SparSNP — Workflow example

Gad Abraham gad.abraham@unimelb.edu.au

April 15, 2014

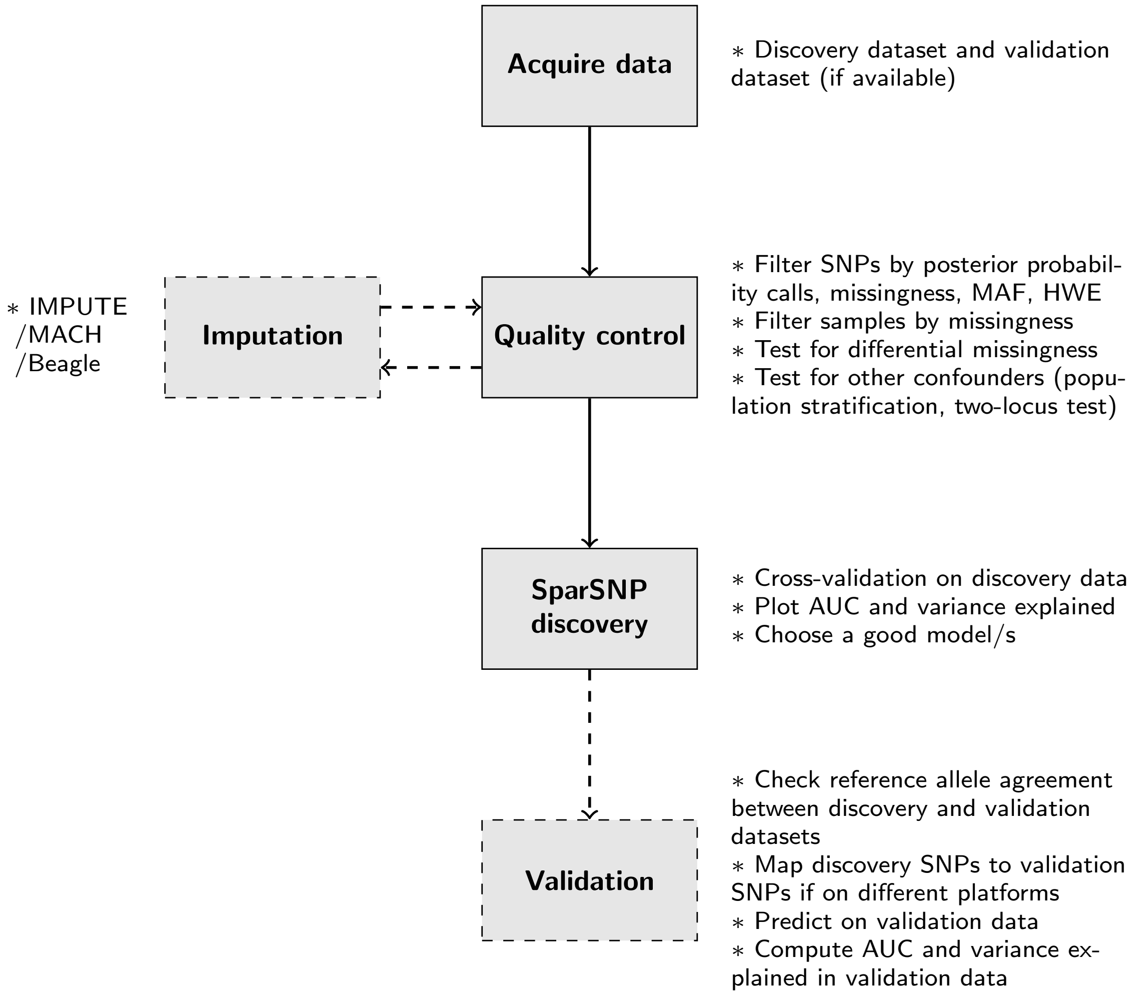
Acquire data
∗ Discovery dataset and validation dataset (if available)
∗ IMPUTE /MACH /Beagle
Imputation
Quality control
∗ Filter SNPs by posterior probability calls, missingness, MAF, HWE
∗ Filter samples by missingness
∗ Test for differential missingness
∗ Test for other confounders (population stratification, two-locus test)
SparSNP discovery
∗ Cross-validation on discovery data
∗ Plot AUC and variance explained
∗ Choose a good model/s
Validation
∗ Check reference allele agreement between discovery and validation datasets
∗ Map discovery SNPs to validation SNPs if on different platforms
∗ Predict on validation data
∗ Compute AUC and variance explained in validation data

