1 Introduction

We provide here documentation supporting PAINTOR (V2.1). Full descriptions of the methods can be found in:

Kichaev G, Yang WY, Lindstrom S, Hormozdiari F, Eskin E, Price AL, Kraft P, Pasaniuc B. Integrating functional data to prioritize causal variants in statistical fine-mapping studies. ***PLoS genetics 10.10 (2014): e1004722.***

Kichaev G and Pasaniuc B. Leveraging functional annotation data in trans-ethnic fine-mapping studies. ***American Journal of Human Genetics (2015):***

PAINTOR is a command line tool written in C++. It is capable of conducting fine-mapping on either single populations or multiple populations simultaneously and integrate functional annotations. For quick start simply type `PAINTOR` for list of options.

1.1 Updates

- V2.1.0 (7/16/15) Update to handling of NCPs. See supplementary section for details on impact on performance.
- V2.0.0 (06/08/15) Major update. Added functionality for multi-ethnic fine-mapping.
- V1.1.1 (11/19/14) Minor I/O bug fix
- V1.1.0 (11/10/14) Cholesky decomposition of the LD matrix to improve stability and performance.
- V1.0 (07/23/14) First stable release of the PAINTOR software.

2 Installation

The early release of PAINTOR is optimized to run on a UNIX-like system. To install the software, unpack the zip file in the directory you want PAINTOR installed. CD into the PAINTOR folder and run the installation script:

```
> tar -xvf PAINTOR_FineMapping*.tar.gz
> cd PAINTOR
> bash install.sh
```

This will unpack and compile all the necessary dependencies and create an executable `PAINTOR`. The two main libraries that PAINTOR uses are Eigen V3.2 and NLopt V2.4.2.

```
http://eigen.tuxfamily.org/index.php?title=Main_Page
http://ab-initio.mit.edu/wiki/index.php/NLopt
```

3 Input

For each locus of interest, there are three input files needed to run the software [File Dimensions]:

1. A *Locus file* that contains the Z-scores from all the population of interest [N+1 x F]
2. *LD matrix file(s)* (need multiple matrices if doing multi-ethnic fine-mapping) [N x N]

3. An *Annotation matrix file* with annotation indicators [N+1 x A]

N = # of SNPs at a locus, F = # of fields in the Locus file, A = total # number of annotations. Sample files are included in the `SampleData/` directory.

3.1 File Formats

All file formats are assumed to be single space delimited. If your file is tab-delimited you can use the following command to modify:
`> sed -i 's/\t/ /g' <filename>`

3.1.1 Locus file

The locus file should at the very minimum contain the Z-scores for all the populations, though metadata on each SNP such as chromosome, position, and rsid are recommended. The top line of the locus file must contain a header with the names of the fields. The Z-score of a SNP is the Wald statistic ($\frac{\hat{\beta}}{SE(\hat{\beta})}$) obtained from standard regression of the phenotype onto the SNP. If a SNP is monomorphic or missing in one of the populations, the corresponding Z-score must be either a `0` or `NA` for the software to run as intended.

Example: A Locus file corresponding to 4 SNPs across two populations with SNP 1 monomorphic in population 2

```
CHR POS RSID ZSCORE.P1 ZSCORE.P2
chr1 10 rs1 1.5 NA
chr1 15 rs2 -3.2 -1.5
chr1 20 rs3 4.5 5.5
chr1 25 rs4 0.8 -0.5
```

Note: An arbitrary number of fields can be included in the Locus files which we leave to the discretion of the user. The Z-score headers are specified with the `-Zhead` flag.

3.1.2 LD matrix file

The LD file(s) contains a symmetric matrix of Pearson correlation coefficients where entry i,j will correspond to the correlation between SNPs i and j (r_{ij}). White space must separate individual columns of the matrix. This file has no header.

If doing fine-mapping over multiple populations, each population must have its own LD matrix file of the same size. We note there will be times where a SNP may be monomorphic (or missing) in one of the populations. These SNPs will be encoded as a 1 in entry (i, i) and 0 for all other entries (i, j) & (j, i) for $j \neq i$.

Example: 4 SNP LD matrices for two populations, where SNP 1 is monomorphic in population 2

```
Population 1

1.0 0.5 0.5 0.2
0.5 1.0 0.3 0.1
0.5 0.3 1.0 0.9
0.2 0.1 0.9 1.0
```

```
Population 2

1.0 0.0 0.0 0.0
0.0 1.0 0.2 0.1
0.0 0.2 1.0 0.3
0.0 0.1 0.3 1.0
```

Note: **VERY IMPORTANT!** Particular care must be taken when computing LD from a reference panel such the 1000 genomes. It is imperative that all the reference and alternate alleles for SNPs from which the Z-scores were computed match the reference and alternate alleles of the reference panel. The output of PAINTOR will not be correct if there are mismatches of this type in the data.

