JAM: Joint Analysis of Marginal statistics

Paul Newcombe

19 November 2015

Contents

Introduction

Genetic association studies typically only publish results from one at a time tests of genetic variants, or

Genetic association studies typically only publish results from one-at-a-time tests of genetic variants, or "SNP"s. Confounding due to linkage disequilibrium, i.e. genetic correlations, can cloud the location of a truly causal SNP, since correlated variants appear associated too. JAM (Joint Analysis of Marginal statistics) is an algorithm which enables inference of joint multi-SNP models from one-at-a-time summary statistics. Correlations are adjusted for according to estimates from an external dataset. Models and SNPs which best explain the complete joint pattern of marginal effects are highlighted via an integrated Bayesian penalized regression framework.

Pre-requisites

The underlying JAM algorithm is coded in Java. Therefore you must have a Java JDK installed on your system (note that a Java runtime environment alone is not sufficient). The Java JDK may be downloaded from https://www.java.com/download/. R2BGLiMS - a general Bayesian model selection package which contains the JAM algorithm, may be installed from github as follows (requires the "devtools" library):

```
install.packages("devtools")
library(devtools)
install_github("pjnewcombe/R2BGLiMS")
```

Quick reference example

The main arguments are:

- `marginal.betas`: The pubished SNP effecs from one at-a-time regressions
- `X.ref`: A reference genotype matrix
- `model.space.prior`: Prior over the model space to use for Bayesian model selection
- `n`: The number of individuals analysed to infer marginal.betas

```
library(R2BGLiMS) # Load package
data(JAM_Example) # Load example data
jam.results <- JAM(
  marginal.betas=marginal.betas[snps.region1],
  X.ref=X.ref.region1,
  model.space.prior = list("Rate"=0.1, "Variables"=snps.region1),
  n=n)
PrettyResultsTable(jam.results)
```

More detail on JAM's arguments, and obtaining inference follow with a worked example below.

Worked example - 1 region

We start by loading the example simulated dataset included in the R2BGLiMS package.

```
library(R2BGLiMS)
data(JAM_Example)
```

One-a-a-time SNP effect estimates from an analysis of `n=15,000` people for 20 SNPs, divided into two regions (`snps.region1` and `snps.region2`) are stored in `marginal.betas`. The true effects used in the simulations are stored in `true.betas`. You can see the level of confounding due to LD by comparing `marginal.betas` vs `true.betas`. For region 1 a single effect at SNP 5 was simulated, and for region 2 two effects were simulated.

JAM's main arguments

JAM requires the following key arguments:

- `marginal.betas`: The pubished SNP effecs from one at-a-time regressions
- `X.ref`: A reference genotype matrix
- `model.space.prior`: Prior over the model space to use for Bayesian model selection
- `n`: The number of individuals analysed to infer marginal.betas

`marginal.betas`

`marginal.betas` is simply a named vector of reported SNP effects.

`X.ref`

`X.ref`, the reference individual level genotype matrix, must be numerically coded as risk allele counts 0/1/2 whereby the risk allele corresponds to `marginal.betas`. See, for example, the first 3 rows of the example reference matrix for region 1:

```
X.ref.region1[1:3,snps.region1]
```

```
##      SNP1 SNP2 SNP3 SNP4 SNP5 SNP6 SNP7 SNP8 SNP9 SNP10
## [1,]    1    1    2    0    1    1    1    1    1     1
## [2,]    1    1    2    0    0    0    0    1    1     1
## [3,]    1    2    1    2    1    1    1    1    1     0
```

`model.space.priors`

To facilitate the Bayesian model selection, a probabilistic prior over the set of possible models, i.e. causal SNP combinations, must be specified using `model.space.prior`. JAM allows two opions:

1) A fixed prior proportion of "causal" SNPs, which results in a Poisson prior probability distribution over different model sizes. This can be provided to JAM as a list, e.g:

```
model.space.prior = list("Rate"=0.1, "Variables"=snps.region1)
```

