Genetics and Population Analysis

Implementation of Algorithms for Inter- and Intrapopulation Genomic SNP Analysis

Ryota Ashizawa[1]*, Jacob Crosser[1*], Sergei Kotelnikov[1], Arjun Rao[2], Niven Singh[1]

[1]Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY.
[2]Department of Computer Science, Stony Brook University, Stony Brook, NY.

*To whom correspondence should be addressed.

Abstract

Motivation: Specific SNPs play a crucial role in determining population health. Studying all possible sites of mutation can be prohibitively costly, motivating the need for tools to identify a (useful) subset with which the remaining SNP information may be reconstructed. An algorithm to accomplish such a reconstruction was proposed by Pashou et al. in 2007. However, a publicly available version of this algorithm was not published alongside it. Towards the dissemination of these efficient genomics tools, we discuss the implementation of their algorithms in the language of Python.

Results: The application of the recreated algorithms on a dataset similar to that of the original study reproduced several large-scale trends. The presence of these trends in the data produced by the current implementation algorithms supports their effectiveness.

Availability: https://github.com/arjunhrao/interpopulation_SNP_genotype_reconstruction
Contact: ryota.ashizawa@stonybrook.edu; jacob.crosser@stonybrook.edu
Supplementary information: N/A

Introduction

Single nucleotide polymorphisms (SNPs) account for a large portion of the genetic variation in the human genome. For a stretch of DNA in the genome, a specific set of bases may vary between individuals in a population, and these single-nucleotide differences may lead to significant differences in expressed phenotypes. These SNPs may lead to heritable diseases or affect the susceptibility of an individual to other afflictions (LaFramboise, 2009). Exhaustive studies searching for specific causal SNPs to a statistically-reliable degree would be both cost- and time-prohibitive; genotyping a specific group of individuals of significant sample size would be extremely costly even before attempting to identify a handful of SNPs from the millions present in the human genome (Paschou et al., 2007). Luckily, the identity of SNPs at neighboring sites in the genome are not strictly independent due to the linkage disequilibrium observed in the genome. It should therefore be possible to identify a more manageable subset of SNPs that captures or approximates the genetic information found over all of the SNPs, hereon referred to as tagging SNPs (tSNPs) (Paschou et al., 2007). In 2007, Paschou et al. constructed a method for computationally identifying a set of these tSNPs through inter- and intra genotype reconstruction from a set of population data.

The Paschou et al. algorithms represent population-level SNP data as simple matrices, allowing them to utilise linear algebra methodologies. The method put forward takes advantage of the spectral

decomposition of the data matrices; the algorithms identify a small set of columns from the data matrices that may be used to reconstruct the rest of the matrix to a predetermined degree of accuracy. These determined columns correspond to the set of tSNPs identified by the algorithm. The authors go on to demonstrate the ability to reconstruct SNP data for individuals which was not present when determining these tSNPs. Finally, the authors studied the degree to which tSNPs identified in one population could recreate the genetic diversity in a separate population.

Our goal was to reconstruct the matrix algorithms developed by Paschou et al. and apply them to a new data set. We isolated targeted tSNPs throughout multiple geographic regions in an attempt to create inter- and intrapopulation genotypic representations using a subset of the 1000 Genomes Project dataset (Birney and Soranzo, 2015). We used the 1000 Genomes Project dataset as a substitute since the HapMap database has been retired by the NIH. We discuss the implementation of previously described algorithms for tSNP identification in Python. This takes advantage of several aspects of the language, namely the efficient linear algebra operations from the Numpy package, the efficient parsing of data files, and the accessibility of the language to the general public. These include: an encoding algorithm, a method to evaluate the linear structure in SNP matrices, a multipass greedy algorithm, and a reconstruction algorithm for unassayed SNPs. We retrieved the 1000 Genomes Project dataset corresponding to the genomic regions studied previously, tested our instantiation of the algorithms on the data, and cross-examined the results while performing a trend comparison despite utilising different datasets.

Results

The analysis discussed here was conducted on the data from the 1000 Genome Project as opposed to the HapMap and Yale datasets. This was done, in part, because the HapMap repository has been decommissioned since the Paschou et al. paper was published. Making this change brings into question the comparability of results between the original results and those obtained in this study. Differences in the datasets include the number and identity of assayed individuals, the boundary definitions for each region, and the number of identified SNPs within each region. Despite these differences, the results obtained from our application of the algorithms were qualitatively similar to those observed in the original study.

