Jaccard Kernel PCA in genotype and gene-burden data for ALS

Author: Francisco Simoes[1] Supervisor: Dr. Kevin Kenna[1]

December 2020

[1]Department of Translational Neuroscience, UMC Utrech Brain Center, University Medical Center Utrecht, Utrecht University, Universiteitsweg 100, 3584 CG Utrecht, the Netherlands

Abstract

The goal of the project was to aid ongoing large scale genetic studies of Amyotrophic lateral sclerosis (ALS) by improving on its population stratification control method. Since genetic differences between population groups confound the statistical methods, it is necessary to control for population stratification. The standard way to do this involves applying principle components analysis (PCA) to common genetic variants. However, in ALS the relevant genetic variants are rare, creating a need for methods that can robustly evaluate population stucture among rare variants. A good candidate is Jaccard principal component analysis (jPCA), a particular case of kernel principal component analysis.

I developed a set of scripts capable of running jPCA on very large datasets (through parallelization) and another one allowing for arbitrary positive integers burden values - thus working not only on genotype data but also on gene-burden data. My implementation of jPCA was tested by checking that its output is identical to the output of another, less general implementation of jPCA when applied to the 1000 genomes project data [1].

Applying jPCA on ALS exome data led to surprising results: although jPCA captures population stratification on the 1000 genomes data [1], it does not capture the sample ancestry structure in a clear way when applied to the exome data explored here, whether we use common or rare variants. The problem seems to lie either on the specific distribution of the exome variants or the data collection limitations. Exactly how this happens is left as an open question for future work.

1 Introduction

Amyotrophic lateral sclerosis (ALS) is a neurodegenerative disease mostly affecting motor neurons, causing progressive weakness of voluntary muscles and eventually death by respiratory arrest [2]. Its cumulative lifetime risk is approximately $\frac{1}{300}$.

It is of obvious importance to identify what genes[1] are associated with ALS. Nowadays it is possible to sequence the entire genome, and use statistical methods to analyze the data in so-called genome wide association studies (GWAS) and gene burden analyses (see §1.2). The distribution of disrupting variants can be compared to population controls to identify genes with an excess of potentially pathogenic variants in sick patients. But these datasets are enormous, and the results may be prone to misinterpretation: so many sequences can be considered that by chance alone one could find a certain sequence overrepresented in sick patients. Auxiliary genetic information can help reducing the search space, *e.g.* one may prioritize variants with higher predicted pathogenicity based for example on alterations of the encoded proteins. Furthermore, mutations of the non-exome part of the genome are even harder to interpret since we don't understand many of its functions. For these reasons, and for its lower cost (and thus higher availability), we use exome sequencing data.

Some problems arise specifically in a ALS GWAS because the variants contributing to risk are rare variants. This complicates matters for a few reasons [2]:

1. When the disease-causing mutations are rare one needs more data to ensure that each mutation is present in enough number to do statistics.
2. Rare variants may be population-specific, making replication difficult.
3. The non-presence of rare possibly pathogenic variants in patients does not mean that we can discard it - after all, it is rare.
4. Rare variants may have different distributions than common variants [3], rendering useless some methods usually applied to common variants.

Some possible solutions are [2]:

[1]For the reader unfamiliar with biology lingo, there is a glossary in §5.2 with some of the most important terms used in this text.

1. Use extremely large datasets from global efforts like Project MinE, which is the largest genetic study for Amyotrophic Lateral Sclerosis.

2. Support the results with biochemical functional analysis.

3. Use gene-burdens (see § 1.1): lower resolution data may identify pathogenic genes, even if sacrificing a nucleotide-scale view of the data.

Let us briefly discuss the type of data we are dealing with and identify the problem to solve.

1.1 The genotype matrix and gene-burdens

A gene can be seen as a sequence of nucleotides from {A, T, G, C}. Each person has two copies of each gene, one from the mother and one from the father.

from mother:	A	T	T	G	A	C	$C\ldots$
from father:	A	A	T	G	G	C	$C\ldots$

Single nucleotide polymorphisms (SNPs) are positions in the genome where there are variations within the sampled population.

person 1:	G	T	**C**	A	A	**C**	C
person 2:	G	T	**A**	A	A	**G**	C
person 3:	G	T	**C**	A	A	**G**	C
person 4:	G	T	**C**	A	A	**C**	C

A SNP has at least two alleles, but can have up to four: one for each distinct nucleotide. The variant with the highest frequency in the population is called the *major allele*. All other variants are *minor alleles*.

For the individual i and SNP j, the *genotype value* (or *SNP-burden*) $b_{i,j}$ is:

$$b_{i,j} = \begin{cases} 0, & \text{no minor alleles at } j \text{ on father and mother genes.} \\ 1, & \text{a minor allele at } j \text{ on one of the parents' genes.} \\ 2, & \text{minor alleles at } j \text{ on the genes of both parents.} \end{cases}$$

the matrix $[b_{i,j}]$ is called the *genotype matrix*.

As discussed in §1, when dealing with rare variants one may want to consider *gene-burdens*, which are simply sums of the SNP-burdens in a gene.

1.2 GWAS and controlling for population stratification

In a *genome-wide association study (GWAS)*, one tries to assess the pathogenicity of the SNP variants w.r.t. a certain disease, using all the SNPs of the genome.

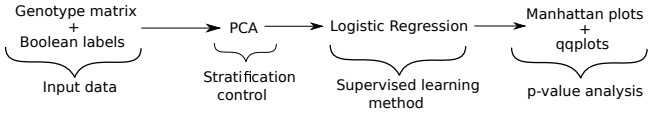
In the language of statistics, the predictors are the genotype values $b_{\cdot,j}$ and the response is a "is sick?" boolean, called the *phenotype*.