2) Treat the prior proporion of "causal" SNPs, as unknown with a Beta hyper-prior. A Beta(1,1) prior corresponds to a flat uniform distribution and may be a sensible choice. To impose sparsity, an informative prior for smaller values of θ can be used by choosing higher values of the second hyper-parameter, eg Beta(1,10) or Beta(1,100). This choice is specified by using a list of the format:

```
model.space.prior = list("a"=1, "b"=1, "Variables"=snps.region1)
```

Running JAM

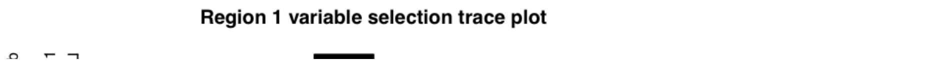
Stochastic model search

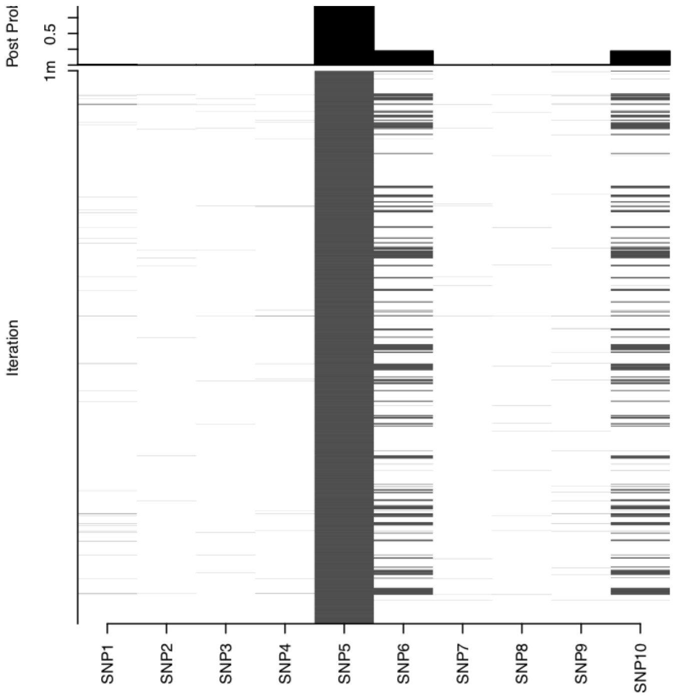
A re-analysis of the example region 1 marginal effects proceeds as follows. Note that an uniformative beta-binomial prior is used for the model space:

```
region1.results <- JAM(
  marginal.betas=marginal.betas[snps.region1],
  X.ref=X.ref.region1,
  model.space.prior = list("a"=1, "b"=1, "Variables"=snps.region1),
  n.mil=1,
  n=n)
```

Posterior probabilities for different SNPs and SNP combinations are obtained, by default, from `n.mil` million iterations of a stochastic model search. Convergence/mixing can be checked using the `AutocorrelationPlot` function. The plot indicates inclusion in the model of the different covariates at different iterations; iterations are ordered vertically, and covariates along the X-axis. If the reversible jump MCMC is mixing well (as below), covariates should be regularly jumping in and out, and not "sticking". The top part of the plot indicates the marginal posterior probability of association for each covariate. Note that JAM has correctly identified the 5th SNP as the single simulated causal effect.

```
AutocorrelationPlot(region1.results, plot.title="Region 1 variable selection trace plot")
```





A nice summary table of the evidence for each SNP can be obtained using the `PrettyResultsTable` function:

```
PrettyResultsTable(region1.results)
```

```
##       Posterior Probability Bayes Factor
## SNP1                   0.01         <0.1
## SNP2                  <0.01         <0.1
## SNP3                  <0.01         <0.1
## SNP4                  <0.01         <0.1
## SNP5                   1.00          Inf
## SNP6                   0.22          0.3
## SNP7                  <0.01         <0.1
## SNP8                  <0.01         <0.1
## SNP9                   0.01         <0.1
## SNP10                  0.22          0.3
```

