A network-driven approach for genome-wide association mapping

Bioinformatics, 32, 2016, i164–i173
ISMB 2016

linwei@20160722

Outline

- Introduction
- Method
- Results
- Summary

Introduction

- Single Nucleotide Polymorphism (SNP)

  A variation in a single base (A, T, C or G) within a sequence of DNA. (occurs in >1% of the population)

- Genome-wide association mapping

  Searches the genome for SNPs that occur more frequently in people with a particular disease than in people without the disease.

- Genotypes+Gene expression data

  ->Phenotypes

Introduction

- Sparse regression method to evaluate the effect of each SNP in the context of all other SNPs

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

Phenotype data

Genotype data

J SNPs

$$\boldsymbol{y} = \begin{pmatrix} y^1 \\ \vdots \\ y^N \end{pmatrix} \quad N \text{ individuals}$$

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_1^1 & \cdots & x_J^1 \\ \vdots & \vdots & & \vdots \\ 1 & x_1^N & \cdots & x_J^N \end{pmatrix} \quad N \text{ individuals}$$

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_J \end{pmatrix}$$

Augmented input feature corresponding to β_0

– Sparsity constraint: Only few SNPs are influencing the given phenotype and the rest of the SNPs have effect size 0

Introduction

- Linear Regression

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon,$$

$$\hat{\boldsymbol{\beta}} = \operatorname{argmin}_{\beta} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2$$

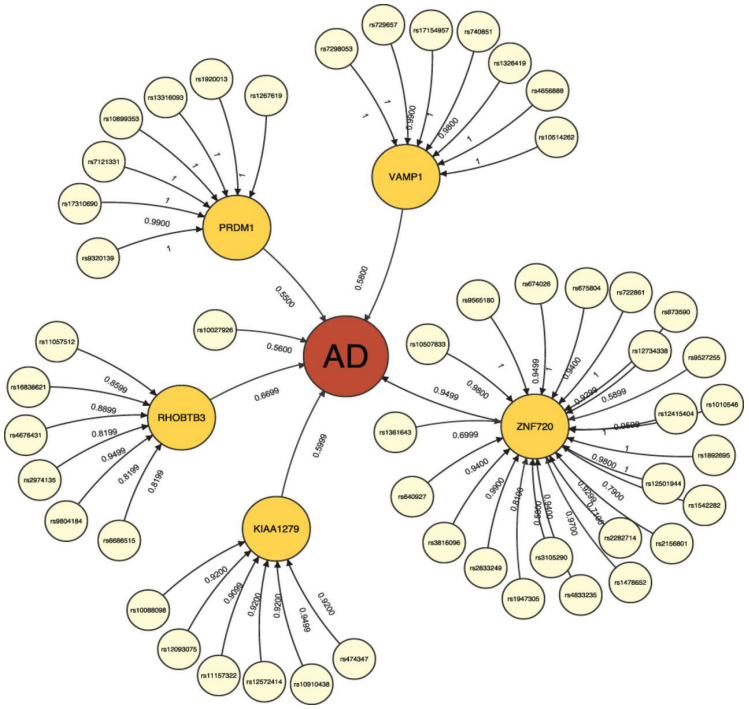
- L1 Regularization (LASSO)

$$\hat{\beta}(\lambda) = \arg\min_{\beta} (\|Y - X\beta\|^2 + \lambda \|\beta\|_1)$$

where $\|Y - X\beta\|^2 = \sum_{i=1}^{n} (Y_i - (X\beta)_i)^2$, $\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$

Method

- Association network construction
  - Sparse regression method to do association mapping
  - Stability selection for false positive control
- Finding significant paths using K-shortest paths algorithm



Method

➢ Association network construction

- Sparse regression method to do association mapping

To identify edges in an association network, we use SRMs as follows:

$$\min_{\mathbf{B}} \frac{1}{2} \|\mathbf{Y} - \mathbf{XB}\|_F^2 + \Omega(\mathbf{B}), \tag{1}$$

$$\min_{\mathbf{D}} \sum_m \sum_i -\log p(z_m^i | \mathbf{y}^i; \mathbf{d}_m) + \Omega(\mathbf{D}), \tag{2}$$

where $\mathbf{B} \in \mathbb{R}^{J \times K}$ and $\mathbf{D} \in \mathbb{R}^{K \times M}$ are regression coefficient matrices whose non-zeros encode associations. For example, $d_m^k \neq 0$ implies association between the k-th gene trait and the m-th phenotype. Also, $p(z_m^i = 1 | \mathbf{y}^i; \mathbf{d}_m) = \frac{1}{1+\exp(-\mathbf{y}^i \mathbf{d}_m)}$, and $\Omega(\cdot)$ is regularizer that induces sparsity in the coefficient matrix. Using Equation (1), we find

Method

➢ Association network construction

- Stability selection

Identifying the most “stable” model structure by using the idea that the same algorithm should yield similar results on similar datasets if the results are “stable”.

