Pangenome-Wide Association Studies with Frequented Regions

Buwani Manuweera
Gianforte School of Computing
Montana State University
Bozeman, MT, USA
buwani.manuweera@student.montana.edu

Joann Mudge
National Center for Genome Resources
Santa Fe, NM, USA
jm@ncgr.org

Indika Kahanda
Gianforte School of Computing
Montana State University
indika.kahanda@montana.edu

Brendan Mumey
Gianforte School of Computing
Montana State University
brendan.mumey@montana.edu

Thiruvarangan Ramaraj
National Center for Genome Resources
tr@ncgr.org

Alan Cleary
National Center for Genome Resources
acleary@ncgr.org

ABSTRACT

Connecting genetic variation (genotype) to trait variation (phenotype) is a critical but often difficult step in genetic research. A genome-wide association study (GWAS) is a common approach to connect underlying genetic variation to complex phenotypic traits, allowing for phenotypic prediction. GWAS is important in many disciplines, including identifying genetic risk factors for common, complex diseases, identifying genes underlying important traits and predicting phenotypes from genotypes. GWAS is limited, though, in that the types of variations typically studied are single nucleotide polymorphisms (SNPs) identified relative to a single reference genome. These limitations lead to bias and preclude GWAS from studies across related species. The advent of next-generation sequencing has brought an exponential growth in DNA sequence data. This has led to the more comprehensive pangenomics approach, where the entire sequence content and variation of a population are succinctly represented independent of a reference. In prior work, we developed a method for identifying genomic regions that characterize complex variations within pangenomic data and showed that these regions provide a more general way to study genetic variation than existing approaches. This work describes our initial results to develop new methods for a new branch of genomic analysis called pangenome-wide association studies (PWAS) that generalizes GWAS to pangenomic datasets both within and across species. We make use of recently developed algorithms for fast compressed De Bruijn graph construction and identifying frequented regions in these graphs that can be used as machine-learning features to identify pangenomic regions, overlaid with gene annotations, that relate to complex phenotypic traits. Initial results on a pangenome composed of 100 yeast indicate that frequented region features provide better machine-learning regression models than SNPs for predicting phenotypic traits.

ACM-BCB '19, September 7–10, 2019, Niagara Falls, NY, USA

ACM ISBN 978-1-4503-6666-3/19/09...$15.00.
https://doi.org/10.1145/3307339.3343478

KEYWORDS

pangenomics; regression; machine-learning

ACM Reference format:
Buwani Manuweera, Joann Mudge, Indika Kahanda, Brendan Mumey, Thiruvarangan Ramaraj, and Alan Cleary. 2019. Pangenome-Wide Association Studies with Frequented Regions. In *Proceedings of 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, Niagara Falls, NY, USA, September 7–10, 2019 (ACM-BCB '19),* 6 pages.
https://doi.org/10.1145/3307339.3343478

1 INTRODUCTION

A pangenome represents the collective genomic information of multiple individuals or organisms from a related group or species [25]. While we can decode genomes, reading and interpreting those genomes to predict physical characteristics is still difficult. One way to connect an organism's DNA sequence to its physical characteristics is by doing a genome-wide association study (GWAS). This method uses a population of organisms from the same species and tries to correlate any of the DNA differences among individuals with differences in their physical characteristics. The full DNA sequence is obtained for one individual (the reference) but other individuals only get small DNA sequence samples at intervals along the chromosome for cost-effectiveness. One source of bias in a GWAS analysis is that an organism's DNA differences are all defined in relation to the reference. This is akin to trying to define all fruits by how they compare to, for instance, seedless grapes. Comparing an apple to the reference grape allows you to describe differences in their stem, skin, and flesh. But because seeds are not in the reference fruit, we cannot describe them at all and the fact that apples have seeds is lost from the analysis. Furthermore, for fruits such as apples that look similar to the reference, we are able to describe more similarities and differences with the reference than more unusual fruits, such as pineapples, where many of the differences are so different that they cannot be matched up to the reference and, therefore, are lost. Likewise, in DNA analysis, highly evolved or even novel DNA regions without a good match in the reference become lost to the analysis. But, it is these very regions that are most likely to contain genes or control regions that account for important differences in physical characteristics, including those that allow the organism to adapt to extreme conditions and affect susceptibility to disease and chronic conditions, or even survivability.