and a summary of the top supported SNP combinations can be obtained using the `TopModels` command. As expected the top model is one which only includes SNP 5:

```
TopModels(region1.results)
```

```
##       SNP1 SNP2 SNP3 SNP4 SNP5 SNP6 SNP7 SNP8 SNP9 SNP10 Post Prob
##  [1,]    0    0    0    0    1    0    0    0    0     0    0.7692
##  [2,]    0    0    0    0    1    1    0    0    0     1    0.2074
##  [3,]    1    0    0    0    1    1    0    0    1     1    0.0024
##  [4,]    1    0    0    0    1    1    0    0    0     0    0.0020
##  [5,]    0    0    0    0    1    1    0    0    1     0    0.0018
##  [6,]    0    0    0    0    1    0    0    0    0     1    0.0016
##  [7,]    0    0    0    0    1    0    0    1    0     1    0.0014
##  [8,]    0    0    0    1    1    1    0    0    0     0    0.0014
##  [9,]    0    0    0    0    1    0    0    0    1     1    0.0012
## [10,]    1    0    0    1    1    1    0    0    0     1    0.0010
## [11,]    0    1    0    0    1    1    0    0    0     0    0.0008
## [12,]    1    0    0    0    1    1    0    0    0     1    0.0008
## [13,]    1    0    1    0    1    1    0    0    0     1    0.0008
## [14,]    0    0    0    0    1    0    1    0    0     1    0.0006
## [15,]    0    0    0    0    1    1    0    0    0     0    0.0006
## [16,]    0    0    0    0    1    1    0    1    0     0    0.0006
## [17,]    0    0    0    0    1    1    1    0    0     0    0.0006
## [18,]    0    1    0    0    1    0    0    0    0     1    0.0006
## [19,]    1    0    0    0    1    0    0    0    0     1    0.0006
## [20,]    0    0    0    1    1    1    1    0    0     1    0.0004
```

Posterior inference by exhaustive enumeration of models

Optionally, posterior probabilities can be inferred by exhaustively assessing every model up to a chosen dimenion specified by `enumerate.up.to.dim`. This is an approximate method (since not all possible models are evaluated). However, inference no longer relies on convergence of a stochastic search. This can provide a good check of the stochastic search inference.

```
region1.results.enum <- JAM(
  marginal.betas=marginal.betas[snps.region1],
  X.ref=X.ref.region1,
```

```
  model.space.prior = list("a"=1, "b"=1, "Variables"=snps.region1),
  enumerate.up.to.dim=3,
  n=n)
```

Encouragingly, inference is equivalent…

```
PrettyResultsTable(region1.results.enum)
```

```
##       Posterior Probability Bayes Factor
## SNP1                  <0.01         <0.1
## SNP2                  <0.01         <0.1
## SNP3                  <0.01         <0.1
## SNP4                  <0.01         <0.1
## SNP5                   1.00     109263.0
## SNP6                   0.24          0.3
## SNP7                  <0.01         <0.1
## SNP8                  <0.01         <0.1
## SNP9                  <0.01         <0.1
## SNP10                  0.24          0.3
```

Worked example - multiple regions

JAM is designed for simulataneous analysis of multiple regions. The example dataset contains marginal SNP effects, and a reference genotype matrix for a second region with an effect simulated at SNP16.

