Compact Genome Representations for Machine Learning

Fermi Ma FERMIM@PRINCETON.EDU
Cyril Zhang CYRIL.ZHANG@PRINCETON.EDU
35 Olden Street, Princeton, NJ 08544

Abstract

A database of human genomes, represented as a table of variants, exhibits sparsity and redundancy; as a result, numerous lossless compression schemes for genomic data have been proposed. However, to perform useful operations, this data typically needs to be decompressed. In this paper, we give a lossy genome compression scheme that can be useful in its compressed form, which we demonstrate by classifying ethnic groups. Our method determines a ranking of alleles by importance for ethnicity classification, and achieves significantly stronger results than other ranking heuristics. We also discuss a speculative application in large-scale data mining.

1. Introduction

1.1. Compression of genomic data

In recent years, the development of high-throughput DNA sequencing technology that can process thousands of genomes in parallel has significantly lowered the cost of DNA sequencing. As a result, storing and transferring enormous amounts of genomic data has become a central problem. Ordinary data compression tools such as zip and rar are not tailored to take full advantage of the structure of the genome, and thus a wave of new genome-specific compression methods have sprung up. In this paper, we approach this compression problem from a machine learning perspective, using data from the 1000 Genomes Project.

1.2. The 1000 Genomes data set

The 1000 Genomes Project is an open-source data set consisting of genome data from 2504 individuals spread over 26 worldwide ethnic groups (Consortium et al., 2012). The project aims to include all alleles that appear with greater than 1% frequency in each population. The data set does not come with any phenotype information other than population label and kin relations.

The full data set, in its final version, consists of 84,739,846 genetic variants. In our studies, as a representative subset of autosomal genome data, we focus on contiguous sites on chromosome 22 for all 2504 individuals at which there is a single allelic variant (rather than multi-allelic sites). We choose to work with 5000 alleles at a time for ease of computation.

1.3. Rarity of high-variance alleles

Figure 1 shows that over 40% of variants occur *only once* across the sampled individuals; in fact, around 90% of variants occur fewer than 10 times. This observation is central to representing genome data concisely: there exists a small subset of alleles that exhibits a high degree of variation.

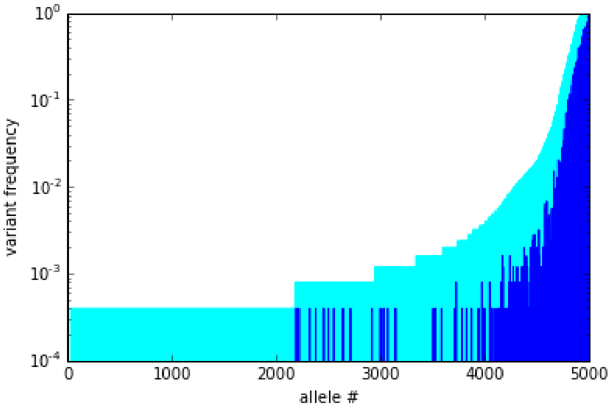

Figure 1. Sorted frequencies for 5000 allelic variants (blue = homozygous, cyan = heterozygous).

The publishers of the 1000 Genotypes data set note that rarer alleles tend to indicate more recent mutations, and thus are generally restricted to just one population. As a result, these rare alleles encode a disproportionately high amount of information about ethnicity. We will qualitatively verify this claim in this paper, and this will be an

Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

important fact in determining feature ranking in our compression scheme.

Another useful exploratory view of the data set is the variant bitmap, shown in Figure 2. This provides a visual verification of the sparsity structure of our data: a large region of isolated points (rare variants) dominates the grid, while the common variants form a thin, dense region of high frequency and variance. Additionally, it is interesting to note the emergence of a discrete band structure. The individuals (vertical axis entries) in Figure 2 are grouped by ethnicity, which strongly suggests that a linear model will successfully classify between these clusters. Each ethnic group evidently possesses a visually identifiable *fingerprint*.

