However, this can be challenging due to a severe lack of prior biological knowledge at the OTU level. This is because this type of measurement of microbial communities has only become routine in the past decade. When such knowledge does exist it is often not at OTU-level resolution. For example, a previously identified association may not apply to every member of a genus. These challenges are compounded by the fast rate of bacterial evolution, which means that the references databases used to name OTUs often contain multiple sequences per species.

OTU names in 16S rRNA datasets are highly redundant (e.g. $Pseudomonas_1$, $Pseudomonas_2$, $Pseudomonas_3$, ...) with hundreds or even thousands of indistinguishable names. While these organisms are assigned to distinct OTUs it is not possible to establish whether they are truly biologically or functionally distinct or whether the differences in their sequences is trivial. The fast rate of evolution also means that functionally or biologically equivalent organisms may be assigned to different to OTUs in two different samples. This is the main reason that the agglomeration to higher-level taxonomic ranks (as was done in Chapter 3) is such a popular strategy.

In summary, the variables in microbial studies have complex phylogenetic relationships that go beyond more traditional statistical challenges such as collinearity and zero-inflation (although these factors also exist in the data). Due to technical effects of the sequencing modality and data preprocessing, it is possible (or even likely) that functionally and evolutionarily equivalent microbes are assigned to separate OTUs, either within or sample or between samples. It is therefore important to model these phylogenetic relationships between variables in any statistical approach that utilises a concept of distance, as treating all OTUs as independent variables ignores a crucial part of the underlying process (the evolutionary relationships).

2.4 Compositionality

In addition to phylogenetic relationships there are characteristics of 16S rRNA count data that introduce particular challenges to analysing microbial datasets. One such characteristic that is increasingly being recognised in the literature is the compositional nature of OTU counts - that they only carry relative information on the abundance of different taxa. The remainder of this section describes the basics of compositional data analysis.

There is a growing consensus that microbiome datasets are compositional in nature (Gloor et al., 2017; Quinn, Erb, et al., 2018). For any composition vector x,

$$x = (x^{(1)}, \ldots, x^{(p)}), \tag{5.1}$$

$$x^{(j)} > 0, \quad j = 1, \ldots, p, \tag{5.2}$$

$$\sum_{j=1}^{p} x^{(j)} = \kappa, \tag{5.3}$$

where each $x^{(j)}, j = 1, \ldots, p$, are called components or parts. The key feature of compositional data is that they only carry relative information. 16S rRNA data is compositional as each microbial sample is collected as a random sample of the entire bacterial population using a sequencer that has a fixed maximum capacity. As it is not possible to recover the unobserved absolute abundances from the observed abundances, it is only appropriate to draw conclusions about the ratios of two components (taxa).

A simple example is shown in Figure 5.2, which illustrates the limitations of observed abundance in the context of clustering. Figure 5.2(A) shows the absolute abundances of three clusters in a two-component system, which are perfectly separable using absolute abundance. Given a measuring device with a fixed capacity κ, the expectation of the measured values for an absolute abundance $(\tilde{x}^{(1)}, \tilde{x}^{(2)})$ is the intersection of two lines:

$$x^{(1)} + x^{(2)} = \kappa\,, \tag{5.4}$$

$$x^{(2)} = \frac{\tilde{x}^{(2)}}{\tilde{x}^{(1)}} x^{(1)}\,, \tag{5.5}$$

where (5.4) is the simplex defined by the measurement capacity and (5.5) is the line from the origin to $(\tilde{x}^{(1)}, \tilde{x}^{(2)})$. The solution to (5.4)-(5.5) is

$$x^{(1)} = \frac{\kappa}{1 + \tilde{x}^{(2)}/\tilde{x}^{(1)}}\,, \quad x^{(2)} = \frac{\tilde{x}^{(2)}}{\tilde{x}^{(1)}} \frac{\kappa}{1 + \tilde{x}^{(2)}/\tilde{x}^{(1)}}\,, \tag{5.6}$$

which depends only on the ratio $\tilde{x}^{(2)}/\tilde{x}^{(1)}$. The solutions (5.6) are displayed in Figure 5.2(B) and they illustrate two points:

1. clusters 2 and 3 have the same ratios of the two components and so cannot be distinguished using the measured abundances (which only contain relative information); and
2. the value of κ is an artefact of the measuring device and so is arbitrary.

The value of κ defines the unit of measurement of the composition. It is common practice in microbial studies to assume $\kappa = 1$ without loss of generality, in which case the components of x are relative abundances. Due to this sum constraint compositional data live in the p-simplex $\mathcal{S}^p$, where

$$\mathcal{S}^p = \{x = (x^{(1)}, \ldots, x^{(p)}) \mid x^{(j)} > 0,\ \sum_{j=1}^{p} x^{(i)} = \kappa\}\,. \tag{5.7}$$

While the value of κ is arbitrary it is desirable when analysing a set of compositions that they all live on the same simplex. The closure operation, given by

$$\text{closure}(x) = \frac{\kappa}{\sum_{j=1}^{p} x^{(j)}} x\,, \tag{5.8}$$

projects x onto the p-simplex defined by κ.

Throughout this chapter *absolute abundance* refers to the unobserved true abundances of taxa. The observed abundances are referred to as counts (or simply abundance), which contain relative information under the assumptions of compositional data analysis. Finally, *relative abundance* refers to the closure of the observed abundances using $\kappa = 1$.

Compositional data analysis (CoDA, Aitchison, 1982) provides a framework to analyse and interpret compositional data in terms of the relative importance of their components. Given that compositional data contain relative information it is natural to work with log-ratios of their components, which live in Euclidean space. One popular log-ratio is the centred log-ratio (CLR) transform, which is given by

$$\text{clr}(x) = \left(\log \frac{x^{(1)}}{g(x)}, \dots, \log \frac{x^{(p)}}{g(x)} \right), \tag{5.9}$$

where $g(x) = (\prod_{j=1}^{p} x^{(j)})^{1/p}$ is the geometric mean of the composition (Aitchison, 1982). For the CLR transform the components of x are interpreted relative to the geometric mean of the sample. One of the most widely-used approaches to compositional data analysis is to apply (5.9) to transform the observed abundances to Euclidean space and then apply standard statistical methodology. The results are then interpreted in terms of ratios (Quinn and Erb, 2020). However, the covariance matrix of the CLR-transformed data is at most rank $p - 1$ due to the sum constraint, which follows from the sum constraint as it implies that the p components are not independent (given $p - 1$ components the final component is determined by the sum constraint).

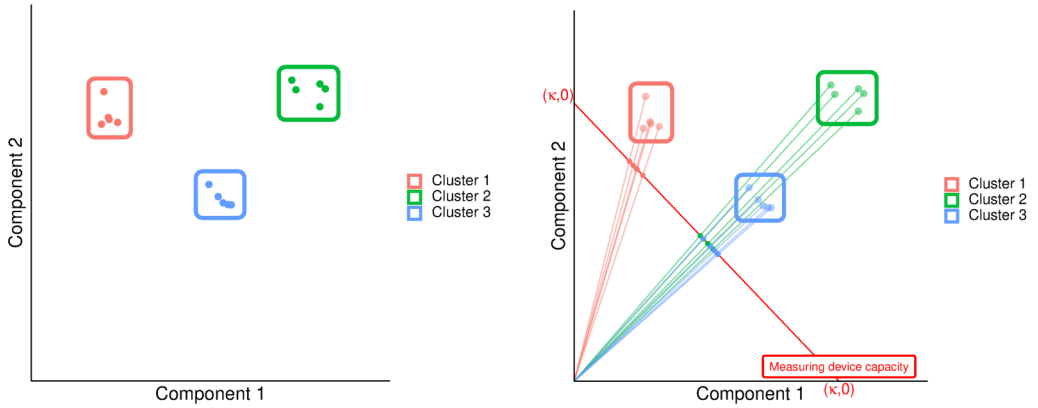
Zero-handling is an important consideration when analysing compositional data as zeros give undefined results as inputs to logarithms or denominators of ratios. Zeros in compositional data can be classed as

1. true zeros - where a component is truly missing from the composition, or
2. rounded zeros - where a component is present in the composition but is below the detection limit of the measuring device.

16S rRNA counts commonly contain up to 90% zeros (Weiss et al., 2017) and so zero-replacement strategies that preserve the structure of the underlying ground-truth (which is unavailable in practice) are required. A simulation study by Lubbe et al. (2021) compared the effect of several strategies on the reconstructed count data and its correlation/distance matrices and found that replacing zeros with values sampled from a uniform distribution on $[0.1a, a]$, for detection limit a provided a good balance of simplicity while recovering the true distance matrix with acceptable accuracy. This is the zero-replacement strategy used throughout this chapter whenever the CLR transform is applied.

3 Simulating 16S rRNA data

In order to develop novel statistical tools (or benchmark existing tools), it is important to be able to simulate realistic 16S rRNA data. These simulated datasets should capture the important characteristics of the OTU counts themselves while also including the phylogenetic relationships present in real datasets.



(A) All three clusters are separable using absolute abundance

(B) Clusters 2 and 3 are not separable using the proportions of the two components.

FIGURE 5.2: The ground truth contains three clusters that are separable using (unobserved) absolute abundance (plot A). However, Clusters 2 and 3 have the same proportions of the two components and so are not separable using the measured abundance (plot B). This is a feature of compositional data and cannot be overcome using statistical methods.

In a 16S rRNA dataset OTUs are related to one another by the phylogenetic tree, which is the result of a complex evolutionary process. The OTU abundances also have specific characteristics beyond their phylogenetic relationships. Like many types of sequence count data they are zero-inflated and overdispersed, containing a large number of low-abundance OTUs and a small number of highly-abundant OTUs. Distinguishing between true zeros, sampling zeros (which occur when the maximum capacity of the measurement device is reached) or technical zeros (zeros that occur due to another bias or limitation of the measurement procedure) is a difficult task that is beyond the scope of this study (Silverman, Roche, et al., 2020).

It is common practice in the literature to overcome these challenges by basing simulated datasets heavily on a single real 16S rRNA dataset. This is the approach utilised here. Given such an observed dataset it is possible to (i) use its phylogenetic tree and (ii) fit a parametric distribution to the observed OTU abundances in order to sample new fictitious counts. This approach avoids having to simulate an evolutionary process to generate synthetic OTUs and also removes the requirement to model the data collection and pre-processing steps, which contain many hard to quantify technical biases (Kennedy et al., 2014; Tremblay et al., 2015; Villette et al., 2021; Silverman, Bloom, et al., 2021). Host phenotypes can then be simulated from the fictitious abundances using a specified model. This approach has been utilised for microbial network inference (Kurtz et al., 2015), association testing (C. Wu et al., 2016), host phenotype prediction (X. Gao et al., 2017; Xiao et al., 2018) and differential abundance testing (Jun Chen, King, et al., 2018), as well as in dedicated 16S rRNA simulation tools (Patuzzi et al., 2019; Rong et al., 2021; S. Ma et al., 2021).

The negative binomial distribution is a popular choice for overdispersed count data but its assumptions are often violated by real sequence count datasets, including 16S rRNA (Hawinkel et al., 2020). The same study found that zero-inflated extensions (the zero-inflated negative binomial) did not significantly improve the fits to real datasets. A log-normal distribution (Fernandes et al., 2014; Prost et al., 2021) and its

zero-inflated counterpart (Ai et al., 2019) have also been used to model count data, motivated by ideas from compositional data analysis.

3.1 Dirichlet-multinomial models of OTU abundance

16S rRNA abundances are commonly modelled using the Dirichlet-multinomial (DMN) distribution for generative clustering (I. Holmes et al., 2012), association testing (La Rosa et al., 2012; C. Wu et al., 2016; Tang et al., 2017), benchmarking statistical tools (Calgaro et al., 2020) and host phenotype prediction (Jun Chen and H. Li, 2013b; X. Gao et al., 2017; Xiao et al., 2018; Koslovsky and Vannucci, 2021). DMNs have also been highlighted as the preferred distribution for molecular ecology in general and lung 16S rRNA modelling in particular (De Valpine and Harmon-Threatt, 2013; Harrison et al., 2020). They have also been shown to result in accurate inference for different zero-inflation models (Silverman, Roche, et al., 2020) and various degrees of compositionality effects (Fernandes et al., 2014). This chapter follows these studies by using DMNs to generate fictitious but realistic OTU counts.

The DMN(N, α) is a compound distribution over non-negative integers $\mathbb{Z}_{\geq 0}$ that is parametrised by a vector of concentrations $\alpha = (\alpha_1, \ldots, \alpha_K) \in \mathbb{R}_+^K$ and $N \in \mathbb{Z}^n$ trials, where K is the number of categories (Mosimann, 1962). A sample $x \in \mathbb{Z}_{\geq 0}^K$ is modelled as

$$\theta \sim \text{Dirichlet}(\alpha), \quad \alpha_1, \ldots, \alpha_K > 0, \tag{5.10}$$

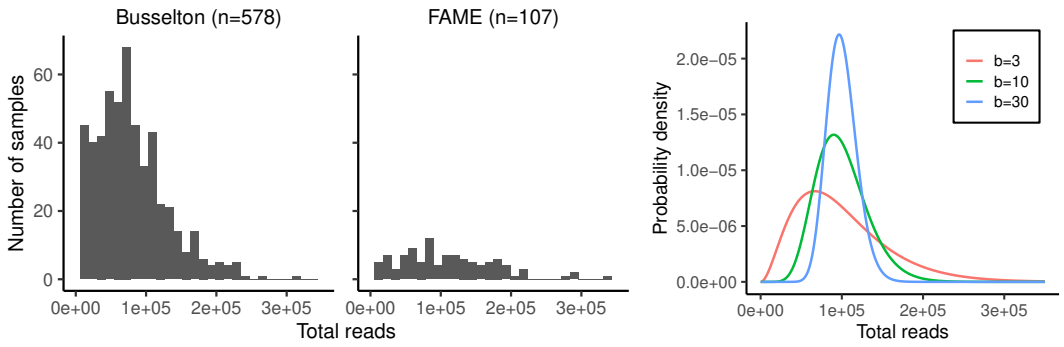
$$x \sim \text{Multinomial}(N, \theta), \quad \sum_{j=1}^{K} \theta_j = 1, \tag{5.11}$$

where $\theta_j, j = 1, \ldots, K$ are multinomial probabilities. The multinomial probabilities $\{\theta_j\}_{j=1}^K$ are constrained to the p-simplex and therefore incorporate compositional effects, while the subsequent multinomial sampling step simulates the observed counts.

In microbiome applications the number of categories K corresponds to the number of OTUs (p), while the number of trials N is the total number of reads per sample. Maximum likelihood estimates of α from two real lung microbiome datasets are used to generate fictitious OTU abundance tables in subsequent sections of this chapter. The datasets are described in Section 3.2.

Modelling a variable number of reads per sample

The number of trials $N \in \mathbb{Z}^n$ can itself be modelled as a random variable to emulate the common scenario where different samples contain different numbers of reads, with a negative binomial being a popular choice (Xiao et al., 2018). Throughout these experiments N is drawn from a negative binomial $N \sim \text{NB}(a, b)$, where a is the mean and b the dispersion. This is the standard parametrisation of the negative binomial in ecology (Lindén and Mäntyniemi, 2011). Figure 5.3(A) shows the empirical reads per sample in the two datasets while Figure 5.3(B) shows the negative binomial distributions used to simulate the total reads per sample in these simulations, which fix $a = 10^5$ and use $b \in \{3, 10, 30\}$. Smaller values of b result in datasets where the reads per sample are more left-skewed.



(A) Observed numbers of reads per sample in the two datasets.

(B) Negative binomials with different values of the dispersion, b

FIGURE 5.3: 16S rRNA commonly exhibit variable numbers of read per sample (plot A). This is emulated in the simulated datasets by modelling the number of reads per sample, N, as being drawn from a negative binomial NB(10^5, b) with different values of the dispersion parameter b (plot B).

3.2 Real datasets

The two simulation studies in this chapter (on the kernel two-sample test and host trait prediction using Gaussian process regression) use two real lung microbiome datasets to (i) generate realistic OTU counts via Maximum Likelihood estimates of the DMN concentrations and (ii) obtain realistic phylogenetic relationships from their inferred phylogenetic tree. The characteristics of the original datasets are displayed in Table 5.1 . The collection and pre-processing of these datasets is not part of the contributions of this chapter and was performed by collaborators.

TABLE 5.1: Real lung microbiome datasets used in this chapter to simulate OTU abundances. The phylogenetic tree is also used in order to have realistic phylogenetic relationships between the OTUs. The original study groups are not used in the simulations but are included here for completeness.

Dataset name	n	p	Original study groups	Citation
FAME (bacterial)	107	1,189	Cystic fibrosis and non-cystic fibrosis bronchiectasis	Ish-Horowicz, Cuthbertson, et al. (2022)
Busselton	578	1,689	Asthma and healthy controls	McBrien (2020)

4 Kernel two-sample testing for 16S rRNA data

A key research question in microbial studies in biology is to determine whether two groups of samples are drawn from distinct distributions (the two-sample test). In most cases the two groups correspond to disease or treatment groups and it is of

interest to establish whether the two groups have distinct microbial communities. Given two sets of samples $X = \{x_i\}_{i=1}^{n_x}$ and $Y = \{y_i\}_{i=1}^{n_y}$, where

$$X \sim P\,, \quad Y \sim Q\,, \tag{5.12}$$

the two-sample (hypothesis) test is

$$H_0 : P = Q\,, \quad H_1 : P \neq Q\,, \tag{5.13}$$

where H_0 and H_1 are the null and alternative hypotheses. Univariate two-sample testing forms the basis of differential expression/abundance analysis, which is a cornerstone of biological research (Soneson and Robinson, 2018; Nearing et al., 2022). Such a univariate test on the means of P and Q aims to identify individual taxa (in microbiome applications) that are significantly differentially abundant between the two groups. However, there may not be a significant difference between two populations even if individual taxa are differentially abundant between them. A multivariate test is therefore required to establish if there are community-level differences between the two groups.

As discussed in Section 3, microbiome samples are high-dimensional, zero-inflated and likely to violate parametric assumptions. These characteristics presents statistical challenges such as the curse of dimensionality and sparsity, in addition to challenges due to the complex nature of the underlying biology (the phylogenetic relationships between variables). The maximum mean discrepancy (MMD, Gretton et al., 2012) measures the distance between two distributions in an inner product space defined a kernel function $k(x, x')$. Using MMD for two-sample testing is therefore an appealing option for microbiome studies as it is non-parametric, well-suited to high-dimensional data and provides a simple way of encoding prior domain knowledge. The behaviour of the test can therefore be carefully controlled by the choice of kernel function. This section briefly outlines the mathematical background of kernel-based two-sample testing and motivates the use of kernels that model the phylogenetic similarity between OTUs.

4.1 Kernel mean embeddings

Chapter 2 (Section 1.1) described the kernel trick, which enables the use of arbitrarily complex feature mappings by specifying a kernel function. A valid kernel function $k(\cdot, \cdot)$ satisfies

$$k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}\,, \tag{5.14}$$

for a feature map $\phi : \mathcal{X} \to \mathcal{H}$ which induces the reproducing kernel Hilbert space (RKHS) $\mathcal{H}$. In Chapter 2 the power of the kernel trick was illustrated with an example from binary classification, where applying an appropriate kernel function caused a non-linear boundary between two classes to become linear. In this setting (the two-sample test) the implicit mapping using $\phi(\cdot)$ on each data point is generalised to allow an entire distribution to be mapped to a point in an RKHS. These so-called kernel mean embedding of a distribution P is given by

$$\mu_P = \mathbb{E}_{x\sim P}[\phi(x)]\,, \tag{5.15}$$

and is the expectation of the distribution in $\mathcal{H}$, the RKHS induced by $\phi(\cdot)$ (Smola et al., 2007). By representing P as an element of $\mathcal{H}$ it is possible to compute common operations such as inner products and distances via μ_P. Furthermore, the properties of $\mathcal{H}$ are fully determined by $\phi(\cdot)$, which is specified using $k(\cdot,\cdot)$ due to the kernel trick.

4.2 Maximum mean discrepancy

For two distributions P and Q their distance in $\mathcal{H}$ is simply the distance between their embeddings,

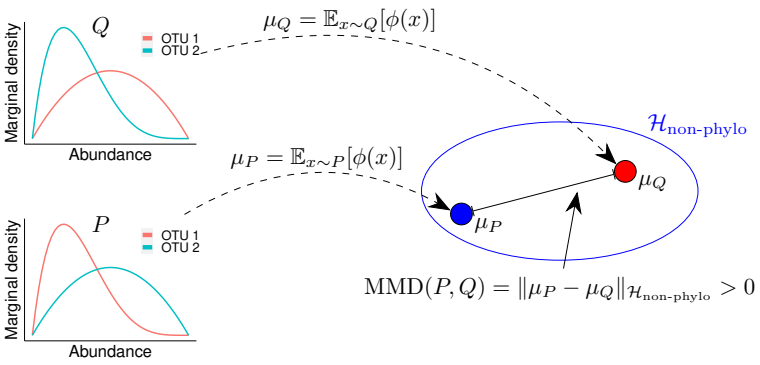
$$\begin{aligned} \mathrm{MMD}(P,Q) &= \|\mathbb{E}_{x\sim P}[\phi(x)] - \mathbb{E}_{y\sim Q}[\phi(y)]\|_{\mathcal{H}} && (5.16)\\ &= \|\mu_P - \mu_Q\|_{\mathcal{H}}\,, && (5.17)\end{aligned}$$

which is called the maximum mean discrepancy (MMD, Gretton et al., 2012). The kernel two-sample test uses $\mathrm{MMD}(P,Q)$ as the test statistic and assesses statistical significance using a permutation test.

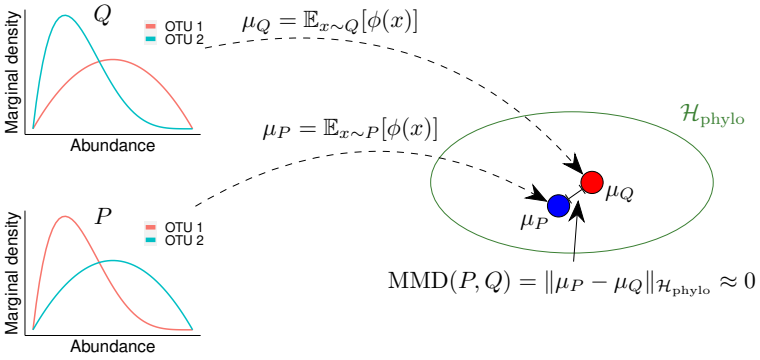
An important element of MMD two-sample testing is therefore the choice of kernel function as it determines the properties of $\mathcal{H}$ and so the behaviour of (5.16). The choice of kernel function provides a simple mechanism by which to encode complex prior knowledge. Furthermore, kernels are well-suited for the analysis of discrete data structures (such as trees and strings), which makes them an attractive option for modelling 16S rRNA datasets.

As a simple example consider the linear kernel $k(x,y) = x^T y$, for which the feature map is $\phi(x) = x$. In this case the $\mathrm{MMD}(P,Q) = \sqrt{(\mathbb{E}_{x\sim P}[x] - \mathbb{E}_{y\sim Q}[y])^2}$, which is zero if and only if P and Q have different means. A linear kernel therefore induces a test on the means of P and Q, while tests with higher order polynomial kernels test for differences in the higher-order moments. $\mathrm{MMD}(P,Q)$ therefore computes the difference in the feature means of P and Q for the feature means are defined by $\phi(\cdot)$.

For *characteristic* kernels $\mathrm{MMD}(P,Q)$ is equal to zero if and only if $P = Q$. This result follows from the fact that the mean embeddings of P and Q in $\mathcal{H}$ (μ_P and μ_Q) are injective when $\mathcal{H}$ is defined by a characteristic kernel (Gretton et al., 2012). Two of the most popular kernels, the radial basis function and Matern class, are characteristic. The feature map for these two kernels are infinite-dimensional and so are guaranteed to detect a difference between P and Q unless all possible moments are the same. However, these simulations will show that two-sample test with a characteristic kernel is not well-suited to 16S rRNA datasets.



