Fast sparse canonical correlation with flashpca — Supplementary Material

Gad Abraham and Michael Inouye

February 19, 2016

1 Reproducibility

Code to reproduce these experiments is at `https://github.com/gabraham/scca-paper`.

2 HapMap data preprocessing and quality control

The HapMap3 phase III [1] genotypes were obtained from `ftp://ftp.ncbi.nlm.nih.gov/hapmap/genotypes/2009-01_phaseIII/plink_format/`. Gene expression levels were obtained from `http://www.ebi.ac.uk/arrayexpress/experiments/E-MTAB-264`.

We excluded individuals who were non-founders, had genotyping missingness >1%, or did not have matching gene expression data, resulting in 601 individuals. We excluded non-autosomal SNPs, SNPs with MAF <5%, missingness >1%, and deviation from Hardy-Weinberg equilibrium $P<5 \times 10^{-6}$ using PLINK 1.9 [2, 3], leaving 1,088,401 autosomal SNPs (see below). The remaining missing genotypes were randomly imputed according to the frequencies of the non-missing observations.

For the gene expression data, we used a subset consisting of the 21,800 probes that were analysed by [4], utilising the original authors' normalised data. Following [4], we performed PCA on the genotypes within each population, and for the GIH, MEX, MKK, and LWK regressed out 10 PCs of the genotypes (as well as intercept) from the corresponding gene expression levels, in order to adjust for the higher levels of admixture within these populations. We further filtered probes with low variance (std. dev. <0.1), leaving 18,193 probes. Both the gene expression levels and the genotypes were standardised to zero-mean and unit-variance.

3 Comparison of predictive power in simulation

Utilising 10,000 SNPs from chromosome 1, we simulated 1,000 gene expression levels as

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E},$$

where $\mathbf{X}$ are the genotypes ($n \times p$ matrix), $\mathbf{B}$ is a $p \times m$ matrix of weights, and $\mathbf{E}$ is an $n \times m$ matrix representing the error (noise). To match the sparsity assumptions of SCCA, $\mathbf{B}$ was chosen to be a mixture of weights $\{0.001, 1\}$ with proportions 0.9999 and 0.0001 (across all $n \times m$ entries), respectively, (0.001 rather than zero was used to prevent some probes from having zero genetic variance). Each column $k = 1, \ldots, m$ of $\mathbf{E}$ was $E_k \sim \mathcal{N}(0, \frac{1-h^2}{h^2}\text{var}((\mathbf{XB})_k))$, and $h^2 = 0.1$.

We used 3-fold cross-validation to compare `flashpcaR::scca` and `PMA::CCA`, over a 2D grid of 30×25 penalties, estimating one pair of canonical vectors. The final predictive power was computed as the average Pearson correlation $\bar{\rho}$ in the $k = 1, \ldots, 3$ test folds:

$$\bar{\rho} = \frac{1}{3}\sum_{k=1}^{3} \text{Cor}(\mathbf{X}^k_{test}u^k, \mathbf{Y}^k_{test}v^k).$$

4 Timing experiments

For timing of `flashpcaR::scca` and `PMA::CCA`, we used contiguous subsets of chromosome 1 (1000, 5000, 10,000, 20,000, and 50,000 SNPs, out of 18,193 SNPs in total) and contigous subsets of the 18,193 gene expression probes (1000, 10,000, and all 18,193 probes).

We used the R package `microbenchmark` [5] to run 30 replications of each timing experiment. For all experiments we estimated one pair of canonical vectors (u_1, v_1). For the results in the main text, we initialised ("warm started") v_1 to a standard normally-distributed vector of variates $\sim \mathcal{N}(0, 1)$. `PMA::CCA` and `flashpcaR::scca` (but not the commandline version `flashpca`) allow the user to provide their own initialisation, and we experimented with other forms, including using the column means of the gene expression data and the rank-1 singular value decomposition (SVD) $\mathbf{X}^T\mathbf{Y} \approx u_1 d_1 v_1^T$) using `flashpcaR::flashpca`. The overall trend of `flashpca` being several-fold faster than `PMA` was consistent across all three initialistion methods (Figure 1).

All experiments were run in R 3.2.2 [6] (with the original LAPACK and BLAS libraries included in R) on 64-bit Ubuntu Linux 12.04 on an Intel Xeon CPU E7-4830 v2 @ 2.20GHz. Time for the commandline `flashpca` include loading of data into RAM. We used flashpca v1.2.6 (`https://github.com/gabraham/flashpca`) and PMA v1.0.9 [7]. For `PMA::CCA`, we increased the maximum number of iterations to match that used by `flashpcaR::scca` (default=1000), in order to prevent early termination of the algorithm before adequate numerical convergence was achieved.