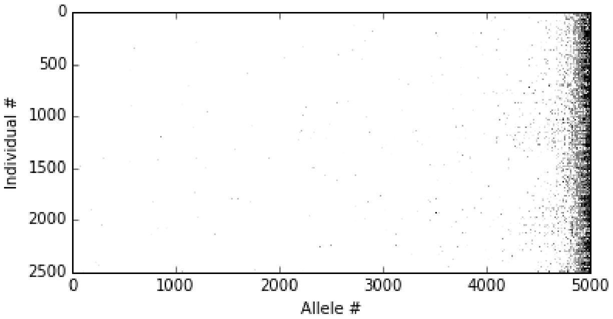

Figure 2. Variant bitmap for 5000 alleles across 2504 individuals (white = homozygous reference, black = homozygous variant, grey = heterozygous). Alleles are sorted in order of increasing variant count; the thin black band indicates that a small fraction of alleles have a high number of variants, while most other alleles rarely appear in their variant forms.

It is not immediately clear, however, that the sparse region also encodes phenotypic information. In our methods, we recover features that are important in determining ethnicity as the components that arise from sparse SVM classification. We find that the maximally informative sparse genetic fingerprints do incorporate a mixture of high-variance and low-variance alleles.

From the perspective of compression, the sparse and dense regions can be represented using distinct schemes. As a first-cut measure, we can store the sparse components (variants which occur with frequency below some threshold α) as a variable-length list of nonzero coordinates, and the dense region as a bitmap. Both formats are very easy to parse in isolation, in comparison to a run-length encoding approach (Layer et al., 2015) We reason that this simple compression scheme is sufficient to exploit the sparsity of the data; such strategies as run-length encoding, delta encoding, and Burrows-Wheeler transform preprocessing are designed to exploit further structure of the bitmaps as strings. In the interest of implementation simplicity, we do not examine these approaches.

1.4. The ethnicity classification problem

The ethnicity labels provided by the 1000 Genomes data set allow us to conveniently benchmark our methods; we can evaluate the viability of our compressed data representations by attempting to distinguish between ethnicities, which are well-separated in a simple linear model in the original vector space.

More specifically, we wish for our data representation to have the property that a short prefix of it (essentially truncating the coordinates of the genotype vectors corresponding to each individual) is sufficient to separate clusters; as we read more input coordinates, our solution becomes more refined.

As mentioned, each genotype vector is labeled with one of 26 ethnic groups. The number of samples within each group ranges from 61 (ASW, African Americans in the southwestern US) to 113 (GWD, Gambians in the Western Division of the Gambia).

2. Related work

2.1. Data representation and compression

Human genotypes are typically stored as a table of variations relative to a fixed reference genome, stored elsewhere. This is a ubiquitous practice, and is regarded as part of the data reduction rather than compression, after sequence alignment. The result of this process is usually a table in **variant call format** (vcf), in which each row corresponds to a point of variation, and the columns give the identity of that allele across each individual in the sample set. In this paper, we will be concerned with further compressing the genome database, starting from the vcf representation.

Layer et al. give a heuristic method for genome compression (Layer et al., 2015). They take the transpose of the vcf table, and permute the columns (now corresponding to genotypes at each allele) in increasing order of allele count in the sample population. This greatly increases compressibility; they subsequently apply a simple run-length encoding scheme so that the data can be indexed, locally decompressed, and queried efficiently. The authors show that this gives a lossless compression rate roughly on par with the LZ77 algorithm, reducing the space requirement of the 1.3TB raw vcf files to a much more manageable 14GB.

Christley et al. gave an alternative method for genome compression. Their method avoids storing the full position of each single nucleotide polymorphism (SNP) along the chromosome, but instead the distance from the previous SNP, a much smaller number. They use Huffman coding as a final step, and achieve a 1000-fold compression on James

Watson's genome (Christley et al., 2009). This work has inspired a number of other papers, such as one by Pavlichin et al., that slightly improve the compression ratio by taking advantage of known genome structure (Pavlichin et al., 2013).