(A) Non-phylogenetic, characteristic kernel



(B) Phylogenetic kernel

FIGURE 5.4: Visualisation of the kernel mean embeddings of two distributions P and Q, where each contain the marginal densities of two indistinguishable OTUs. A characteristic kernel leads to a large MMD (plot A) but if the kernel models phylogenetic relationships it correctly finds that the distance between P and Q is small (plot B).

4.3 Kernels for 16S rRNA two-sample testing

As outlined in Section 2.3 interpreting OTUs can pose a challenge in 16S rRNA datasets due to the redundancy in their identification. This leads to a large number of OTUs that are effectively indistinguishable - they may be functionally or biologically equivalent but any difference cannot be established using 16S rRNA data. This has clear implications for the two-sample test as an appropriate test should not reject H_0 on the basis of differences that are below the resolution of the sequencing modality.

One way of constructing such a test is by using a kernel that explicitly models phylogenetic similarity in a kernel two-sample test. Some such kernels are described in Section 6 but here the focus is on the desirable characteristics of the RKHS they induce using two toy examples. The first is illustrated in Figure 5.4. Both P and Q consist of the marginal distributions for two OTUs that represent biologically indistinguishable organisms. In plot A the two-sample test is performed using a non-phylogenetic, characteristic kernel, which results in a large value for $\text{MMD}(P, Q)$ that would likely lead to a rejection of H_0. In plot B a kernel that models phylogeny is used, which induces an RKHS in which $\text{MMD}(P, Q)$ is close to zero, which is clearly preferable. Note that such a phylogenetic kernel is not characteristic by definition as the kernel mean embeddings are surjective by design.

For a slightly more complex example, consider a toy dataset of four OTUs with the marginal OTU distributions shown in Figure 5.5. The populations P and Q are constructed such that

$$P = (p_1(x), p_0(x), p_2(x), p_0(x)) \,, \tag{5.18}$$
$$Q = (p_0(x), p_1(x), p_0(x), p_2(x)) \,, \tag{5.19}$$

where $p_1(x)$ and $p_2(x)$ are marginal probability mass functions that place some mass on $x > 0$ and satisfy $p_1(x) \neq p_2(x)$, while $p_0(x)$ places all its mass at zero. Now consider two scenarios where:

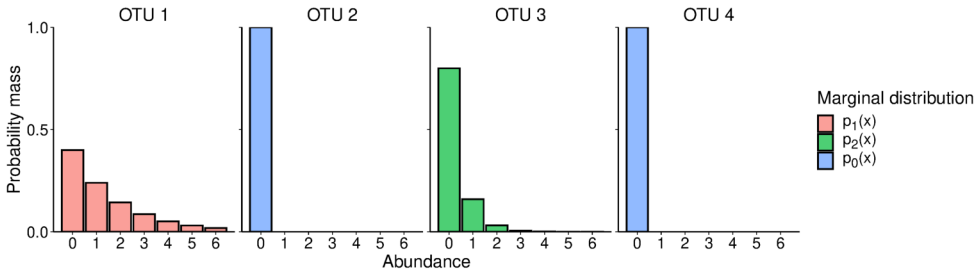
1. all four OTUs are biologically distinct; and
2. the phylogenetic relationships between the four OTUs are as follows:
   - OTU 1 has 96% sequence similarity to OTU 2;
   - OTU 3 has 96% sequence similarity to OTU 4; and
   - OTUs 1 and 2 are distantly related to OTUs 3 and 4.

In the first scenario it is clear that $P \neq Q$ and that a two-sample test should reject H_0 (assuming a sufficiently large sample size). However, in the second scenario the dataset only contains four OTUs because a 97% similarity threshold is used to define OTUs. If the convention was to use a 96% similarity threshold then the dataset would contain only two OTUs and performing the same two-sample test on this collapsed (but biologically equivalent) two-OTU dataset would have a well-calibrated Type I error. Such a high degree of dependence of test behaviour on an arbitrary technical feature of the preprocessing pipeline (the similarity threshold) is clearly undesirable, but is unavoidable if the test is not aware of the phylogenetic relationships between the OTUs.

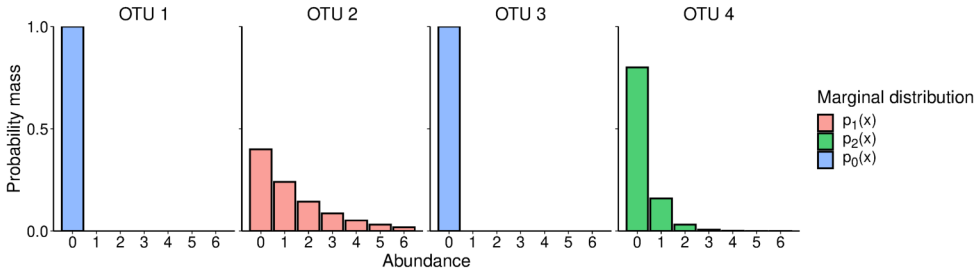
5 Relevant work

Previous applications of kernel methods to 16S data focus on two tasks: (i) host-trait prediction from bacterial community composition and (ii) differential abundance analysis. For host-trait prediction the most common approach is to combine a kernel with a support vector machine (Ning and Beiko, 2015; Jasner et al., 2021; Giliberti et al., 2022) or kernel regression (Jun Chen and H. Li, 2013a; Randolph et al., 2018; Xiao et al., 2018). The majority of studies use a radial basis function or linear kernel. Some studies also include UniFrac kernels (described in Section 6) to explicitly model phylogeny.

Kernel two-sample tests have previously been used with 16S rRNA data for testing associations between community composition and host phenotype in the MiRKAT method (microbiome regression-based kernel association test N. Zhao et al., 2015), which has also been extended to survival times (Plantinga et al., 2017) as well as to other outcome types (Koh et al., 2019). The work by Koh et al. (2019) incorporates phylogeny into its test by using the UniFrac kernel. A second method called



(A) Marginal OTU distributions in P.



(B) Marginal OTU distributions in Q.

FIGURE 5.5: Underlying marginal distributions for 4 OTUs in a toy example with two populations P (panel A and (5.18)) and Q (panel B and (5.19)). Distributions with the same colour are identical. In a scenario where OTUs 1 and 2 and OTUs 3 and 4 are biologically equivalent a two-sample test should not reject the null hypothesis.

AMDA (the adaptive multivariate two-sample test for microbiome differential analysis Banerjee et al., 2019) also uses MMD but with an additional filtering step to identify candidate taxa that are carried forward to the MMD two-sample test.

This work focuses on the behaviour of string kernels as a method for modelling phylogeny in 16S rRNA datasets. As each OTU is defined by a representative DNA sequence of ∼200 base pairs their similarity can be quantified using string kernels, which were developed in natural language processing for text classification (Lodhi et al., 2002). These kernels moved naturally from applications such as spam email detection (Amayri and Bouguila, 2009) to biological fields in which sequence data is ubiquitous. String kernels were successfully used to classify DNA and protein sequences with state-of-the-art performance in many cases when combined with support vector machines (Kuksa, 2013; Ghandi et al., 2014; Nojoomi and Koehl, 2017). In recent years the popularity of string kernels has decreased due to their lack of scalability to longer sequences and the rise of deep learning models. However, recent work developing approximate and scalable string kernels has shown that string kernels can outperform both convolutional and recurrent neural networks for protein sequence classification when used with support vector classifiers (Blakely et al., 2020).

6 Description of phylogenetic kernels

The aim of this study is to investigate the benefits of explicitly modelling the phylogenetic relationships between OTUs using kernels. This section will describe two approaches to constructing phylogenetic kernels:

- via the UniFrac distance (C. Lozupone and Knight, 2005), a popular distance metric in microbial research; and
- via three types of string kernels (Spectrum, Mismatch and Gappy pair), which measure the similarity between the representative sequences of OTUs.

As noted in the previous section, other studies have utilised the UniFrac kernel to model phylogeny in a kernel framework.

Before introducing these phylogenetic kernels it is useful to discuss the other (non-phylogenetic) kernels with which they will be compared. The first two are the radial basis function (RBF) kernel,

$$k(x, x') = \sigma^2 \exp\left(\frac{-\|x - x'\|_2^2}{2l^2}\right), \tag{5.20}$$

and the Matern32 kernel,

$$k(x, x') = \sigma^2 \left(1 + \frac{\sqrt{3}\|x - x'\|_2^2}{l^2}\right) \exp\left(\frac{-\sqrt{3}\|x - x'\|_2^2}{l^2}\right), \tag{5.21}$$

where the signal variance $\sigma^2 > 0$ and lengthscale $l > 0$ are hyperparameters. Both the RBF and Matern32 are stationary kernels as the similarity they compute is a function of $\|x - x'\|$. The fact that both kernels depend only on the Euclidean distance between samples illustrates the motivation behind using a phylogenetic kernel - the distance $\|x - x\|_2^2$ assumes that the samples are expressed in an orthonormal basis, which is clearly inappropriate given the phylogenetic relationships that define the OTUs. The non-phylogenetic linear kernel is computed using

$$k(x, x') = \sigma^2 x x'^T, \tag{5.22}$$

which is proportional to the dot product of the two samples. The Linear kernel is a special case of the String kernels that will be introduced in Section 6.2.

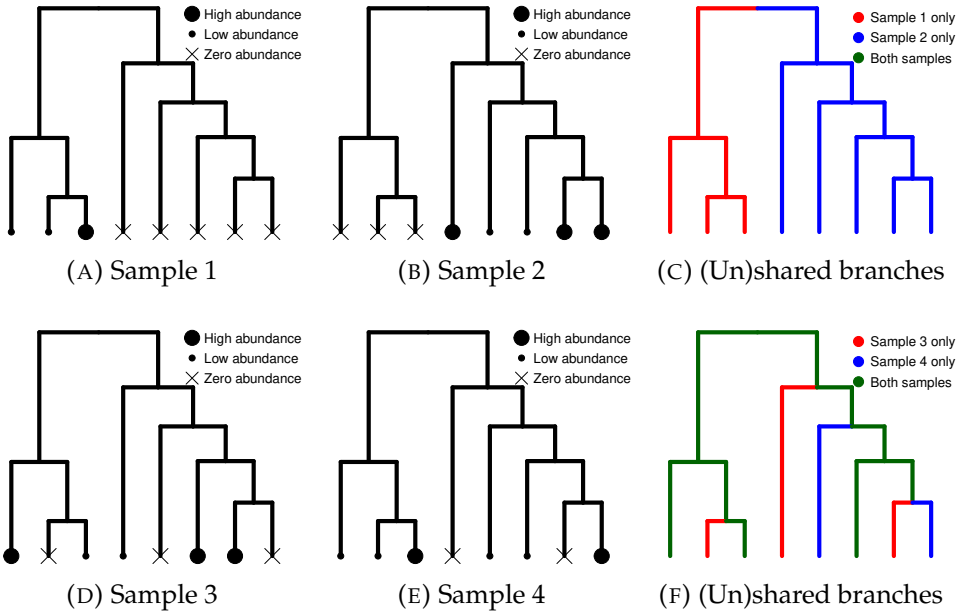
6.1 UniFrac kernel

UniFrac is a distance metric for pairs of microbial samples that incorporates the phylogenetic relationships between OTUs using the branch lengths of the phylogenetic tree (C. Lozupone and Knight, 2005). The unweighted UniFrac distance between two samples x and x' is the ratio of unshared branch lengths between the two samples to the total branch lengths in the tree,

$$d_{\text{uf-uw}}(x, x') = \frac{\sum_{j=1}^{p} l_j |\mathbb{1}(x^{(j)} > 0) - \mathbb{1}(x'^{(j)} > 0)|}{\sum_{j=1}^{p} l_j \max(\mathbb{1}(x^{(j)} > 0), \mathbb{1}(x'^{(j)} > 0))}, \tag{5.23}$$

where p is the number of taxa (OTUs) in the tree, l_j is the branch length between taxa j and the root and $\mathbb{1}(x^{(j)} > 0)$ is an indicator function for whether taxa j appears in sample x.

The weighted UniFrac distance (C. A. Lozupone et al., 2007) is weighted by the sample abundances,



(A) Sample 1 (B) Sample 2 (C) (Un)shared branches

(D) Sample 3 (E) Sample 4 (F) (Un)shared branches

FIGURE 5.6: Calculating the UniFrac distance between pairs of samples. Both the weighted and unweighted UniFrac distance between Sample 1 (A) and Sample 2 (B) is 1 as the two samples do not share any branches (C). The weighted and unweighted UniFrac distances between Sample 3 (D) and Sample 4 (E) are both less than 1, as they share some branches (F), but they will not be equal to one another as the samples have different abundances.

$$d_{\text{uf-w}}(x, x') = \frac{\sum_{j=1}^{p} l_j |x^{(j)} - x'^{(j)}|}{\sum_{j=1}^{p} l_j (x^{(j)} + x'^{(j)})}. \tag{5.24}$$

Figure 5.6 illustrates how both variants of the UniFrac distance are calculated. It is common to practice to calculate both the weighted and unweighted versions of the UniFrac distance as they can lead to different conclusions. Both the unweighted and unweighted UniFrac distances take values in [0,1], with both being equal to one when two samples do not share any non-zero OTUs. Unweighted UniFrac is equal to zero when the two samples contain identical OTUs, but weighted UniFrac further requires that the abundances are equal for its distance to be zero. Low-abundance OTUs have a smaller effect on the weighted UniFrac distance, which is dominated by OTUs with large abundances, while unweighted UniFrac is more sensitive to rare taxa. This makes unweighted UniFrac better suited to quantifying community structure.

6.2 String kernels

Each OTU is defined by a representative DNA sequence of ∼200 base pairs, which means that OTU-wise similarity can be quantified using string kernels. As discussed in the previous section, string kernels were developed in natural language processing for text classification (Lodhi et al., 2002) and have been widely applied to classify biological sequences such as proteins. However, in sequence classification tasks the samples themselves are strings, while in 16S rRNA datasets samples are count vectors whose elements (the OTUs) are related to one another by their representative

sequences. This distinction means that the string kernels in this study are used to construct an inner product space in which sample similarity is computed.

Relationship between positive-definite matrices and kernels

Given a valid kernel function $s(\cdot,\cdot)$ and a set of strings $\{z_i\}_{i=1}^{p}$ it is possible to construct a positive definite matrix S whose ij^{th} element is given by

$$(S)_{ij} = s(z_i, z_j)\,, \quad i,j = 1,\ldots,p\,. \tag{5.25}$$

Here, $s(\cdot,\cdot)$ is used to denote a kernel function that operates on pairs of variables (representative sequences of OTUs) as opposed to $k(\cdot,\cdot)$ which operates on pairs of observations. The matrix S defines an inner product $\langle x, x'\rangle_S = x'^T S x$, from which the kernel matrix XSX^T can be constructed given an $n \times p$ design matrix X. Furthermore, this kernel is satisfies

$$XSX^T = -\frac{1}{2}\mathcal{J}_n \Delta^S \mathcal{J}_n\,, \tag{5.26}$$

where $\mathcal{J}_n = I_n - \frac{1}{n}1_n 1_n^T$ is the $n \times n$ centring matrix and $\Delta^S \in \mathbb{R}_+^{n\times n}$ is a matrix of sample-wise squared distances in S (Randolph et al., 2018). The matrix Δ^S has elements

$$\begin{aligned}(\Delta^S)_{ij} &= \|x_i - x_j\|_S^2 &(5.27)\\ &= \langle x_i - x_j, x_i - x_j\rangle_S\,, &(5.28)\end{aligned}$$

where the distances are calculated in the inner product space defined by S. From (5.26) it can be seen that the linear kernel is proportional to XX^T and so is a specific case of the String kernel with $S = I$ (up to a multiplicative constant). The Linear kernel therefore assumes independence between all OTUs.

For 16S rRNA datasets it is likely that S is not positive definite as the feature map $\phi(\cdot)$ may produce identical outputs for closely-related OTUs. In this case S, the $p \times p$ matrix of OTU similarities, is rank-deficient ($\text{rank}(S) << p$) and therefore only positive semi-definite. However, this is not a problem in practice as S itself is not used as a kernel - the sample-wise kernel is XSX^T, which is a square matrix of size $n << p$. Even if S is rank deficient it is unlikely that the rank of X, which has shape $n \times p$, will be sufficiently low to cause the rank XSX^T to be less than n.

Spectrum Kernel

The Spectrum kernel (Leslie, Eskin, and Noble, 2001) is defined by a feature mapping that counts the number of k-mers that appear in string s and is given by

$$\phi(s) = (h_u^{\text{spec}}(s))_{u\in\mathcal{A}^k}\,, \tag{5.29}$$

where $h_u^{\text{spec}}(\cdot)$ counts the number of occurrences of substring u and $\mathcal{A}^k$ is the set of possible k-mers in alphabet $\mathcal{A}$. When analysing DNA sequences, $\mathcal{A} = \{\text{T},\text{G},\text{C},\text{A}\}$

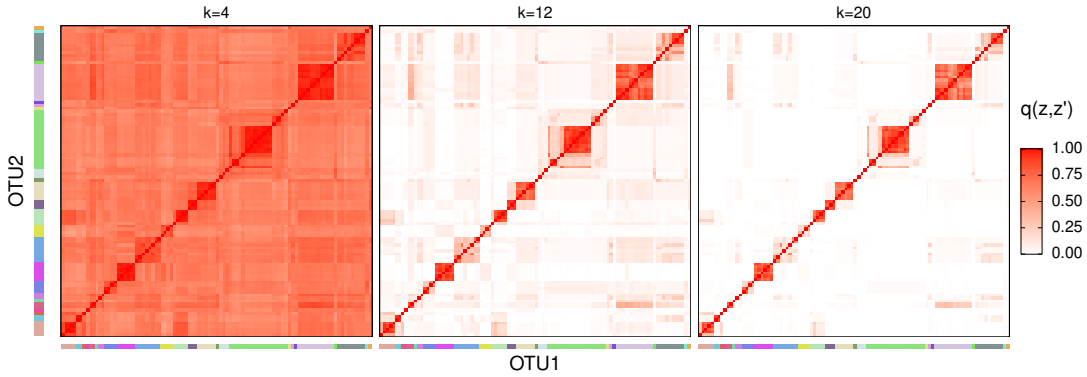


FIGURE 5.7: Spectrum kernels for 100 most abundant OTUs in the Busselton dataset with k-mer length $k \in \{4, 12, 20\}$. Coloured bars indicate the Family of the OTU - these show that the blocks of highly similar OTUs correspond to taxonomic classifications. The value of k can be tuned to correspond to different taxonomic classifications.

for the four nucleotide and so the k-mer feature space $\mathcal{A}^k$ has size 4^k. k is a hyperparameter that must be tuned (e.g. by optimising the marginal likelihood in supervised GP applications). The resulting kernel is the inner product

$$s(z, z') = \langle \phi(z), \phi(z') \rangle_{\mathcal{A}^k}, \tag{5.30}$$

where z, z' are the representative sequences of two OTUs. Figure 5.7 illustrates the $S = (s(z_i, z_j))_{i,j=1}^{p}$ matrices for Spectrum kernels with lengthscales $k \in \{4, 12, 20\}$. Smaller values of k produce a matrix with many non-zero elements while larger values of k induce a block diagonal structure, with blocks corresponding to clades of related OTUs.

Mismatch Kernel

DNA sequences undergo mutation, mainly in the form of insertions/deletions (indels) and substitutions, while mismatches between identical sequences can also be falsely recorded due to sequencing errors. Such similarities would not be recognised by the Spectrum kernel. The Mismatch kernel (Leslie, Eskin, Weston, et al., 2003) addresses this by allowing for mismatches in k-mers of length m, which is an additional hyperparameter whose maximum value is $k - 1$. Its feature map is given by

$$\phi(s) = (h^{\text{mis}}_{u,m}(s))_{u \in \mathcal{A}^k}, \tag{5.31}$$

where $h^{\text{mis}}_{u,m}(s)$ counts the number of occurrences of any substring with at most m mismatches with u.

Gappy Pair Kernel

The Gappy Pair kernel (Leslie and Kuang, 2003) allows for matches between a pair of k-mers with up to g gaps, where g is an additional hyperparameter. Its feature map is

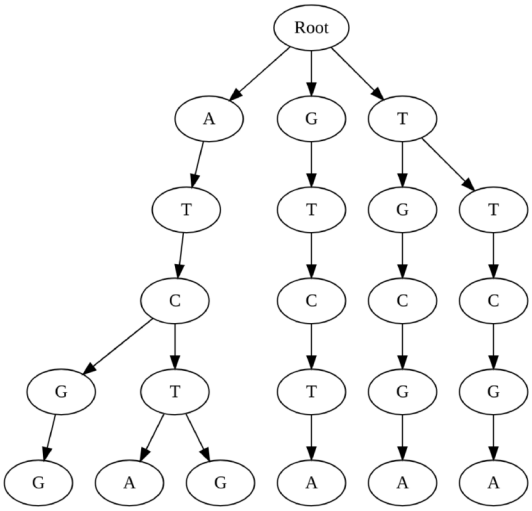


FIGURE 5.8: An example of a trie for the 5-mers ATCTA, ATCTG, GTCTA, ATCGG, TTCGA and TGCGA. Any of the 5-mers can be represented by a depth-first traversal of the trie. Tries enable efficient computation of string kernels.

$$\phi(s) = (h^{\text{gap}}_{u,g}(s))_{u \in \mathcal{A}^k}, \tag{5.32}$$

where $h^{\text{gap}}_{u,g}(s)$ counts the number of occurrences of any substring with that matches u with at most g gaps.

Computing String kernels

Efficient implementations of String kernels rely on tries, a tree data structure whose leaves represent a set of sequences and where all the children of an internal node have the same prefix. Given a set of sequences, it is possible to reconstruct every individual sequence via a depth-first traversal of the corresponding trie. Given two sequences and k-mer length, trie-based algorithms construct a single trie whose leaves represent every element in $\mathcal{A}_k$. The trie is then used to count the occurrences of each k-mer in either sequence in order to calculate $\phi(s)$. An example trie constructed for a set of 5-mers is shown in Figure 5.8.

Tries allow for far more efficient k-mer lookups than a naive search in the size of the k-mer space $|\mathcal{A}_k| = 4^k$, which is exponential in k. When using tries the time complexity to compute one element in a Spectrum kernel is $\mathcal{O}(k(|z| + |z'|))$ for k-mer length k and sequences z, z' with lengths $|z|, |z'|$, which is linear in k (Shawe-Taylor, Cristianini, et al., 2004).

The time complexity of the Mismatch kernel is $\mathcal{O}(k^{m+1}|\mathcal{A}_k|(|z| + |z'|))$, meaning that including the mismatches increases the time complexity by a factor of $k^m 4^k$ relative to the Spectrum kernel, restricting the values of k for which the Mismatch kernel can feasibly be calculated (Shawe-Taylor, Cristianini, et al., 2004). For a single element of the Gappy pair kernel the running time is $\mathcal{O}(k^g(|z| + |z'|))$, which is an increase by a factor of k^{g-1} relative to the Spectrum kernel with the same k-mer length (Leslie and Kuang, 2003).

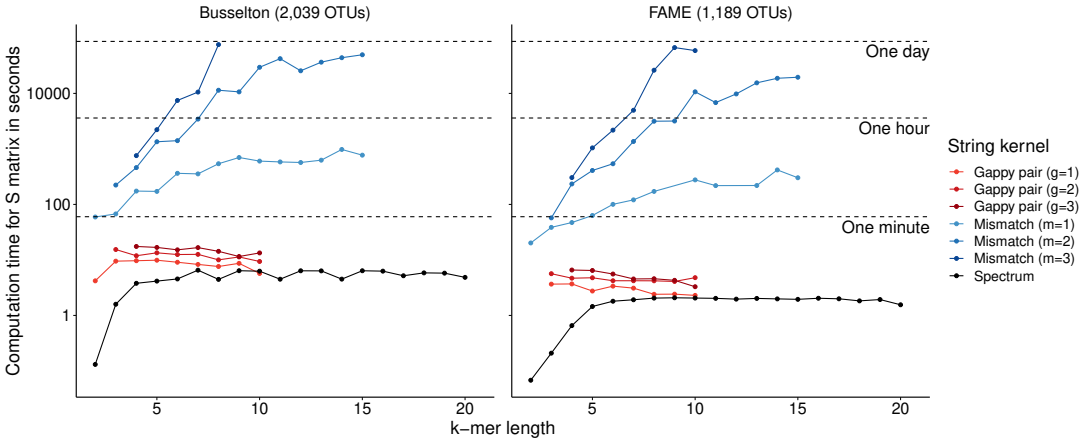


FIGURE 5.9: Empirical computation times for the string similarity matrix S for different hyperparameter values. Calculations were run on 8 cores using the Kebabs package for R (Palme et al., 2015).