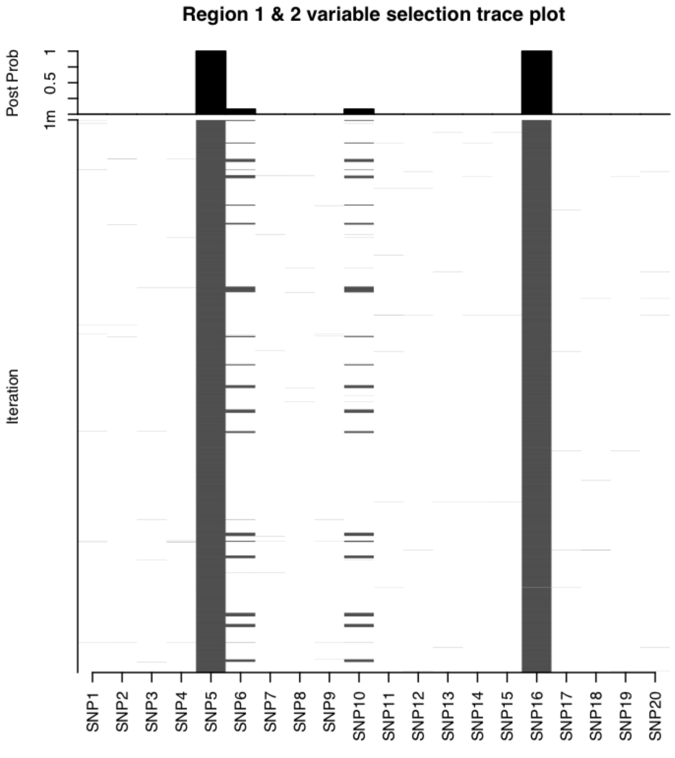
Stochastic search

Running JAM with both regions is straight forward. We start by analysing the data using a stochastic model search. Notice that X.ref is now specified as a list - each element of which contains the reference genotype matrix for a different region:

```
two.regions.results <- JAM(
  marginal.betas=marginal.betas,
  X.ref=list(X.ref.region1, X.ref.region2),
  model.space.prior = list("a"=1, "b"=9, "Variables"=c(snps.region1,snps.region2) ),
  n.mil=1,
  n=n)
```

The autocorrelation plot suggests good mixing:

```
AutocorrelationPlot(
  two.regions.results,
  plot.title="Region 1 & 2 variable selection trace plot")
```



and JAM has correctly placed compelling evidence at both causal SNPs:

```
PrettyResultsTable(two.regions.results)
```

```
##       Posterior Probability Bayes Factor
## SNP1                  <0.01         <0.1
## SNP2                  <0.01         <0.1
## SNP3                  <0.01         <0.1
## SNP4                  <0.01         <0.1
## SNP5                   1.00          Inf
## SNP6                   0.08          0.8
## SNP7                  <0.01         <0.1
## SNP8                  <0.01         <0.1
## SNP9                  <0.01         <0.1
## SNP10                  0.08          0.8
## SNP10                    0.08          0.8
## SNP11                   <0.01         <0.1
## SNP12                   <0.01         <0.1
## SNP13                   <0.01         <0.1
## SNP14                   <0.01         <0.1
## SNP15                   <0.01         <0.1
## SNP16                    1.00      22491.0
## SNP17                   <0.01         <0.1
```

```
## SNP18                   <0.01         <0.1
## SNP19                   <0.01         <0.1
## SNP20                   <0.01         <0.1
```

Exhaustive model enumaration

As a sensitivity analysis we re-analyse the two regions using model enumeration. For multiple regions, under the block independence assumption, the enumeration is carried out within each region independently and then posterior probabilities are combined:

```
two.regions.results.enum <- JAM(
  marginal.betas=marginal.betas,
  X.ref=list(X.ref.region1, X.ref.region2),
  model.space.prior = list("a"=1, "b"=9, "Variables"=c(snps.region1,snps.region2) ),
  enumerate.up.to.dim=3,
  n=n)
```

Results are very similar to above:

```
PrettyResultsTable(two.regions.results.enum)
```

```
##        Posterior Probability Bayes Factor
## SNP1                   <0.01         <0.1
## SNP2                   <0.01         <0.1
## SNP3                   <0.01         <0.1
## SNP4                   <0.01         <0.1
## SNP5                    1.00     765118.9
## SNP6                    0.08          0.8
## SNP7                   <0.01         <0.1
## SNP8                   <0.01         <0.1
## SNP9                   <0.01         <0.1
## SNP10                   0.08          0.8
## SNP11                  <0.01         <0.1
## SNP12                  <0.01         <0.1
## SNP13                  <0.01         <0.1
## SNP14                  <0.01         <0.1
## SNP15                  <0.01         <0.1
## SNP16                   1.00      64566.9
## SNP17                  <0.01         <0.1
## SNP18                  <0.01         <0.1
## SNP19                  <0.01         <0.1
## SNP20                  <0.01         <0.1
```

Application to binary data (log-odds ratios)

JAM is formulated as a linear model (with Gaussian residuals), and this formulation is responsible for the efficiency due to the conjugate results available for this class of models. In order to apply JAM to binary data (i.e. marginal Odds Ratios from logistic regressions) it is necessary to invoke a transformation between the logistic and linear scales. Fortunately, an approximate transformation has been derived by Pirinen et al (Pirinen, Donnelly, and Spencer 2013).