Our work differs markedly from these approaches to genome compression. Current compression schemes are lossless and rely on a full reference genome in order to decompress the data. Furthermore, it seems difficult to reach any useful conclusions by looking only at the compressed files. In this paper, we focus on achieving a lossy compressed data representation that can be useful in its compressed form. Our data representation will allow us to better use the genotype vectors for linear models in machine learning. Unfortunately, since the 1000 Genomes Project data includes no phenotypic information other than ethnicity, we can only train and test our data representation on the ethnicity classification task.

2.2. Compression-based machine learning

One conceivable way to perform machine learning on compressed data is to use the symbols from the compression algorithm directly as features. This has been explored in natural language processing to some success. Sculley and Brodley show that the symbols from Lempel-Ziv and predictive prefix matching (PPM) family serve as effective feature vectors (in an implicit high-dimensional vector space) for a document classification problem. (Sculley & Brodley, 2006). Under this benchmark, they find that these vectors outperform the n-gram and binary bag-of-words models by a small margin, suggesting that the symbols encode the structure of the text.

In this paper, we examine an analogous method. However, we find that compression of genome data does not significantly preserve cluster structure. This is perhaps unsurprising: whereas the compressibility of natural language text derives from redundancy, the compressibility of genome data is largely due to the rarity of most alleles.

3. Methods and Assessment

In the remainder of this paper, we freely interchange the terminology of alleles, features, and coordinates (of genotype vectors), depending on whether context calls for us to emphasize the perspectives of applications, machine learning, or compression. These terms refer identically to the genetic variations at a single allelic site. When we map this to a number, we are referring to the number of copies of the variant allele possessed by an individual, which can be 0, 1, or 2.

3.1. PCA indicates high linear separability

One way to immediately verify the linear separability of ethnic group clusters is to run principal component analysis (PCA) on the genotype vectors (after z-scoring each coordinate). Even in two dimensions, some pairs of clusters – for example, the Japanese and African Caribbean groups – are clearly well-separated (Figure 3). Although this degree of separability under this two-dimensional projection is atypical, this suggests that the linear model is faithful to the structure of ethnic clusters. In particular, we may use the separation of the JPT and ACB clusters as a benchmark for the viability of a linear model with features derived from any particular lossy compression scheme. In the following section, we use this benchmark to show a negative result for features derived from text compression.

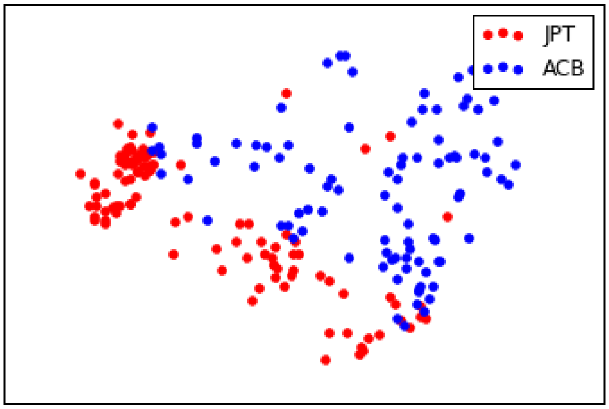

Figure 3. Plot of z-scored genotype vectors labeled JPT (Japanese in Tokyo) and ACB (African Caribbeans in Barbados), reduced to 2 dimensions via PCA. x-axis is the component along the first factor; y-axis is the component along the second.

3.2. Clustering with compression-based features

We attempted to replicate the approach described by Sculley and Brodley, in which we take the keys of a compression algorithm's symbol table as the set of coordinates in an implicit feature space, with components set as the appearance counts of these symbols. Embedding the benchmark clusters that were well-separated under 2-dimensional PCA on the raw data, we see in Figure 4 that the clusters are not recovered.