Since the number of SNPs in the human population is enormous, very large quantities of data are needed, usually from individuals around the world. This is true even when one restricts oneself to the exome, which is only about 1% of the human genome, as we do. In such a scenario, one cannot simply run a classification model - say, multivariate logistic regression - and expect good results: the ancestry differences between individuals, which are highly correlated with their geographical differences, is a confounding variable that we must account for. Indeed, correcting for population stratification is now common practice in GWASs.

In [4], it is shown that the first two principal components of the data obtained from principal component analysis (PCA) on genotype data from the 1000 genome project (OTGP) are highly correlated with geographical axes. This means that one can use the principal components of a population's genotype data to control for ancestral differences - whose effect on the population genome is known as *population stratification* - by simply taking those principal components as covariates in whatever classification model we choose to use.

1.3 The GWAS pipeline

Given what we said above, the pipeline for a GWAS can be separated in four parts: The input data; a method for stratification control (*e.g.* PCA); a supervised classification model (*e.g.* logistic regression); and some methods for analysing the resulting p-values (*e.g.* manhattan plots and qqplots, which we will not describe here).



Each of these parts can be modified. In this work, we only tinker with the first two:

- Instead of a genotype matrix, my supervisor proposed the use of a matrix of gene-burdens (see §2) for the reasons stated in §1. Thus I tested my methods on both genotype and gene-burden data.

- Instead of using PCA to summarise variance across genetic variants, one must use methods better suited for rare variants. As we will see shortly, I ended up using two types of so-called jaccard PCA, which effectively ignores the nucleotides with no mutations (see §2.2).

2 Data and methods

As explained in §1, I tested my methods to capture population structure in both exome genotype and gene-burden data. The gene-burden matrix was obtained from the genotype data by adding the SNP-burdens of each gene, obtaining as many rows as there are genes in the human exome. Instead of this matrix, I used filtered versions of

it, containing only mutations that affect the protein encodings.

Concretely, this data is divided in three datasets:

1. Genotype data from the *MinE* dataset, corresponding to 1343816 exome and non-exome SNPs from 30820 individuals, approximately 23% of which are ALS patients (phenotype = 1). For the parts where we only wanted rare variants we extracted them from the genotype matrix.

2. The *LOF* dataset: a gene-burden matrix with 12802 exome genes from 30820 individuals. This dataset was constructed by computing gene-burden values from the MinE dataset. It only takes into account rare and severe mutations that are known to cause a loss of function of the encoded protein. Also, this data has been mean-imputated[2].

3. The *moderate* dataset: a gene-burden matrix with 17881 exome genes from 30820 individuals. This dataset was constructed by computing gene-burden values from the MinE dataset. It takes into account moderate (mostly missense) rare mutations. Like LOF, this data has been mean-imputated.

2.1 Alternatives to PCA

PCA selects the directions for which the variance is maximized. Those directions are called the *principal components (PCs)* of the data.

The PCA algorithm always gives us the (orthogonal) PCs of the data, but they are not always meaningful or useful.

PCA is useful when the PCs include most or all of the interesting information in the data. More precisely, this means that the PCA captures enough of the population structure for our purposes.

For PCA to be useful, the following must be true: [5]

1. Linearity: expressing the data in terms of the PCs corresponds to a change of basis.

2. PCs are orthogonal: this helps with the linear algebra problem of finding the PCs.

3. Directions of largest variance are the important[3]ones. This means that signal to noise ratio is high - directions with lower variance correspond to noise.

Remark: The first and second assumptions are automatically satisfied for (multivariate) normal distribution of the data, and the third assumption is usually true for not-too-noisy data. So this justifies the common claim that PCA works for Gaussian data.

[2]Missing values were replaced with the average of their columns.

[3]In the current context of controlling for population stratification, "important directions" are the ones which capture the ancestry differences.

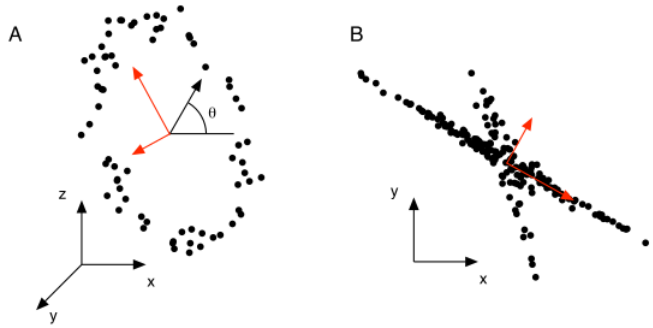


Figure 1: Examples of when PCA fails; taken from [5]. The red arrows represent the two first PCs. A: circular data violates the linearity assumption; the angle θ (from polar coordinates) would be a much better coordinate to use as PC than the PCs from PCA. B: Non-orthogonal PCs - PCA cannot recognize that the PCs are not orthogonal, since it only seaches for othogonal PCs.

Given the good results in [4] (mentioned in §1.2), it seems that the three assumptions above are at least approximately satisfied. But there was no 'a priori' reason to think that this would be the case. For instance, the genotype data is clearly not Gaussian distributed, but Binomially distributed[4].

When dealing with rare variants only, things get worse: it was shown in [3] that, if the genotype data uses rarer alleles, then: the ratio $\frac{\text{inter-population-variance}}{\text{intra-population-variance}}$ diminishes; the distance between populations decreases; and the variance explained by the first PCs also decreases.

Thus, it is clear that we must find an alternative to PCA that is appropriate for sparse rare variant data, as opposed to the non-sparse common variant level genetic relatedness that we usually assess.