The empirical compute times for the kernel matrix of the Busselton and FAME datasets are shown in Figure 5.9, which show that the Mismatch kernel requires at least 3 orders of magnitude more time than a Spectrum or Gappy pair kernel for the same k-mer length. For the Spectrum, Gappy pair kernels and Mismatch kernels with $m \leq 2$ the compute time plateaus once it reaches some value of k (the specific value depends on the type of kernel). This is because for all any moderately large k the number of leaves in the trie (which is 4^k) is far larger than the number of k-mers actually present in the two strings z and z', meaning that large parts of the tree are unpopulated. These unpopulated subtrees are pruned before conducting the k-mer search and so increasing the value of k does not increase the size of the search in practice (Shawe-Taylor, Cristianini, et al., 2004).

While the time complexity of computing String kernels can be restrictive this is mitigated by a combination of two factors. Firstly, the elements of a kernel are independent computations and so the computational time can be easily reduced using distributed computing infrastructure (so-called embarrassingly parallel computations). Secondly, the nature of microbiome dataset analysis means that the definitions of the OTUs (via their representative sequences) are fixed once the initial pre-processing has been completed. The entire kernel matrix can therefore be computed in advance and stored for future use, and so a computation time on the order of days (or even weeks) is feasible as it only has to be performed once.

6.3 Implications of compositionality for kernel methods

A common strategy for analysing compositional data is to apply a log-ratio transformation and then apply a statistical method to the transformed data, which live in Euclidean space (Quinn and Erb, 2020). This is the strategy followed in this study.

Kernel methods centre around inner products in the feature space as any kernel function $k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$ for a feature map $\phi(\cdot)$ and RKHS $\mathcal{H}$. As compositional data live on the simplex, calculating Euclidean distances and inner products on the observed abundances may give misleading results, but this is mitigated by CLR transformation prior to computing the kernel function. For kernels that use the Euclidean distance (such as RBF and Matern32) this is clearly appropriate as the CLR transform maps the compositions to Euclidean space. The only requirement is

to bear in mind that the transformed variables represent the abundance of an OTU relative to the sample geometric mean when interpreting the results.

String kernels are computed using

$$k(x, x') = x' S x^T = \sum_{i=1}^{p} \sum_{j=1}^{p} x'^{(i)} \, S_{ij} \, x^{(j)} \, . \tag{5.33}$$

When using absolute abundances each term weights the similarity of OTU i and OTU j by their absolute abundance in x and x'. After applying the CLR transform the i^{th} sample contains the relative importance of OTU i and so it is valid to naively apply (5.33) to the CLR-transformed abundances. The Linear kernel falls into this category as it is simply a String kernel with $S = I$.

This is not the case for the UniFrac kernels as the UniFrac kernel is an inherently non-compositional distance metric (Gloor et al., 2017). Both the weighted and unweighted variants depend on which branches of the phylogenetic tree have zero abundance in two samples and zeros are not preserved by the combination of a zero-replacement strategy and CLR transform. Therefore neither of two UniFrac kernels are used with the CLR transform in this study.

7 Two-sample testing simulation study

7.1 Simulation aims

Given two sets of 16S rRNA samples, $X = \{x_i\}_{i=1}^{n_x} \sim P$ and $Y = \{y_i\}_{i=1}^{n_y} \sim Q$, where each $x_i, i = 1, \ldots, n_x, y_i, i = 1, \ldots, n_y$ are p-dimensional vectors and a kernel function $k(\cdot, \cdot)$, the following set of simulations aims to investigate the performance of the test statistic $\text{MMD}_k(P, Q)$, the MMD between P and Q estimated using $k(\cdot, \cdot)$. This is estimated from the samples in X and Y using

$$\widehat{\text{MMD}}_k(X, Y) = \left(\frac{1}{n_x^2} \sum_{i,j=1}^{n_x} k(x_i, x_j) + \frac{1}{n_y^2} \sum_{i,j=1}^{n_y} k(y_i, y_j) - \frac{2}{n_x n_y} \sum_{i,j=1}^{n_x, n_y} k(x_i, y_j) \right)^{\frac{1}{2}} , \tag{5.34}$$

which is the biased estimator of $\text{MMD}_k(P, Q)$ but has minimum variance (Gretton et al., 2012). This estimator is simply the sum of the within-group similarities in $\mathcal{H}$ minus the between-group similarities. Statistical significance is assessed using a permutation test with N_{perm} permutations, where the p-value is given by

$$p_{\text{perm}} = \frac{\sum_{i=1}^{N_{\text{perm}}} \mathbb{1}(\widehat{\text{MMD}}_k(X_i^*, Y_i^*) \geq \widehat{\text{MMD}}_k(X, Y))}{1 + N_{\text{perm}}} , \tag{5.35}$$

where $\{(X_i^*, Y_i^*)\}_{i=1}^{N_{\text{perm}}}$ is formed by permuting the combined samples of X and Y (Phipson and Smyth, 2010).

This study requires a simulation strategy that:

1. uses a P and Q that generate realistic OTU counts;

2. uses realistic phylogenetic relationships between the OTUs; and
3. offers control of the scale of phylogenetic differences between P and Q.

The first requirement is achieved by simulating the OTU counts from a Dirichlet-multinomial $\mathrm{DMN}(N, \hat{\alpha})$, where $\hat{\alpha}$ are Maximum likelihood estimates of the concentrations from a real 16S rRNA dataset. This also addresses the second requirement, as it allows the use of the accompanying phylogenetic tree. The following section describes how to address the third point by constructing P and Q such that they are identical up to a given phylogenetic scale.

7.2 Controlling the phylogenetic differences between P and Q

In this simulation study the two populations are described by

$$P = \mathrm{DMN}(N, \alpha_1)\,, \quad Q = \mathrm{DMN}(N, \alpha_2)\,, \tag{5.36}$$

meaning that the difference between P and Q is fully defined by the relationship between α_1 and α_2. Consider a scenario where each OTU is assigned to one of a set of clusters $\mathcal{C}$, where

$$\mathcal{C} = \{c_1\,, \ldots\,, c_{|\mathcal{C}|}\}\,. \tag{5.37}$$

As each OTU is assigned to a single cluster it is possible to write the elements of α_1 as the union of disjoint subsets $\{\alpha^{(c_k)}\}_{k=1}^{|\mathcal{C}|}$, where each subset contains the DMN concentrations corresponding to a single cluster of $\mathcal{C}$.

It is then possible to define a family of permutation operations $\pi_{\mathcal{C}}(\cdot)$, for which

$$\alpha_2 = \pi_{\mathcal{C}}(\alpha_1) \implies \alpha_1^{(c_k)} = \alpha_2^{(c_k)}\,, \quad \forall c_k \in \mathcal{C}\,. \tag{5.38}$$

This ensures that the set of concentrations assigned to a cluster in P are identical to the concentrations for that cluster in Q. The specific OTUs to which a concentration is assigned may differ between P and Q if the cluster contains more than one item. If the clustering $\mathcal{C}$ is constructed based on the phylogenetic distances between OTUs then the difference between P and Q will be restricted to the same phylogenetic scale as the OTU cluster assignments.

7.3 Phylogeny-aware clustering of OTUs

Controlling the scale of phylogenetic difference between P and Q therefore requires defining clusters of phylogenetically similar OTUs. This can be achieved by clustering the OTUs based on their phylogenetic distances, which are available via the phylogenetic tree.

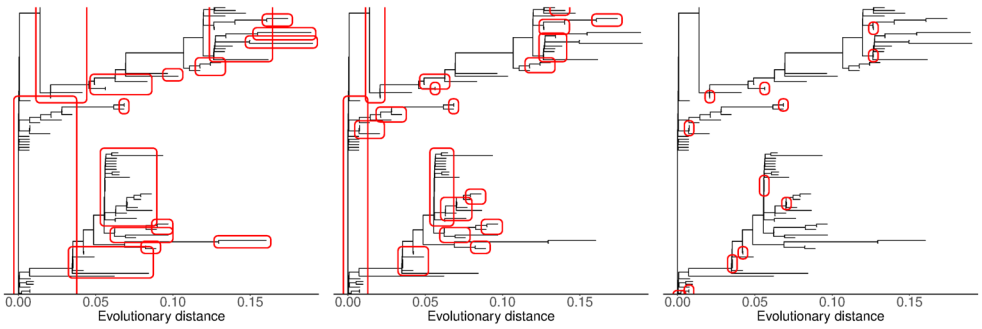
Consider a dataset of p OTUs, with accompanying phylogenetic tree $\tau = \{V, E\}$, where the nodes $V = \{v_1\,, \ldots\,, v_{|V|}\}$ are indexed such that $\{v_i \mid i \leq p\}$ are the leaves (OTUs) and $\{v_i \mid i > p\}$ are internal nodes (common ancestors). The tree τ can be expressed as the symmetric matrix $\Delta^\tau \in \mathbb{R}_+^{p \times p}$ whose elements are

$$(\Delta^\tau)_{ij} = d^\tau(v_i, v_j)\,, \quad i, j = 1, \ldots, p\,, \tag{5.39}$$

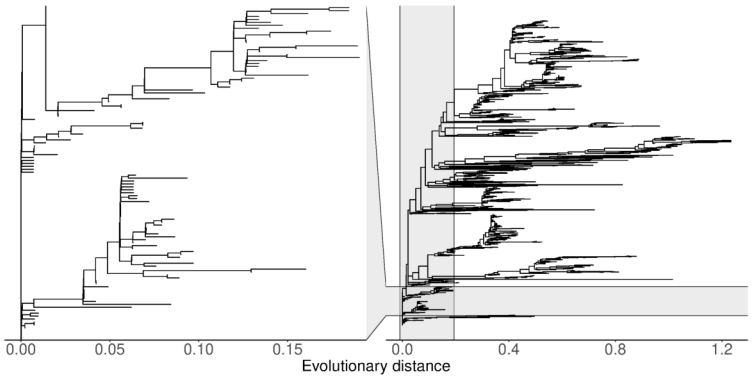
where $d^\tau(v_i, v_j)$ is the distance between leaves i and j along the branches of τ. This implies that the diagonal elements $(\Delta^\tau)_{ii} = 0, i = 1, \ldots, p$. Since the leaves are the first p nodes of the tree Δ^τ only encodes relationships between the observed OTUs and excludes the imputed common ancestors. Given Δ^τ, there exists a set of OTU clusters $\mathcal{C}_\varepsilon = \{c_1, \ldots, c_{|\mathcal{C}_\varepsilon|}\}$ that satisfies

$$(\Delta^\tau)_{ij} \leq \varepsilon \Delta^\tau_{\max}\,, \quad \forall i, j \in c_k\,, \quad \forall c_k \in \mathcal{C}_\varepsilon\,, \tag{5.40}$$

where $\Delta^\tau_{\max}$ is the largest element of Δ^τ. That is to say, for a set of clusters $\mathcal{C}_\varepsilon$, no two OTUs within a cluster have a pairwise phylogenetic distance greater than $\varepsilon\Delta^\tau_{\max}$. Figure 5.10(A-C) illustrates the OTU clusters for a subset of OTUs (panel D) from FAME dataset for $\varepsilon \in \{0.03, 0.01, 0.003\}$. As the value of ε decreases there are a larger number of clusters, each of which contains a smaller number of OTUs. This can also be seen in the distribution of cluster sizes in Figure 5.11 - for small values of ε the majority of OTUs are in singleton clusters.



(A) $\varepsilon = 0.03$ (B) $\varepsilon = 0.01$ (C) $\varepsilon = 0.003$



(D) The region of the tree shown in (A), (B) and (C) in the context of the entire tree.

FIGURE 5.10: A-C: Clusters of OTUs for $\varepsilon \in \{0.03, 0.01, 0.003\}$ for a subset of the FAME phylogenetic tree. Red boxes indicate clusters of OTUs and singleton clusters are not marked. D: the region of the tree in panels A-C in the context of the entire tree.

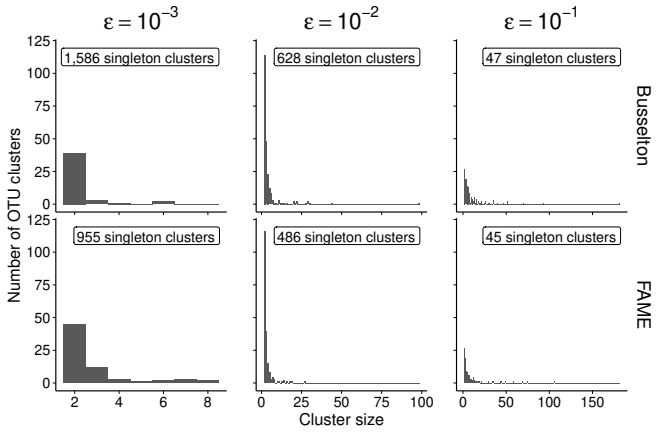


FIGURE 5.11: Distribution of OTU cluster sizes for the two datasets at different values of the phylogenetic distance threshold ε (non-singleton clusters only).

By combining the cluster definitions (5.40) with the permutation $\pi_{\mathcal{C}}(\cdot)$ (defined in (5.38)) it is possible to construct two populations of OTU samples, P and Q, where the differences between P and Q occur on a phylogenetic scale less than ε. The permutation corresponding to the clustering $\mathcal{C}_\varepsilon$ is denoted $\pi_\varepsilon(\cdot)$ from this point onwards, which is to say $\pi_\varepsilon(\cdot) := \pi_{\mathcal{C}_\varepsilon}(\cdot)$. This is illustrated in Figure 5.12.

This control results from the fact that $\pi_\varepsilon(\cdot)$ depends on the clustering $\mathcal{C}_\varepsilon$. Smaller values of ε induce an OTU clustering that restricts any swaps from P to Q to occur amongst closely-related OTUs. Two limiting cases are when

- $\varepsilon = 0$, when $\mathcal{C}_0$ contains p singleton clusters and π_1 has no effect.
- $\varepsilon = 1$, when $\mathcal{C}_1$ contains 1 cluster of size p and π_0 is a full permutation (it does not consider the phylogenetic tree).

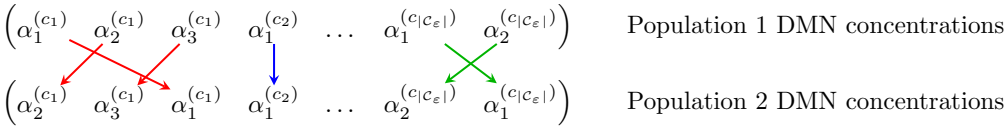


FIGURE 5.12: The difference between the two populations in the two-sample test simulation study is a permutation that restricts swaps to those within a set of clusters $\mathcal{C}_\varepsilon = \{c_1, \ldots, c_{|\mathcal{C}_\varepsilon|}\}$. Here $\alpha_i^{(c_k)}$ is the DMN concentration of the i^{th} OTU in cluster c_k. In this example the clusters c_1, c_2 and $c_{|\mathcal{C}_\varepsilon|}$ have sizes 3, 1 and 2 respectively.

7.4 Simulation Setup

The final simulation setup is

$$X = \{x_i\}_{i=1}^{n_x} \sim \text{DMN}(N, \alpha_1), \tag{5.41}$$
$$Y = \{y_i\}_{i=1}^{n_y} \sim \text{DMN}(N, \alpha_2), \tag{5.42}$$
$$N \sim \text{NB}(10^5, b), \tag{5.43}$$
$$\alpha_2 = \pi_\varepsilon(\alpha_1)\,, \tag{5.44}$$

where the scale of phylogenetic differences are controlled by ε. Throughout these experiments $n_x = n_y = n$, where n is the group size and the dispersion parameter b for the total reads per sample takes one value from $b \in \{3, 10, 30\}$. For all non-UniFrac kernels the effect of transforming the counts using $\log(x+1)$ and $\text{clr}(x)$ are investigated, while only $\log(x+1)$ is used with UniFrac kernels. Applying a $\log(x+1)$ transform is a popular pre-processing step in biological data analysis as it preserves zeros and can reduce variance (Changyong et al., 2014). The full simulation procedure is stated in Algorithm 7.1.

Algorithm 7.1 Simulation procedure for MMD two-sample tests (Section 7.4).

Require: empirical DMN concentrations $\hat{\alpha}$, phylogenetic tree τ, phylogenetic distance thresholds $\{\varepsilon_1, \ldots, \varepsilon_n\}$, kernel $k(\cdot,\cdot)$, reads per sample dispersion b
for $i = 1, \ldots, n_{\text{replicates}}$ **do**
  $\alpha_1 = \hat{\pi}_1(\hat{\alpha}), \quad \hat{\pi}_1 \sim \pi_1$ ▷ Population 1 DMN concentrations
  for $\varepsilon = \{\varepsilon_1, \ldots, \varepsilon_n\}$ **do**
    $\alpha_2 = \hat{\pi}_\varepsilon(\alpha_1), \quad \hat{\pi}_\varepsilon \sim \pi_\varepsilon$ ▷ Population 2 DMN concentrations
    $N \sim \text{NB}(10^5,\, b)$ ▷ Total reads per sample
    $X \sim \text{DMN}(N, \alpha_1)$
    $Y \sim \text{DMN}(N, \alpha_2)$
    Calculate $\widehat{\text{MMD}}_k(X, Y)$ ▷ Test statistic, using (5.34)
    Calculate p-value ▷ 100-permutation test, using (5.35)
  end for
end for

The aim of the study is to investigate the behaviour of the two-sample test with $\text{MMD}_k(X, Y)$ as the test statistic. An appropriate kernel induces a two-sample test which has well-calibrated Type I error and high power, but is also sensitive to the value of ε. These experiments use the following kernels:

- Spectrum kernel with $k \in \{2, \ldots, 20\}$;
- Mismatch kernel with $k \in \{2, \ldots, 7\}$ and $m \in \{1, 2, 3\}$;
- Gappy pair kernel with $k \in \{2, \ldots, 10\}$ and $g \in \{1, 2, 3\}$;
- UniFrac kernel (Weighted and Unweighted);
- RBF and Matern32 kernels (with median heuristic lengthscale, Flaxman et al., 2016); and
- Linear kernel.

7.5 Simulation results I: Type I error and power

The metric used to evaluate the behaviour of the two-sample test with a given kernel is the fraction of replicates in which H_0 is rejected at a nominal significance level of 0.1, for which a well-calibrated test rejects H_0 close to 10% of the time when data are simulated under the null hypothesis. If the observed rate of H_0 rejections is higher or lower than 10% then the Type I error of the test is poorly-calibrated.

When $\varepsilon = 0$, a test with any phylogenetic kernel has well-calibrated Type I error

Figure 5.13 shows the H_0 rejection rate for the Spectrum kernel with $k = 20$ and the Unweighted and Weighted UniFrac kernels. These kernels are included in Figure 5.13 as they are the phylogenetic kernels with the highest power (when $\varepsilon > 0$). The behaviour of other phylogenetic kernels (Spectrum kernels with other values of k, Mismatch and Gappy pair kernels) are discussed later in this Section. For these three kernels the Type I error is close to the nominal significance level of 0.1 when $\varepsilon = 0$ (Figure 5.13(A) and (B), left-hand column).

The Spectrum ($k = 20$) kernel and two UniFrac kernels have high power when $\varepsilon \geq 10^{-2}$

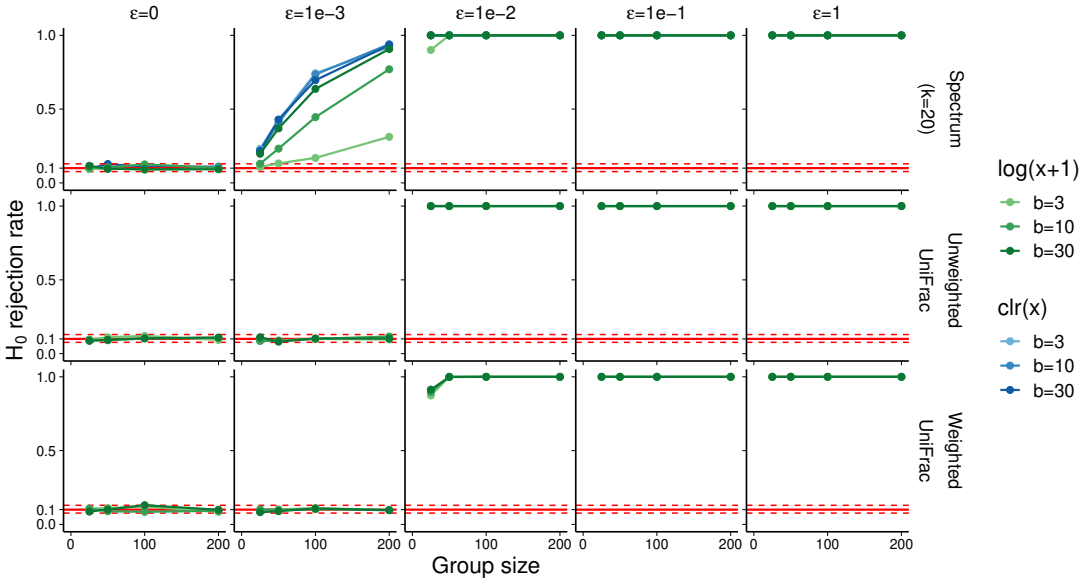
Figure 5.13 also shows that when $\varepsilon \geq 10^{-2}$, the power of the Spectrum ($k = 20$) and two UniFrac kernels quickly converges to one (H_0 is always rejected) as the group size increases. For $\varepsilon \in \{10^{-1}, 1\}$ all three of these phylogenetic kernels have power equal to one in both datasets, irrespective of the group size or transformation.

The power of the Spectrum ($k = 20$) kernel depends on the choice of transformation when $\varepsilon = 10^{-3}$

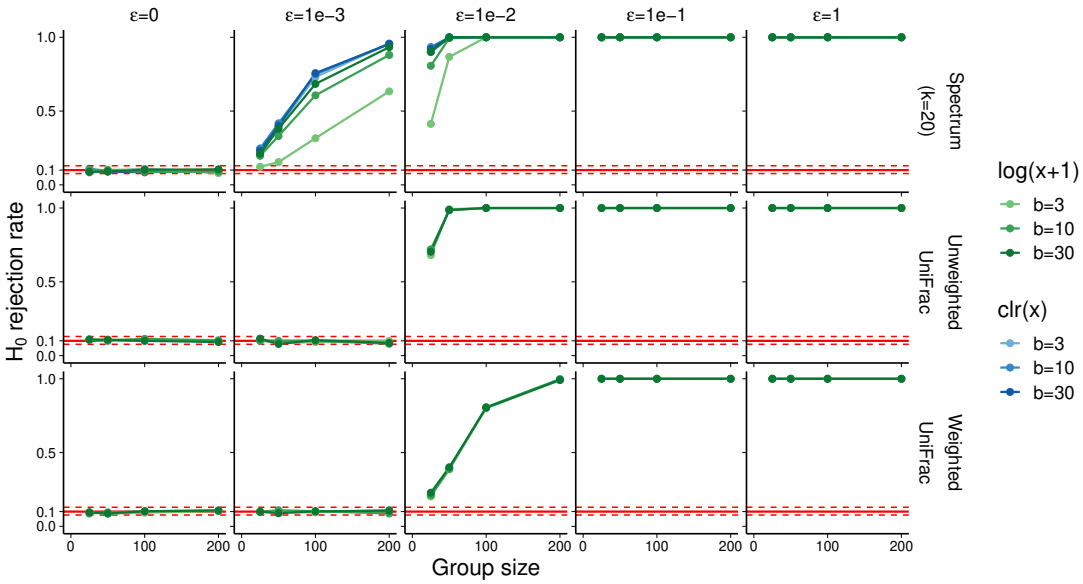
Only the Spectrum ($k = 20$) kernel has non-zero power when $\varepsilon = 10^{-3}$ (Figure 5.13). In this case the power of the Spectrum ($k = 20$) kernel depends on the transformation used - for the $\log(x + 1)$ transform the power depends on the value of b, the dispersion parameter for distribution of the total reads per sample, with lower values of b decreasing the power of the test. Using the CLR transform removes this dependence on b increases the power of the Spectrum ($k = 20$) kernel.