We found that although string compression algorithms do successfully produce a shorter representation of the genome data, the symbols obtained are not reliably informative of the structure of an individual's genome or of ethnicity. The symbols that this method produces can be reliable for learning about text, as we can use them for important information about redundancies. However, the primary

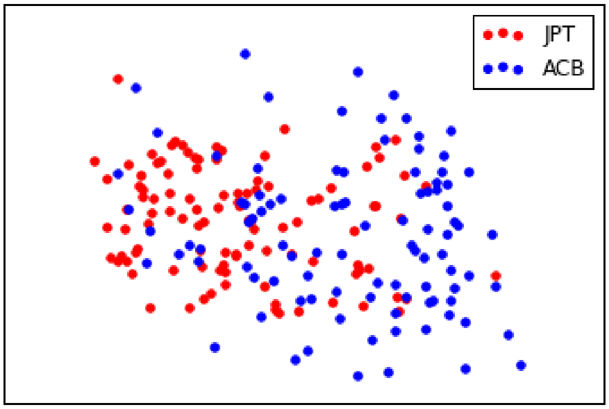

Figure 4. 2-dimensional PCA embedding of JPT and ACB clusters, using LZW symbol table-derived feature space.

redundancy in our sparse genome data is due to long strings of zeros. As a result, the symbols depend heavily on the positions of nonzero entries, which we do not believe encodes important genomic information. This is likely why we cannot recover clusters in Figure 4.

We hypothesize that the compression-based features may be salvageable for different problems in bioinformatics. Although it is difficult to justify interpreting a variant bitmap as a string, it is entirely natural to interpret a raw sequence of base pairs in the framework of string redundancy. As such, this approach may be fruitful in classifying between snippets of DNA belonging to very different organisms based on the statistical properties of their base pair strings.

3.3. Finer-grained cluster separation via SVMs

We consider the problem of finding hyperplanes to separate ethnic groups in our data, with the hope that normal vectors to these hyperplanes will give useful information for a compression scheme.

To find hyperplanes that separate the ethnic clusters, we can use the linear support vector machine (SVM) model.

Since each allele corresponds to a feature, the dimensionality of the data set is extremely high; the features far outnumber the sample set. This will be true in any genomic data set in the foreseeable future, as it would be prohibitively expensive to perform whole exome sequencing on the order of 10^8 individuals. Even in our representative sample of 5000 alleles, careful attention must be taken as to prevent overfitting of models. Thus, we focus on the scope of high-dimensional data with low sample size.

Dimension reduction (via, for instance, PCA) is a common answer to these constraints, and is shown above to preserve separating hyperplanes. However, when we perform such a drastic reduction, we lose the informative value of specific coordinates.

Instead of finding a separating hyperplane, we strengthen the requirement significantly: we seek a separating hyperplane whose normal vector has support over a small subset of coordinates. This is the classic problem of sparse regression. We found that applying the well-known LASSO ℓ_1 regularization technique indeed produced sparse separators. For the JPT and ACB groups ($N = 104 + 96$), we obtained a linear combination of 30 alleles (out of 5000) which almost completely separated the clusters (Figure 5).

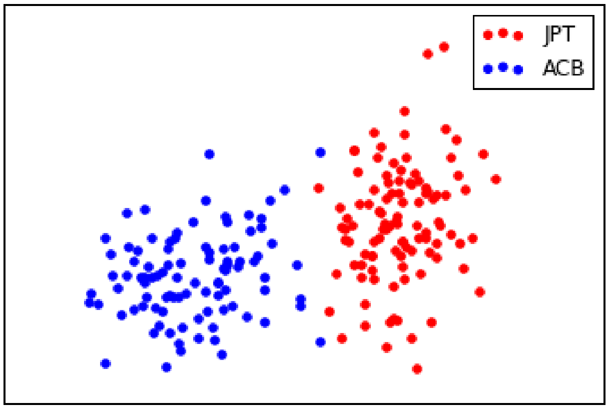

Figure 5. Plot of JPT and ACB clusters, with x-axis determined by component along sparse separating hyperplane normal. Y-axis is a random projection, for visualization purposes.

3.4. Cross-validation of sparse linear separators

We tested our sparse SVM classifier for the possibility of overfitting. We tested the model with the upper bound on the ℓ_1 norm set to 0.3, which in this case returned a separating hyperplane with 42 nonzero components. For this experiment, we again focused on classifying the JPT and ACB populations.

We performed 10-fold cross validation, and found near-perfect separation on the unseen testing data for all 10 partitions. In Figure 6, we show the results of 2 of the 10 iterations of the 10-fold cross validation scheme.