The analysis conducted here identified similar numbers of eigenSNPs and tSNPs as the original study. The current implementation of the linear structure algorithm found ~2-3 eigenSNPs in the PAH locus of each population and roughly two to three times as many in the SORCS3 region for the same population (~5-10); the 17q25 locus, in turn, had approximately two to two and a half times as many eigenSNPs (~15-25) as in the SORCS3 locus (**Fig. 1**). The last part of this observed trend is supported qualitatively by the results of the Paschou et al. study, but the original study reported similar numbers of eigenSNPs in the PAH and SORCS3 loci. The trend in tSNPs identified by the *Greedy Multipass* algorithm observed here is that, for the SORCS3 and 17q25 loci, the number of tSNPs is roughly 50% higher than the number of eigenSNPs for most of the populations. In contrast, the algorithm identified no more tSNPs than the number of eigenSNPs for all populations in the PAH locus. The original study found similar behaviors in their sample populations, but these behaviors presented across all four loci. This provides mixed support for the current implementation.

The results of the interpopulation reconstruction of genomic SNP data are presented in **Fig. 2**. The scaling of the heat map coloration in the present diagrams span are different than that used in the original study, which should be kept in mind when reading the presented data. The diagonal entries of these diagrams correspond to the final reconstruction error for a population when using their data to identify tSNPs, which should result in sub-threshold levels of error; this is the trend observed in the