Tests using non-phylogenetic kernels have high power for all $\varepsilon > 0$, but are not sensitive to ε

Figure 5.14 shows that the RBF, Matern32 and Linear kernels have well calibrated Type I error ($\varepsilon = 0$) and high power ($\varepsilon > 0$). In fact, these three non-phylogenetic kernels have higher power than the three phylogenetic kernels shown in Figure 5.13. However, this behaviour is a reflection of an undesirable feature of the non-phylogenetic kernels - there is no sensitivity to the value of ε in the behaviour of the two-sample test. This is to be expected as these kernels do not model any phylogenetic relationships and weight all differences between OTUs equally. They are therefore very likely to reject H_0 based on differences between very closely-related (and often indistinguishable) OTUs.

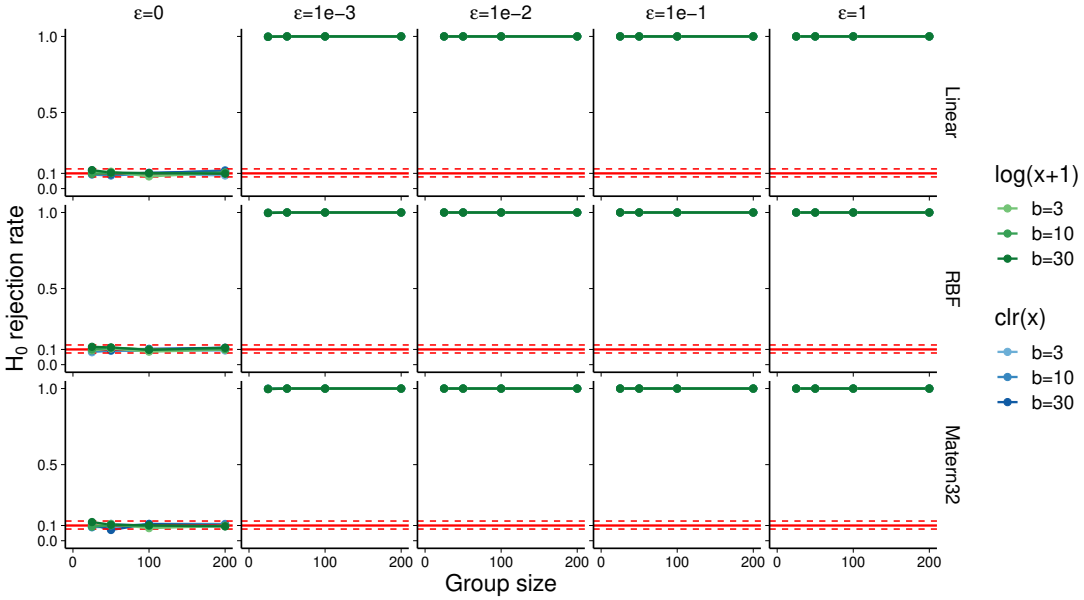


(A) Busselton ($p = 1,689$)

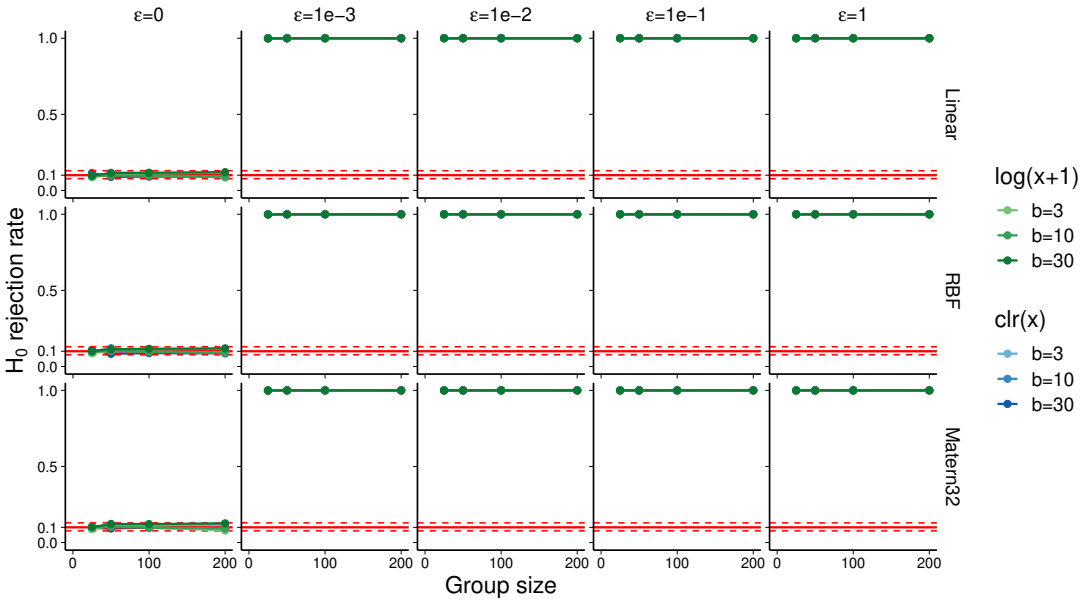


(B) FAME ($p = 1,147$)

FIGURE 5.13: Rate of null hypothesis rejections at a significance level of 0.1 for MMD two-sample tests with the highest-power phylogenetic kernels. The solid red line denotes the nominal significance level (0.1) and the dashed lines show its 95% binomial proportion confidence interval. Data were simulated using the phylogenetic trees and DMN concentrations of the Busselton (A) and FAME (B) datasets. Generated from 1,000 replicates of Algorithm 7.1.



(A) Busselton ($p = 1,689$)



(B) FAME ($p = 1,147$)

FIGURE 5.14: Rate of null hypothesis rejections at a significance level of 0.1 for MMD two-sample tests with non-phylogenetic kernels. The solid red line denotes the nominal significance level (0.1) and the dashed lines show its 95% binomial proportion confidence interval. Data were simulated using the phylogenetic trees and DMN concentrations of the Busselton (A) and FAME (B) datasets. Generated from 1,000 replicates of Algorithm 7.1.

For phylogenetic kernels differences in MMD are driven by phylogeny

The MMD measures the distance between P and Q in the RKHS defined by the chosen kernel. As every kernel function (defined by a kernel and its hyperparameters) induces a distinct RKHS it is not possible to compare MMD values calculated using different kernels. However, it is possible to inspect the properties of a specific RKHS by comparing its empirical MMD results for different values of ε.

For a single replicate of Algorithm 7.1 the DMN concentrations α_1 are fixed, from which α_2 are obtained using $\pi_\varepsilon(\cdot)$ using a sequence of ε values. Therefore, an appropriate RKHS for microbiome applications should produce larger MMD values when $\varepsilon = 1$ than when $\varepsilon = 0.1$. The two scenarios represented by these values of ε are very different, as $\varepsilon = 1$ imposes no phylogenetic restrictions on the differences between the P and Q, but $\varepsilon = 0.1$ forces any differences to occur amongst OTUs that are most 10% of the total phylogenetic variation apart.

Figure 5.15 shows the ratio of MMD values when $\varepsilon = 0.1$ to its value when $\varepsilon = 1$ for a selection of kernels. The Spectrum ($k = 20$) and UniFrac kernels (top row of plots) exhibit this desirable behaviour, while non-phylogenetic kernels do not (bottom row of plots).

For non-phylogenetic kernels differences in MMD are driven by the size of the permutation space $\pi_\varepsilon(\cdot)$

The median ratio of in Figure 5.15 is less than 1 for the non-phylogenetic kernels but this is not due to phylogenetic differences as is the case for the phylogenetic kernels. The decrease in MMD from $\varepsilon = 1$ to $\varepsilon = 0.1$ is caused by the relative sizes of the set of permutations $\pi_\varepsilon(\cdot)$ and not phylogenetic differences between the P and Q. Recall that

$$\alpha_2 = \pi_\varepsilon(\alpha_1)\,, \tag{5.45}$$

where $\pi_\varepsilon(\cdot)$ is the family of permutations that leaves the elements of the set $\mathcal{C}_\varepsilon$ unchanged.

Larger values of ε define a small number of large OTU clusters, while smaller values define a large number of small clusters with many singleton clusters (see Figure 5.11). Given a set of clusters $\mathcal{C}_\varepsilon = \{c_1\,, \ldots\,, c_{|\mathcal{C}_\varepsilon|}\}$, the size of the permutation space $\pi_\varepsilon(\cdot)$ is $\sum_{c\in\mathcal{C}_\varepsilon} |c|!$, which grows quickly with ε due to the factorial dependence (see Table 5.2).

An important driver of the size of $\pi_\varepsilon(\cdot)$ is the number of singleton clusters as any OTUs in singleton clusters have the same marginal distribution in both P and Q. As smaller values of ε result in more singleton clusters it follows to expect larger MMD values for larger ε, irrespective of phylogeny. This is because there are a larger number of possible permutations contained in $\pi_\varepsilon(\cdot)$, which is denoted $|\pi_\varepsilon(\cdot)|$.

The relative importance of phylogeny and $|\pi_\varepsilon(\cdot)|$ in controlling the magnitude of MMD values can be established by comparing the MMD when $\alpha_2 = \pi_\varepsilon(\alpha_1)$ with those calculated using $\pi_{\tilde{\varepsilon}}(\cdot)$, where $\pi_{\tilde{\varepsilon}}(\cdot)$ is the set of permutations defined by a set of clusters with the same sizes as $\mathcal{C}_\varepsilon$, but whose labels are assigned at random (without using the phylogenetic tree). In other words, given a set of phylogenetic

TABLE 5.2: The size of the permutation set $\pi_\varepsilon(\cdot)$ for different ε.

	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-1}$
Busselton	3×10^3	3×10^{153}	$> 10^{308}$
FAME	1×10^5	1×10^{28}	1×10^{170}

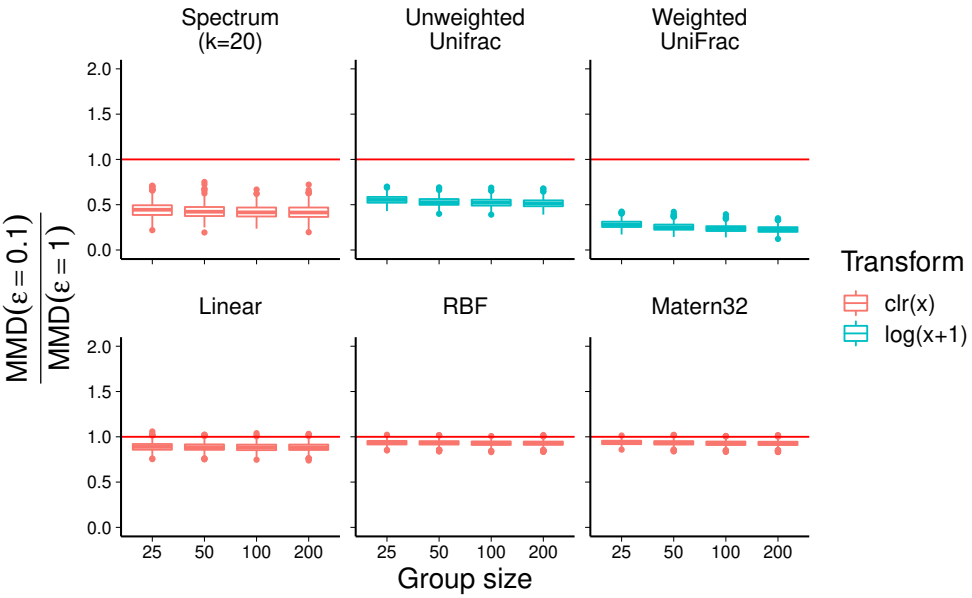
clusters $\mathcal{C}_\varepsilon$, the set of permutations $\pi_{\tilde{\varepsilon}}(\cdot)$ simply shuffles the cluster labels amongst the OTUs. The result is a set of permutations with the same size as $\pi_\varepsilon(\cdot)$ that have no relation to phylogeny.

Figure 5.16 compares MMD values calculated when α_1 and α_2 are related to one another by one of $\pi_\varepsilon(\cdot)$ or $\pi_{\tilde{\varepsilon}}(\cdot)$. In Figure 5.16 the permutation that defines α_2 is either constructed using phylogeny ($\pi_\varepsilon(\cdot)$) or uses an equivalent random clustering ($\pi_{\tilde{\varepsilon}}(\cdot)$). MMDs for the Spectrum ($k = 20$) and two UniFrac kernels have distinct MMD distributions across for the two permutations, but non-phylogenetic kernels have identical distributions. This demonstrates that in an RKHS defined by a non-phylogenetic kernel, MMD values are determined by $|\pi_\varepsilon(\cdot)|$ and not by the phylogenetic relationships encoded by $\pi_\varepsilon(\cdot)$.

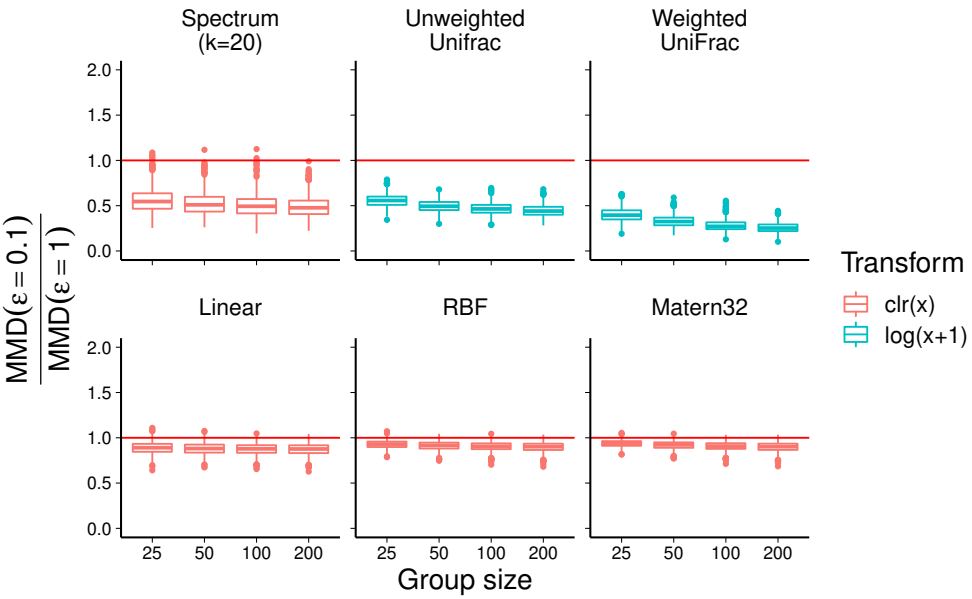
Larger k-mer lengths increase power for String kernels

Before applying String kernels it is necessary to select the k-mer length as well as the number of mismatches (m, for the Mismatch kernel) or number of gaps (g, for the Gappy pair kernel). Figure 5.17 shows that the String kernels all have well-calibrated Type I error for any choice of hyperparameters. However, the power of the test depends critically on the choice of k (Figure 5.18). The larger the value of k, the more powerful the test for all three variants of the String kernel. For the Mismatch and Gappy pair kernels, the effect of k is larger than that of their additional hyperparameter (m or g). In addition, the Mismatch kernel has lower power than the Spectrum or Gappy pair kernel for a fixed value of k, irrespective of the choice of m.

This dependence of power on k can be explained by considering the role of k-mer length when computing String kernels. A String kernel computes $k(x, x') = xSx'^T$, where the the length of k-mer controls the entries of S. Small values of k (e.g. $k \leq 4$) result in an S matrix that has few non-zero entries, effectively modelling all OTUs as highly related to one another (see Figure 5.7). This means that larger values of ε or larger group sizes are required for a statistically significant MMD value, as differences between OTU abundances in X and Y are "smoothed" by the S matrix. As k increases S approaches a block-diagonal structure, where the only non-zero entries are those corresponding to clusters of OTUs with very similar sequences. These S matrices only smooth differences in P and Q if they occur between closely-related OTUs, resulting in tests with higher power.

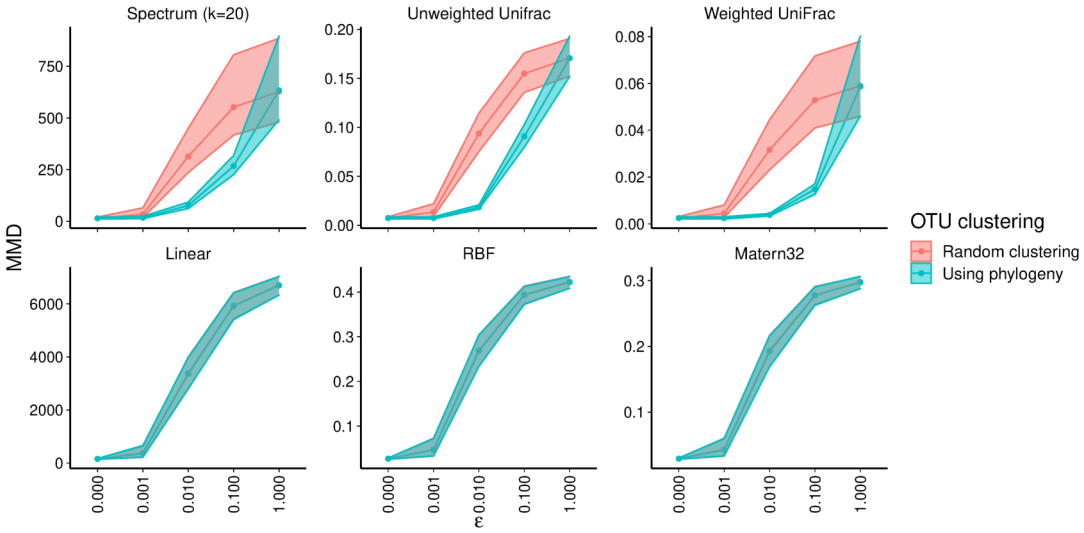


(A) Busselton ($p = 1,689$)

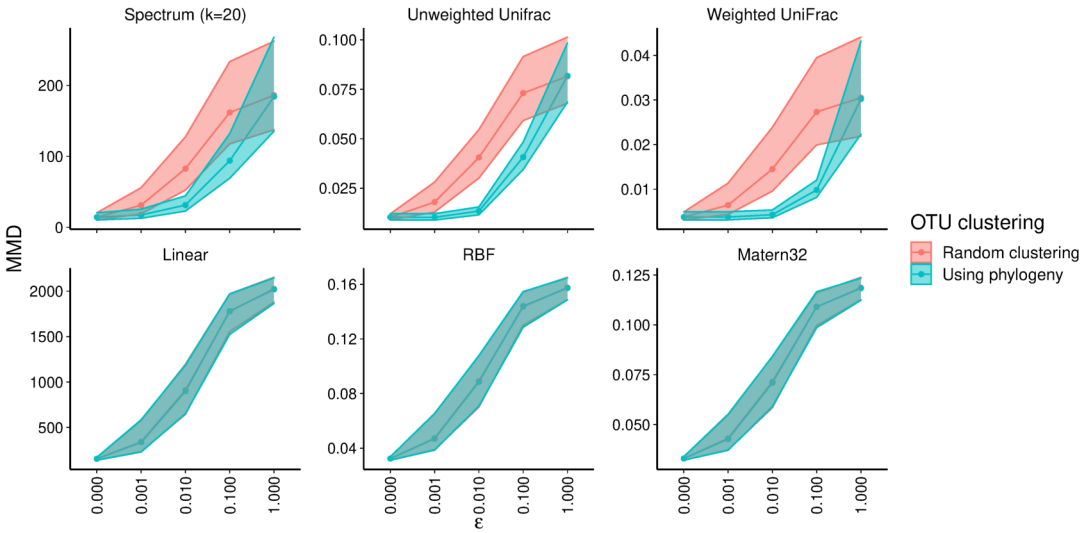


(B) FAME ($p = 1,147$)

FIGURE 5.15: The ratio of the empirical MMD when $\varepsilon = 0.1$ to when $\varepsilon = 1$ across 1,000 replicates. The red line indicates equality between the MMD in the two scenarios. The top row contains kernels that exhibit desirable behaviour (phylogenetic kernels with well-selected hyperparameters) while the bottom row contains kernels which do not exhibit this behaviour (non-phylogenetic kernels).



(A) Busselton ($p = 1,689$)



(B) FAME ($p = 1,147$)

FIGURE 5.16: Distributions of MMD values from 1,000 replicates of Algorithm 7.1. The shaded area denotes the 2.5%, 50.0%, 97.5% percentiles across the replicates. These results are for $b = 10$ but are representative of the other values tested. The CLR transformed is used for all non-UniFrac kernels. UniFrac kernels use the $\log(x + 1)$ transform.

The behaviour of the phylogenetic kernels is stable between the two datasets

The behaviour of the Type I error (when $\varepsilon = 0$) and statistical power (when $\varepsilon > 0$) follow the same trends when using the Busselton or FAME OTUs, suggesting that this modelling approach is capturing relevant characteristics of these two lung 16S rRNA datasets. It is also observed in both datasets that:

- the UniFrac kernels have zero power for $0 < \varepsilon \leq 10^{-2}$ and high power for $\varepsilon > 10^{-2}$;
- the unweighted UniFrac is at least as powerful as the weighted UniFrac kernel;
- larger values of k increase the power of all three String kernels;
- the Mismatch kernel has lower power than the Spectrum and Gappy pair kernels for fixed ε, sample size and k;
- the power of the Gappy pair and Mismatch kernels are more dependent on k than on g or m;
- of the phylogenetic kernels, only the Spectrum ($k = 20$) kernel has non-zero power when $\varepsilon = 10^{-3}$; and
- when $\varepsilon = 10^{-3}$ the power of the Spectrum ($k = 20$) kernel is inversely proportional to b when using $\log(x + 1)$ to transform OTU abundances, but this dependence is removed when using the CLR transform.

8 Host trait prediction using Gaussian process regression

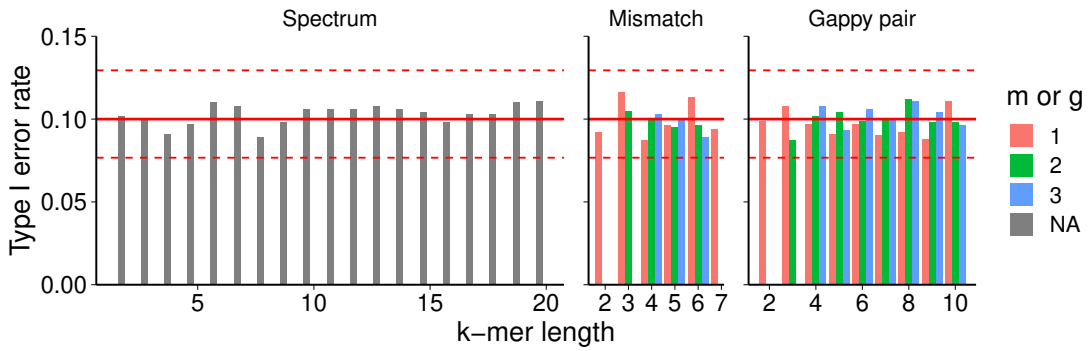
8.1 Simulation aims

One of the most common applications of supervised learning in microbiome studies is host trait prediction, which aims to to predict host phenotype from microbial community composition. Constructing such predictive models is often the first step of a pipeline that includes a variable importance analysis for association testing and/or biomarker identification. The longer-term aims of such studies are in the field of personalised/precision medicine, which aims to tailor treatments more specifically to individual patients.