Method

- Stability selection

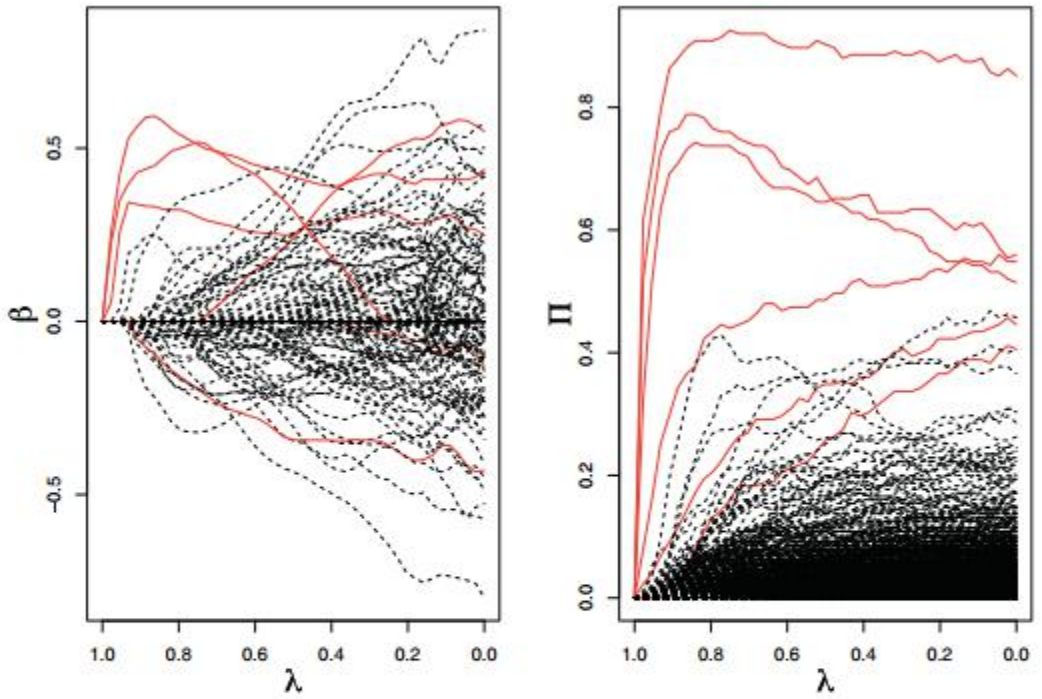


Fig. 10.1 Left: the Lasso regularization path for the riboflavin production data set with $n = 115$ and $p = 4088$. The paths of the 6 non-permuted variables (genes) are plotted as solid, red lines, while the paths of the 4082 permuted genes are shown as broken, black lines. Selecting a model including all 6 non-permuted variables (genes) invariably means selecting a large number of irrelevant noise variables. Right: the stability path of the Lasso. The first 4 variables chosen with stability selection are truly non-permuted variables. The figure is taken from Meinshausen and Bühlmann (2010).

Method

- Association network construction

Algorithm 1: Sparse Regression Under Stability Selection.

Input: **P**: Input matrix, **O**: Output matrix, where $(\mathbf{P}, \mathbf{O}) \in \{(\mathbf{X}, \mathbf{Y}), (\mathbf{Y}, \mathbf{Z}), (\mathbf{X}, \mathbf{Z})\}$, $\mathcal{S}_d$: selected SNPs for the d-th output by screening, π_{thr}: threshold for stability selection ($0.5 \leq \pi_{thr} \leq 1$), T: total number of random samples

Output: $\{\mathcal{U}_d\}$: selected inputs for the d-th output with scores ($d = 1, \ldots, D$)

1. $\Pi_l^d = 0, l \in \mathcal{S}_d$ for $d = 1, \ldots, D$
2. Randomly select $\lfloor N/2 \rfloor$ samples from N samples without replacement
3. Given the $\lfloor N/2 \rfloor$ subsamples and a SRM (i.e. Equation (1) or (2)), find $\{\lambda_d\}$ using cross-validation, denoted by $\{\lambda_d^*\}$
4. $s_l^d = 0, \forall l \in \mathcal{S}_d$ for $d = 1, \ldots, D$
5. **for** $t = 1$ *to* T **do**
6. Randomly select $\lfloor N/2 \rfloor$ samples from N samples without replacement
7. Given the $\lfloor N/2 \rfloor$ subsamples, solve a SRM with $\{\lambda_d^*\}$
8. $s_l^d = s_l^d + 1$ for all selected terms l for the d-th output
9. $\Pi_l^d \leftarrow \frac{s_l^d}{T}, \forall l \in \mathcal{S}_d$ for $d = 1, \ldots, D$
10. $\mathcal{U}_d = \{(l, \Pi_l^d) : \Pi_l^d \geq \pi_{thr}\}$ for $d = 1, \ldots, D$