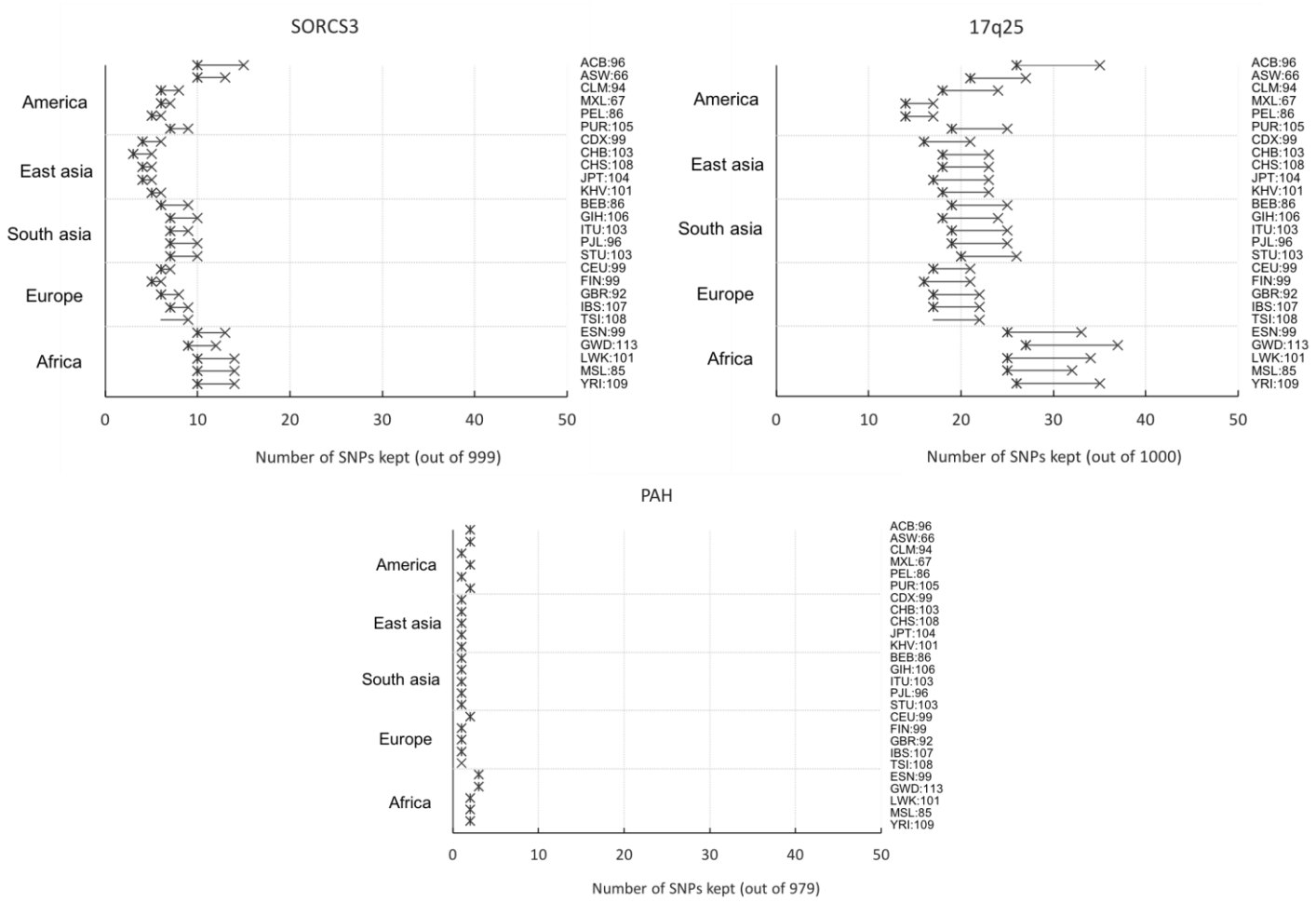

Figure 1. Number of EigenSNPs computed with 'Evaluating the Linear Structure' algorithm (left) vs Number of EigenSNPs computed with the Multipass Greedy algorithm (right). The number of individuals in each population sample is denoted next to the population's abbreviation. Populations are ordered from bottom to top based on geographic regions (abbreviations used are shown in parentheses). **Africa**: Yoruba (YRI), Mende (MSL), Luhya (LWK), Gambian (GWD) and Esan (ESN); **Europe**: Toscani (TSI), IBS, GBR, FIN, and CEU; **South Asia**: STU, PJL, ITU, GIH, and BEB; **East Asia**: KHV, JPT, CHS, CHB, and CDX; **Americas**: PUR, PEL, MXL, CLM, ASW, and ACB.

presented data. When looking at the interpopulational reconstruction within a region, populations from the African region tend to have the highest error. This is a trend also observed in the original study. It is also observed in the present data, across all three loci, that the error in reconstruction of data from populations in the Americas is lowest when the reference population is from the European or East Asian regions (upper middle regions of the heat maps). This trend is present in the original study as well. Additionally, it was observed in both studies that there was a relatively low reconstruction error when using the data from Asian populations to reconstruct data from European populations, at least in the SORCS3 and PAH loci. Finally, it was observed that the error in interpopulational reconstruction was the highest overall when looking at the 17q25 locus.