3.1.3 Annotation Matrix File

There should be an annotation file for each locus that contains a matrix of annotations that are typically binary. The rows of the matrix correspond to SNPs at that locus and columns represent unique annotations. For example, if the first column of the matrix represented "coding" region, and entry [1,1] of the matrix was equal to 1, this would signify that SNP 1 falls within a coding region. The first line in the file must be header identifying the annotations. Each annotation must have a unique identifier.

Example: A locus with 4 SNPs and three potential annotations {Coding, DHS, Enhancer}

```
Coding DHS1 DHS2
1 0 1
0 1 0
1 1 1
1 0 0
```

3.2 File Naming conventions for input

PAINTOR is designed to run on multiple loci and/or populations simultaneously. The program takes as input a single file with a list of file names that contain the Z-scores (and other metadata) for all the loci the user wants to consider jointly. By convention, these names serve as the prefix for the entire locus. The files containing the corresponding LD matrix/matrices and annotation file will have suffixes appended to them that are specified in the command (see flags described below). For example, if one wanted to run PAINTOR on three loci, the input file would look like this:

```
> cat input.file
Locus1
Locus2
Locus3
```

Note: The input file name is specified with `-input` flag. The directory that contained all the files for both populations would have the following:

```
> ls RunDirectory/
Locus1
Locus1.LD1
Locus1.LD2
Locus1.annotations
```

```
Locus2
Locus2.LD1
Locus2.LD2
Locus2.annotations
Locus3
Locus3.LD1
Locus3.LD2
Locus3.annotations
```

Note: The input directory can be specified using the `-in` flag

4 Output

4.1 Posterior Probabilities

The posterior probabilities for snps to be causal will be output to a .results file for each locus and will contain the input data with an additional Posterior_Prob column appended.

Example:A results file for a locus of 4 SNPs

```
CHR POS RSID ZSCORE.P1 ZSCORE.P2 Posterior_Prob
chr1 10 rs1 1.5 NA 0.013
chr1 15 rs2 -3.2 -1.5 0.130
chr1 20 rs3 4.5 5.5 0.980
chr1 25 rs4 0.8 -0.5 0.002
```

Note: The default filename will be the locus name with ".results" appended to it. User can change the name using the `-OUTname` flag. To designate an output directory use the `-out` flag.

4.2 Gamma Estimates

A file that has the effect size estimates for each of the annotation(s) used. PAINTOR automatically estimates the baseline annotation (A0) and will always output this values as the first line in the file.

Example: Estimates for 2 annotations (+1 baseline) {Baseline, Coding, DHS1}

```
Baseline Coding DHS1
5.2 -1.3 2.0
```

These effect sizes can be converted to probabilities using the expit transformation. For the preceding output the corresponding prior probabilities can be calculated as follows:

The baseline prior probability for any SNP in the fine-mapping dataset to be causal is obtained as:

$$\begin{aligned} \frac{1}{1+exp(\gamma_0)} &= \frac{1}{1+exp(5.2)} \\ &= 0.0055 \end{aligned}$$

The prior probability for a SNP in Coding coding:

$$\frac{1}{1+exp(\gamma_0+\gamma_1)} = \frac{1}{1+exp(5.2+(-1.3))}$$
$$= 0.01984$$

The relative probability for a SNP to be causal given that it is in Coding is simply computed as

$$\frac{0.01984}{0.0055} = 3.6$$

Note: The default filename is Enrichment.Parameters. User can change name using the `-Gname` flag

4.3 Final log-likelihood

A file that has the final log likelihood of the PAINTOR model. This can be used in subsequent steps to conduct a likelihood ratio test (LRT) for significance of the annotation effect sizes. To test the marginal significance of a single annotation (A1) one would first fit a PAINTOR model with just the baseline annotation A0 (M0) then fit a joint model with both annotations A0, A1 (M1). The resultant log likelihoods for each model can be used to compute an LRT statistic which will be distributed asymptotically χ^2 with degrees of freedom = 1 (under the null).

Example: Testing significance of annotation A1

Model 1 log likeilhood with only baseline annotation (A0)
-10039
Model 2 log likeilhood with both annotations(A0, A1)
-10036

$$\begin{aligned} LRT &= -2[ln(likelihood(M0)) - ln(likelihood(M1))] \\ &= -2[-10039-(-10036)] \\ &= 6 \end{aligned}$$

$\sim \chi^2$ (df=1) p-value= 0.0143

5 Running software

5.1 PAINTOR

Usage: `PAINTOR -input.files [input filename] -in [input directory] -out [output directory] -Zhead [Zscore header(s)] -LDname [LD suffix(es)] -annotations [annotation1,annotation2...] <other options>`

`OPTIONS: -flag Description [default setting]`

`-input` `(required) Filename of the input file containing the list of the fine-mapping loci [default: input.files ]`

`-Zhead` `(required) The name(s) of the Zscore column in the header of the locus file`

(comma separated) [default: N/A]

-LDname (required) Suffix(es) for LD files. Must match the order of Z-scores in which the -Zhead flag is specified (comma separated) [Default:N/A]