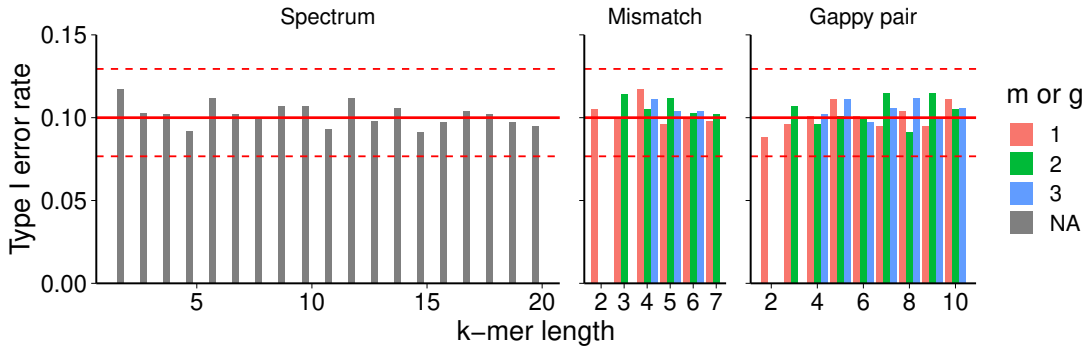
The aim of this set of simulations is to identify scenarios under which a phylogenetic kernel improves the training data fit of a GP regression model and the predictive performance. Once this has been achieved these results then show how to estimate the degree to which OTU effects are related to 16S rRNA gene sequence similarity by comparing the log-marginal likelihood of GP regression models with phylogenetic and non-phylogenetic kernels in a Bayesian hypothesis testing framework.

8.2 Simulation setup

In these simulations the OTU abundances $X \in \mathbb{Z}_{\geq 0}^{n \times p}$ are sampled from a single population with DMN concentrations α, which are a permutation of Maximum likelihood concentration estimates from one the of the two real datasets. The phenotype

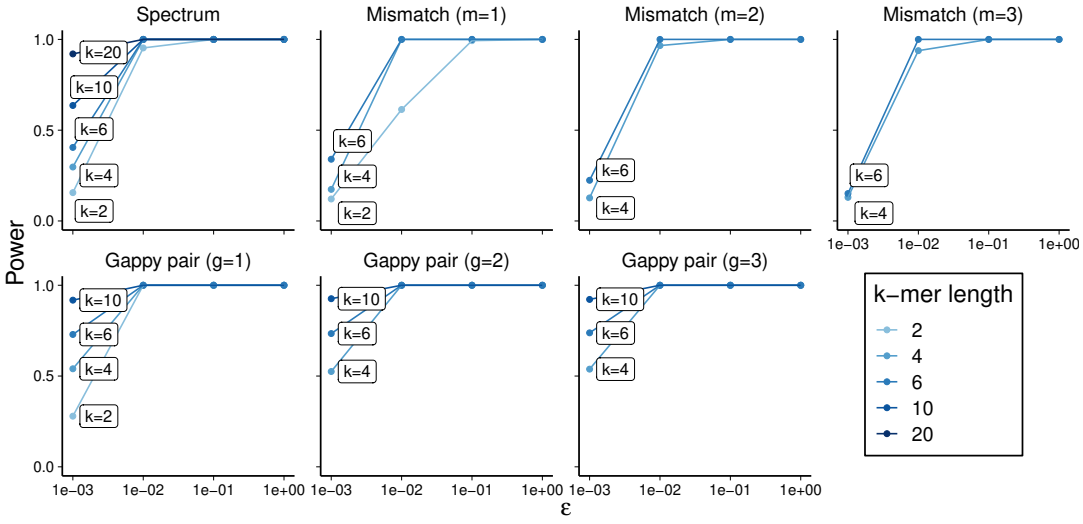


(A) Busselton ($p = 1,689$)

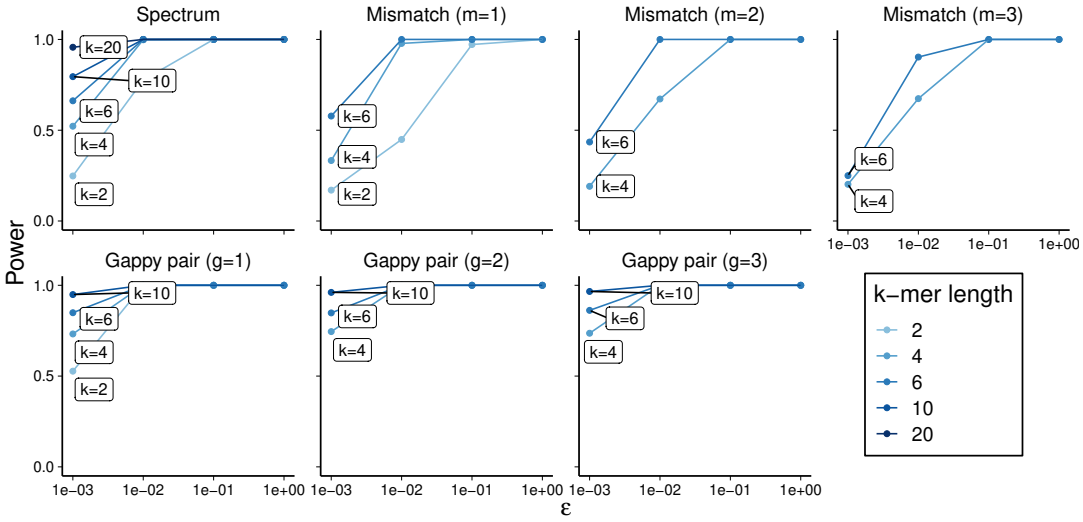


(B) FAME ($p = 1,147$)

FIGURE 5.17: Type I error rate of string kernels with different hyperparameters at a nominal significance level of 0.1. The solid red line denotes the nominal significance level (0.1) and the dashed lines show its 95% binomial proportion confidence interval. These results are for a group size of 200 using the CLR transform and $b = 3$ but are representative of all simulation scenarios tested. Generated from 1,000 replicates of Algorithm 7.1.



(A) Busselton ($p = 1,689$)



(B) FAME ($p = 1,147$)

FIGURE 5.18: Power of string kernels with different hyperparameters at a nominal significance level of 0.1. These results are for a group size of 200 using the CLR transform and $b = 3$ but are representative of all simulation scenarios tested. Generated from 1,000 replicates of Algorithm 7.1.

model (5.46) follows Xiao et al. (2018) and assumes that the relative abundance of each taxa in a sample is the relevant quantity in determining phenotype. A fictitious continuous host phenotype $y \in \mathbb{R}^n$ is generated from $Z \in [0,1]^{n \times p}$ using a linear model of the form

$$y = \beta Z + \eta\,, \quad \eta \sim \mathcal{N}(0, \rho^2)\,, \tag{5.46}$$

where $\beta \in \mathbb{R}^p$ are effect sizes, Z contains relative abundances satisfying $\sum_{j=1}^{p} Z_{ij} = 1\,, j = 1\,, \ldots\,, n$ and η is observation noise with variance ρ^2. The variance of βZ is fixed to 1 throughout and two noise-levels defined by one of $\rho \in \{0.3, 0.6\}$ were tested, corresponding to signal to noise ratios of $\frac{10}{3}$ and $\frac{10}{6}$.

OTU effect sizes

The phylogenetic component of the simulation is introduced via the OTU effect sizes β, which are assigned to clusters of OTUs in two scenarios, each of which represents a distinct hypothesis:

1. OTU effects are driven by the 16S rRNA gene sequence and so phylogenetically similar OTUs have similar effects; or
2. OTU effects are assigned at random and are unrelated to the tree and 16S rRNA gene sequence.

Scenario 1 is achieved by clustering OTUs in the same manner used in the two-sample test simulations with $\varepsilon = 0.1$ while Scenario 2 assigns clusters at random. The cluster sizes in the two scenarios have the same distribution. Given a set of clusters, a set of ten causal clusters are sampled without replacement and assigned cluster-level effects $\tilde{\beta} \sim \mathcal{N}(0, 10\, I_{10})$. The OTU level effects are given by

$$\beta_j = \begin{cases} \tilde{\beta}_k & \text{if OTU } j \text{ is in cluster } k \\ 0 & \text{otherwise} \end{cases}\,, \tag{5.47}$$

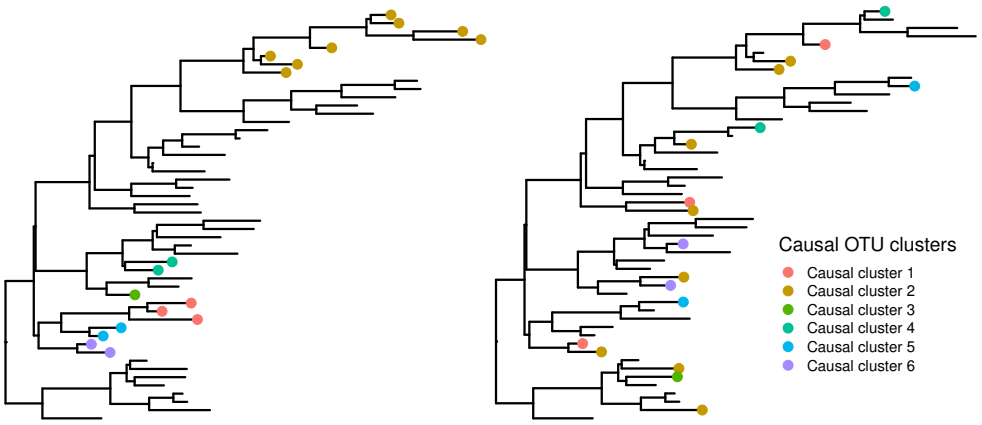
which results in a sparse β. The distribution of OTU effect sizes in the two scenarios is illustrated in Figure 5.19.

8.3 Gaussian process regression model

Given the OTU relative abundances Z and continuos host phenotype y, the aim is to investigate the performance of the GP regression model

$$f \sim \mathcal{GP}(0, k(\cdot, \cdot))\,, \tag{5.48}$$

for different kernel functions $k(\cdot, \cdot)$. Similarly to the two-sample testing simulations, the kernels include three non-phylogenetic kernels: (i) Linear, (ii) RBF and



(A) Effect sizes follow tree (B) Effect sizes unrelated to tree

FIGURE 5.19: Generating OTU effect sizes that are related to phylogeny (plot A) or are unrelated to phylogeny (plot B). Unmarked leaves denote OTUs with zero effect size in the phenotype model.

(iii) Matern32. As the phenotype model (5.46) explicitly assumes that relative abundances are driving the phenotype the UniFrac kernels are not included in these experiments.

The only phylogenetic kernel included here is the String kernel, with all three variants (Spectrum, Mismatch and Gappy pair) considered together as a single kernel. Unlike in the two-sample test example it is possible to consider these three variants as a single kernel in supervised learning as the hyperparameters k, m and g can be selected using the marginal likelihood. The same marginal likelihood optimisation procedure is used to learn the signal and noise variance estimates for the RBF, Matern32 and Linear kernels. In addition, the RBF and Matern32 kernels also learn a single lengthscale for all dimensions and using the median heuristic as the starting guess. The models are trained on the relative abundances Z.

The GP models are evaluated using their log-marginal likelihood (LML, calculated on the training set) and their log-predictive density (LPD) on the test set. The training set contains 80% of the samples while the test set contains the remaining 20%.

8.4 Full simulation procedure

A single replicate of the simulation setup proceeds as follows. Starting with a set of DMN concentrations estimated using one of the observed datasets α and the accompanying phylogenetic tree, $X \sim \text{DMN}(\hat{\pi}_1(\alpha))$, where $\hat{\pi}_1(\cdot)$ performs a full permutation of the OTUs ($\varepsilon = 1$ places all OTUs in a single cluster). The OTUs are then placed into clusters under one of the two hypotheses (using the phylogenetic tree or at random) and the cluster-level effect sizes are sampled from $\tilde{\beta} \sim \mathcal{N}(0, 10\, I_{10})$. The continuous phenotype is then generated from these effect sizes and relative abundances Z using (5.46) and a fixed noise variance $\rho \in \{0.3, 0.6\}$. A GP regression model is then trained with 80% of the samples using each of the four kernels and the LML and test LPD are recorded. This is repeated for 1,000 replicates.

8.5 Results

Log-marginal likelihoods

Figure 5.20 shows the difference in LML between a GP using a String kernel and between a GP using one of the three non-phylogenetic kernels (Linear, RBF and Matern32). The LML quantifies the fit of the GP to the training data with larger values indicating a better fit. The difference in LMLs between two GP regression models is the Bayes factor, which quantifies the relative strength of the hypotheses represented by the two models.

When the effect sizes of OTUs are assigned using the phylogenetic tree a GP with a string kernel has a larger LML value than any of the three non-phylogenetic kernels. This is observed in both the low-noise (top row) and high-noise (bottom row) settings and for all combinations of sample size $n \in \{200, 400\}$ and sample read dispersion $b \in \{3, 10, 30\}$. The String kernel has a larger LML than the RBF or Matern32 kernels under both scenarios because the underlying phenotype model is linear. However, the benefit of using a String kernel is larger when the OTU effects are related to phylogeny.

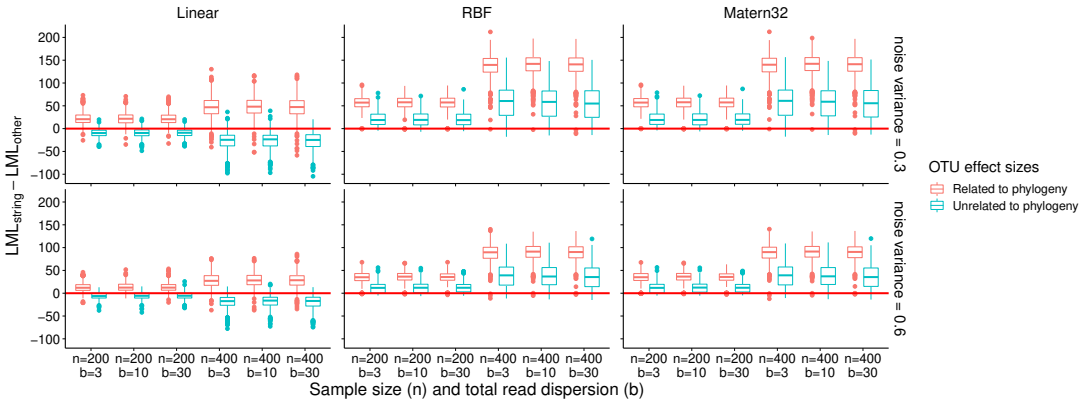
When OTU effect sizes are unrelated to phylogeny, using the Linear kernel results in larger LMLs than the String kernel. This is to be expected as in this case the Linear kernel represents the true model, in which case there is a linear relationship between OTU relative abundance and phenotype and no relationship between phylogenetic similarity and OTU effect size. However, in the case where the effect sizes depend on phylogeny the String kernel is a better model and so has a larger LML.

The two stationary kernels (RBF and Matern32) are included here as they are the most popular for GP regression modelling both generally and for 16S rRNA data specifically. These results show that a naive application of these kernels is not appropriate in this setting. Furthermore, given that the true phenotype model is linear it is expected that the Linear kernel should perform better than RBF or Matern32, which can both capture linear effects but are generally preferred because of their ability to capture higher-order interactions. As these higher-order interactions are not included in the phenotype model the additional complexity of the RBF and Matern32 kernels are a hindrance rather than a help.

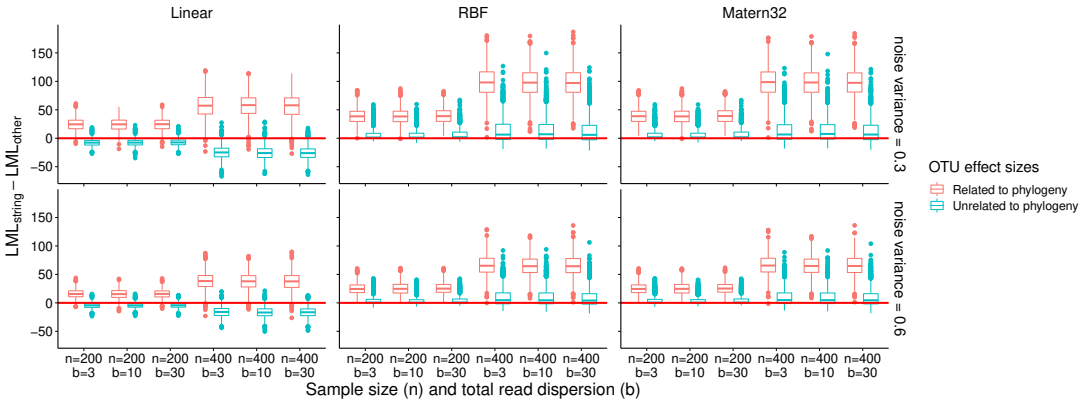
These results suggest a promising avenue for investigating different hypothesis about the nature of the relationship between community structure and phenotype in a given dataset. In these simulations the difference in LML between a GP with a String kernel and a GP with a Linear kernel is a reliable indicator of the extent to which the OTU effects are distributed according to the 16S rRNA sequences (see Figure 5.21). Such an analysis can therefore be used to identify whether the factors controlling a host trait are related to the observed 16S rRNA sequence or if they are driven by other factors (such as areas of the bacterial genome that have not been sequenced or environmental factors).

Log-predictive densities

The second quantity of interest when evaluating GP models is the LPD, which quantifies the predictive performance of the GP model on the held-out test data. The difference in LPD values for a GP using a String kernel and one using one of a Linear, RBF or Matern32 kernels are shown in Figure 5.22, which show similar behaviour to

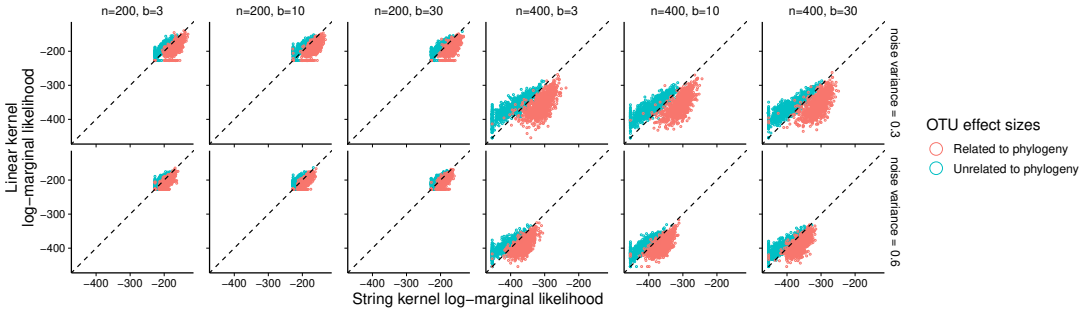


(A) Busselton ($p = 1,689$)

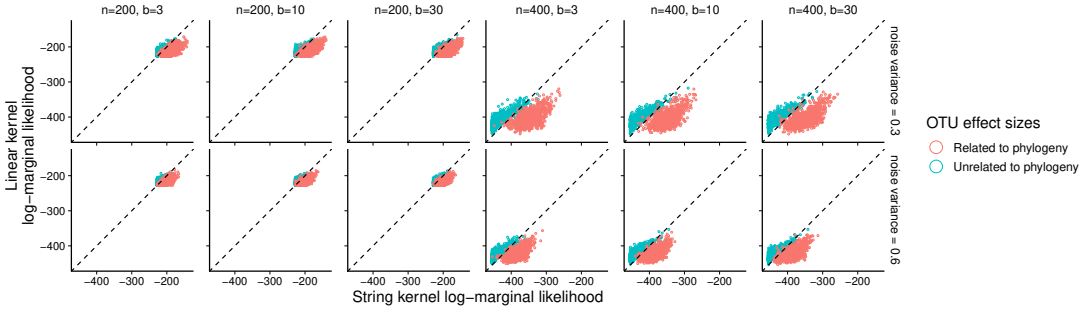


(B) FAME ($p = 1,147$)

FIGURE 5.20: $\text{LML}_{\text{string}} - \text{LML}_{\text{other}}$, where LML_k is the log-marginal likelihood of a GP regression with kernel k. The red line indicates where both kernels have the same log-marginal likelihood.



(A) Busselton ($p = 1,689$)



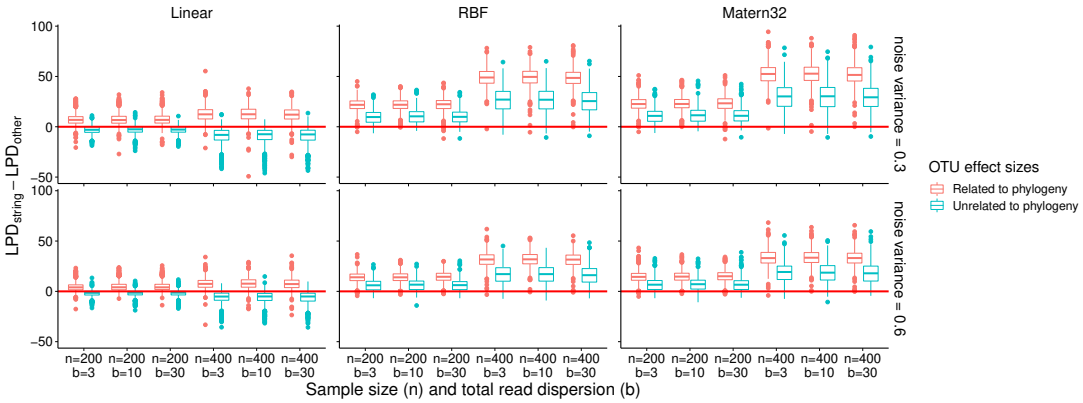
(B) FAME ($p = 1,147$)

FIGURE 5.21: Comparing the LML (log-marginal likelihood) of GP models trained with a String or Linear kernel differentiates between the two hypotheses. The dashed line indicates when the two kernels result in the same LML. The difference in LMLs between the two models is a Bayes factor.

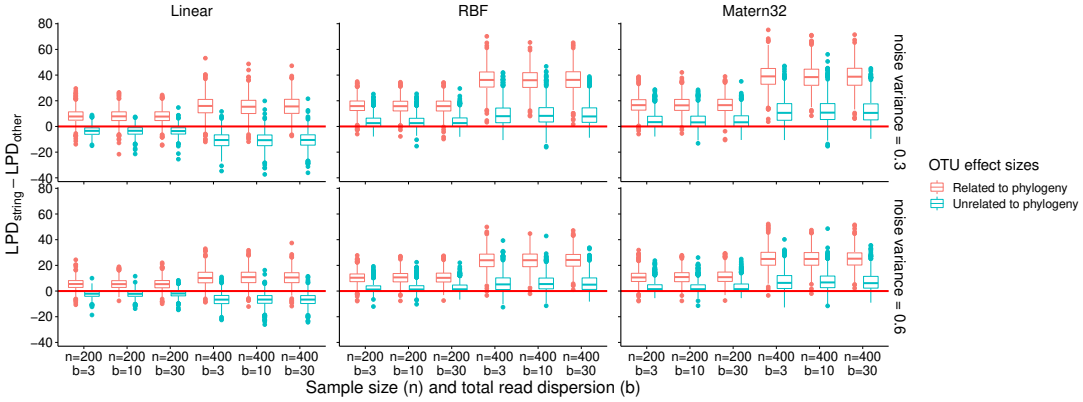
the LMLs. The Linear kernel exhibits better predictive performance than the String kernel (a higher median LPD across replicates) when the effect sizes are unrelated to phylogeny, while the String kernel has a higher median LPD when closely-related OTUs have identical effects. Both the RBF and Matern32 kernels exhibit worse predictive performance than the String kernel in both effect size scenarios. In some replicates the RBF or Matern32 have larger LML but lower LPD than the String kernel, which indicates that they are over-fitting the training data. Again, the benefit of using a String kernel over the RBF or Matern32 is larger when the OTU effect sizes are related to phylogeny.