This work studies a recently-introduced reference-free method for finding genomic features [4], combined with machine-learning integration, for the purposes of discovering phenotype associations in a pangenomic data set. This type of analysis is termed *Pangenome-wide Association Study* (PWAS). In this work, we compared the results found to more traditional GWAS approaches based on SNPs.

In Section 2 we discuss related work and in Section 3 we provide an overview of frequented regions and machine-learning methods. In Section 4 we discuss our experiments and examine the results in Section 5. Lastly, in Section 6 we discuss future work and make closing remarks.

2 RELATED WORK

There exist a variety of tools and techniques for performing genome-wide association studies [2, 26]. Unfortunately, these typically only take SNPs into consideration, disregarding structural variation. Additionally, there has been limited exploration of the application of machine-learning to GWAS [19, 24], and less exploration of the application of machine-learning in the pangenomics space [4, 10, 12]. Though there has been work on *pangenome*-wide association studies, it is limited in scalability and does not leverage the graphical representation of pangenomic data [9, 13]. Lastly, tools designed for the analysis of graphical pangenomes are currently limited to the construction of pangenomic graphs [1, 7, 17] or moving fundamental bioinformatic analyses into the graphical space, such as read mapping and variant calling [8, 11]. In this work, we leverage the FR-finding algorithm described in [4]. Other tools exist for finding syntentic regions in pangenomes including [18] and could also be used for PWAS studies.

3 METHODS

3.1 Frequented Regions

Frequented regions were introduced in [4] as a method to identify "hotspot" regions in a compressed De Bruijn graph (cDBG) that are co-visited by a set of supporting paths from individual sequences in the pangenome. Here we provide a brief restatement of the approach, as well as discuss several adaptations that were made for the PWAS application. Full algorithmic details can be found in [4]. The input used is a cDBG graph G and set of paths P within G (the `cdbg` package [1] was used to construct G). Nodes in G represent specific k-mers (or $\geq k$-mers if the graph has been compressed). An edge (u, v) is present provided the last $k-1$ nucleotides of u match the first $k-1$ nucleotides of v. The objective is to identify *frequented regions* (FRs) (C, S) which are composed of a set of nodes C and set of subpaths from P that each approximately traverse the nodes in C. There are two parameters used in the definition: (1) the *penetrance* parameter α which is the minimum fraction of the nodes in C that each subpath must contain, and (2) the *maximum insertion* parameter κ, which is the maximum number of nodes outside of C that can be visited by a supporting subpath before the path must return to C. Subpaths that meet these conditions are called (α, κ)-supporting subpaths. Following [4], we define a *frequented region* (FR) as a tuple (C, S), where C is a set of De Bruijn nodes and S is a set of (α, κ)-supporting subpaths of paths from P.

For each $p \in S$, let strain(p) be the specific pangenomic strain that p belongs to. Since we will be doing machine-learning experiments, we assume that there is a set of strains T that is the *training set* of strains; only the paths associated with these strains are used for scoring potential FRs. We define $S_T = \{p \in S : \text{strain}(p) \in T\}$ be the set of supporting subpaths of the FR (C, S) that belong to the training set. We introduce another measure of an FR that was not present in our original definition: For an FR (C, S) and each node $n \in C$, we define $\beta(n) = \frac{|\{p \in S_T : n \in P\}|}{|S_T|}$ as the fraction of subpaths in S_T that include the node n. We then compute the geometric mean over all the nodes in C to define the *coverage* of the FR (C, S) as $\beta(C, S) = \prod_{n \in C} \beta(n)^{\frac{1}{|C|}}$ (the geometric mean was chosen over the arithmetic mean as it will give more penalty to nodes that have low $\beta(n)$ values). Finally, we define the *support* of an FR (C, S) as

$$\text{support}(C, S) = \beta(C, S)|S_T|. \tag{1}$$

The set of sequences P is also checked for reverse-complement support, with the assumption that some of the assembled contigs may be in reverse-complement orientation relative to the majority in some overlapping regions. If a given sequence $p \in P$ has more support in the reverse-complement direction we assume that that is the correct orientation for that sequence and only consider its FR support in that orientation.