Figure 1: Analysis workflow

Here we describe a typical analysis using SparSNP. We assume that we have two datasets in PLINK (Purcell et al., 2007) BED/BIM/FAM format, named `discovery` and `validation`, on the same SNP microarray platform. The phenotype in the FAM file can be discrete (1 for controls and 2 for cases) or continuous. In the following, we assume the user is working with a Unix-like command line shell such as Bash.

1. **Installation**

   Requirements:

   - Linux / Unix / Mac operating system, preferably 64-bit.
   - R, with packages ggplot2 ($\geq$ 0.9.3), scales, grid, abind, ROCR
   - C compiler, gcc or clang
   - make

   Download and install:

   ```
   git clone git://github.com/gabraham/SparSNP
   cd SparSNP
   make
   ```

   Prior to running SparSNP you must include the directory you unpacked it in, in the path:

   ```
   export PATH=<PATH_TO_SPARSNP>:$PATH
   ```

2. **Quick Start**

   Assuming the data is called `my-data.bed`, `my-data.bim`, `my-data.fam` (case sensitive) and it's case/control data:

   - `export PATH=<PATH_TO_SPARSNP>:$PATH`
   - `crossval.sh my-data sqrhinge 2>&1 | tee log`
   - `eval.R` to plot cross-validation AUC results in `discovery/discovery_AUC.pdf`

3. **Quality Control**
   SparSNP implements simple imputation of genotypes such that missing genotypes are randomly assigned the value {0,1,2} with probability proportional to the genotypes in non-missing samples of each SNP[1], which does not introduce substantial bias to the model when the proportion of missingness is small and the genotypes are missing at random. We recommend removing samples and SNPs with high missingness, and testing for differential missingness between cases and controls.

   - Remove markers with missingness ≥ 0.01, MAF ≤ 0.01, and Hardy-Weinberg test for equilibrium in controls of $p \leq 10^{-4}$, and samples with missingness ≥ 0.01:
     ```
     $ plink --noweb --bfile my-data --geno 0.01 --mind 0.01 \
     --maf 0.01 --hwe 0.0001 \
     --make-bed --out my-data-filtered
     ```

[1] Earlier versions assigned equal probability to each genotype.

- Test for differential missingness:
  ```
  $ plink --noweb --bfile my-data-filtered --missing
  ```
  (remove SNPs as indicated by output)
- Test for sample relatedness with a threshold of $\hat{\pi} = 0.05$:
  ```
  $ plink --noweb --bfile my-data-filtered --Z-genome --min 0.05
  ```
  (remove related samples as indicated by output)
- Check for stratification using PCA, for example, using `smartpca` in Eigensoft (Price et al., 2006).
- Two locus test for detecting batch effects (Lee et al., 2010).
  (remove SNPs as indicated by output)