8.6 Effect of string kernel hyperparameters

The previous results presented the LMLs and LPDs of the String kernel that maximised the marginal likelihood on the training data. However, it is of interest to investigate the behaviour of the GP model with respect to the String kernel hyperparameters k, m and g. Figure 5.23 shows the number of times each value of k, m and g were chosen during GP regression model selection using two datasets. For both datasets there is a preference for larger k-mer length, with values of $k < 5$ never chosen in either dataset when using the Spectrum kernel. There is also a dependence on the sample size, as when $n = 400$ the Gappy pair ($g = 3$) kernel is more likely to have the largest log-marginal likelihood than when $n = 200$. The Mismatch kernel is almost never selected in either dataset, suggesting that using a Spectrum or Gappy

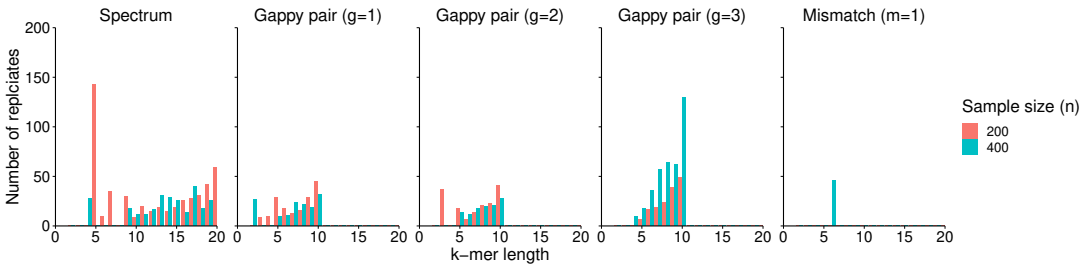


(A) Busselton ($p = 1,689$)

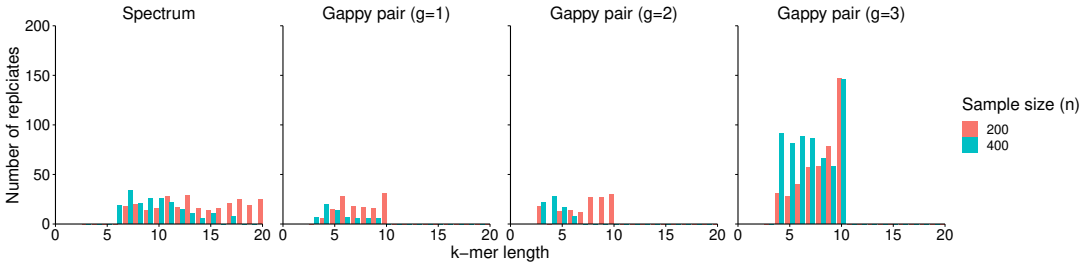


(B) FAME ($p = 1,147$)

FIGURE 5.22: $\text{LPD}_{\text{string}} - \text{LPD}_{\text{other}}$, where LPD_k is the test log-predictive density of a GP regression with kernel k. The red line indicates where both kernels have the same LPD on the test set.



(A) Busselton ($p = 1,689$)



(B) FAME ($p = 1,147$)

FIGURE 5.23: Number of times different String kernel hyperparameters are selected in 1,000 replicates of the GP regression experiments. String kernel hyperparameters are selected using the log-marginal likelihoods of the resulting GP model. These plots are for $b = 10$ and $\sigma^2 = 0.3$ but are representative of the results with other values.

pair kernel is always the preferred option as they are both cheaper to compute and lead to a larger LML.

9 Discussion

These results demonstrate the utility of using kernels to model the phylogenetic relationships present in microbial datasets in two tasks: (i) the kernel two-sample test and (ii) host-trait prediction using Gaussian process regression. Modelling phylogenetic relationships when performing the two-sample test results in a test that is sensitive to the phylogenetic scale of the differences between two populations. Two-sample tests using either the RBF or Matern32 kernel, which are the most commonly applied with 16S rRNA data in the literature, are not sensitive to phylogenetic scales as they weight all differences between OTUs equally, which can lead to misleading results. The regression simulations showed that when the effect sizes are assigned to OTUs based on phylogenetic relationships GP regression models with a String kernel have has a larger LML than a model using a non-phylogenetic kernel.

The two-sample test simulations demonstrated that popular characteristic kernels may not be appropriate for two-sample tests with 16S rRNA data, at least under the assumptions of these simulations. The most restrictive of these assumptions is the fact that the two groups were exactly equal in size throughout, which is clearly not realistic. They also only considered scenarios where differences between P and Q occurred through permutations of the underlying α, when there are many other ways for two populations to differ. However, this simulation setup was constructed to demonstrate the undesirable behaviours of the RBF and Matern32 kernel in this setting, as well as show that the phylogenetic kernels do not exhibit these behaviours.

This aim was achieved and these findings are sufficient to warn against using RBF and Matern32 kernels in a two-sample test on OTU-level data (or at least to exercise caution when performing such tests).

However, several hurdles remain in developing an ideal two-sample test based on String kernels. These results show that String kernels show sensitivity to the value of ϵ in the two-sample test but a method for tuning the String kernel hyperparameters to be sensitive to a desired value of ϵ is still required. This is left for future work.

The most interesting finding from the GP regression simulations is that a comparison of the LMLs from a string and linear (i.e. a Bayes factor) is a precise indicator of the distribution of OTU effects across the phylogenetic tree. As the tree is constructed from the 16S rRNA gene sequences this summary statistic therefore quantifies the degree to which the OTU effects are explained by 16S rRNA gene sequence variation. If a GP with a non-phylogenetic kernel has a larger LML than one with a phylogenetic kernel then the OTU effects must be explained by (i) variation in parts of the microbial sequence that have not been collected or (ii) by non-sequence (e.g. environmental) factors. This is therefore a novel way to approach this biologically relevant question in a Bayesian hypothesis framework.

However, this approach for hypothesis testing has only been shown to be effective when the assumptions of the simulation are met. The most important of these is that the relative abundance is the relevant quantity when relating community composition to host trait. While this is not especially restrictive and is commonly assumed in most analyses, it is still worth stating as a limitation. The GP regression simulations also assume a linear dependence (with sparse OTU effects) between relative abundance and the host trait. An interesting option for future work is to investigate the robustness of the results to mis-specification of the phenotype model (when the phenotype model contains non-linear dependencies but the phylogenetic kernel remains linear).

One of the benefits of this approach is that it can be applied as-is even as sequencing technologies improve. For example, there is a growing movement to replace OTUs (which use 97% sequence identity) with amplicon sequence variants (ASVs) which use 100% identity. Since ASVs are still defined by a single sequence the same analysis pipelines used here can be applied directly. This also extends to alternative sequences technologies for microbial datasets. While 16S rRNA is still the mainstay of bacterial sequencing it is limited by the relatively short region of bacterial DNA that is sequenced, which results in limited resolution (Jeong et al., 2021) as well as preventing more detailed functional analysis. Whole genome sequencing (WGS) offers many advantages over 16S rRNA, resulting from the fact that it provides the entire genomic sequence (Ranjan et al., 2016). However, WGS is far more expensive than 16S rRNA leading to some researchers adopting a hybrid approach where the two modalities are used to complement one another where possible (Regalado et al., 2020). Applying String kernels to WGS data would be prohibitively expensive in its current form due to the increased length of the representative sequences ($\sim$250 for 16S rRNA and $> 10^5$ for WGS). However, this could be achieved using existing work on Monte Carlo approximations to String kernels (Blakely et al., 2020), which would act as a drop-in replacement for the exact kernels used here. These analyses could also be extended easily to biological fields where variables are defined by strings, such as genetics, proteomics and transcriptomics.

The two datasets come from two very different studies (for example, FAME only contains sample from two severe lung diseases while Busselton contains relatively

mild asthmatics and healthy controls) and were collected and pre-processed separately. However, the results of the two-sample test and GP regression simulations showed high-levels of consistency between the two datasets. This suggests that this kernel-based approach may be widely applicable to a range of microbial studies, including other lung disease datasets and other non-lung sample sites (such as the gut). This can be explored further by validating these analysis pipelines on more datasets, such as those contained in the Microbiome Learning Repository (Vangay et al., 2019).

A final limitation of these experiments is that they focus on modelling the phylogenetic relationships amongst the OTUs and have largely neglected some other important features of OTU count data: sparsity and zero-inflation. While these features were present in the simulated OTU tables they were not explicitly modelled in the MMD two-sample test nor the GP regression models. Recent work has developed zero-inflated Gaussian processes where the kernels incorporate a latent model that predicts the presence or absence of a zero (Hegde et al., 2018). One of the many benefits of kernel methods is their modularity - it is straightforward to construct a GP that models both zero-inflated counts and phylogenetic relationships by taking a product or sum of the appropriate kernels. This modularity means kernel methods are a popular approach for biological data integration as their additive property (the sum of two kernel matrices is a valid kernel) enables the straightforward combination of heterogeneous data types (Daemen et al., 2009; Hériché et al., 2014; Mariette and Villa-Vialaneix, 2018).

Chapter 6

Discussion and Conclusions

This thesis describes three research projects concerning the application of non-parametric predictive models to biological sequence data. While the work in Chapters 3-5 is largely separate, they cover several closely-related themes that arise from the application of these methods in biological research.

Supervised learning forms a large part of the work in all three results chapters. The three chapters each reflect a different aspect of the main motivation for constructing predictive models in biomedical applications. These models usually act as an abstraction of the data-generating process and are used to gain biological insights into the underlying system rather than make predictions on unseen data.

In Chapter 3 a random forest classifier is used as a proxy for the two-sample test. The same classifiers are then used for differential abundance analysis via variable importance. This chapter is an empirical study of the behaviour of this common approach using a real dataset, which demonstrated that the results of a random forest-based two-sample test are largely robust to the choice of data transformation. This robustness was observed in the predictive performance of the model as well as in the results of popular hypothesis tests on the receiver-operating-characteristic curves. The behaviour of a differential abundance analysis (using variable importance analysis) was also explored for the first time using microbiome data and found to be largely robust, with some notable exceptions. These exceptions are data-dependent and so investigating the effect of data perturbations and transformations is required to achieve robust results in practice.

Random forests are sometimes considered interpretable due to their ability to compute these variable importance scores. A variable importance analysis is one of the most useful statistical tools for extracting biological knowledge from a predictive model and the ability to compute these scores which distinguishes random forests from other black box methods. This motivates Chapter 4, which presents a method of calculating *post-hoc* grouped variable importance scores for Bayesian neural networks and sparse Gaussian process regression models. The resulting method, GroupRATE, is able to effectively prioritise causal groups in two different simulation studies.

Chapter 5 also includes a supervised learning analysis using Gaussian process regression. Similarly to the other chapters, these models are not used explicitly for prediction but rather to investigate the role phylogenetic relationships in lung microbiome datasets using simulation studies. The predictive model in this chapter was used in a Bayesian hypothesis testing framework to investigate how taxa effects on host traits are distributed across the genetic tree. This is achieved using kernels that model the phylogenetic similarity between taxa using their 16S rRNA gene

sequence and string kernels. These kernels are also utilised in Chapter 5 in an alternative approach to the two-sample test explored in Chapter 3. Rather than an empirical study of a commonly-used approach, Chapter 5 describes a novel approach that considers phylogenetic relationships via these string kernels. The resulting test is sensitive to the phylogenetic scale of the difference between the two populations. This makes it more appropriate for use with 16S rRNA sequencing data than other popular kernels, with are liable to reject the null hypothesis over biologically irrelevant differences.

In addition to the future work outlined in the individual chapters, one common avenue for extending all three chapters is fully-Bayesian inference using Markov Chain Monte Carlo (MCMC). This provides a principled way to encode prior knowledge into these types of data-driven analyses, which is clearly desirable in this setting. While data-driven approaches are most useful when prior knowledge is unavailable or hard to specify, as more is learned about these problems it becomes more useful to encode this knowledge in predictive models.

While there is existing work on Bayesian non-parametric equivalents the random forest models in Chapter 3 (Matthew et al., 2015), the most popular Bayesian decision tree ensemble is Bayesian additive regression trees (BART, Chipman et al., 2010). A fully Bayesian treatment for the last layer Bayesian networks in Chapter 4 (as opposed to variational inference) would allow asymptotically exact inference for non-conjugate priors for the final layer. While priors for the features of a large neural network would be difficult to specify, this could be mitigated by enforcing a sparse structure on the inner layer weights using biological annotations (Demetci et al., 2021). A fully Bayesian treatment of the Gaussian process used for trait prediction in Chapter 5 would give posterior samples of the kernel hyper-parameters using MCMC, which would allow variable selection to be combined with the phylogenetic modelling provided by string kernels.

In conclusion, non-parametric predictive models are an increasingly useful tool for data-driven analyses of biological systems using sequence data. As the volume and complexity of biological datasets continue to increase the main bottleneck in biological research has moved from data acquisition to data analysis (J. Chang, 2015). Extensive future research is therefore required to both better understand these non-parametric methods and increase their utility as they continue to increase in popularity.

Bibliography

Abadi, Martın et al. (2016). "{TensorFlow}: A System for {Large-Scale} Machine Learning". In: *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pp. 265–283.

Adebayo, Julius et al. (2018). "Sanity checks for saliency maps". In: *Advances in Neural Information Processing Systems* 31.

Ai, Dongmei et al. (2019). "Identifying gut microbiota associated with colorectal cancer using a zero-inflated lognormal model". In: *Frontiers in Microbiology* 10, p. 826.

Aitchison, John (1982). "The statistical analysis of compositional data". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 44.2, pp. 139–160.

Alqaraawi, Ahmed et al. (2020). "Evaluating saliency map explanations for convolutional neural networks: a user study". In: *Proceedings of the 25th International Conference on Intelligent User Interfaces*, pp. 275–285.

Altmann, André et al. (2010). "Permutation importance: a corrected feature importance measure". In: *Bioinformatics* 26.10, pp. 1340–1347.

Amayri, Ola and Nizar Bouguila (2009). "Improved online support vector machines spam filtering using string kernels". In: *Iberoamerican Congress on Pattern Recognition*. Springer, pp. 621–628.

Amin, Reshma et al. (2010). "The effect of chronic infection with Aspergillus fumigatus on lung function and hospitalization in patients with cystic fibrosis". In: *Chest* 137.1, pp. 171–176.

Andrianakis, Ioannis and Peter G Challenor (2012). "The effect of the nugget on Gaussian process emulators of computer models". In: *Computational Statistics & Data Analysis* 56.12, pp. 4215–4228.

Archer, Kellie J and Ryan V Kimes (2008). "Empirical characterization of random forest variable importance measures". In: *Computational Statistics & Data Analysis* 52.4, pp. 2249–2260.

Arun, Nishanth et al. (2021). "Assessing the trustworthiness of saliency maps for localizing abnormalities in medical imaging". In: *Radiology: Artificial Intelligence* 3.6, e200267.

Ba, Jimmy and Rich Caruana (2014). "Do deep nets really need to be deep?" In: *Advances in Neural Information Processing Systems*, pp. 2654–2662.

Bach, Sebastian et al. (2015). "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation". In: *PloS one* 10.7, e0130140.

Badri, Michelle et al. (2018). "Normalization methods for microbial abundance data strongly affect correlation estimates". In: *BioRxiv*, p. 406264.

Banerjee, Kalins et al. (2019). "An adaptive multivariate two-sample test with application to microbiome differential abundance analysis". In: *Frontiers in genetics* 10, p. 350.

Barber, David and Christopher M Bishop (1998). "Ensemble Learning in Bayesian Neural Networks". In: *NATO ASI Series F Computer and Systems Sciences* 168, pp. 215–238.

Bardenhorst, Sven Kleine et al. (2021). "Data Analysis Strategies for Microbiome Studies in Human Populations—a Systematic Review of Current Practice". In: *Msystems* 6.1.

Behnamian, Amir et al. (2017). "A systematic approach for variable selection with random forests: achieving stable variable importance values". In: *IEEE Geoscience and Remote Sensing Letters* 14.11, pp. 1988–1992.

Ben-Hur, Asa et al. (2008). "Support vector machines and kernels for computational biology". In: *PLoS Computational Biology* 4.10, e1000173.

Bergen, Giel HH van et al. (2020). "Bayesian neural networks with variable selection for prediction of genotypic values". In: *Genetics Selection Evolution* 52.1, pp. 1–14.

Bien, Jacob et al. (2013). "A lasso for hierarchical interactions". In: *Annals of Statistics* 41.3, p. 1111.

Blakely, Derrick et al. (2020). "FastSK: fast sequence analysis with gapped string kernels". In: *Bioinformatics* 36.Supplement_2, pp. i857–i865.

Blei, David M et al. (2017). "Variational inference: A review for statisticians". In: *Journal of the American statistical Association* 112.518, pp. 859–877.

Blundell, Charles et al. (2015). "Weight uncertainty in neural network". In: *International Conference on Machine Learning*. PMLR, pp. 1613–1622.

Bolyen, Evan et al. (2019). "Reproducible, interactive, scalable and extensible microbiome data science using QIIME 2". In: *Nature Biotechnology* 37.8, pp. 852–857.

Bonferroni, Carlo (1936). "Teoria statistica delle classi e calcolo delle probabilita". In: *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze* 8, pp. 3–62.

Boyd, Kendrick et al. (2013). "Area under the precision-recall curve: point estimates and confidence intervals". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 451–466.

Breiman, Leo et al. (1984). *Classification and regression trees*. CRC Press.

Brosse, Nicolas et al. (2020). "On last-layer algorithms for classification: Decoupling representation from uncertainty estimation". In: *arXiv preprint arXiv:2001.08049*.

Budden, Kurtis F et al. (2017). "Emerging pathogenic links between microbiota and the gut–lung axis". In: *Nature Reviews Microbiology* 15.1, pp. 55–63.

Bzdok, D et al. (2018). *Statistics versus machine learning. Nat. Meth. 15 (4), 233–234 (2018)*.

Cai, Haiyan et al. (2020). "Two-sample test based on classification probability". In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 13.1, pp. 5–13.

Calders, Toon and Szymon Jaroszewicz (2007). "Efficient AUC optimization for classification". In: *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, pp. 42–53.

Calgaro, Matteo et al. (2020). "Assessment of statistical methods from single cell, bulk RNA-seq, and metagenomics applied to microbiome data". In: *Genome Biology* 21.1, pp. 1–31.

Calle, M Luz and Vıctor Urrea (2011). "Stability of Random Forest importance measures". In: *Briefings in Bioinformatics* 12.1, pp. 86–89.

Cao, Quy et al. (2021). "Effects of rare microbiome taxa filtering on statistical analysis". In: *Frontiers in Microbiology*, p. 3203.

Cekikj, Miodrag et al. (2022). "Understanding the role of the microbiome in cancer diagnostics and therapeutics by creating and utilizing ML models". In: *Applied Sciences* 12.9, p. 4094.

Chang, Jeffrey (2015). "Core services: reward bioinformaticians". In: *Nature* 520.7546, pp. 151–152.

Changyong, FENG et al. (2014). "Log-transformation and its implications for data analysis". In: *Shanghai archives of psychiatry* 26.2, p. 105.

Che, Zhengping et al. (2016). "Interpretable deep models for ICU outcome prediction". In: *AMIA Annual Symposium Proceedings*. Vol. 2016. American Medical Informatics Association, p. 371.

Chen, Jun, Emily King, et al. (2018). "An omnibus test for differential distribution analysis of microbiome sequencing data". In: *Bioinformatics* 34.4, pp. 643–651.

Chen, Jun and Hongzhe Li (2013a). "Kernel methods for regression analysis of microbiome compositional data". In: *Topics in Applied Statistics*. Springer, pp. 191–201.

— (2013b). "Variable selection for sparse Dirichlet-multinomial regression with an application to microbiome data analysis". In: *The Annals of Applied Statistics* 7.1.

Chen, Lijuan et al. (2020). "Comparative Analysis of Soil Microbiome Profiles in the Companion Planting of White Clover and Orchard Grass Using 16S rRNA Gene Sequencing Data". In: *Frontiers in Plant sSience*, p. 1431.

Chen, Xi and Hemant Ishwaran (2012). "Random forests for genomic data analysis". In: *Genomics* 99.6, pp. 323–329.

Chipman, Hugh A et al. (2010). "BART: Bayesian additive regression trees". In: *The Annals of Applied Statistics* 4.1, pp. 266–298.

Chotirmall, Sanjay H et al. (2010). "Sputum Candida albicans presages FEV1 decline and hospital-treated exacerbations in cystic fibrosis". In: *Chest* 138.5, pp. 1186–1195.

Clark, Sarah E (2020). "Commensal bacteria in the upper respiratory tract regulate susceptibility to infection". In: *Current Opinion in Immunology* 66, pp. 42–49.

Clarridge III, Jill E (2004). "Impact of 16S rRNA gene sequence analysis for identification of bacteria on clinical microbiology and infectious diseases". In: *Clinical Microbiology Reviews* 17.4, pp. 840–862.

Consortium, Tabula Muris et al. (2018). "Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris." In: *Nature* 562.7727, p. 367.

Corrigan, A et al. (2018). "The use of random forests modelling to detect yeast-mannan sensitive bacterial changes in the broiler cecum". In: *Scientific Reports* 8.1, pp. 1–13.

Couronné, Raphael et al. (2018). "Random forest versus logistic regression: a large-scale benchmark experiment". In: *BMC Bioinformatics* 19.1, pp. 1–14.

Crawford, Lorin et al. (2019). "Variable prioritization in nonlinear black box methods: A genetic association case study". In: *The Annals of Applied Statistics* 13.2, p. 958.

Cryan, John F et al. (2019). "The microbiota-gut-brain axis". In: *Physiological Reviews*.

Cuthbertson, Leah, Imogen Felton, et al. (2021). "The fungal airway microbiome in cystic fibrosis and non-cystic fibrosis bronchiectasis". In: *Journal of Cystic Fibrosis* 20.2, pp. 295–302.

Cuthbertson, Leah, Sofia Forslund, et al. (2022, Manuscript in prepration). "The genomics of airway microbiota".

Cybenko, George (1989). "Approximation by superpositions of a sigmoidal function". In: *Mathematics of Control, Signals and Systems* 2.4, pp. 303–314.

Daemen, Anneleen et al. (2009). "A kernel-based integration of genome-wide data for clinical decision support". In: *Genome Medicine* 1.4, pp. 1–17.

Das, A et al. (2021). "The fecal mycobiome in patients with Irritable Bowel Syndrome". In: *Scientific Reports* 11.1, pp. 1–9.

Davoodi, Raheleh and Mohammad Hassan Moradi (2018). "Mortality prediction in intensive care units (ICUs) using a deep rule-based fuzzy classifier". In: *Journal of Biomedical Informatics* 79, pp. 48–59.

De Valpine, Perry and Alexandra N Harmon-Threatt (2013). "General models for resource use or other compositional count data using the Dirichlet-multinomial distribution". In: *Ecology* 94.12, pp. 2678–2687.

Degenhardt, Frauke et al. (2019). "Evaluation of variable selection methods for random forests and omics data sets". In: *Briefings in Bioinformatics* 20.2, pp. 492–503.

DeGrave, Alex J et al. (2021). "AI for radiographic COVID-19 detection selects shortcuts over signal". In: *Nature Machine Intelligence* 3.7, pp. 610–619.

DeLong, Elizabeth R et al. (1988). "Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach". In: *Biometrics*, pp. 837–845.

Demetci, Pinar et al. (2021). "Multi-scale inference of genetic trait architecture using biologically annotated neural networks". In: *PLoS Genetics* 17.8, e1009754.

Demler, Olga V et al. (2012). "Misuse of DeLong test to compare AUCs for nested models". In: *Statistics in Medicine* 31.23, pp. 2577–2587.

Deveau, Aurelie et al. (2018). "Bacterial–fungal interactions: ecology, mechanisms and challenges". In: *FEMS Microbiology Reviews* 42.3, pp. 335–352.

Dickson, Robert P et al. (2016). "The microbiome and the respiratory tract". In: *Annual review of physiology* 78, pp. 481–504.

Ding, Li et al. (2021). "Pathogen Metagenomics Reveals Distinct Lung Microbiota Signatures Between Bacteriologically Confirmed and Negative Tuberculosis Patients". In: *Frontiers in Cellular and Infection Microbiology* 11.

Doshi-Velez, F and B Kim (2017). "Roadmap for a Rigorous Science of Interpretability. arxiv". In: *arXiv preprint arXiv:1702.08608*.