Our previous work focused on finding FRs that have high support and high average supporting subpath length. In this work, we seek FRs that are useful as machine-learning features, so we introduce a new approach to computing *interesting FRs* (iFRs) that is based on which strains provide support to each FR. For each FR (C, S) we define

$$\text{strains}(C, S) = T \cap \bigcup_{p \in S} \text{strain}(p), \tag{2}$$

as the training strains that are present among the supporting subpaths of the FR. The `FindFRs` software [4], uses a bottom-up approach to identify FRs; first, an approximate maximal weigh matching of the existing FRs (initially individual cDBG nodes) is computed, and then new FRs are created by merged matched edges. This is repeated until no new FRs with positive support are found. For each observed strain set, we keep track of the FR with that strain set that has the greatest support. Since there are potentially a large number of strain subsets, we also assume that a limit M is set, such that if more than M strain subsets are seen, then those with the least support are dropped. In this way, at most M iFRs are reported. In this work, M was set to 50,000.

3.2 Machine-learning with FRs

In order to explore the utility of FRs for deciphering genotype to phenotype relationships, we use FRs in a supervised machine-learning setting in which FRs are used as features for predicting yeast phenotypes. Here, each instance (i.e. example) represents an individual yeast strain. Each example is annotated with a series of phenotypes (see Table 1). Each phenotype is a continuous target variable. Therefore, we modeled this task of predicting phenotypes as a multi-output regression [15] problem.

As our machine-learning algorithm, we use the random forest (RF) [3] algorithm for the regression problem. The Random Forest algorithm has been used with genomic data in several studies [3, 5, 27]. It known to work well with high-dimensional genomic

datasets [22]. In this case, a separate single-output random forest regressor is learned for each phenotype from the training data.

The outline of the regression model can be given as follows. The input to the regression model consists of m examples/strains. Each of those examples have n features which are, in this case, iFRs or, for comparison, single nucleotide polymorphisms (SNPs) for each strain. Therefore, each example is represented as an n-dimensional vector (n is the total number of FRs used) in which each individual component of a vector corresponds to the number of times a certain FR occurs with that genome. We use these vectors as input to our regression model for training. Then the trained model is used for making predictions on the phenotype values for a previously unseen set of strain genomes by the model. Performance is evaluated by comparing the predicted phenotype values with the actual values (see Section 5 for results of this study).

4 EXPERIMENTS

4.1 Data

Sequencing and phenotypic data were obtained from [23] for 100 yeast strains and 49 phenotypes. Data for 49 phenotypic traits across the 100 yeast strains was quantile normalized using the `preprocessCore` R library [16].

In order to compare to standard GWAS methods, we called biallelic SNPs to create a set that could be used both in the GWAS analysis as well as machine-learning. Unlike in [23], we chose not to include additional features in the GWAS, such as the presence and absence of genes, in order to replicate a more typical GWAS experiment, in which low coverage sequencing is sufficient to identify SNPs in the genome but not to generate whole genome assemblies.

To generate SNPs, sequencing read pairs were first individually aligned to the reference yeast strain (*Saccharomyces cerevisiae* S288C, baker's yeast) using the BWA [14] alignment program with default parameters. `FreeBayes` [7], a Bayesian genetic variant detector designed to find SNPs was used to generate variant calls. All alignments files were called at the same time so that FreeBayes could use information from all the strains to report SNPs. A total of 489,150 SNPs were reported across all 99 samples compared to the reference. Genotypes for the reference strain were coded as homozygous for the reference allele, making a total of 100 yeast strains. A subset of 50,000 random SNPs were used for all analyses.

GWAS, including phenotypic prediction, was performed using a Baysian sparse linear mixed model implemented in GEMMA [28], with 250,000 burn-in steps. Population structure is corrected for by GEMMA using a centered relatedness matrix. Phenotypic prediction used the estimated SNP breeding values. GWAS and phenotypic prediction was run using the same tests sets described for the machine-learning experiments (see below), with the other 80 samples used for the training set.

4.2 Experimental Setup

In the experimental setup, the inputs to the regression model consist of the set of features and labels. The FR data for each yeast genome are used as features and each yeast strain considered as an example in the dataset creating 100 examples. Since this is supervised learning, 49 phenotype values corresponding to the 100 yeast strains are used as labels. For comparison, we also used SNPs as features as well as traditional SNP-based GWAS and prediction.

We used the random forest regressor with default parameters for our experiments. The implementation of the machine-learning process was done using `Scikit-learn` [20] library for Python programming language.