Possible alternatives

One may start exploring PCA alternatives that fit our needs with the question: how to extract the population stratification in the OTGP data *with rare variants only*? The answer may suggest methods to use in our own rare variant data.

Fortunately this was already answered in [6]. Two methods stand out: *Logistic PCA (LPCA)* and *Jaccard PCA (jPCA)*, with LPCA giving slightly better results.

LPCA rests on the assumption that the data follows a binomial distribution. At the start of this project the main goal was to control for population stratification *in the gene-burden version of the data*, which is not binomially distributed. Therefore I decided to explore jPCA, and apply it both to the gene-burden and the genotype versions of the data.

[4]The other type of data that we use (gene-burdens) has all the same issues. In particular, it is straightforward to check that the distribution of the gene-burden values is not Gaussian (it is a zero inflated negative binomial distribution).

2.2 Jaccard (Kernel) PCA

Before discussing jPCA one must introduce the *Jaccard index*.

Jaccard index

Given two genotype vectors (two rows of a genotype matrix) a and b, their *Jaccard index* $j(a,b)$ is schematically given by

$$j(a,b) = \frac{|(a_{>0} \cap b_{>0})|}{|(a_{>0} \cup b_{>0})|} \tag{1}$$

where $|(a_{>0} \cap b_{>0})|$ is the number of SNPs where a and b have equal, non-zero genotype values, and $|(a_{>0} \cup b_{>0})|$ is the number of SNPs where either a or b (or both) have non-zero genotype values.

The motivation for defining such an index is the following: when comparing two genotype vectors, a first naive idea is to define their similarity index as the ratio of entries on which they agree. Then, $a = (0,0,0,1,0)$ and $b = (2,0,0,0,0)$ would have a similarity index of $\frac{3}{5}$ since neither has mutations on the second, third and fifth nucleotides.

One can see how this reasoning can go terribly wrong when dealing with rare mutations. In fact, one may get vectors a and b with zeros in thousands of entries, and just a couple of non-zero values in different SNPs. Then, the naive similarity index will be extremely high - which for our purposes is nonsensical: a and b should be considered identical only when they have mutations in the same SNPs. The Jaccard index takes this into account by only considering the entries with positive values.

The *Jaccard matrix* is the similarity matrix

$$J := [j(a_k, a_l)]_{k,l \in \{1 \dots n\}} \tag{2}$$

where n is the number of samples/individuals.

Notice that we defined the Jaccard index in a way that generalizes easily to gene-burdens, since nothing stops the vectors' entries from having values larger than 2. This contrasts with *e.g.* [1], where the author only accounts for burden values in $\{0,1\}$. When clarification is necessary, I will refer to (1) as the *generalized Jaccard index*.

Kernel PCA

jPCA is sometimes described as "applying PCA on the Jaccard matrix" but this is slightly misleading (see §5). Actually, it is a particular type of the so-called Kernel PCA, which we introduce now.

Kernel PCA (kPCA) is a non-linear dimensionality reduction method where the data is non-linearly mapped to a higher-dimensional space (*feature space*) where PCA is finally applied [7][8]. It can be especially useful when trying to separate data that is not separable using standard PCA.

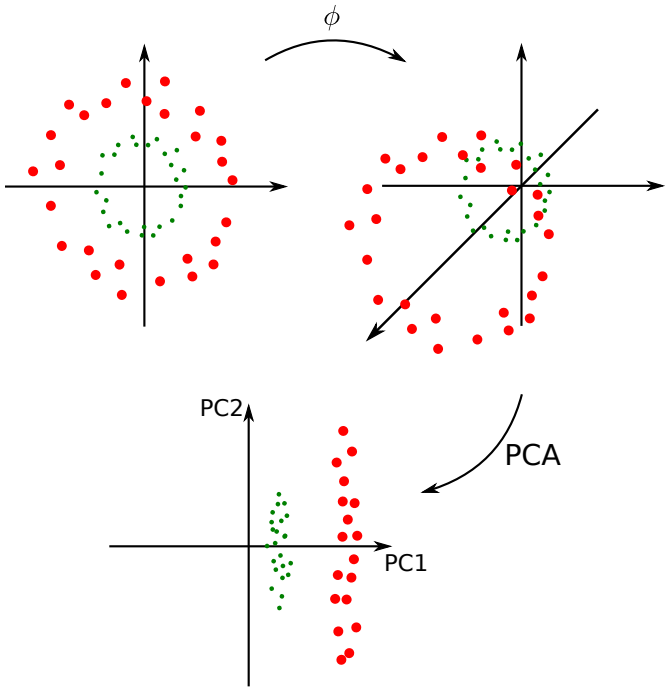


Figure 2: Scheme of kPCA applied to a dataset of concentric circles.

To illustrate kPCA, consider[5] a dataset consisting of two concentric circles (plus noise) - see figure 2. Clearly, PCA would approximately give the original axes as the first two principal components. Hence, because of the non-linearity of the data, PCA is incapable of capturing its structure.

Instead, we can apply a non-linear map $\phi\colon \mathbb{R}^2 \to \mathbb{R}^3$ given by $\phi(x,y) = (x, y, x^2+y^2)$. This captures the structure of the data in the third dimension: the smaller dots from the inner circle will be separated from the bigger dots from the outer circle along the third dimension, since it measures the square of the distance of the original points to the origin.

PCA can now handle the new data and separate the two clusters along one of the first principal components. When applying PCA on a dataset X, one obtains the PCs from X^tX: they are the eigenvectors of X^tX. So in our case one must find the eigenvectors of $\phi(X)^t\phi(X)$: they are the kPCA PCs.