Esteva, Andre et al. (2017). "Dermatologist-level classification of skin cancer with deep neural networks". In: *Nature* 542.7639, pp. 115–118.

Faner, Rosa et al. (2017). "The microbiome in respiratory medicine: current challenges and future perspectives". In: *European Respiratory Journal* 49.4, p. 1602086.

Farquhar, Sebastian et al. (2020). "Try Depth Instead of Weight Correlations: Mean-field is a Less Restrictive Assumption for Deeper Networks". In: *arXiv preprint arXiv:2002.03704*.

Fawcett, Tom (2006). "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8, pp. 861–874.

Fernandes, Andrew D et al. (2014). "Unifying the analysis of high-throughput sequencing datasets: characterizing RNA-seq, 16S rRNA gene sequencing and selective growth experiments by compositional data analysis". In: *Microbiome* 2.1, pp. 1–13.

Fernández-Delgado, Manuel et al. (2014). "Do we need hundreds of classifiers to solve real world classification problems?" In: *The Journal of Machine Learning Research* 15.1, pp. 3133–3181.

Filos, Angelos et al. (2019). "A Systematic Comparison of Bayesian Deep Learning Robustness in Diabetic Retinopathy Tasks". In: *arXiv preprint arXiv:1912.10481*.

Flaxman, Seth et al. (2016). "Bayesian learning of kernel embeddings". In: *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pp. 182–191.

Folgoc, Loic Le et al. (2021). "Is MC Dropout Bayesian?" In: *arXiv preprint arXiv:2110.04286*.

Friedman, Jerome H (2001). "Greedy function approximation: a gradient boosting machine". In: *Annals of Statistics*, pp. 1189–1232.

Fu, Yu et al. (2020). "Pan-cancer computational histopathology reveals mutations, tumor composition and prognosis". In: *Nature Cancer* 1.8, pp. 800–810.

Gagnon-Bartsch, Johann and Yotam Shem-Tov (2016). "The classification permutation test: A nonparametric test for equality of multivariate distributions". In: *arXiv preprint arXiv:1611.06408*.

Gal, Yarin and Zoubin Ghahramani (2016). "Dropout as a bayesian approximation: Representing model uncertainty in deep learning". In: *International Conference on Machine Learning*, pp. 1050–1059.

Galili, Tal and Isaac Meilijson (2016). "Splitting matters: how monotone transformation of predictor variables may improve the predictions of decision tree models". In: *arXiv preprint arXiv:1611.04561*.

Gao, Xiang et al. (2017). "A dirichlet-multinomial bayes classifier for disease diagnosis with microbial compositions". In: *Msphere* 2.6, e00536–17.

Garg, Manik et al. (2021). "Tumour gene expression signature in primary melanoma predicts long-term outcomes". In: *Nature Communications* 12.1, pp. 1–14.

Genuer, Robin et al. (2010). "Variable selection using random forests". In: *Pattern Recognition Letters* 31.14, pp. 2225–2236.

Ghandi, Mahmoud et al. (2014). "Enhanced regulatory sequence prediction using gapped k-mer features". In: *PLoS Computational Biology* 10.7, e1003711.

Ghorbani, Amirata et al. (2019). "Interpretation of neural networks is fragile". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01, pp. 3681–3688.

Ghosh, Soumya et al. (2019). "Model Selection in Bayesian Neural Networks via Horseshoe Priors". In: *Journal of Machine Learning Research* 20.182, pp. 1–46.

Giliberti, Renato et al. (2022). "Host phenotype classification from human microbiome data is mainly driven by the presence of microbial taxa". In: *PLoS Computational Biology* 18.4, e1010066.

Gilpin, Leilani H et al. (2018). "Explaining explanations: An overview of interpretability of machine learning". In: *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*. IEEE, pp. 80–89.

Ginsburg, Geoffrey S and Kathryn A Phillips (2018). "Precision medicine: from science to value". In: *Health Affairs* 37.5, pp. 694–701.

Gloor, Gregory B et al. (2017). "Microbiome datasets are compositional: and this is not optional". In: *Frontiers in Microbiology* 8, p. 2224.

Goodfellow, Ian et al. (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Graves, Alex (2011). "Practical Variational Inference for Neural Networks". In: *Advances in Neural Information Processing Systems*, pp. 2348–2356.

— (2016). "Stochastic backpropagation through mixture density distributions". In: *arXiv preprint arXiv:1607.05690*.

Graves, Alex et al. (2013). "Speech recognition with deep recurrent neural networks". In: *2013 IEEE international Conference on Acoustics, Speech and Signal Processing*. IEEE, pp. 6645–6649.

Greener, Joe G et al. (2022). "A guide to machine learning for biologists". In: *Nature Reviews Molecular Cell Biology* 23.1, pp. 40–55.

Greenwell, Brandon and Maintainer Brandon Greenwell (2021). "Package 'fastshap'". In:

Gregorutti, Baptiste et al. (2015). "Grouped variable importance with random forests and application to multiple functional data analysis". In: *Computational Statistics & Data Analysis* 90, pp. 15–35.

— (2017). "Correlation and variable importance in random forests". In: *Statistics and Computing* 27.3, pp. 659–678.

Gretton, Arthur et al. (2012). "A kernel two-sample test". In: *The Journal of Machine Learning Research* 13.1, pp. 723–773.

Guo, Chuan et al. (2017). "On calibration of modern neural networks". In: *International Conference on Machine Learning*. PMLR, pp. 1321–1330.

Guo, Fangjian et al. (2016). "Boosting variational inference". In: *arXiv preprint arXiv:1611.05559*.

Guyon, Isabelle et al. (2002). "Gene selection for cancer classification using support vector machines". In: *Machine Learning* 46.1, pp. 389–422.

Halsey, Lewis G (2019). "The reign of the p-value is over: what alternative analyses could we employ to fill the power vacuum?" In: *Biology Letters* 15.5, p. 20190174.

Harrison, Joshua G et al. (2020). "Dirichlet-multinomial modelling outperforms alternatives for analysis of microbiome and other ecological count data". In: *Molecular Ecology Resources* 20.2, pp. 481–497.

Hawinkel, Stijn et al. (2020). "Sequence count data are poorly fit by the negative binomial distribution". In: *PloS one* 15.4, e0224909.

He, Bobby et al. (2020). "Bayesian deep ensembles via the neural tangent kernel". In: *Advances in Neural Information Processing Systems* 33, pp. 1010–1022.

Hediger, Simon et al. (2022). "On the use of random forest for two-sample testing". In: *Computational Statistics & Data Analysis*, p. 107435.

Hegde, Pashupati et al. (2018). "Variational zero-inflated Gaussian processes with sparse kernels". In: *arXiv preprint arXiv:1803.05036*.

Heinze, Georg et al. (2018). "Variable selection–A review and recommendations for the practicing statistician". In: *Biometrical Journal* 60.3, pp. 431–449.

Hensman, James et al. (2015). "Scalable variational Gaussian process classification". In: *Artificial Intelligence and Statistics*. PMLR, pp. 351–360.

Hériché, Jean-Karim et al. (2014). "Integration of biological data by kernels on graph nodes allows prediction of new genes involved in mitotic chromosome condensation". In: *Molecular Biology of the Cell* 25.16, pp. 2522–2536.

Hicks, Steven A et al. (2022). "On evaluation metrics for medical applications of artificial intelligence". In: *Scientific Reports* 12.1, pp. 1–9.

Higgins, Irina et al. (2016). "beta-vae: Learning basic visual concepts with a constrained variational framework". In: *International Conference on Learning Representations (ICLR)*.

Hilty, Markus et al. (2010). "Disordered microbial communities in asthmatic airways". In: *PloS one* 5.1, e8578.

Hinton, Geoffrey E and Drew Van Camp (1993). "Keeping Neural Networks Simple by Minimizing the Description Length of the Weights". In: *Proceedings of the Sixth Annual Conference on Computational Learning Theory*. ACM, pp. 5–13.

Hinton, Geoffrey, Nitsh Srivastava, et al. (2012). "Neural networks for machine learning". In: *Coursera, video lectures* 264, p. 1.

Hinton, Geoffrey, Oriol Vinyals, et al. (2015). "Distilling the knowledge in a neural network". In:

Hoerl, Arthur E and Robert W Kennard (1970). "Ridge regression: Biased estimation for nonorthogonal problems". In: *Technometrics* 12.1, pp. 55–67.

Holmes, Ian et al. (2012). "Dirichlet multinomial mixtures: generative models for microbial metagenomics". In: *PloS one* 7.2, e30126.

Hooker, Giles et al. (2021). "Unrestricted permutation forces extrapolation: variable importance requires at least one more model, or there is no free variable importance". In: *Statistics and Computing* 31.6, pp. 1–16.

Hormozdiari, Farhad et al. (2015). "Identification of causal genes for complex traits". In: *Bioinformatics* 31.12, pp. i206–i213.

Huang, Chunrong et al. (2020). "Fungal and bacterial microbiome dysbiosis and imbalance of trans-kingdom network in asthma". In: *Clinical and Translational Allergy* 10.1, pp. 1–13.

Huang, Ying (2016). "Evaluating and comparing biomarkers with respect to the area under the receiver operating characteristics curve in two-phase case–control studies". In: *Biostatistics* 17.3, pp. 499–522.

Huggins, Jonathan et al. (2020). "Validated variational inference via practical posterior error bounds". In: *International Conference on Artificial Intelligence and Statistics*, pp. 1792–1802.

Ish-Horowicz, Jonathan, Leah Cuthbertson, et al. (2022). "Machine learning for exploring microbial inter-kingdom associations in Cystic Fibrosis and Bronchiectasis". In: *bioRxiv*.

Ish-Horowicz, Jonathan, Evgeny Tankhilevich, et al. (2020). "GpABC: a Julia package for approximate Bayesian computation with Gaussian process emulation". In: *Bioinformatics* 36.10, pp. 3286–3287.

Ish-Horowicz, Jonathan, Dana Udwin, et al. (2019). "Interpreting deep neural networks through variable importance". In: *arXiv preprint arXiv:1901.09839*.

Ishwaran, Hemant, Udaya B Kogalur, Xi Chen, et al. (2011). "Random survival forests for high-dimensional data". In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 4.1, pp. 115–132.

Ishwaran, Hemant, Udaya B Kogalur, Eiran Z Gorodeski, et al. (2010). "High-dimensional variable selection for survival data". In: *Journal of the American Statistical Association* 105.489, pp. 205–217.

Ishwaran, Hemant and Min Lu (2019). "Standard errors and confidence intervals for variable importance in random forest regression, classification, and survival". In: *Statistics in Medicine* 38.4, pp. 558–582.

Janitza, Silke et al. (2018). "A computationally fast variable importance test for random forests for high-dimensional data". In: *Advances in Data Analysis and Classification* 12.4, pp. 885–915.

Jasner, Yoel et al. (2021). "Microbiome Preprocessing Machine Learning Pipeline". In: *Frontiers in Immunology* 12.

Jeong, Jinuk et al. (2021). "The effect of taxonomic classification by full-length 16S rRNA sequencing with a synthetic long-read technology". In: *Scientific Reports* 11.1, pp. 1–12.

Jong, Tristan V de et al. (2019). "Gene expression variability: the other dimension in transcriptome analysis". In: *Physiological Genomics* 51.5, pp. 145–158.

Jospin, Laurent Valentin et al. (2022). "Hands-on Bayesian neural networks—A tutorial for deep learning users". In: *IEEE Computational Intelligence Magazine* 17.2, pp. 29–48.

Jović, Alan et al. (2015). "A review of feature selection methods with applications". In: *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. Ieee, pp. 1200–1205.

Jumper, John et al. (2021). "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873, pp. 583–589.

Kennedy, Katherine et al. (2014). "Evaluating bias of Illumina-based bacterial 16S rRNA gene profiles". In: *Applied and Environmental Microbiology* 80.18, pp. 5717–5722.

Kermany, Daniel S et al. (2018). "Identifying medical diagnoses and treatable diseases by image-based deep learning". In: *Cell* 172.5, pp. 1122–1131.

Kim, Been et al. (2016). "Examples are not enough, learn to criticize! Criticism for interpretability". In: *Advances in Neural Information Processing Systems* 29.

Kim, Ilmun et al. (2021). "Classification accuracy as a proxy for two-sample testing". In: *The Annals of Statistics* 49.1, pp. 411–434.

Kim, Juhyun et al. (2018). "MGLM: an R package for multivariate categorical data analysis". In: *The R journal* 10.1, p. 73.

Kimmel, Alan R and Brian Oliver (2006). *DNA Microarrays, Part B: Databases and Statistics*. Elsevier.

Kindermans, Pieter-Jan et al. (2016). "Investigating the influence of noise and distractors on the interpretation of neural networks". In: *arXiv preprint arXiv:1611.07270*.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Kingma, Diederik P and Max Welling (2013). "Auto-encoding variational Bayes". In: *arXiv preprint arXiv:1312.6114*.

Kingma, Durk P et al. (2015). "Variational dropout and the local reparameterization trick". In: *Advances in Neural Information Processing Systems*, pp. 2575–2583.

Koh, Hyunwook et al. (2019). "A distance-based kernel association test based on the generalized linear mixed model for correlated microbiome studies". In: *Frontiers in Genetics* 10, p. 458.

Komiyama, Yusuke et al. (2016). "Automatic generation of bioinformatics tools for predicting protein–ligand binding sites". In: *Bioinformatics* 32.6, pp. 901–907.

Koslovsky, Matthew D and Marina Vannucci (2021). "Dirichlet-Multinomial Regression Models with Bayesian Variable Selection for Microbiome Data". In: *Statistical Analysis of Microbiome Data*. Springer, pp. 249–270.

Kristiadi, Agustinus et al. (2020). "Being bayesian, even just a bit, fixes overconfidence in relu networks". In: *International Conference on Machine Learning*. PMLR, pp. 5436–5446.

Krizhevsky, Alex et al. (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*, pp. 1097–1105.

Kuhn, Max (2015). "Caret: classification and regression training". In: *Astrophysics Source Code Library*, ascl–1505.

Kuksa, Pavel P (2013). "Biological sequence classification with multivariate string kernels". In: *IEEE/ACM transactions on computational biology and bioinformatics* 10.5, pp. 1201–1210.

Kummerer, Matthias et al. (2018). "Saliency benchmarking made easy: Separating models, maps and metrics". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 770–787.

Kurtz, Zachary D et al. (2015). "Sparse and compositionally robust inference of microbial ecological networks". In: *PLoS Computational Biology* 11.5, e1004226.

La Rosa, Patricio S et al. (2012). "Hypothesis testing and power calculations for taxonomic-based human microbiome data". In: *PloS one* 7.12, e52078.

Lakshminarayanan, Balaji et al. (2017). "Simple and scalable predictive uncertainty estimation using deep ensembles". In: *Advances in Neural Information Processing Systems* 30.

Larsen, Nadja et al. (2010). "Gut microbiota in human adults with type 2 diabetes differs from non-diabetic adults". In: *PloS one* 5.2, e9085.

Lázaro-Gredilla, Miguel and Anıbal R Figueiras-Vidal (2010). "Marginalized neural network mixtures for large-scale regression". In: *IEEE transactions on neural networks* 21.8, pp. 1345–1351.

Leclercq, Mickael et al. (2019). "Large-scale automatic feature selection for biomarker discovery in high-dimensional OMICs data". In: *Frontiers in genetics* 10, p. 452.

LeCun, Yann (1998). "The MNIST database of handwritten digits". In: *http://yann. lecun. com/exdb/mnist/*.

LeDell, Erin, Maya Petersen, and Mark van der Laan (2015). "Computationally efficient confidence intervals for cross-validated area under the ROC curve estimates". In: *Electronic Journal of Statistics* 9.1, p. 1583.

LeDell, Erin, Maya Petersen, Mark van der Laan, and Maintainer Erin LeDell (2022). *Package 'cvAUC'*.

Leibig, Christian et al. (2017). "Leveraging uncertainty information from deep neural networks for disease detection". In: *Scientific Reports* 7.1, pp. 1–14.

Leiner, Tim et al. (2021). "Bringing AI to the clinic: blueprint for a vendor-neutral AI deployment infrastructure". In: *Insights into Imaging* 12.1, pp. 1–11.

Leonelli, Sabina (2016). "Data-centric biology". In: *Data-Centric Biology*. University of Chicago Press.

Leslie, Christina, Eleazar Eskin, and William Stafford Noble (2001). "The spectrum kernel: A string kernel for SVM protein classification". In: *Biocomputing 2002*. World Scientific, pp. 564–575.

Leslie, Christina, Eleazar Eskin, Jason Weston, et al. (2003). "Mismatch string kernels for SVM protein classification". In: *Advances in Neural Information Processing Systems*, pp. 1441–1448.

Leslie, Christina and Rui Kuang (2003). "Fast kernels for inexact string matching". In: *Learning Theory and Kernel Machines*. Springer, pp. 114–128.

Lim, Michael and Trevor Hastie (2015). "Learning interactions via hierarchical group-lasso regularization". In: *Journal of Computational and Graphical Statistics* 24.3, pp. 627–654.

Lin, Huang and Shyamal Das Peddada (2020). "Analysis of microbial compositions: a review of normalization and differential abundance analysis". In: *NPJ Biofilms and Microbiomes* 6.1, pp. 1–13.

Lindén, Andreas and Samu Mäntyniemi (2011). "Using the negative binomial distribution to model overdispersion in ecological count data". In: *Ecology* 92.7, pp. 1414–1421.

Lippert, Christoph et al. (2011). "FaST linear mixed models for genome-wide association studies". In: *Nature Methods* 8.10, pp. 833–835.

Lipton, Zachary C (2018). "The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery." In: *Queue* 16.3, pp. 31–57.

Liu, DC and J Nocedal (1989). "On the limited memory method for large scale optimization: Mathematical Programming B". In:

Liu, Xiaoqing et al. (2021). "Advances in deep learning-based medical image analysis". In: *Health Data Science* 2021.

Lodhi, Huma et al. (2002). "Text classification using string kernels". In: *Journal of Machine Learning Research* 2.Feb, pp. 419–444.

Lopatkin, Allison J and James J Collins (2020). "Predictive biology: modelling, understanding and harnessing microbial complexity". In: *Nature Reviews Microbiology* 18.9, pp. 507–520.

Louizos, Christos and Max Welling (2016). "Structured and efficient variational deep learning with matrix gaussian posteriors". In: *International Conference on Machine Learning*, pp. 1708–1716.

— (2017). "Multiplicative normalizing flows for variational bayesian neural networks". In: *arXiv preprint arXiv:1703.01961*.

Love, Michael I et al. (2014). "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2". In: *Genome Biology* 15.12, pp. 1–21.

Lozupone, Catherine A et al. (2007). "Quantitative and qualitative β diversity measures lead to different insights into factors that structure microbial communities". In: *Applied and Environmental Microbiology* 73.5, pp. 1576–1585.

Lozupone, Catherine and Rob Knight (2005). "UniFrac: a new phylogenetic method for comparing microbial communities". In: *Applied and Environmental Microbiology* 71.12, pp. 8228–8235.

Lu, Zhou et al. (2017). "The expressive power of neural networks: A view from the width". In: *Advances in Neural Information Processing Systems*, pp. 6231–6239.

Lubbe, Sugnet et al. (2021). "Comparison of zero replacement strategies for compositional data with large numbers of zeros". In: *Chemometrics and Intelligent Laboratory Systems* 210, p. 104248.

Luecken, Malte D and Fabian J Theis (2019). "Current best practices in single-cell RNA-seq analysis: a tutorial". In: *Molecular Systems Biology* 15.6, e8746.

Lundervold, Alexander Selvikvåg and Arvid Lundervold (2019). "An overview of deep learning in medical imaging focusing on MRI". In: *Zeitschrift für Medizinische Physik* 29.2, pp. 102–127.

Ma, Siyuan et al. (2021). "A Statistical Model for Describing and Simulating Microbial Community Profiles". In: *PLoS Computational Biology* 17.9, e1008913.

Mac Aogáin, Micheál et al. (2021). "Integrative microbiomics in bronchiectasis exacerbations". In: *Nature Medicine* 27.4, pp. 688–699.

Máiz, Luis et al. (2018). "Fungi in bronchiectasis: a concise review". In: *International Journal of Molecular Sciences* 19.1, p. 142.

Mandrekar, Jayawant N (2010). "Receiver operating characteristic curve in diagnostic test assessment". In: *Journal of Thoracic Oncology* 5.9, pp. 1315–1316.

Mariette, Jérôme and Nathalie Villa-Vialaneix (2018). "Unsupervised multiple kernel learning for heterogeneous data integration". In: *Bioinformatics* 34.6, pp. 1009–1015.

Maselli, Diego J et al. (2017). "Suspecting non-cystic fibrosis bronchiectasis: What the busy primary care clinician needs to know". In: *International Journal of Clinical Practice* 71.2, e12924.

Matthew, Taddy et al. (2015). "Bayesian and empirical Bayesian forests". In: *International Conference on Machine Learning*. PMLR, pp. 967–976.

Matthews, Alexander G de G et al. (2017). "GPflow: A Gaussian Process Library using TensorFlow." In: *J. Mach. Learn. Res.* 18.40, pp. 1–6.

McBrien, Claire Nichola (2020). "The bacterial and fungal communities in the airways of adults with asthma and eosinophilic lung diseases". In:

McCallum, Gabrielle B and Michael J Binks (2017). "The epidemiology of chronic suppurative lung disease and bronchiectasis in children and adolescents". In: *Frontiers in Pediatrics* 5, p. 27.

McMurdie, Paul J and Susan Holmes (2013). "phyloseq: an R package for reproducible interactive analysis and graphics of microbiome census data". In: *PloS one* 8.4, e61217.

Miller, Andrew C et al. (2017). "Variational boosting: Iteratively refining posterior approximations". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, pp. 2420–2429.

Miller, Tim (2019). "Explanation in artificial intelligence: Insights from the social sciences". In: *Artificial Intelligence* 267, pp. 1–38.

Mishkin, Aaron et al. (2018). "Slang: Fast structured covariance approximations for bayesian deep learning with natural gradient". In: *Advances in Neural Information Processing Systems*, pp. 6245–6255.

Moe, YM (2022). *group-lasso*. https://github.com/yngvem/group-lasso.

Molnar, Christoph (2020). *Interpretable Machine learning*. Lulu. com.

Monod, Mélodie et al. (2021). "Age groups that sustain resurging COVID-19 epidemics in the United States". In: *Science* 371.6536, eabe8372.

Moreno-Indias, Isabel et al. (2021). "Statistical and machine learning techniques in human microbiome studies: contemporary challenges and solutions". In: *Frontiers in Microbiology* 12, p. 277.

Mosimann, James E (1962). "On the compound multinomial distribution, the multivariate β-distribution, and correlations among proportions". In: *Biometrika* 49.1/2, pp. 65–82.

Muir, Paul et al. (2016). "The real cost of sequencing: scaling computation to keep pace with data generation". In: *Genome Biology* 17.1, pp. 1–9.

Murdoch, W James et al. (2019). "Definitions, methods, and applications in interpretable machine learning". In: *Proceedings of the National Academy of Sciences* 116.44, pp. 22071–22080.

Nagpal, Ravinder et al. (2020). "Gut mycobiome and its interaction with diet, gut bacteria and alzheimer's disease markers in subjects with mild cognitive impairment: A pilot study". In: *EBioMedicine* 59, p. 102950.

Nearing, Jacob T et al. (2022). "Microbiome differential abundance methods produce different results across 38 datasets". In: *Nature Communications* 13.1, pp. 1–16.

Nembrini, Stefano et al. (2018). "The revival of the Gini importance?" In: *Bioinformatics* 34.21, pp. 3711–3718.

Nemeth, Christopher and Paul Fearnhead (2021). "Stochastic gradient Markov chain Monte Carlo". In: *Journal of the American Statistical Association* 116.533, pp. 433–450.

Nicodemus, Kristin K (2011). "On the stability and ranking of predictors from random forest variable importance measures". In: *Briefings in Bioinformatics* 12.4, pp. 369–373.