3.5. Compression method by ranking

We considered the following compression scheme:

1. Divide the data into *blocks* of 5000 alleles.
2. In each block, identify the alleles most useful for ethnicity classification by finding (sparse normal) sepa-

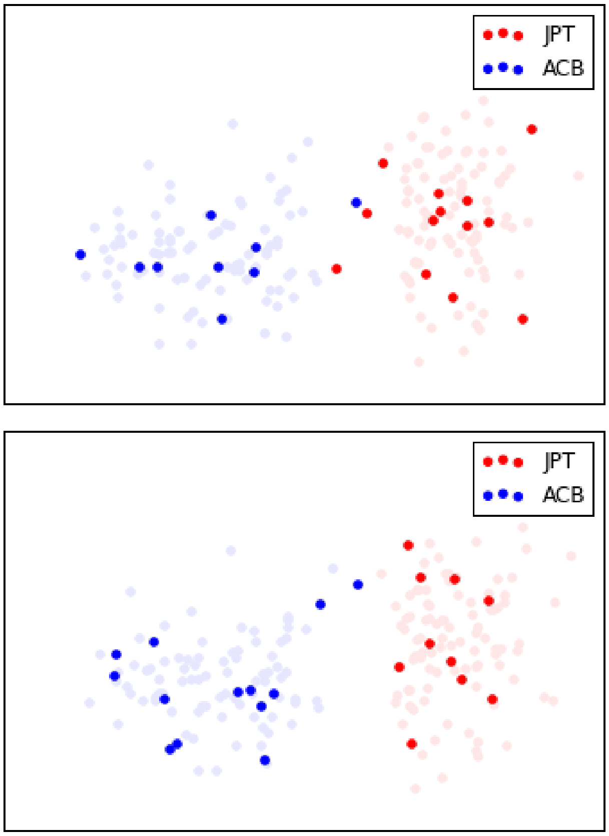

Figure 6. Plots of 2 of the 10 partitions in 10-fold cross validation between JPT and ACB clusters. The x-axis is determined by the component along the separating hyperplane from a sparse SVM classifier trained only from the training set. The y-axis is a random projection, for visualization purposes. Faded points are the training set (the remainder of the points) for each fold.

rating hyperplanes for each pair of ethnicity clusters; call the normal vector for the jth hyperplanes $\{n_j\}$.

3. For the ith allele, let $\hat{n}_i = \sum_j |n_j|$. This is a rough measure of the importance of that allele.

4. Within each block, separate the alleles $\{i\}$ into two groups: $A^+ = \{i|\hat{n}_i > 0\}$, and $A^0 = \{i|\hat{n}_i = 0\}$. Sort A^+ in decreasing order of $\hat{n}_i$.

5. Let F_1 be the interleaved rows of all the $\{A^+\}$ from each block. Store this as a dense matrix.

6. Let F_2 be the rows of all the $\{A^0\}$ with more than ϵn entries (we try $\epsilon = 0.1$), ordered by decreasing variant count. Store this as a dense matrix.

7. Let F_3 be the remainder of the data, the sparse rows of $\{A^0\}$, ordered by decreasing variant count. Store this as a sparse matrix.

8. Output F_1, then F_2, then F_3 as the reordered, compressed data set.

This orders the alleles by their importance when available, and heuristically by count otherwise. We tested our method on 20 contiguous blocks of chromosome 22, for a total of 100,000 alleles.

3.6. Assessment of compression efficiency

After processing the entire data set, one can find the size of our compact representation, and relate it to the total vcf size. This gives an empirical compression efficiency, which can be compared to others in practice.

On a segment of the data set, the compression ratio can be measured as

$$\frac{\text{size of dense data}}{\text{size of dense } F_1, F_2 + \text{size of sparse } F_3}.$$

3.7. Assessment of permutation quality

Our method can be evaluated against two other permutations of the data: random, and sorted by decreasing variance. We employ the following methodology:

1. Take the first $L = 10$ lines of the permutation. Train an ℓ_1-regularized SVM classifier on the ethnicity classification test problem.