4. **SparSNP on discovery data**

   - Note: your PLINK bed/bim/fam files must have a lower-case suffix (i.e., bed not BED).
   - Run cross-validation. By default, 10 × 10-fold cross-validation will be performed, allowing up to $\min\{N, p\}$ SNPs in the model, using a case/control phenotype:
     for classification (case/control)
     ```
     $ crossval.sh my-data-filtered sqrhinge
     ```
     and for continuous outputs (linear regression)
     ```
     $ crossval.sh my-data-filtered linear
     ```
     The `crossval.sh` script can be modified to perform more cross-validation folds or to tune the model parameters.
     You can run SparSNP on 10 cores at once using
     ```
     $ NUMPROCS=10 crossval.sh my-data-filtered sqrhinge
     ```
     You can run change the default output directory `discovery` to `my-discovery` using
     ```
     $ DIR=my-discovery-dir crossval.sh my-data-filtered sqrhinge
     ```
   - Plot AUC and explained phenotypic and genetic variance in the cross-validation replications:
     ```
     $ eval.R
     ```
     Optionally, supply population prevalence $K = 1\%$ and heritability $h^2_L = 50\%$:
     ```
     $ eval.R prev=0.01 h2l=0.5
     ```
     The plots `discovery_AUC.pdf`, `discovery_VarExp.pdf`, and `discovery_GenVarExp.pdf` will be produced in the directory `discovery` (the last two plots will be produced only if you supply prevalence and/or heritability). The raw AUC and explained variance data is stored in the `.Rdata` file named `discovery.RData`. If used a non-standard directory, you need to specify it:
     ```
     $ eval.R dir=my-discovery-dir prev=0.01 h2l=0.5
     ```
     Note that currently for `linear` models `eval.R` only computes R^2 and not the explained heritability etc.
   - The set of models with best predictive ability will automatically be chosen from the results, based on smoothing of the AUC[2]. The SNPs appearing in these models will be tabulated according to how many time they were included over all

[2] The λ grid must be fine enough to produce good smooths. The default `NLAMBDA1=30` is usually sufficient but may need to be increased in some cases.

cross-validation folds. To inspect the SNPs selected by models that maximise the predictive ability:

```
head discovery/topsnps.txt
```

```
RS Counts Proportion Replications
rs2050189 60 1 60
rs2187668 60 1 60
rs9357152 60 1 60
rs7774954 60 1 60
rs3129763 58 0.966666666666667 60
...
```

The SNPs are ordered by the number of times they were included in a model with non-zero weight (Counts) out of the total number of cross-validation folds (Replications, i.e. `NREPS`×`NFOLDS`), also shown as a Proportion. SNPs at the top of the list are more stably selected by the lasso and are potentially more robust markers than SNPs at the bottom of the list.

5. **Optional: Apply models to validation data**

   SparSNP models trained on the discovery dataset can be tested on an independent dataset, if one is available. Datasets can differ in terms of which SNPs are assayed, their ordering in the PLINK files, and their minor allele frequencies[3]. SparSNP is completely oblivious to these differences and cannot detect them, hence we use PLINK scoring. The one caveat is that PLINK scoring does not have an intercept term, which we must add on. The process is:

   - Extract a consensus model from the training data using `getmodels.R`, requesting a certain number of SNPs in the model:

     ```
     $ getmodels.R nzreq=256
     ```

     And if you used a non-standard discovery directory:

     ```
     $ getmodels.R nzreq=256 dir=my-discovery-dir
     ```

   - Call `predict.sh` on the validation dataset `my-validation-data` which will call PLINK scoring internally:

     ```
     $ predict.sh my-validation-data
     ```

     If you used a non-standard discovery directory:

     ```
     $ DIR=my-discovery-dir predict.sh my-validation-data
     ```

     Optionally, you can declare OUTDIR to specify a directory other than `predict` (say `my-validation-dir`), and use multi-processing if you have multiple cores:

     ```
     $ NUMPROCS=10 OUTDIR=my-validation-dir \
     predict.sh my-validation-data
     ```

   - Call `evalprofile.R` to compute ROC/AUC on the validation data, optionally with the prevalence to allow plotting PPV/NPV:

     ```
     $ evalprofile.R model=sqrhinge prev=0.01
     ```

[3] Strand flips will cause some predictor SNPs to be ignored, see `http://pngu.mgh.harvard.edu/~purcell/plink/dataman.shtml#flip` to fix this prior to running the prediction stage.