Nicodemus, Kristin K et al. (2010). "The behaviour of random forest permutation-based variable importance measures under predictor correlation". In: *BMC Bioinformatics* 11.1, pp. 1–13.

NIH Human Microbiome Portfolio Analysis Team (2019). "A review of 10 years of human microbiome research activities at the US National Institutes of Health, Fiscal Years 2007-2016". In: *Microbiome* 7, pp. 1–19.

Nilsson, R Henrik et al. (2019). "Mycobiome diversity: high-throughput sequencing and identification of fungi". In: *Nature Reviews Microbiology* 17.2, pp. 95–109.

Ning, Jie and Robert G Beiko (2015). "Phylogenetic approaches to microbial community classification". In: *Microbiome* 3.1, pp. 1–13.

Nojoomi, Saghi and Patrice Koehl (2017). "String kernels for protein sequence comparisons: improved fold recognition". In: *BMC Bioinformatics* 18.1, pp. 1–15.

Notley, Stephen and Malik Magdon-Ismail (2018). "Examining the use of neural networks for feature extraction: A comparative analysis using deep learning, support vector machines, and k-nearest neighbor classifiers". In: *arXiv preprint arXiv:1805.02294*.

O'Dwyer, David N et al. (2016). "The lung microbiome, immunity, and the pathogenesis of chronic lung disease". In: *The Journal of Immunology* 196.12, pp. 4839–4847.

Osband, Ian (2016). "Risk versus uncertainty in deep learning: Bayes, bootstrap and the dangers of dropout". In: *NIPS Workshop on Bayesian Deep Learning*. Vol. 192.

Ovadia, Yaniv et al. (2019). "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift". In: *Advances in Neural Information Processing Systems*, pp. 13991–14002.

Palme, Johannes et al. (2015). "KeBABS: an R package for kernel-based analysis of biological sequences". In: *Bioinformatics* 31.15, pp. 2574–2576.

Papamarkou, Theodore et al. (2019). "Challenges in Bayesian inference via Markov chain Monte Carlo for neural networks". In: *arXiv preprint arXiv:1910.06539*.

Parracho, Helena MRT et al. (2005). "Differences between the gut microflora of children with autistic spectrum disorders and that of healthy children". In: *Journal of Medical Microbiology* 54.10, pp. 987–991.

Patuzzi, Ilaria et al. (2019). "metaSPARSim: a 16S rRNA gene sequencing count data simulator". In: *BMC Bioinformatics* 20.9, pp. 1–13.

Paudel, Keshav Raj et al. (2020). "Role of lung microbiome in innate immune response associated with chronic lung diseases". In: *Frontiers in medicine* 7, p. 554.

Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Pendleton, Kathryn M et al. (2017). "The significance of Candida in the human respiratory tract: our evolving understanding". In: *Pathogens and Disease* 75.3, ftx029.

Peterson, Roseann E et al. (2019). "Genome-wide association studies in ancestrally diverse populations: opportunities, methods, pitfalls, and recommendations". In: *Cell* 179.3, pp. 589–603.

Phipson, Belinda and Gordon K Smyth (2010). "Permutation P-values should never be zero: calculating exact P-values when permutations are randomly drawn". In: *Statistical Applications in Genetics and Molecular Biology* 9.1.

Plantinga, Anna et al. (2017). "MiRKAT-S: a community-level test of association between the microbiota and survival times". In: *Microbiome* 5.1, pp. 1–13.

Price, Alkes L et al. (2006). "Principal components analysis corrects for stratification in genome-wide association studies". In: *Nature Genetics* 38.8, pp. 904–909.

Price, Morgan N et al. (2010). "FastTree 2–approximately maximum-likelihood trees for large alignments". In: *PloS one* 5.3, e9490.

Prost, Vincent et al. (2021). "A zero inflated log-normal model for inference of sparse microbial association networks". In: *PLOS Computational Biology* 17.6, e1009089.

Purcell, Shaun et al. (2007). "PLINK: a tool set for whole-genome association and population-based linkage analyses". In: *The American Journal of Human Genetics* 81.3, pp. 559–575.

Qi, Yanjun (2012). "Random forest for bioinformatics". In: *Ensemble Machine Learning*. Springer, pp. 307–323.

Qian, Xubo et al. (2020). "Gut microbiota in children with juvenile idiopathic arthritis: characteristics, biomarker identification, and usefulness in clinical prediction". In: *BMC Genomics* 21.1, pp. 1–13.

Quinn, Thomas P and Ionas Erb (2020). "Interpretable log contrasts for the classification of health biomarkers: a new approach to balance selection". In: *Msystems* 5.2, e00230–19.

Quinn, Thomas P, Ionas Erb, et al. (2018). "Understanding sequencing data as compositions: an outlook and review". In: *Bioinformatics* 34.16, pp. 2870–2878.

Ran, Di and Z John Daye (2017). "Gene expression variability and the analysis of large-scale RNA-seq studies with the MDSeq". In: *Nucleic Acids Research* 45.13, e127–e127.

Randolph, Timothy W et al. (2018). "Kernel-penalized regression for analysis of microbiome data". In: *The Annals of Applied Statistics* 12.1, p. 540.

Ranganathan, Yuvaraj and Renee M Borges (2011). "To transform or not to transform: that is the dilemma in the statistical analysis of plant volatiles". In: *Plant Signalling & Behavior* 6.1, pp. 113–116.

Ranjan, Ravi et al. (2016). "Analysis of the microbiome: Advantages of whole genome shotgun versus 16S amplicon sequencing". In: *Biochemical and Biophysical Research Communications* 469.4, pp. 967–977.

Ratti, Emanuele (2015). "Big data biology: Between eliminative inferences and exploratory experiments". In: *Philosophy of Science* 82.2, pp. 198–218.

Regalado, Julian et al. (2020). "Combining whole-genome shotgun sequencing and rRNA gene amplicon analyses to improve detection of microbe–microbe interaction networks in plant leaves". In: *The ISME Journal* 14.8, pp. 2116–2130.

Remeseiro, Beatriz and Veronica Bolon-Canedo (2019). "A review of feature selection methods in medical applications". In: *Computers in Biology and Medicine* 112, p. 103375.

Rezende, Danilo Jimenez et al. (2014). "Stochastic backpropagation and approximate inference in deep generative models". In: *arXiv preprint arXiv:1401.4082*.

Ribeiro, Marco Tulio et al. (2016). ""Why should I trust you?" Explaining the predictions of any classifier". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144.

Riquelme, Carlos et al. (2018). "Deep Bayesian bandits showdown". In: *International Conference on Learning Representations*.

Ritchie, Matthew E et al. (2015). "limma powers differential expression analyses for RNA-sequencing and microarray studies". In: *Nucleic Acids Research* 43.7, e47–e47.

Robin, Xavier et al. (2011). "pROC: an open-source package for R and S+ to analyze and compare ROC curves". In: *BMC Bioinformatics* 12.1, pp. 1–8.

Roguet, Adélaıde et al. (2018). "Fecal source identification using random forest". In: *Microbiome* 6.1, pp. 1–15.

Roimi, Michael et al. (2020). "Early diagnosis of bloodstream infections in the intensive care unit using machine-learning algorithms". In: *Intensive Care Medicine* 46.3, pp. 454–462.

Rong, Ruichen et al. (2021). "MB-GAN: Microbiome Simulation via Generative Adversarial Network". In: *GigaScience* 10.2, giab005.

Ronneberger, Olaf et al. (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 234–241.

Roscher, Ribana et al. (2020). "Explainable machine learning for scientific insights and discoveries". In: *IEEE Access* 8, pp. 42200–42216.

Rosenblatt, Jonathan D et al. (2021). "Better-than-chance classification for signal detection". In: *Biostatistics* 22.2, pp. 365–380.

Rossi, Robert M et al. (2019). "Predictive model of factors associated with maternal intensive care unit admission". In: *Obstetrics & Gynecology* 134.2, pp. 216–224.

Rudin, Cynthia (2019). "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead". In: *Nature Machine Intelligence* 1.5, pp. 206–215.

Saeys, Yvan et al. (2007). "A review of feature selection techniques in bioinformatics". In: *Bioinformatics* 23.19, pp. 2507–2517.

Samek, Wojciech et al. (2016). "Evaluating the visualization of what a deep neural network has learned". In: *IEEE Transactions on Neural Networks and Learning Systems* 28.11, pp. 2660–2673.

Sandri, Marco and Paola Zuccolotto (2008). "A bias correction algorithm for the Gini variable importance measure in classification trees". In: *Journal of Computational and Graphical Statistics* 17.3, pp. 611–628.

Santus, William et al. (2021). "Crossing kingdoms: how the mycobiota and fungal-bacterial interactions impact host health and disease". In: *Infection and Immunity* 89.4, e00648–20.

Saporta, Adriel et al. (2021). "Deep learning saliency maps do not accurately highlight diagnostically relevant regions for medical image interpretation". In: *medRxiv*.

Schölkopf, Bernhard et al. (2004). *Kernel methods in computational biology*. MIT Press.

Scott, James A et al. (2021). "Epidemia: An R Package for Semi-Mechanistic Bayesian Modelling of Infectious Diseases using Point Processes". In: *arXiv preprint arXiv:2110.12461*.

Shad, Rohan et al. (2021). "Designing clinically translatable artificial intelligence systems for high-dimensional medical imaging". In: *Nature Machine Intelligence* 3.11, pp. 929–935.

Shawe-Taylor, John, Nello Cristianini, et al. (2004). *Kernel methods for pattern analysis*. Cambridge University Press.

Shi, Bibo et al. (2018). "Prediction of occult invasive disease in ductal carcinoma in situ using deep learning features". In: *Journal of the American College of Radiology* 15.3, pp. 527–534.

Shrikumar, Avanti et al. (2017). "Learning important features through propagating activation differences". In: *International Conference on Machine Learning*. PMLR, pp. 3145–3153.

Silverman, Justin D, Rachael J Bloom, et al. (2021). "Measuring and mitigating PCR bias in microbiota datasets". In: *PLoS Computational Biology* 17.7, e1009113.

Silverman, Justin D, Kimberly Roche, et al. (2020). "Naught all zeros in sequence count data are the same". In: *Computational and structural biotechnology journal* 18, p. 2789.

Simon, Noah et al. (2013). "A sparse-group lasso". In: *Journal of Computational and Graphical Statistics* 22.2, pp. 231–245.

Simonyan, Karen et al. (2014). "Deep inside convolutional networks: Visualising image classification models and saliency maps". In: *In Workshop at International Conference on Learning Representations*. Citeseer.

Smilkov, Daniel et al. (2017). "Smoothgrad: removing noise by adding noise". In: *arXiv preprint arXiv:1706.03825*.

Smola, Alex et al. (2007). "A Hilbert space embedding for distributions". In: *International Conference on Algorithmic Learning Theory*. Springer, pp. 13–31.

Snelson, Edward and Zoubin Ghahramani (2005). "Sparse Gaussian processes using pseudo-inputs". In: 18.

Snoek, Jasper, Hugo Larochelle, et al. (2012). "Practical Bayesian optimization of machine learning algorithms". In: *Advances in Neural Information Processing Systems* 25.

Snoek, Jasper, Oren Rippel, et al. (2015). "Scalable Bayesian optimization using deep neural networks". In: *International Conference on Machine Learning*. PMLR, pp. 2171–2180.

Soneson, Charlotte and Mark D Robinson (2018). "Bias, robustness and scalability in single-cell differential expression analysis". In: *Nature Methods* 15.4, pp. 255–261.

Soret, Perrine et al. (2020). "Respiratory mycobiome and suggestion of inter-kingdom network during acute pulmonary exacerbation in cystic fibrosis". In: *Scientific reports* 10.1, pp. 1–14.

Springenberg, Jost Tobias et al. (2015). "Striving for simplicity: The all convolutional net". In: *In Workshop at International Conference on Learning Representations*. Citeseer.

Srivastava, Nitish et al. (2014). "Dropout: a simple way to prevent neural networks from overfitting". In: *The Journal of Machine Learning Research* 15.1, pp. 1929–1958.

Ssekagiri, AT et al. (2017). "microbiomeSeq: An R package for analysis of microbial communities in an environmental context". In: *ISCB Africa ASBCB Conference, Kumasi, Ghana. https://github. com/umerijaz/microbiomeSeq*. Vol. 10.

Statnikov, Alexander et al. (2013). "A comprehensive evaluation of multicategory classification methods for microbiomic data". In: *Microbiome* 1.1, pp. 1–12.

Strobl, Carolin, Anne-Laure Boulesteix, Thomas Kneib, et al. (2008). "Conditional variable importance for random forests". In: *BMC Bioinformatics* 9.1, pp. 1–11.

Strobl, Carolin, Anne-Laure Boulesteix, Achim Zeileis, et al. (2007). "Bias in random forest variable importance measures: Illustrations, sources and a solution". In: *BMC Bioinformatics* 8.1, p. 25.

Štrumbelj, Erik and Igor Kononenko (2014). "Explaining prediction models and individual predictions with feature contributions". In: *Knowledge and Information Systems* 41.3, pp. 647–665.

Sudlow, Cathie et al. (2015). "UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age". In: *PLoS Medicine* 12.3, e1001779.

Sundararajan, Mukund et al. (2017). "Axiomatic attribution for deep networks". In: *International Conference on Machine Learning*. PMLR, pp. 3319–3328.

Supplitt, Stanislaw et al. (2021). "Current achievements and applications of transcriptomics in personalized cancer medicine". In: *International Journal of Molecular Sciences* 22.3, p. 1422.

Svetnik, Vladimir et al. (2004). "Application of Breiman's random forest to modeling structure-activity relationships of pharmaceutical molecules". In: *International Workshop on Multiple Classifier Systems*. Springer, pp. 334–343.

Szymczak, Silke et al. (2016). "r2VIM: A new variable selection method for random forests in genome-wide association studies". In: *BioData Mining* 9.1, pp. 1–15.

Tang, Zheng-Zheng et al. (2017). "A general framework for association analysis of microbial communities on a taxonomic tree". In: *Bioinformatics* 33.9, pp. 1278–1285.

Tedjo, Danyta I et al. (2016). "The fecal microbiota as a biomarker for disease activity in Crohn's disease". In: *Scientific Reports* 6, p. 35216.

Tenesa, Albert and Chris S Haley (2013). "The heritability of human disease: estimation, uses and abuses". In: *Nature Reviews Genetics* 14.2, pp. 139–149.

The scikit-bio development team (2020). *scikit-bio: A Bioinformatics Library for Data Scientists, Students, and Developers*. Version 0.5.5. URL: http://scikit-bio.org.

Thompson, Jaron et al. (2019). "Machine learning to predict microbial community functions: an analysis of dissolved organic carbon from litter decomposition". In: *PLoS One* 14.7, e0215502.

Tibshirani, Robert (1996). "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1, pp. 267–288.

Ting, Daniel Shu Wei et al. (2017). "Development and validation of a deep learning system for diabetic retinopathy and related eye diseases using retinal images from multiethnic populations with diabetes". In: *JAMA* 318.22, pp. 2211–2223.

Titsias, Michalis (2009). "Variational learning of inducing variables in sparse Gaussian processes". In: *Artificial Intelligence and Statistics*. PMLR, pp. 567–574.

Toivonen, Jussi et al. (2019). "Radiomics and machine learning of multisequence multiparametric prostate MRI: Towards improved non-invasive prostate cancer characterization". In: *PLoS One* 14.7, e0217702.

Tolosana-Delgado, R et al. (2019). "On machine learning algorithms and compositional data". In: *Proceedings of the 8th International Workshop on Compositional Data Analysis, Terrassa, Spain*, pp. 3–8.

Topçuoğlu, Begüm D et al. (2020). "A framework for effective application of machine learning to microbiome-based classification problems". In: *MBio* 11.3, e00434–20.

Torrenté, Laurence de et al. (2020). "The shape of gene expression distributions matter: how incorporating distribution shape improves the interpretation of cancer transcriptomic data". In: *BMC Bioinformatics* 21.21, pp. 1–18.

Tremblay, Julien et al. (2015). "Primer and platform effects on 16S rRNA tag sequencing". In: *Frontiers in Microbiology* 6, p. 771.

Turnbaugh, Peter J, Micah Hamady, et al. (2009). "A core gut microbiome in obese and lean twins". In: *Nature* 457.7228, pp. 480–484.

Turnbaugh, Peter J, Ruth E Ley, et al. (2007). "The human microbiome project". In: *Nature* 449.7164, pp. 804–810.

Unwin, H Juliette T et al. (2020). "State-level tracking of COVID-19 in the United States". In: *Nature Communications* 11.1, pp. 1–9.

Vangay, Pajau et al. (2019). "Microbiome Learning Repo (ML Repo): A public repository of microbiome regression and classification tasks". In: *Gigascience* 8.5, giz042.

Vapnik, Vladimir (1991). "Principles of risk minimization for learning theory". In: *Advances in Neural Information Processing Systems* 4.

Villette, Remy et al. (2021). "Refinement of 16S rRNA gene analysis for low biomass biospecimens". In: *Scientific Reports* 11.1, pp. 1–12.

Visscher, Peter M et al. (2017). "10 years of GWAS discovery: biology, function, and translation". In: *The American Journal of Human Genetics* 101.1, pp. 5–22.

Wang, Hong and Gang Li (2017). "A selective review on random survival forests for high dimensional data". In: *Quantitative bio-science* 36.2, p. 85.

Wang, Huazhen et al. (2016). "An experimental study of the intrinsic stability of random forest variable importance measures". In: *BMC Bioinformatics* 17.1, pp. 1–18.

Wang, Yan and Lloyd H Kasper (2014). "The role of microbiome in central nervous system disorders". In: *Brain, Behavior, and Immunity* 38, pp. 1–12.

Watson, Joe et al. (2021). "Latent derivative bayesian last layer networks". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1198–1206.

Weber, Noah et al. (2018). "Optimizing over a bayesian last layer". In: *NeurIPS workshop on Bayesian Deep Learning*.

Wehenkel, Marie et al. (2018). "Random forests based group importance scores and their statistical interpretation: application for Alzheimer's disease". In: *Frontiers in Neuroscience* 12, p. 411.

Weiss, Sophie et al. (2017). "Normalization and microbial differential abundance strategies depend upon data characteristics". In: *Microbiome* 5.1, pp. 1–18.

Wickham, Hadley et al. (2019). "Welcome to the tidyverse". In: *Journal of Open Source Software* 4.43, p. 1686. DOI: 10.21105/joss.01686.

Williams, Christopher and Carl Rasmussen (1995). "Gaussian processes for regression". In: *Advances in Neural Information Processing Systems* 8.

— (2006). "Gaussian Processes for Machine Learning". In: *The MIT Press* 2.3, p. 4.

Williams, Christopher and Matthias Seeger (2000). "Using the Nyström method to speed up kernel machines". In: *Advances in Neural Information Processing Systems* 13.

Worby, Colin J. et al. (2022). "Longitudinal multi-omics analyses link gut microbiome dysbiosis with recurrent urinary tract infections in women". In: *Nature Microbiology* 7.5, pp. 630–639. ISSN: 2058-5276. DOI: 10.1038/s41564-022-01107-x. URL: https://doi.org/10.1038/s41564-022-01107-x.

Wright, Marvin N and Andreas Ziegler (2015). "ranger: A fast implementation of random forests for high dimensional data in C++ and R". In: *arXiv preprint arXiv:1508.04409*.

WTCCC et al. (2007). "Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls". In: *Nature* 447.7145, p. 661.

Wu, Chong et al. (2016). "An adaptive association test for microbiome data". In: *Genome Medicine* 8.1, pp. 1–12.

Wu, Tong Tong et al. (2009). "Genome-wide association analysis by lasso penalized logistic regression". In: *Bioinformatics* 25.6, pp. 714–721.

Xiao, Jian et al. (2018). "Predictive modeling of microbiome data using a phylogeny-regularized generalized linear mixed model". In: *Frontiers in Microbiology* 9, p. 1391.

Xicota, Laura et al. (2019). "Multi-omics signature of brain amyloid deposition in asymptomatic individuals at-risk for Alzheimer's disease: The INSIGHT-preAD study". In: *EBioMedicine* 47, pp. 518–528.

Yao, Yuling et al. (2018). "Yes, but did it work?: Evaluating variational inference". In: *arXiv preprint arXiv:1802.02538*.

Young, Vincent B (2017). "The role of the microbiome in human health and disease: an introduction for clinicians". In: *Bmj* 356.

Yu, Guangchuang et al. (2017). "ggtree: an R package for visualization and annotation of phylogenetic trees with their covariates and other associated data". In: *Methods in Ecology and Evolution* 8.1, pp. 28–36.

Yuan, Ming and Yi Lin (2006). "Model selection and estimation in regression with grouped variables". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.1, pp. 49–67.

Yun, Yong-Huan et al. (2016). "Variable importance analysis based on rank aggregation with applications in metabolomics for biomarker discovery". In: *Analytica Chimica Acta* 911, pp. 27–34.

Zeiler, Matthew D (2012). "Adadelta: an adaptive learning rate method". In: *arXiv preprint arXiv:1212.5701*.

Zemanick, Edith T and Lucas R Hoffman (2016). "Cystic fibrosis: microbiology and host response". In: *Pediatric Clinics* 63.4, pp. 617–636.

Zeng, Jiaming et al. (2018). "The relevance of Bayesian layer positioning to model uncertainty in deep Bayesian active learning". In: *arXiv preprint arXiv:1811.12535*.

Zhang, Mo and Wenjiao Shi (2019). "Systematic comparison of five machine-learning methods in classification and interpolation of soil particle size fractions using different transformed data". In: *Hydrology and Earth System Sciences Discussions*, pp. 1–39.

Zhang, Quanshi et al. (2019). "Interpreting CNNs via decision trees". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6261–6270.

Zhang, Weiwei et al. (2008). "Cat head detection-how to effectively exploit shape and texture features". In: *European Conference on Computer Vision*. Springer, pp. 802–816.

Zhao, Guoyan et al. (2017). "Intestinal virome changes precede autoimmunity in type I diabetes-susceptible children". In: *Proceedings of the National Academy of Sciences* 114.30, E6166–E6175.

Zhao, Ni et al. (2015). "Testing in microbiome-profiling studies with MiRKAT, the microbiome regression-based kernel association test". In: *The American Journal of Human Genetics* 96.5, pp. 797–807.

Zhou, Yi-Hui and Paul Gallins (2019). "A review and tutorial of machine learning methods for microbiome host trait prediction". In: *Frontiers in Genetics*, p. 579.

Ziegler, Andreas and Inke R König (2014). "Mining data with random forests: current options for real-world applications". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 4.1, pp. 55–63.

Zou, Hui and Trevor Hastie (2005). "Regularization and variable selection via the elastic net". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2, pp. 301–320.

Appendix: List of software packages

This appendix lists all the software packages used in each of the three results chapters (Chapters 3, 4 and 5).

Data wrangling and plotting in all chapters was done using the tidyverse suite of packages (Wickham et al., 2019). Microbial datasets were managed using phyloseq (McMurdie and S. Holmes, 2013).

Chapter 3

Random forest models were fitted in ranger (M. N. Wright and Ziegler, 2015) and cross-validated using caret (Kuhn, 2015). ROC analysis was peformed using pROC (Robin et al., 2011). Shapley values were computed using fastshap (B. Greenwell and M. B. Greenwell, 2021). LeDell's confidence intervals were calculated using cvAUC (E. LeDell, Petersen, Laan, and M. E. LeDell, 2022).

Chapter 4

The Bayesian neural network models were implemented using TensorFlow probability (Abadi et al., 2016) and the sparse Gaussian process regression model used GPFlow (Matthews et al., 2017). The GroupLasso model was trained using the group-lasso Python package (Moe, 2022). The random forest models were fit using Scikit-learn (Pedregosa et al., 2011).

Chapter 5

The string kernels were computed using the KeBABs package (Palme et al., 2015). Tree visualisations were produced using the gtree package (Yu et al., 2017). Maximum likelihood estimates for the Dirichlet multinomial models were computed using the MGLM package (J. Kim et al., 2018). All kernels were implemented using the GPFlow framework (Matthews et al., 2017). UniFrac kernels also used Scikit-bio (The scikit-bio development team, 2020).