4.3 Validation

In order to evaluate the FRs on their ability to predict phenotypes, we use the 5-fold cross-validation setup shown in Figure 1. The dataset is composed of 100 strains/examples and those strain names were initially randomized and divided into 5 folds. In each iteration, three of the *training* folds (shown in blue) are first used to generate 80 models corresponding to the 80 sets of parameter values for k-mers, α, and κ (i.e. a grid search procedure). These models are compared using the average RMSE taken from the 49 RMSE values (defined below) by testing on the *validation* set (shown in orange) to find the most optimum set of parameter values. Then, a separate model is re-trained on both the training and validation folds using this "best" set of parameters. This final model is then evaluated using RMSE by testing on the *test* set (shown in green). RMSE values across the 5 iterations are averaged to obtain the overall performance metric. This form of "nested" cross-validation procedure [6] produces the most unbiased estimation of the model performance because the test sets are not touched during the internal parameter optimization process. Ideally, each fold used for parameter optimization needs to be iteratively picked as the validation set. However, this was omitted due to the increased runtimes. Similarly,

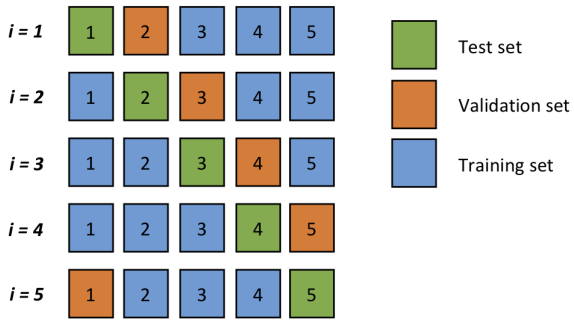

Figure 1: Overview of the 5-fold cross-validation procedure.

using features generated from SNPs in place of FRs, experiments were repeated, for the purpose of comparing the power of FRs vs SNPs. But note that SNPs did not involve parameter tuning, and therefore, the internal grid search was omitted for SNPs.

4.4 Performance Measures

The performance of the regression model was evaluated using RMSE (Root Mean Square Error). RMSE was used to select the best set of parameters as well as to evaluate the performance of the final models.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(O-P)^2} \tag{3}$$

In the above equation, O represents the actual values for the phenotype and P represents the predicted value from the regressor.

Since there are 49 phenotype values predicted, that gave 49 corresponding RMSE values. In order to compare the sets of RMSE values for validation, *average RMSE* was used as the performance measure. For that, the average value of the 49 RMSE values per dataset was calculated and compared to all the other datasets.

5 EXPERIMENTAL RESULTS

5.1 Phenotypic Prediction

As shown in Figure 3, the "best" set of (k, alpha, kappa) parameters found during the five iterations of the cross-validation process are as follows: (100, 0.8, 3), (500, 0.9, 3), (100, 0.7, 0), (25, 0.7, 3), (1000, 0.6, 1).

As shown in Figure 4 and Table 1, FRs have a stronger classification power over SNPs for predicting phenotypes. FRs provide better RMSEs for 42/49 phenotypes. There is a significant difference (p-value: 3.703E-10 from a two-tailed paired t-test) between the average RMSE with FRs (5.38) versus SNPs (5.74). The improvement in predictive accuracy due to FRs is 6%.

FRs (average RMSE=5.38) also had a slightly better classification power, overall, compared to GWAS SNPs (average RMSE=5.41) (Table 1), though for some phenotypes FRs clearly was the better method (an example is shown in Figure 2. Although the difference was not significant (p-value: 0.67), it is encouraging that initial attempts using FRs matched that of GWAS-based predictions.

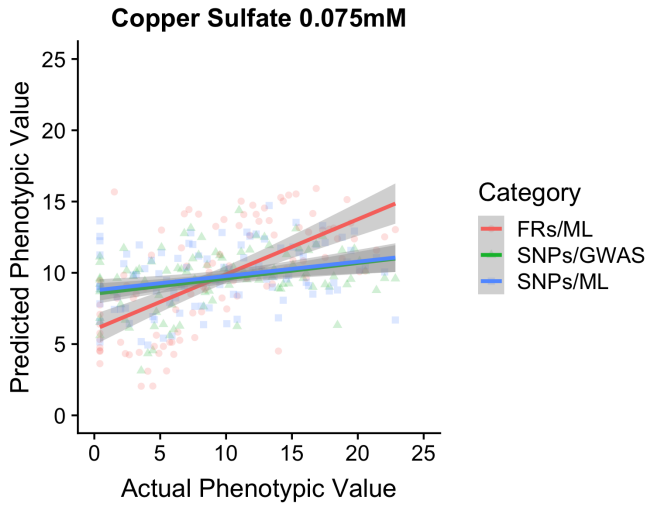

Figure 2: Dotplot comparing actual and predicted values for copper sulfate (0.075 mM) across all 100 samples and the three analysis methods. Each dot represents a single yeast strain and colors indicate the analysis method.