If you used an non-default directory for discovery (say `my-discovery-dir`) and/or non-default for validation (say `my-validation-dir`), then specify it:

```
$ evalprofile.R model=sqrhinge prev=0.01 \
indir=my-discovery-dir outdir=my-validation-dir
```

The results will be saved in the directory you specified, in the file `results.RData`.

You can now load the results (assuming you used the default output directory `predict`), stored as a list, one element per model in the path, plot the results or compute new one curves using the ROCR `pred` object:

```
> load("predict/results.RData")
> length(res)
[1] 30
> names(res[[10]])
[1] "pred"    "perf"    "sens"    "spec"    "cutoffs" "ppv"     "npv"
> library(ROCR)
> plot(res[[10]]$perf) # ROC curve for 10th model
> plot(performance(res[[10]]$pred, "prec", "rec")) # precision-recall
```

- `evalprofile.R` can compute R^2 for linear regression models, in which case prevalence shouldn't be supplied:

  ```
  $ evalprofile.R model=linear outdir=my-validation-dir
  ```

  and if you used a non-standard discovery directory:

  ```
  $ evalprofile.R model=linear \
  indir=my-discovery-dir outdir=my-validation-dir
  ```

  Note that an alternative phenotype file is currently not supported for `evalprofile.R`, that is, the validation FAM file must contain a phenotype (not `-9`).

6. **Other post processing**

   - The model weights are stored in each cross-validation directory
     `discovery/crossvalXX/beta.csv.XX.XX`
     using a sparse text format <index,weight>, where index is the zero-based index of the SNP in the data (0 is the intercept), and weight is the model weight (a real number). The weights can be read into R (using `read.table` with `sep=","`, `header=FALSE`) or any other tool for visualisation or for validating the model on other datasets.

7. **Optional parameters**

   Some options that can be set by setting the environment variables before calling `crossval.sh`:

   - `NUMPROCS`: number of processes to use in parallel, should be between 1 and the number of cross-validation replications `NREPS` (not folds), (default=1)
   - `LAMBDA2`: ℓ_2 penalty for elastic-net (default=0)
   - `NFOLDS`: number of cross-validation folds (default=10)
   - `NREPS`: number of cross-validation replications (default=10)
   - `NZMAX`: maximum number of SNPs to allow in model (default=$\min\{N, p\}$)

- `NLAMBDA1`: number of λ penalties on the grid (default=30)
- `L1MIN`: multiplier on smallest λ used; it should be a some positive fraction such as 0.01. Setting it lower will increase the number of SNPs in the models, but will also increase computational time (default=0.001)
- `PHENO`: for an alternative phenotype file (equivalent to PLINK `--pheno`). The file must be in the format:
  FamilyID IndividualID PhenotypeA PhenotypeB ...
  without a header line. When K phenotypes are available SparSNP will fit a multivariate model equivalent to K separate models.
- `LAMBDA1PATH`: a text file containing the path of λ_1 penalties (decreasing order, one per row), to be used across all cross-validation folds, instead of SparSNP recomputing the path for each fold.

Example of using the optional parameters:

```
NUMPROCS=10 LAMBDA2=0.1 NFOLDS=3 NREPS=10 NZMAX=500 \
LAMBDA1PATH=lambda.txt crossval.sh my-data sqrhinge 2>&1 | tee log
```

8. Acknowledgments

- Thanks to Marco Colombo for patches to enable a consistent λ_1 path.

References

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.

A. L. Price, N. J. Patterson, R. M. Plenge, M. E. Weinblatt, et al. Principal components analysis corrects for stratification in genome-wide association studies. *Nat. Genet.*, 38:904–909, 2006.

S. Lee, D. Nyholt, S. Macgregor, A. Henders, K. Zondervan, G. Montgomery, and P. Visscher. A simple and fast two-locus quality control test to detect false positives due to batch effects in genome-wide association studies. *Genet. Epidemiol*, 34: 854–862, 2010.