2. Record the classification accuracy.

3. Take the next L lines, and add them to the data set. Repeat with these new, higher-dimensional genotype vectors.

However, with this data set, the paucity of phenotypic labels forces us to use the same classification problem to derive and test this permutation. This potential circularity is addressed using a training and testing scheme similar to leave-one-out cross-validation. We validate the results of these experiments as follows:

1. Choose two ethnicities C_1 and C_2 as the test set. Let all other genotypes be in the training set.

2. Run our compression/ordering scheme on the training set.

3. Assess the quality of the permutation for classifying between C_1 and C_2.

This tests if our permutation, trained on a subset of clusters, is suitable for distinguishing between novel clusters.

3.8. The genotype covariance matrix

When considering methods to evaluate the performance of our ranking algorithm, one desirable quality is that it does not appeal to the same classification problem from which the ranking is derived. However, since we only have one type of label, such a method must only involve some function of the genotype vectors only. The space of such "first-principles" objectives is extremely limited.

One (and possibly the only) natural object to consider is the covariance matrix of individuals. In settings such as detecting and removing population stratification effects, a covariance matrix is directly required. However, upon experimentation, it was shown that randomly sampling alleles performed much better than our method or any other heuristic tested. We interpret this negative result as arising from the high autocorrelation of the vectors; in the absence of costlier tools that negate the notion of ranking, sampling alleles from any ranking will result in redundancies that skew the estimator for the covariance matrix. Thus, we proceed with the validation scheme described in the previous section, despite the undesirable property that we cannot be certain that it generalizes to all phenotypes.

3.9. An alternative ranking scheme

Our ranking scheme involves comparing 5000 alleles at a time in 20 separate sparse SVM classifiers, and then weaving together the results to get an overall permutation of 100,000 alleles. Of course, there may be a better choice of parameters, or a smarter way to combine the results of multiple SVM classifiers. Here, we consider the possibility of a more robust scheme for determining the best ranking of alleles

If we view each sparse SVM ranking step as a "competition" among alleles, this ranking problem can be solved with existing methods for developing a ranking based on numerous small competitions. The general framework is that there are a fixed number of contestants who repeatedly compete in ranked competitions involving a subset of the competitors, and the goal is to come up with an overall ordering of the competitors by skill. In the Bradley-Terry model, each contestant (allele) is given a positive-valued parameter γ_i, where $\Pr[i \text{ beats } j] = \gamma_i/(\gamma_i + \gamma_j)$.

We can obtain the maximum-likelihood values of γ through iterative computations given the results of numerous competitions between the alleles (Hunter, 2004). This suggests the following ranking method:

1. Run sparse SVMs on 100 (any small constant will suffice) randomly chosen alleles at a time, and denote the winners of each competition as the alleles whose component in the sparse SVM are non-zero.
2. Calculate the maximum-likelihood values of γ.
3. Rank alleles in decreasing order of γ.

Unfortunately, to calculate values of γ, we have to run enough competitions (sparse SVM ranking steps) for the following assumption to be satisfied: for every partition of the alleles into two sets, one allele in the second set beats some allele in the first set in some competition (Hunter, 2004). Running enough sparse SVM ranking steps to satisfy this assumption for 100,000 alleles is extremely computationally expensive, and thus we were unable to even run this ranking scheme.

4. Results

4.1. Compression ratio

Our preliminary test on 100,000 alleles gives a compression ratio of 2.63, with block size 5000 and $\epsilon = 0.1$. Note that the reference uncompressed size is that of the dense matrix, which is a much more compact representation than the raw vcf files. Applying the method of Layer et al. and measuring compression ratio in the same way gives a ratio of 1.67. This suggests that our method achieves a competitive compression rate.

4.2. Classification efficiency

Figure 7 shows typical results for the clustering problem on prefixes of permutations of the alleles. In each run of the experiment, we recompute the permutation, excluding two of the clusters, then attempt to separate them by a sparse SVM classifier. Our permutation stays reliably ahead of the others in terms of accuracy in low dimensions, and typically reaches 100% accuracy with the fewest features; we demonstrate this more formally in the following section. We thus conclude that our feature ranking method succeeds in nontrivially capturing the structure of the clusters.