5.2 FRs & Annotations

Additionally, we investigated subpaths and FRs that were associated with yeast genes. To do this we used a tool called "intersect" which is a part of the BEDTOOLS [21] package. The intersect command takes a BED (Browser Extensible Data) file generated by the FindFRs program [4] and intersects it with the combined annotation file in

Table 1: Phenotype prediction performance comparison between Machine-learning using FRs (FRs) and SNPs (SNPs) and GWAS SNPs (SNPsG) on the task of predicting phenotypes. Lower RMSEs are better.

Phenotype	FRs	SNPs	SNPsG
Biofilm	5.82	6.06	5.91
0.07mM_copper_sulfate	4.73	5.82	5.79
0.1mM_copper_sulfate	4.74	5.84	5.74
0.25mM_copper_sulfate	5.37	6.08	5.86
amphotericin_B_15mM	5.20	5.75	5.18
cycloheximide_0.25mM	5.55	5.97	5.01
cycloheximide_0.5mM	5.62	6.20	5.39
ketoconazole_10mM	5.17	5.52	4.93
ketoconazole_20mM	5.73	6.10	5.69
natamycin_3mM	5.51	5.96	5.88
flocculation	6.01	5.84	5.93
50mM_lithium_chloride	4.92	5.43	5.41
pH8.0	5.59	6.41	5.84
1M_sodium_chloride	5.68	5.87	5.60
%4-spored,_KAc_plates_25C	5.44	5.49	5.11
%4-spored,_KAc_plates_30C	5.39	5.58	5.41
%4-spored,_diet_KAc_plates_25C	5.63	5.99	5.59
%4-spored,_diet_KAc_plates_30C	5.94	6.24	5.87
%4-spored,_liquid_KAc25C	5.45	5.62	5.30
%4-spored,_liquid_KAc30C	5.36	5.91	5.34
%sporulation,_KAc_plates_25C	5.17	5.49	4.87
%sporulation,_KAc_plates_30C	4.83	5.53	4.75
%sporulation,_diet_KAc_plates_25C	4.88	5.48	4.65
%sporulation,_diet_KAc_plates_30C	5.65	5.74	5.02
%sporulation,_liquid_KAc25C	5.41	5.58	5.12
%sporulation,_liquid_KAc30C	5.40	5.94	5.16
Sulfite 3mM	5.79	6.26	5.93
Sulfite 6mM	5.22	5.73	4.49
Sulfite 9mM	5.59	5.93	5.75
Temperature sd_15C	5.42	6.13	5.50
Temperature sd_37C	5.57	5.17	5.21
Temperature sd_39C	5.18	5.71	5.37
Temperature seg_15C	5.85	6.23	5.81
Temperature seg_37C	6.05	6.00	5.71
Temperature seg_39C	5.50	5.87	5.68
Temperature ypd_15C	5.58	5.71	5.32
Temperature ypd_37C	4.96	5.29	5.03
Temperature ypd_39C	5.53	5.42	5.53
Temperature ypeg_15C	5.93	6.36	6.00
Temperature ypeg_37C	5.50	5.78	5.51
Temperature ypeg_39C	5.04	5.69	5.24
biotin	5.11	5.82	5.10
inositol	4.87	4.87	5.46
niacin	3.47	3.66	4.62
p-aminobenzoic_acid,_-folic_acid	5.58	5.59	5.51
pantothenate	5.44	6.00	5.45
pyridoxine	4.80	5.20	5.11
riboflavin	5.72	5.65	5.66
thiamine	5.96	5.64	5.66
Mean	**5.38**	**5.74**	**5.41**

GFF (General Feature Format) format for the 100 yeast strains used in this study and reports iFRs that overlaps with genic coordinates in the yeast gene annotations. The BED file generated running the FR algorithm with the following parameters, *k*-mer= 25; alpha=0.7; kappa=3; minimum support=1; maximum iFRs to report=50,000 was intersected with the GFF file. Using a minimum overlap of 50 base pairs there were a total of 644,236 sub-paths distributed in 3,115 FRs that overlapped the yeast genes. Out of the 3,115 FRs, 3,078 overlapped with multiple genes, whereas 37 FRs overlapped with exactly one gene. We also ran this analysis on a BED file generated running the FR algorithm for the largest *k*-mer, 1000 base pairs. The rest of the parameters included the following, alpha=0.7; kappa=3; minimum support=1; maximum iFRs to report=50,000 and found out that out of the 3,040,550 subpaths from 50,000 iFRs reported in the BED file, 2,592,521 (85%) subpaths overlapped with a gene with at least a minimum threshold of 50%.