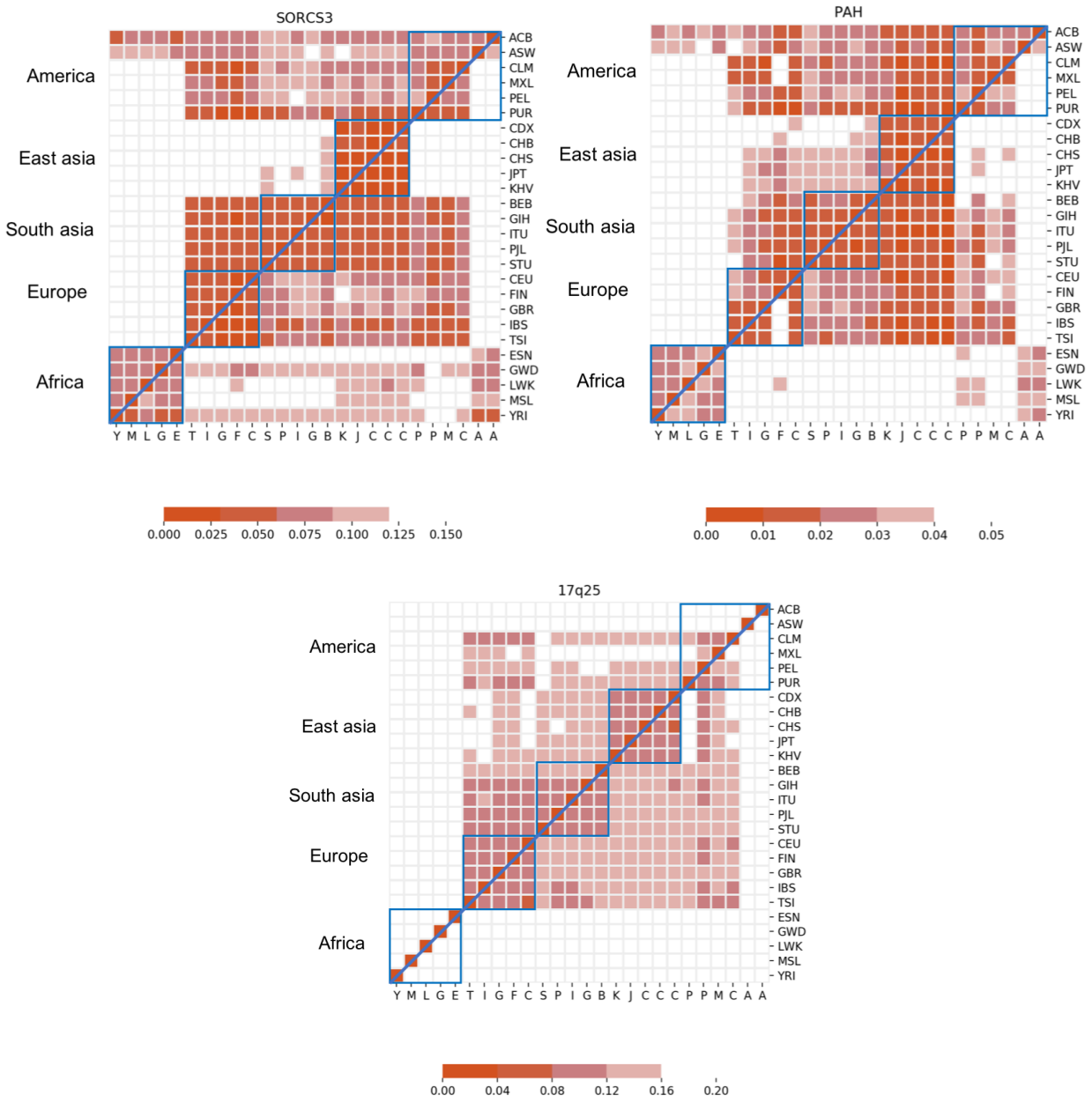

Figure 2. Interpopulational genomic reconstruction. Each entry depicts the reconstruction error that occurs when recreating the data for the population on the *x*-axis with the population on the *y*-axis acting as the reference genome. Data is presented for the SORCS3 (upper left), PAH (upper right), and 17q25 (bottom) loci.

Discussion

While it is hard to say with certainty that the current implementation of the algorithms described by Paschou et al. are completely accurate to the original implementation, the efficacy of the described reconstruction is supported by the shared trends in observed in reported data. As mentioned in the summary of results, the scaling from eigenSNPs determined by singular value decomposition to the number of identified tSNPs is, on the whole, consistent with the scaling seen in the 2007 paper. There are discrepancies in how the scaling behaviors present across the studied loci, but there are a few potential sources for these differences. The most likely cause is that the prefiltering of SNPs for the PAH locus may

have identified a subset containing an abnormally high degree of linear structure, making the *Greedy Multipass* algorithm more effective in identifying tSNPs in this locus. Additionally, the error in interpopulational reconstruction observed in this paper was, on average, much lower than that observed in the original study. This could again be attributed to the prefiltering process, described in the methods section, effectively increased the linear structure found in the analyzed data. Despite these discrepancies, reproduced trends support the idea that the current implementation of the described algorithms was effective. The data provided demonstrated the ability of the algorithms to characterize the amount of linear structure in a matrix of SNP data, identify key tSNPs from a provided set of data, and to reconstruct the data of unassayed populations from that of an assayed one. An immediate extension of this is the ability to reconstruct the data of an individual with low relatively low error when provided with an appropriate reference population. This has implications for the field of medicine, the efficient identification of genetic risk factors given a small amount of input information being an example.

Methods

Datasets and Pre-Analysis Filtering

The data presented in this paper was taken from the database for the 1000 Genomes Project (https://www.ncbi.nlm.nih.gov/variation/tools/1000genomes/). This database contains information regarding the genomic variation in more than 2535 individuals from 26 populations where a large subset of this data corresponds to biallelic SNPs. To establish effective implementation of previously described algorithms (Paschou, 2007), analysis was conducted on the SNPs in the SORCS3, PAH, and 17q25 loci to allow for a more direct comparison to the results of the original paper. The allelic data from these regions was first filtered to only include sites with biallelic, single-nucleotide variations. Each of these regions had on the order of tens of thousands of variation sites, many of which had nearly zero variation across the assayed populations. As the goal of the algorithm set was to recreate the variation in an observed population, we elected to filter these sites to include only ~1000 sites with the most variation as measured by the frequency of the minority allele.

Quantifying Linear Structure

The analysis of linear structure for a given population was conducted in much the same way as previously described. The process as originally described uses the singular value decomposition of the numerical data matrix $\boldsymbol{A^x}$, generating the following matrices:

$$\boldsymbol{A = U\Sigma V^*}$$

The columns of the matrix $\boldsymbol{U}$ correspond to a basis for the columns of $\boldsymbol{A^x}$, meaning that all of the columns of $\boldsymbol{A^x}$ can be recreated from the columns of $\boldsymbol{U}$. The algorithm then finds the number of number of columns of $\boldsymbol{U}$ needed to approximate $\boldsymbol{A^x}$ to a threshold level of error by iterating through an increasing subset of the columns of $\boldsymbol{U}$ and approximating $\boldsymbol{A^x}$ as $\boldsymbol{A^k}$ with these columns using a least-squares fit. The method for determining the least-squares fit, left unspecified in the original paper, also utilizes tools from linear algebra. For a system of the following form:

$$\boldsymbol{A^X = UC}$$

The matrix $\boldsymbol{C}$ corresponding to the best approximation of $\boldsymbol{A^x}$ is given by:

$$\boldsymbol{C = (U^*U)^{-1}U^*A^X}$$

Finally, the error in this approximation was measured as the percentage of entries with an inappropriate value and was compared to the desired error threshold.