4.3. Quantifying difficulty of separation

From Figure 7 and similar plots for other pairs for ethnicities, we remark qualitatively that our ranking does particularly well in separating groups that are "close", according to geography or known migrational history. To test a rigorous claim of this nature without relying on anecdotal evidence, we use the haploid mitochondrial DNA of the individuals, included with the data set.

Indeed, we verified that the (0-1) mitochondrial DNA vectors, with 3874 variants, exhibited high population clustering (mean $F_{ST} \approx 0.123$, compared to 0.075 for the 5000-allele segment of chromosome 22). Furthermore, due to the the fact that mutations occur very infrequently, mitochondrial DNA is known to be a good determinant of ancestral

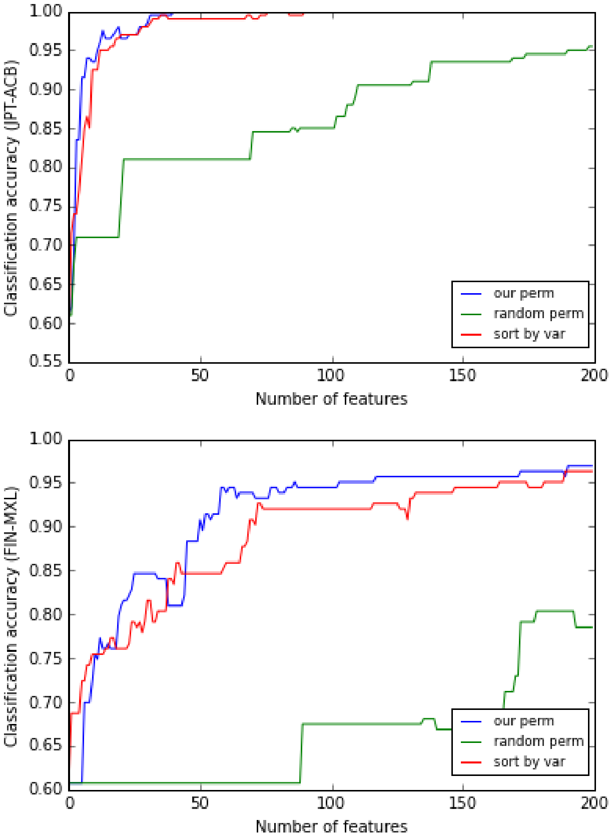

Figure 7. Plots of ℓ_1-regularized classification accuracy vs. number of features in the permutation associated with our method, alleles selected in decreasing order of variance, and a random permutation, for two typical pairs of clusters.

origin, and can be used to inform a notion of "distance" between ethnic groups (Lee et al., 2011). For each ethnic group, we compute the average mitochondrial DNA vector, and then we define the distance between two ethnic groups be the ℓ_2 distance between their corresponding average mitochondrial genome vectors.

Next, we require a consistent measure of the *advantage* of our permutation over the sort-by-variance heuristic. To grant equal weighting to all prefix lengths, we compute for each classification problem the integral of the signed difference between the blue and red curves, as in Figure 7, such that a positive (negative) integral implies that our permutation does better (worse) than sort-by-variance. Some typical examples of these integrands are shown in Figure 8.

Using these measures of cluster distance and advantage, we verified the claim that our method performed better on "harder" problems (differentiating between genetically closer groups). Our results are summarized in Figure 9, where there is a clear negative trend as expected. Furthermore, only 12 out of 325 points lie below the zero line, showing that our permutation does not often confer a disadvantage; when it does, it is a modest one. Interestingly, the pairs of ethnicities for which our permutation comparatively does the worst are almost exclusively between tribal groups in sub-Saharan Africa.