If gene annotation is available for a set of genomes of interest, then using this approach we can effectively use FRs to understand the population of genomes not only at the genome level but also at the gene level and start understanding and associating functions to FRs and their related phenotypes.

6 CONCLUSIONS

Using a pangenomic approach allows for the unbiased incorporation of all the genetic variation present in the sequenced members of a species. Applying this rich source of variation to phenotypic prediction should improve our ability to find genetic variation driving phenotypic differences. Indeed, using FRs as features for machine-learning prediction improved prediction accuracy over using SNPs, which have an inherent bias toward variation seen in the reference, as features. This is especially encouraging due to the use of simple FR counts as features and the default values for the regressor parameters. We expect that as we improve our strategies for applying machine-learning algorithms using FR features, prediction rates should improve and outpace GWAS-based prediction methods. In addition, applying FR-based machine-learning methods to larger training sets should also improve prediction power and take advantage of the strengths of machine-learning.

ACKNOWLEDGMENTS

This work was supported by NSF grant DBI-1759522.

REFERENCES

[1] Timo Beller and Enno Ohlebusch. 2016. A representation of a compressed de Bruijn graph for pan-genome analysis that enables search. *Algorithms for Molecular Biology* 11, 1 (2016), 20.

[2] Liana T Burghardt, Nevin D Young, and Peter Tiffin. 2017. A guide to genome-wide association mapping in plants. *Current Protocols in Plant Biology* 2, 1 (2017), 22–38.

[3] Xi Chen and Hemant Ishwaran. 2012. Random forests for genomic data analysis. *Genomics* 99 (2012), 323–329. https://doi.org/10.1016/j.ygeno.2012.04.003

[4] Alan Cleary, Thiruvarangan Ramaraj, Indika Kahanda, Joann Mudge, and Brendan Mumey. 2018. Exploring frequented regions in pan-genomic graphs. *IEEE/ACM transactions on computational biology and bioinformatics* (2018).

[5] Ramón Díaz-Uriarte and Sara Alvarez de Andrés. 2006. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics* 7, 1 (jan 2006), 3. https://doi.org/10.1186/1471-2105-7-3

[6] Lingraj Dora, Sanjay Agrawal, Rutuparna Panda, and Ajith Abraham. 2018. Nested cross-validation based adaptive sparse representation algorithm and its application to pathological brain classification. *Expert Systems With Applications* 114 (2018), 313–321. https://doi.org/10.1016/j.eswa.2018.07.039

[7] Erik Garrison. 2019. seqwish. https://github.com/ekg/seqwish. (2019).

[8] Erik Garrison, Jouni Sirén, Adam M Novak, Glenn Hickey, Jordan M Eizenga, Eric T Dawson, William Jones, Michael F Lin, Benedict Paten, and Richard Durbin. 2017. Sequence variation aware genome references and read mapping with the variation graph toolkit. *BioRxiv* (2017), 234856.

[9] Andrea Gori, Odile Harrison, Ethwako Mlia, Yo Nishihara, Jacquline Chinkwita-Phiri, Macpherson Mallewa, Queen Dube, Todd D Swarthout, Angela H Nobbs, Martin Maiden, et al. 2019. Pan-GWAS of Streptococcus agalactiae highlights lineage-specific genes associated with virulence and niche adaptation. *bioRxiv* (2019), 574152.

[10] Hsuan-Lin Her and Yu-Wei Wu. 2018. A pan-genome-based machine learning approach for predicting antimicrobial resistance activities of the Escherichia coli strains. *Bioinformatics* 34, 13 (2018), i89–i95.

[11] Mahdi Heydari, Giles Miclotte, Yves Van de Peer, and Jan Fostier. 2018. BrownieAligner: accurate alignment of Illumina sequencing data to de Bruijn graphs. *BMC bioinformatics* 19, 1 (2018), 311.

[12] Erol S Kavvas, Edward Catoiu, Nathan Mih, James T Yurkovich, Yara Seif, Nicholas Dillon, David Heckmann, Amitesh Anand, Laurence Yang, Victor Nizet, et al. 2018. Machine learning and structural analysis of Mycobacterium tuberculosis pan-genome identifies genetic signatures of antibiotic resistance. *Nature communications* 9, 1 (2018), 4306.