But this computation can be expensive. If we instead find a function $k\colon \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ such that[6] $\forall x,y,\ k(x,y) = \phi(x)^t\phi(y)$, then $\phi(X)^t\phi(X) = [k(x_i,x_j)]_{i,j=1,\dots,n} =: K$, where n is the number of samples in the dataset. Hence the kPCA PCs are the eigenvectors of K, and if we have K from the start we never have to deal with the non-linear function ϕ.

Such a function k is called a *kernel* and is often thought

[5] This example is similar to the one given in the excellent video lecture on kPCA by David R. Thompson, from Caltech.

[6] In other words, k acts as a dot product in $\mathbb{R}^3$.

of as a similarity function. The matrix K is sometimes called the *Gram matrix*.

In practice, one rarely chooses an explicit form of ϕ, and instead deals with kernels from the start. One must be careful when choosing a function as a kernel: we cannot just use any function - it must be one for which a ϕ exists such that $\forall x, y,\ k(x, y) = \phi(x)^t \phi(y)$. Fortunately, one *can* be sure that a certain function k is a kernel if it is "positive definite" as discussed in [9].

The next natural question is then: what kernel should we choose for the genotype and gene-burden datasets?

Jaccard PCA (jPCA) is the particular kPCA method for which the kernel k is taken to be the Jaccard index j [1]. We have seen that j is indeed an appropriate measure of similarity between genotype and gene-burden vectors when dealing with rare mutations. Using j as the kernel should effectively help separate samples with different mutations in the feature space even if their genotype/gene-burden vectors coincide in most non-mutated entries. This may help PCA better capture the genetic variation between samples when considering only rare variants. Similar to what happened with regular PCA and common variants data, we expect that genetic variation to be a proxy for population stratification. Furthermore, it turns out that the Jaccard index is positive definite [9], meaning that we can indeed use it as a kernel.

3 Results

I ran my own implementation of jPCA on the datasets described in §2, and tested it on the OTGP data as a control, comparing the results with [1].

3.1 jPCA on the 1000 Genomes Project data

In [1], jPCA was successfully applied to the OTGP data to extract population structure, obtaining better results than PCA when applied to rare variants. That paper uses an open-source R library called *jacpop*, which computes the jPCA principal components from a genotype matrix using the function *generate_pw_jaccard*.

Unfortunately, this library was unfit for this work, because:

1. The MinE dataset is too large to be fed to *generate_pw_jaccard*. So one needs a flexible script that one can parallelize.

2. *generate_pw_jaccard* uses the standard Jaccard index, not the generalized Jaccard index, and is thus unfit for gene-burden data (the LOF and moderate datasets).

So I created a collection of Python scripts that generalized *jacpop* for genotype data by allowing parallelization, and another collection of Python scripts that generalized *jacpop* for gene-burden data by allowing burdens of arbitrary positive value. Clearly, the former should still give the same results as *jacpop* in the OTGP data, so I used both methods using 5e5 randomly chosen variants and verified that they output identical plots (see figure 3).

3.2 Gene burdens

For this part I used the moderate data.

The plots obtained by applying jPCA to the gene-burden data show a highly irregular pattern that does not seem to correlate with sample ancestry: in contrast with the results in [1] and [6], the first two PCs do not show a clear separation between the different populations (figure 4).

Another surprising result is that the jPCA PCs do not correlate at all with the PCs from PCA applied to the genotype data. Since the latter already capture the population structure fairly well, this indicates that jPCA may not be not able to do the same using the gene-burden, rare variants exome data. However, this can also be a symptom of some problems/biases in the quality of the sequencing data. This warrants further investigation in the future.

We may also be interested in knowing if the PCs separate sick and healthy people. They don't: there's no visible separation (see figure[7] 5) and, more rigorously, running logistic regression resulted in very large p-values, and the model simply predicted 0 every time (thus having an accuracy of $\frac{\text{number of healthy individuals}}{\text{number of individuals}} = 77\%$ and null F1-statistic).

This contrasts with results coming from the use of PCs from conventional PCA: in that case the first 5 PCs are significant (with nearly null p-values) and the F1-statistic is nonzero (albeit still unsatisfactory with a value of 0.02).

3.3 Genotype values - rare variants

We now apply jPCA on the MinE genotype data, including only the rare variants.

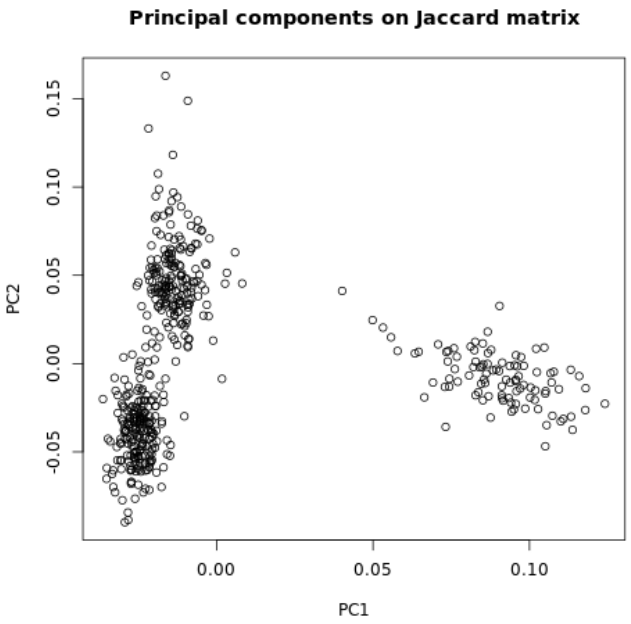
The results still show no clear separation between cohorts (figures 6 and 7). Furthermore, logistic regression (with the phenotype as the response variable) gives similar results to the gene-burden case, and again the jPCA PCs do not correlate with the standard PCs.