Simpler linear transformation

Pirinen et al derived a linear approximation of β, the effect estimates we would obtain analysing binary data with a linear regression model, in terms of γ, the reported summary log-odds ratios:

$$\hat{\beta} = \hat{\gamma}\phi(1-\phi)$$

where ϕ is the proportion of cases in the sample.

Prior for the residual variance

After using the above formula to transform the log-odds ratios to the linear scale used by JAM, we must also consider an appropiate prior for the residual variance parameter. Under JAM, the prior distribution for σ^2 is inverse-gamma:

$$\sigma^2 \sim Inverse - Gamma(a, b)$$

With mean and variance:

$$E(\sigma^2) = \frac{b}{a-1}$$

$$Var(\sigma^2) = \frac{b^2}{(a-1)^2(a-2)}$$

Note that $a > 2$ to insure a non-negative variance. The residual variance for a binary outcome under the null linear model would be equvialent to the bernoulli variance, which is given by:

$$\hat{\sigma^2} = \phi(1-\phi)$$

Since genetic effects are generally small, when applying JAM with binary data we choose an informative prior centred on this bernoulli variance which would be the correct value if the binary data were analysed with JAM under a null model. To insure an informative prior we start with $a = n$. In order to achieve the desired expectation b is then given by:

$$b = (n-1)\phi(1-\phi)$$

Example

The example dataset also includes marginal log-Odds Ratios from analyses of binary data simulated from exactly the same genotype matrix as above for $n = 15,000$, with causal effects placed at the same two SNPs. To analyse these data with JAM and flag them as log-Odds ratios, we must also pass the n.cases argument (included in the example data). This tells JAM to invoke the approximate logistic -> linear transform above, and the corresponding prior for the residual variance.

```r
binary.results <- JAM(
  marginal.betas=marginal.logors,
  X.ref=list(X.ref.region1, X.ref.region2),
  model.space.prior = list("a"=1, "b"=9, "Variables"=c(snps.region1,snps.region2) ),
  n.mil=1,
  n=n,
  n.cases=n.cases)
```

JAM has correctly placed compelling evidence at both causal SNPs:

```
PrettyResultsTable(binary.results)
```

```
##       Posterior Probability Bayes Factor
## SNP1                 <0.01         <0.1
## SNP2                 <0.01         <0.1
## SNP3                 <0.01         <0.1
## SNP4                 <0.01         <0.1
## SNP5                  1.00          Inf
## SNP6                 <0.01         <0.1
## SNP7                 <0.01         <0.1
## SNP8                 <0.01         <0.1
## SNP9                 <0.01         <0.1
## SNP10                <0.01         <0.1
## SNP11                <0.01         <0.1
## SNP12                <0.01         <0.1
## SNP13                 0.01         <0.1
## SNP14                <0.01         <0.1
## SNP15                <0.01         <0.1
## SNP16                 0.99       1314.5
## SNP17                <0.01         <0.1
## SNP18                <0.01         <0.1
## SNP19                <0.01         <0.1
## SNP20                <0.01         <0.1
```

Additions to JAM:

Simply add a n.case and n.control options? Which will then invoke the transformation and set the inverse-gamma prior?

References

Pirinen, Matti, Peter Donnelly, and Chris Spencer. 2013. "Efficient Computation with a Linear Mixed Model on Large-Scale Data Sets with Applications to Genetic Studies." *The Annals of Applied Statistics* 7 (1): 369–90.