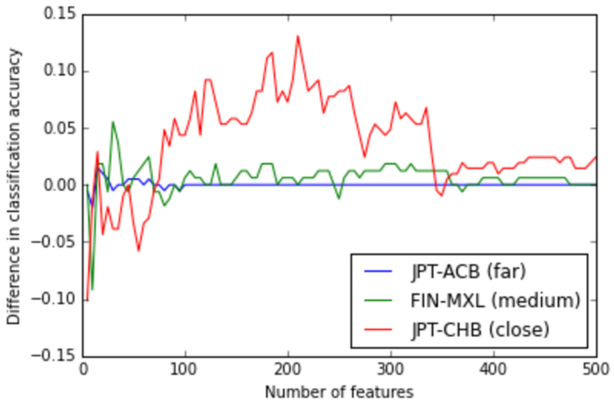

Figure 8. Plot of difference in classification accuracy between our permutation and the sort-by-variance permutation vs. number of features used in classification. Three different pairs of ethnicities with varying mitochondrial distances are considered: Japanese vs. African Caribbean (far), Finnish vs. Mexican Ancestry in Los Angeles (medium), and Japanese vs. Han Chinese (close).

5. Discussion

5.1. Application in big-data genomics

We imagine that our results might be useful in a big-data setting, in which coordinate truncation can be modulated to trade off desired accuracy for running time. In general, our feature selection scheme produces a rough ranking of alleles by importance in separating a set of given labels; thus, in permuting the data set by this ranking, any statistical method can be tuned by taking the first D columns. In the future, as personal whole exome sequencing becomes increasingly affordable and accessible, available data will become richer. We propose the following workflow:

1. Select a phenotype P, and a desired classification error ϵ.
2. Take the first D rows (alleles) from the data set, and train a classifier for P.
3. If the classifier achieves error greater than ϵ, repeat with $2D, 4D, 8D, \ldots$ rows.

At the end of this process, assuming that the set of alleles correlated with the phenotype is not too localized, we have

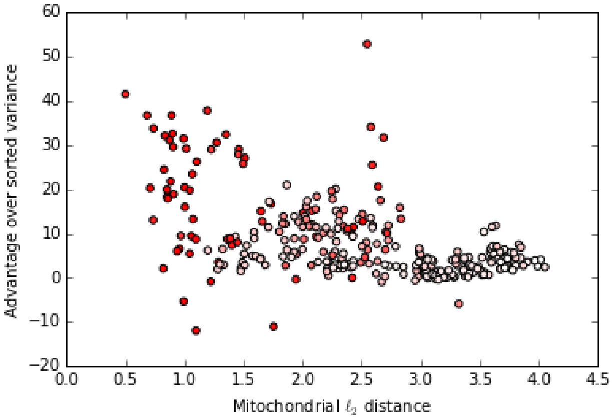

Figure 9. Plot of advantage in classification accuracy vs. mitochondrial distance between ethnic groups. Shading (redness) intensity is proportional to "difficulty" of the classification problem, measured by number of features required to reach 95% classification accuracy.

a truncation of coordinates suitable for our specific problem. We imagine that this may be useful when we wish to use the data set to classify (or regress) a large number of diverse phenotypes.

References

Christley, Scott, Lu, Yiming, Li, Chen, and Xie, Xiaohui. Human genomes as email attachments. *Bioinformatics*, 25(2):274–275, 2009.

Consortium, 1000 Genomes Project et al. An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491(7422):56–65, 2012.

Hunter, David R. Mm algorithms for generalized bradley-terry models. *Annals of Statistics*, pp. 384–406, 2004.

Layer, Ryan M, Kindlon, Neil, Karczewski, Konrad J, ExAC, Exome Aggregation Consortium, and Quinlan, Aaron R. Efficient genotype compression and analysis of large genetic variation datasets. 2015. doi: 10.1101/018259.

Lee, Chih, Măndoiu, Ion I, and Nelson, Craig E. Inferring ethnicity from mitochondrial dna sequence. In *BMC proceedings*, volume 5, pp. S11. BioMed Central Ltd, 2011.

Pavlichin, Dmitri S, Weissman, Tsachy, and Yona, Golan. The human genome contracts again. *Bioinformatics*, 29 (17):2199–2202, 2013.

Sculley, D. and Brodley, C.E. Compression and machine learning: a new perspective on feature space vectors. pp. 332–341, March 2006. ISSN 1068-0314. doi: 10.1109/DCC.2006.13.