3.4 Genotype values - common variants

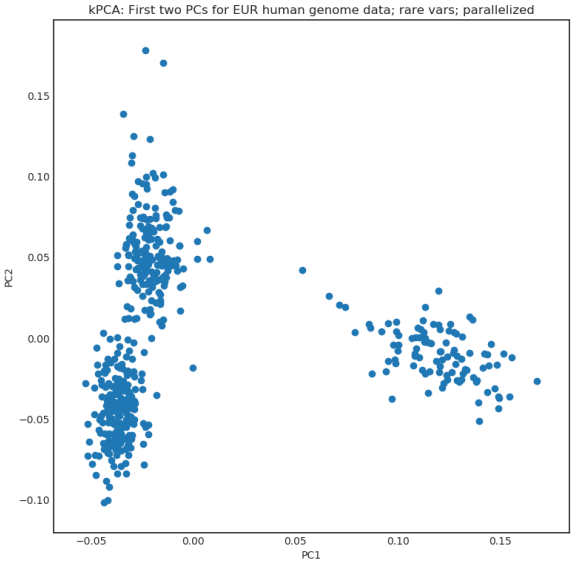
Next we applied jPCA on the MinE genotype data, now using exclusively the common variants.

The first two PCs separate the data in two (see figures 8 and 9). Contrary to our expectations, these two clusters do not have an obvious association with the cohort labels.

[7] You may notice that the two plots from figures 4 and 5 are slightly different. This is because in the first plot I removed the points whose cohort label was "nan".



(a) Plot from *generate_pw_jaccard* using 5e5 variants from the OTGP data.



(b) Plot from my (parallelized) jPCA script using 5e5 variants from the OTGP data.

Figure 3: Comparison between plots generated by *generate_pw_jaccard* and by my script when applied to OTGP data.

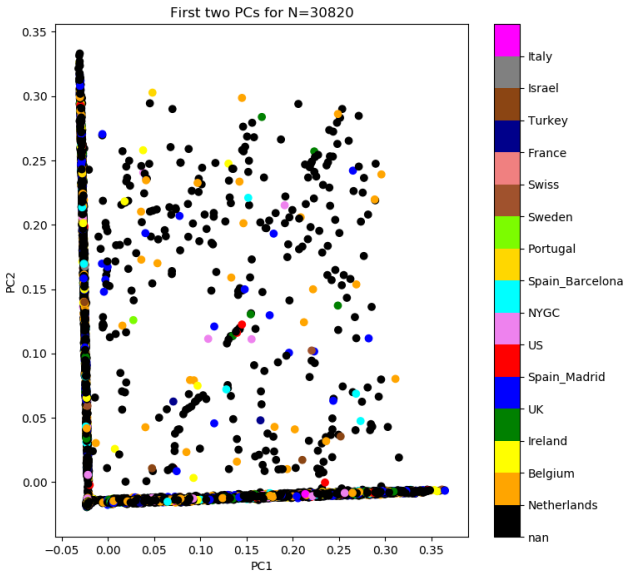


Figure 4: First two jPCA PCs in the Gene-burden data, labelled by cohort.

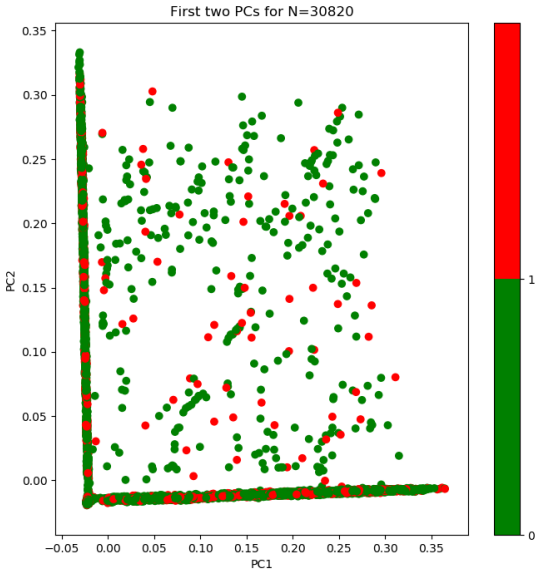


Figure 5: First two jPCA PCs in the Gene-burden data, labelled by phenotype.

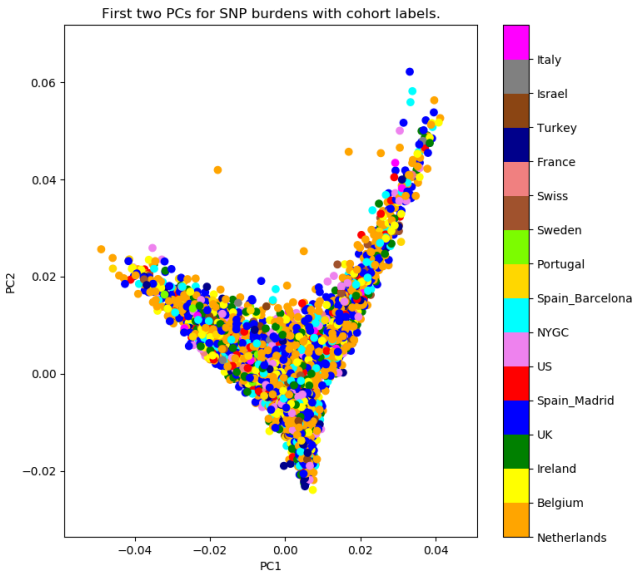


Figure 6: First two jPCA PCs in the genotype data, labelled by cohort.