[13] John A Lees, Marco Galardini, Stephen D Bentley, Jeffrey N Weiser, and Jukka Corander. 2018. pyseer: a comprehensive tool for microbial pangenome-wide association studies. *Bioinformatics* 34, 24 (2018), 4310–4312.

[14] Heng Li and Richard Durbin. 2009. Fast and accurate short read alignment with Burrows–Wheeler transform. *bioinformatics* 25, 14 (2009), 1754–1760.

[15] Guangcan Liu, Zhouchen Lin, and Yong Yu. 2009. Multi-output regression on the output manifold. *Pattern Recognition* 42 (2009), 2737–2743. https://doi.org/10.1016/j.patcog.2009.05.001

[16] Pedro López-Romero. 2011. Pre-processing and differential expression analysis of Agilent microRNA arrays using the AgiMicroRna Bioconductor library. *BMC genomics* 12, 1 (2011), 64.

[17] Ilia Minkin and Paul Medvedev. 2019. Scalable multiple whole-genome alignment and locally collinear block construction with SibeliaZ. *BioRxiv* (2019), 548123.

[18] Ilia Minkin and Paul Medvedev. 2019. Scalable multiple whole-genome alignment and locally collinear block construction with SibeliaZ. *bioRxiv* (2019). https://doi.org/10.1101/548123 arXiv:https://www.biorxiv.org/content/early/2019/02/13/548123.full.pdf

[19] Thanh-Tung Nguyen, Joshua Zhexue Huang, Qingyao Wu, Thuy Thi Nguyen, and Mark Junjie Li. 2015. Genome-wide association data classification and SNPs selection using two-stage quality-based Random Forests. In *BMC genomics*, Vol. 16. BioMed Central, S5.

[20] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830. http://scikit-learn.sourceforge.net.

[21] Aaron R Quinlan and Ira M Hall. 2010. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* 26, 6 (2010), 841–842.

[22] Daniel F. Schwarz, Inke R. König, and Andreas Ziegler. 2010. On safari to Random Jungle: a fast implementation of Random Forests for high-dimensional data. *Bioinformatics* 26, 14 (jul 2010), 1752–1758. https://doi.org/10.1093/bioinformatics/btq257

[23] Pooja K Strope, Daniel A Skelly, Stanislav G Kozmin, Gayathri Mahadevan, Eric A Stone, Paul M Magwene, Fred S Dietrich, and John H McCusker. 2015. The 100-genomes strains, an S. cerevisiae resource that illuminates its natural phenotypic and genotypic variation and emergence as an opportunistic pathogen. *Genome research* 25, 5 (2015), 762–774.

[24] Silke Szymczak, Joanna M Biernacka, Heather J Cordell, Oscar González-Recio, Inke R König, Heping Zhang, and Yan V Sun. 2009. Machine learning in genome-wide association studies. *Genetic epidemiology* 33, S1 (2009), S51–S57.

[25] Hervé Tettelin, Vega Masignani, Michael J Cieslewicz, Claudio Donati, Duccio Medini, Naomi L Ward, Samuel V Angiuoli, Jonathan Crabtree, Amanda L Jones, A Scott Durkin, et al. 2005. Genome analysis of multiple pathogenic isolates of Streptococcus agalactiae: implications for the microbial âĂIJpan-genomeâĂİ. *Proceedings of the National Academy of Sciences* 102, 39 (2005), 13950–13955.

[26] Peter M Visscher, Naomi R Wray, Qian Zhang, Pamela Sklar, Mark I McCarthy, Matthew A Brown, and Jian Yang. 2017. 10 years of GWAS discovery: biology, function, and translation. *The American Journal of Human Genetics* 101, 1 (2017), 5–22.

[27] Jiansheng Wu, Hongde Liu, Xueye Duan, Yan Ding, Hongtao Wu, Yunfei Bai, and Xiao Sun. 2009. Prediction of DNA-binding residues in proteins from amino acid sequences using a random forest model with a hybrid feature. *Bioinformatics* 25, 1 (jan 2009), 30–35. https://doi.org/10.1093/bioinformatics/btn583

[28] Xiang Zhou, Peter Carbonetto, and Matthew Stephens. 2013. Polygenic modeling with Bayesian sparse linear mixed models. *PLoS genetics* 9, 2 (2013), e1003264.