Selection of tSNPs

Our next objective was to select a small set of tSNPs that was representative of the population-level data and use it to reconstruct the genotypes of unassayed individuals. The algorithm for describing the linear structure of $\boldsymbol{A^x}$ estimated the matrix using a subset of the columns of $\boldsymbol{U}$, which are themselves linear combinations of the columns of $\boldsymbol{A^x}$ and thus lack meaning in relation to observed SNPs. In contrast, the *Multipass Greedy* algorithm chooses columns from $\boldsymbol{A^x}$ that can act as a set of tSNPs. The matrix $\boldsymbol{E}$ is initially set equal to $\boldsymbol{A^x}$. As the loop is iterated, the function *f[i]* finds the column *i* of $\boldsymbol{E}$ such that the largest

subset of the columns of ***E*** can be represented by this singular column. Column *i* is added a list of tSNPs and the projection of $\boldsymbol{A^x}$ onto the space spanned by these tSNPs is set as the current approximation of $\boldsymbol{A^x}$. If the error in this approximation is less than the desired threshold (10% or 1%), the algorithm is terminated and the identified set of tSNPs is taken as the output. Otherwise, ***E*** is set to be the difference between $\boldsymbol{A^x}$ and the projection of $\boldsymbol{A^x}$ onto the current set of tSNPs and the process is iterated on the new matrix ***E***. The projections of matrices onto subspaces are, in general, determined by:

$$\boldsymbol{proj}(\boldsymbol{B}\ onto\ \boldsymbol{A}) = \boldsymbol{A}\,(\boldsymbol{A^T A})^{-1}\boldsymbol{A^T B} = \boldsymbol{A} * \boldsymbol{LeastSquareApprox}(\boldsymbol{A},\boldsymbol{B})$$

In the implementation presented here, the projections are calculated using left multiplication of the subspace matrix by the least squares fit of the starting matrix onto the subspace; these operations were carried out using matrix operations defined in the Numpy package.

Data Reconstruction

Reconstructing the SNP data from an unassayed population was accomplished using the tSNPs identified by the *Multipass Greedy* algorithm. For two populations ***X*** and ***Y***, the total SNP data matrices $\boldsymbol{A^x}$ and $\boldsymbol{A^Y}$ can be approximately reconstructed from matrices $\boldsymbol{T^X}$ and $\boldsymbol{T^Y}$ that hold the tSNP data in the following manner:

$$\boldsymbol{A^X} = \boldsymbol{T^X C} \quad where \quad \boldsymbol{C_{fit}} = (\boldsymbol{T^X})^{+}\boldsymbol{A^X}$$

$$\boldsymbol{A^Y} = \boldsymbol{T^Y C}$$

In **+** in the above formula denotes the Moore-Penrose inverse, or pseudoinverse, of $\boldsymbol{T^X}$. Having the full SNP data $\boldsymbol{A^x}$ for population ***X*** and the tSNP data for population ***Y***, an approximation of $\boldsymbol{A^Y}$ can be obtained by:

$$\boldsymbol{A^Y_{approx}} = \boldsymbol{T^Y C_{fit}} = \boldsymbol{T^Y}(\boldsymbol{T^X})^{+}\boldsymbol{A^X}$$

The difference between this formulation and that provided in the original paper is only one of nomenclature. The matrix ***W*** in the original algorithm is here referred to as $\boldsymbol{T^X}$ to reflect the shared informational relevance between $\boldsymbol{T^X}$ and $\boldsymbol{T^Y}$. These computations were also carried out using the Numpy package, but this time featuring an operation to directly determine the pseudoinverse of an input matrix.

References

LaFramboise, T. (2009) Single Nucleotide Polymorphism Arrays: A Decade Of Biological, Computational And Technological Advances. *Nucleic Acids Research* **37**, 4181-4193

Paschou,P., Mahoney,M.W., Javed,A., Kidd,J.R., Pakstis,A.J., Gu,S., Kidd,K.K. and Drineas,P. (2007) Intra- and interpopulation genotype reconstruction from tagging SNPs. *C. S. H. Laboratory Press*, **17**, 96-107

Birney,E. And Soranzo,N. (2015) The end of the start for population sequencing. *Nature* **526**, 52-53