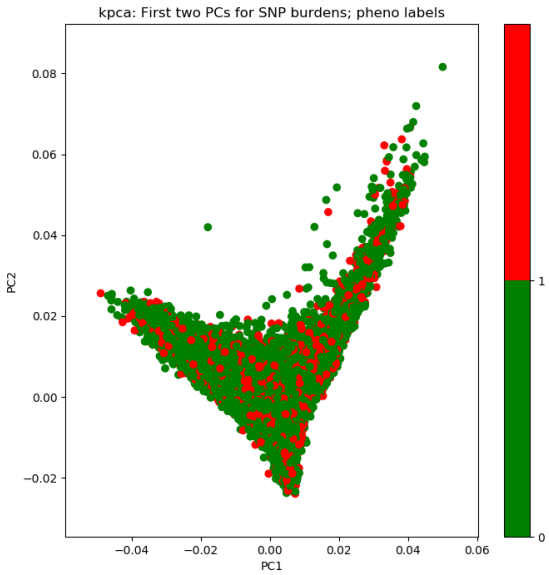


Figure 7: First two jPCA PCs in the genotype data, labelled by phenotype.

Logistic regression with phenotype as the response variable gives similar results to the gene burden case, and again the jPCA PCs do not correlate with the standard PCs.

The natural question to ask is: why does jPCA capture the population structure in the OTGP data (whether we use rare or common variants [1]) but not in the genotype data that we are using (whether we use rare or common variants)?

To understand this, one may start by looking at the differences between these datasets: although they both consist of genotype data, the OTGP data contains the SNPs of the entire genome of the population, while the MinE data only contains the exome SNPs. Furthermore, while the MinE data that we are using is an amalgamation of data from different countries and laboratories, the OTGP data comes from a single source sequencing project, meaning that the MinE data is a lot less uniform than the OTGP data. This can throw off the results. For example, the two clusters in figure 8 may very well be an artifact of mean-imputation (see the appendix §5), similar to what happens in §5.1. This is clear by comparing figures 10 and 12a in light of the discussion in §5.1.

From these results alone it is not possible to conclude which of these explains the unusual and as of yet unexplained structure seen with jPCA.

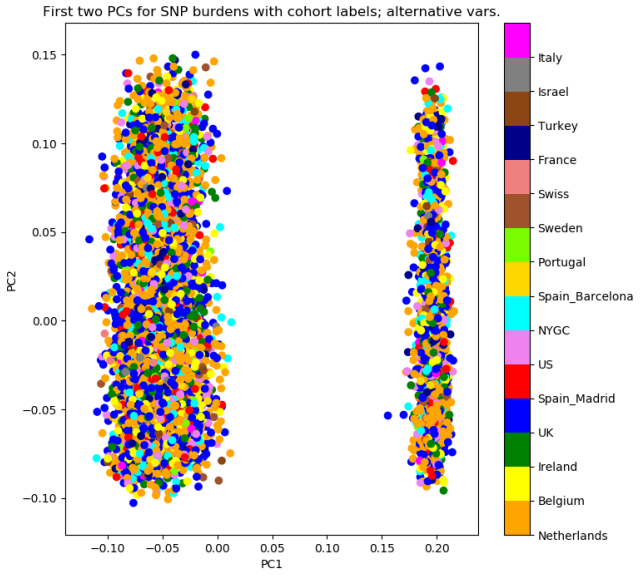


Figure 8: First two jPCA PCs in the genotype data, labelled by cohort.

4 Conclusion

The first two PCs resulting from applying jPCA on moderate (gene-burden) data do not capture the population

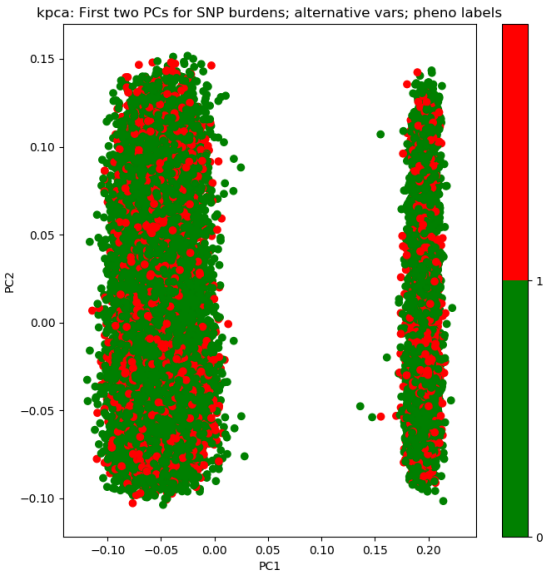


Figure 9: First two jPCA PCs in the genotype data, labelled by phenotype.

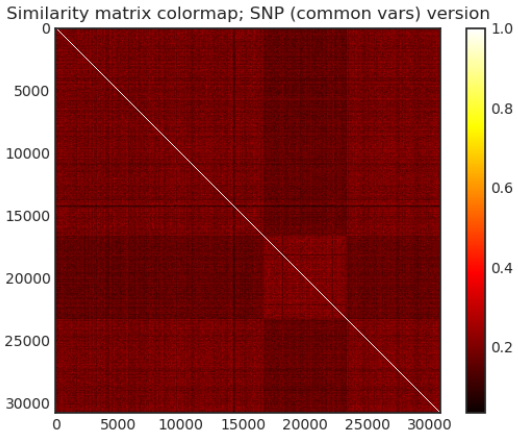


Figure 10: Heatmap for Jaccard scores from the genotype data - common variants.

structure. The same happens on the rare variants of the MinE genotype data, possibly because by using all rare variants one includes extremely rare variants occurring on only one or two individuals each, which can throw off the selection of PCs. Hence further filtration of the data could solve this, for example by removing rare variants that actually result from errors made by DNA sequencing machines.