References

[1] International HapMap 3 Consortium. Integrating common and rare genetic variation in diverse human populations. *Nature*, 467:52–58, 2010.

[2] S. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M. A. Ferreira, D. Bender, J. Maller, P. Sklar, P. I. de Bakker, M. J. Daly, and P. C. Sham. PLINK: a tool set for whole-genome association and population-based linkage analyses. *Am. J. Hum. Genet.*, 81:559–575, 2007.

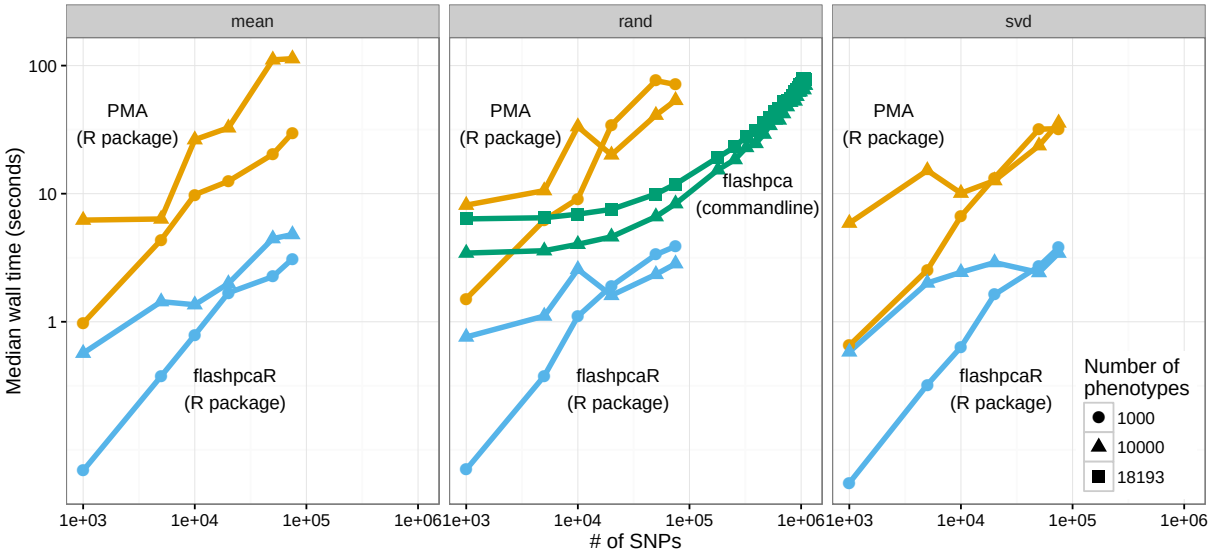

Figure 1: Timing (median of 30 runs) of SCCA implemented in (i) the `flashpcaR` (R package) and (ii) `flashpca` (stand-alone commandline tool), compared with SCCA from `PMA`, using subsets of the HapMap3 dataset with gene expression levels as phenotypes. We compared three schemes for initialising v_1: (i) "mean": column means of the gene expression data; (ii) "rand": normally-distributed variates $\mathcal{N}(0, 1)$; and (iii) "svd": 1st right singular value of $\mathbf{X}^T\mathbf{Y}$.

[3] Christopher Chang, Carson Chow, Laurent Tellier, Shashaank Vattikuti, Shaun Purcell, and James Lee. Second-generation plink: rising to the challenge of larger and richer datasets. *GigaScience*, 4(1):7, 2015. ISSN 2047-217X. doi: 10.1186/s13742-015-0047-8.

[4] B. Stranger et al. Patterns of Cis Regulatory Variation in Diverse Human Populations. *PLoS Genet*, 8(4):e1002639, 2012. doi: 10.1371/journal.pgen.1002639.

[5] Olaf Mersmann. *microbenchmark: Accurate Timing Functions*, 2015. URL `http://CRAN.R-project.org/package=microbenchmark`. R package version 1.4-2.1.

[6] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015. URL `http://www.R-project.org`. ISBN 3-900051-07-0.

[7] D. Witten et al. *PMA: Penalized Multivariate Analysis*, 2013. URL `http://CRAN.R-project.org/package=PMA`. R package version 1.0.9.