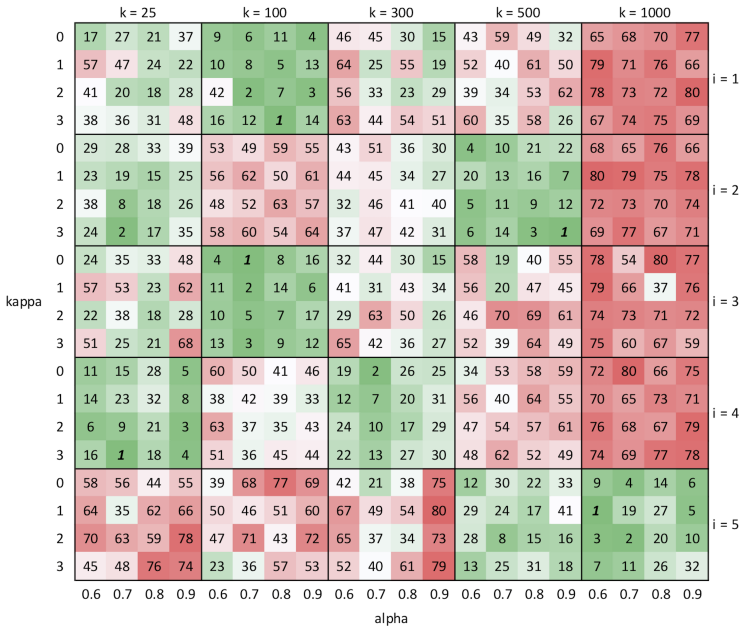

Figure 3: Distribution of the average RMSE per each parameter combination. The alpha and kappa parameters are depicted in x- and y-axis, respectively. Each sub matrix depicts the all parameter combinations of alpha and kappa for a specific k value and the specific iteration, indicated by *i*. Each cell indicates the ranking (1-80) received by a parameter combination within the specific iteration. Green-white-red conditional formatting is used within each iteration and the ranking "1" represents the best combination for each iteration.

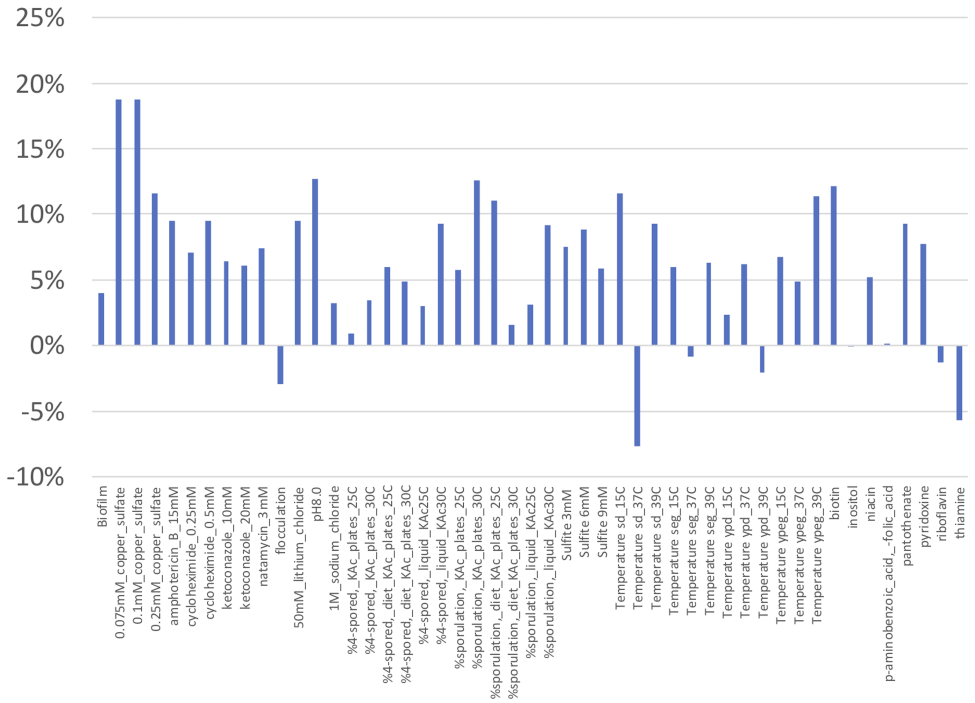

Figure 4: Improvement in performance for FRs over SNPs on the task of predicting phenotypes. $Improvement = 100 \cdot \frac{RMSE_{SNP} - RMSE_{FR}}{RMSE_{SNP}}$.