The true mystery is the fact that jPCA does not capture population structure in the MinE genotype data using the common variants. Here, such filtration problems are inexistent, and there seems to be no good reason for the results observed, given that jPCA does capture said structure when applied to the OTGP data [1][6].

The issue must come from the technical differences between the MinE genotype data and the OTGP data, namely: the MinE data we used contains exome-sequencing only, which may have a different distribution than the genome-sequencing data; the OTGP data is more uniform - the MinE data comes from dozens of different projects, and for example the imputation methods that were used to cope with the missing data may create spurious issues.

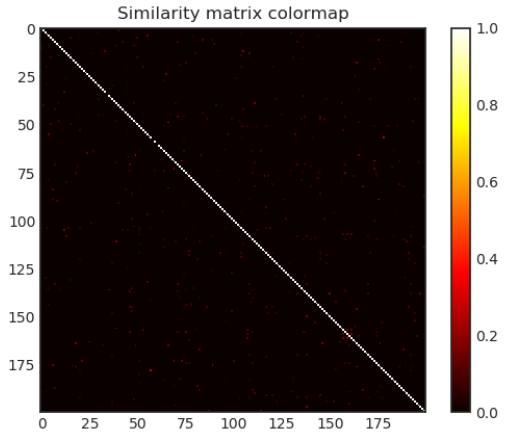
One way to pinpoint the peculiarities of the MinE dataset responsible for these results would be to apply jPCA to the entire genotype data (not just the exome). Furthermore, it could be instructive to investigate the origins of the two clusters on figure 8. It may also be interesting to re-run jPCA on the rare-variants, genotype data after filtering out the extremely rare mutations. Finally, the arguments that I have seen so far in favour of using the Jaccard index as a kernel are not completely convincing. Thus one could explore in detail the geometrical implications of using this kernel, and see if it indeed makes it so that the data in feature space is primed for the use of PCA. These are left for future work, since my internship has come to an end.

5 Appendix

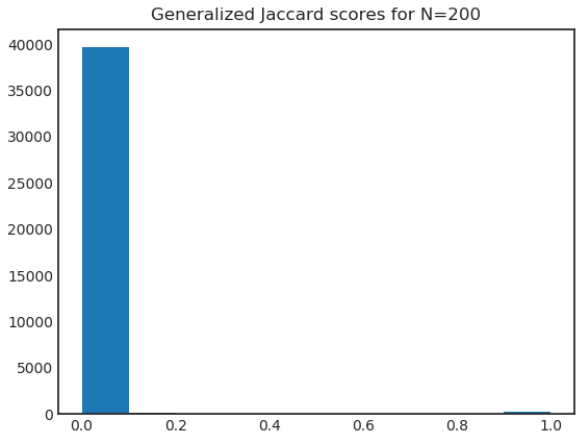
5.1 PCA on Jaccard values

In [6] and [1], the authors describe jPCA as applying PCA on the similarity matrix created using the so-called Jaccard index as the similarity value. This is not exactly what jPCA is, as discussed in the main text. But their abuse of terminology suggested a different approach: why not try PCA on the Jaccard values themselves? After all, the gene-burden data may be approximately Gaussian distributed: each gene-burden is the sum of a large number of genotype values (each binomially distributed), and there *may* be enough consistency between binomial distributions and genes that the central limit theorem suggests an approximate Gaussianity. It was a long shot, but it was worth a try.

One important lesson came from this exploration: how rounding mean-inputed data can give spurious results. I



(a) Heatmap for Jaccard scores using 200 individuals.



(b) Histogram for Jaccard scores using 200 individuals.

Figure 11: Jaccard scores for the LOF data, using 200 individuals.

will illustrate this with the problems that appeared when applying PCA on the Jaccard matrix from the LOF data, but the moderate data case was similar.

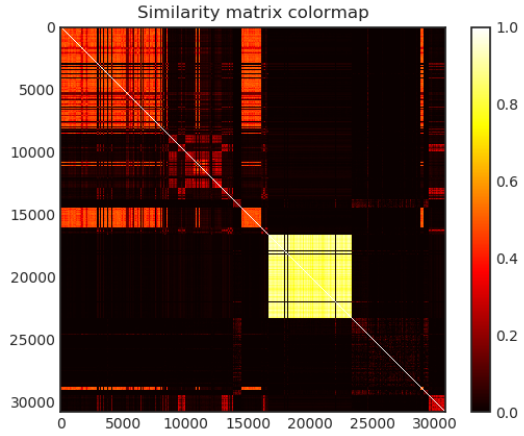
Building the Gram matrix from the LOF data one sees that, as we feared, the data Jaccard values are very low and mostly zero or very close to zero (figure 11).

Looking at the LOF data, one may notice that many values are not integers. This is due to the many mean-inputed values in the dataset. Because of the way Jaccard values are computed, we will get zeros even if the individuals have similar (but non-integer) gene-burden values.

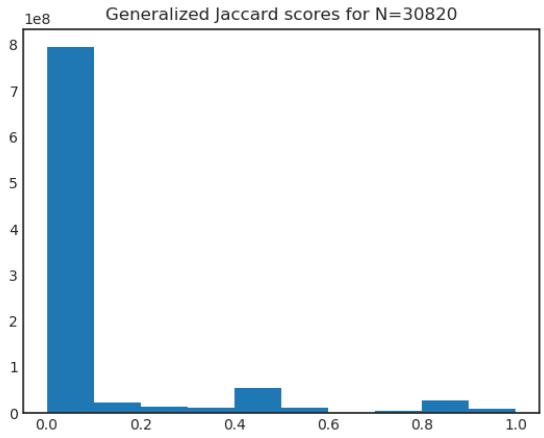
A natural solution is to round these values. This gives interesting results (figure 12), although still heavily skewed towards zero.