-c The number of causal variants to consider per locus [default: 2]

-annotations The names of the annotations to include in model (comma separated) [default: N/A]

-in Input directory with all run files [default: ./]

-out Output directory where output will be written [default: ./]

-Gname Output Filename for enrichment estimates [default: Enrichment.Estimate]

--Lname Output Filename for log likelihood [Default: Log.Likelihood]

-RESname Suffix for output files of results [Default: results]

-ANname Suffix for annotation files [Default: annotations]

-MI Maximum iterations for algorithm to run [Default: 10]

-post1CV Fast conversion of Z-scores to posterior probabilities assuming a single casual variant and no annotations [Default: False]

-GAMinital Initialize the enrichment parameters to a pre-specified value (comma separated) [Default: 0,...,0]

-NCP how to set Non-Centrality Parameter {old, default} [Default: default]

Example: Running PAINTOR with two populations, considering up to three causal variants per locus, and integrating Coding and DHS annotations .
s

```
> ./PAINTOR -input input.files -Zhead ZSCORE.P1,ZSCORE.P2 -LDname LD1,LD2 -in RunDirectory
-out OutDirectory/ -c 3 -annotations Coding,DHS
```

5.2 Suggested Pipeline

In order to determine which annotations are relevant to the phenotype being considered, we recommend running PAINTOR on each annotation independently.

Example: Pipeline for a pool of 100 annotations.

```
> ./PAINTOR -input input.files -Zhead ZSCORE.P1,ZSCORE.P2 -LDname LD1,LD2 -in RunDirectory
-out OutDirectory/ -c 2 -Gname Enrich.Base -Lname Likeli.Base
> ./PAINTOR -input input.files -Zhead ZSCORE.P1,ZSCORE.P2 -LDname LD1,LD2 -in RunDirectory
-out OutDirectory/ -c 2 -annotations A1 -Gname Enrich.A1 -Lname Likeli.A1
```

```
> ./PAINTOR -input input.files -Zhead ZSCORE.P1,ZSCORE.P2 -LDname LD1,LD2 -in RunDirectory
-out OutDirectory/ -c 2 -annotations A2 -Gname Enrich.A2 -Lname Likeli.A2
> ./PAINTOR -input input.files -Zhead ZSCORE.P1,ZSCORE.P2 -LDname LD1,LD2 -in RunDirectory
-out OutDirectory/ -c 2 -annotations A3 -Gname Enrich.A3 -Lname Likeli.A3
.
.
.
> ./PAINTOR -input input.files -Zhead ZSCORE.P1,ZSCORE.P2 -LDname LD1,LD2 -in RunDirectory
-out OutDirectory/ -c 2 -annotations A100 -Gname Enrich.100 -Lname Likeli.100
```

After obtaining the output for all of the annotations marginally, prioritize annotations based on the improvement in the model fit. Take the top annotations (usually no more than 4 or 5) to enter the final model that are roughly uncorrelated with one another. We recommend correlation matrices for this process. Then use those annotations in a final model to compute trait-specific posterior probabilities for causality:

```
> ./PAINTOR -input input.files -Zhead ZSCORE.P1,ZSCORE.P2 -LDname LD1,LD2 -in RunDirectory
-out OutDirectory/ -c 2 -annotations A5,A20,A93 -Gname Enrich.Final -Lname Likeli.Final
```

6 Supplemenatry

In the the updated version 2.1 of the PAINTOR software, we introduce a new way to handle the non-centrality parameters. For legacy purpose, we still allow the user to specify using the -NCP flag if they want to use the NCP handling of versions v1.0 and v2.0. Rather than setting the NCP as a function of the observed Z-scores. We now first solve a system of linear equations to account for LD between causal NCPs. The boost in performance can be see in figure 1. We recommend using this as default.

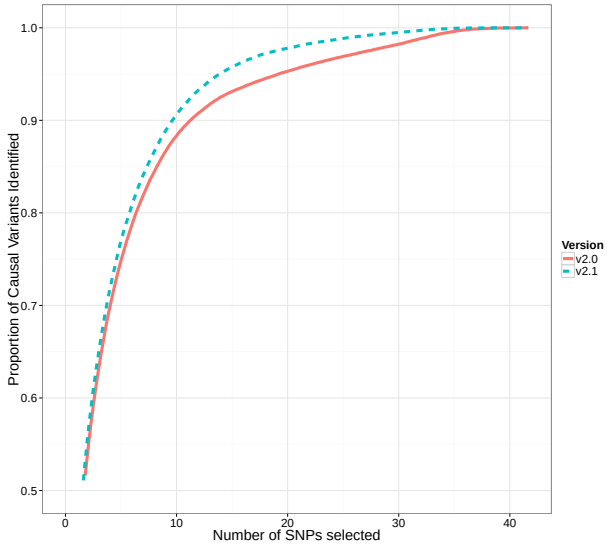

Figure 1: Simulations were done over one hundred 10KB loci with N=10,000 and h_g^2=0.25.