I spent some time zooming in and trying to understand the origins of the non-zero peaks of the histogram.

One may realize that something odd is going on by looking at the well-defined yellow and red squares in the



(a) Heatmap for Jaccard scores using 30820 individuals.



(b) Histogram for Jaccard scores using 30820 individuals.

Figure 12: Jaccard scores for the LOF rounded data, using 30820 individuals.

heatmap. What are the chances that the similar individuals are all in the same region of the dataset? It's much more likely that these apparently interesting results are actually an artifact of mean-imputation followed by rounding the results: many of the individuals with close index numbers are from the same cohort, and thus have missing values in many of the same SNPs, so that mean-imputation + rounding effectively turns two individuals from the same cohort very similar under the Jaccard index.

This is the danger of using mean-inputation (with data with many missing values and/or very rare nonzero values) together with the Jaccard index.

5.2 Glossary

Allele: a variant form of a given gene, meaning it is one of two or more versions of a gene. It can also refer to a region of interest in the genome. In this last sense, alleles can come in different extremes of size. At the lowest possible end one can be the single base choice of a single nucleotide polymorphism (SNP). At the higher end, it can be the sequence variations for the regions of the genome that code for the same protein which can be up to several thousand base-pairs long.

Common variant: SNP variant/allele with an allele frequency smaller than 0.5 in the sample (hence any minor SNP allele is a common variant).

Deoxyribonucleic acid (DNA): is a molecule composed of two polynucleotide chains that coil around each other to form a double helix, being connected by hydrogen bonds at their nucleotides. Its nucleotides are: A, T, C, G.

Gene: a sequence of nucleotides in DNA or RNA that encodes the synthesis of a genetic product, either RNA or protein.

Gene expression: the process by which information from a gene is used in the synthesis of a functional gene product. These products are often proteins, but in non-protein coding genes such as transfer RNA (tRNA) or small nuclear RNA (snRNA) genes, the product is a functional RNA.

Genome: Sum total of an organism's DNA.

Genome-wide association study (GWAS): observational study of a genome-wide set of genetic variants in different individuals to see if any of those variants are associated with a phenotype. Typically, the variants are simply SNPs and the phenotype is the presence or absence of a (human) disease. In more detail: GWA studies compare the DNA of participants with varying phenotypes for a particular trait/disease. The cases are the individuals with the disease, and the controls are the ones without (this is the traditional approach, called phenotype-first). The DNA of every subject is read using snip arrays, letting us know what allele occurs in each person. If one allele is more frequent in cases than in controls, it is said to be associated with the disease.

Nucleotides: molecules consisting of a nucleoside (five-carbon sugar ribose + nitrognous base) and a phosphate group. They are the basic building blocks of DNA and RNA.

Population Structure: A population has structure when there are large-scale systematic differences in ancestry and/or groups of individuals with more recent shared ancestors than one would expect in a randomly mating population.

Rare variant: SNP variant/allele with an allele frequency smaller than 0.01 in the sample.

Variant: An alteration in the most common DNA nucleotide sequence. The term variant can be used to describe an alteration that may be benign, pathogenic, or of unknown significance. The term variant is increasingly being used in place of the term mutation.

Aknowledgements

I would like to thank Dr. Kevin P. Kenna, my supervisor, for giving me this opportunity and helping me navigate bioinformatics without a bioinformatics background. I also want to thank Paul Hop, currently a PhD candidate under Kevin's supervision, who assisted me with many technical and theoretical hiccups. Finally this wouldn't have been possible without the data from Project MinE, containing the ALS pacients in the MinE dataset.

References

[1] Dmitry Prokopenko, Julian Hecker, Edwin K Silverman, Marcello Pagano, Markus M Nöthen, Christian Dina, Christoph Lange, and Heide Loehlein Fier. Utilizing the jaccard index to reveal population stratification in sequencing data: a simulation study and an application to the 1000 genomes project. *Bioinformatics*, 32(9):1366–1372, 2016.

[2] Ammar Al-Chalabi, Leonard H Van Den Berg, and Jan Veldink. Gene discovery in amyotrophic lateral sclerosis: implications for clinical management. *Nature Reviews Neurology*, 13(2):96, 2017.

[3] Shengqing Ma and Gang Shi. On rare variants in principal component analysis of population stratification. *BMC genetics*, 21(1):1–11, 2020.

[4] John Novembre, Toby Johnson, Katarzyna Bryc, Zoltán Kutalik, Adam R Boyko, Adam Auton, Amit Indap, Karen S King, Sven Bergmann, Matthew R Nelson, et al. Genes mirror geography within europe. *Nature*, 456(7218):98–101, 2008.

[5] Jonathon Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.

[6] Asuman S Turkmen, Yuan Yuan, and Nedret Billor. Evaluation of methods for adjusting population stratification in genome-wide association studies: Standard versus categorical principal component analysis. *Annals of human genetics*, 83(6):454–464, 2019.

[7] Ali Ghodsi. Dimensionality reduction a short tutorial. *Department of Statistics and Actuarial Science, Univ. of Waterloo, Ontario, Canada*, 37(38):2006, 2006.

[8] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[9] Thomas Gärtner, Quoc Viet Le, and Alex J Smola. A short tour of kernel methods for graphs. *Under Preparation*, 2006.