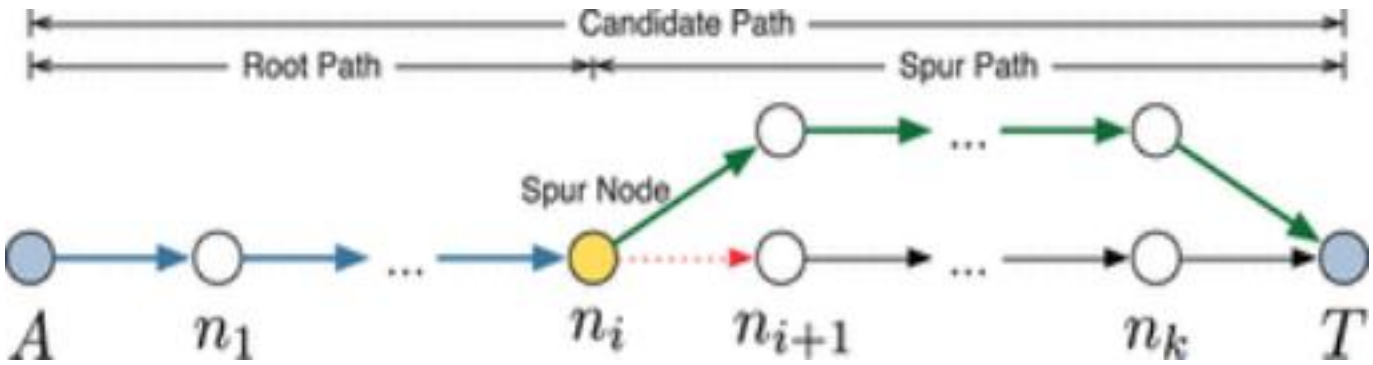
Method

➢ Finding significant paths

- To rank path associations, we further assign scores to paths based on edge scores as follows:

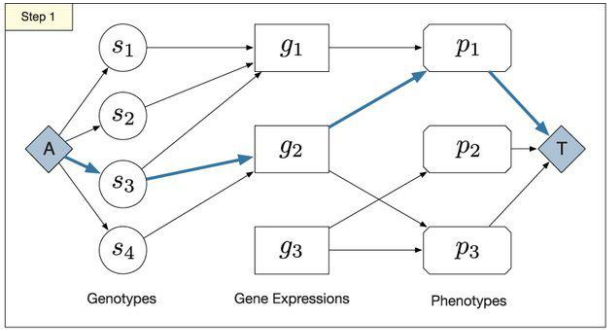
$$score(Path_i) = \prod_{Edge \subseteq Path_i} score(Edge)$$

- K-shortest path algorithm

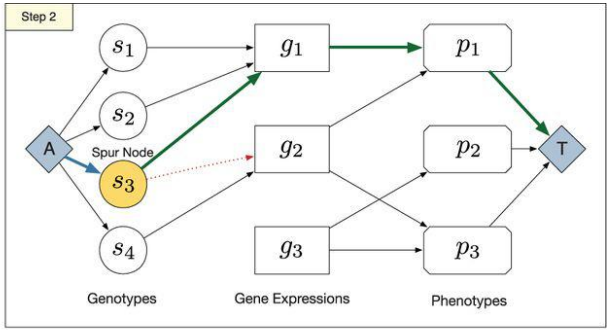


Method

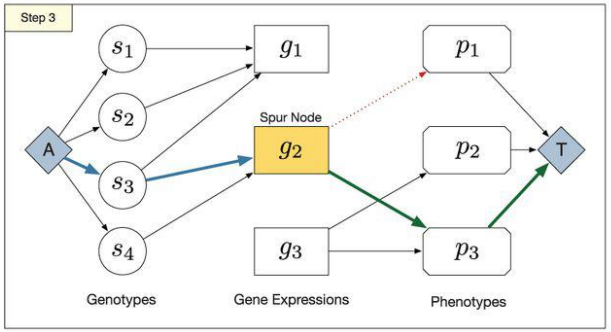
- **Illustration of one iteration of the K-shortest algorithm**



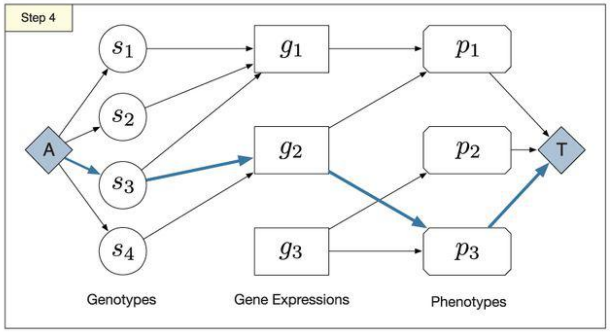
[Step 1] In this example, we assume that we found a shortest path $P = \{A, s_3, g_2, p_1, T\}$ in the previous iteration of the K-shortest path algorithm.



[Step 2] It picks the spur node s_3 and removes the edge $\{s_3, g_2\}$. Then, it finds the spur path $\{s_3, g_1, p_1, T\}$, a shortest path from s_3 to T. Appending the spur path, to the root path $\{A, s_3\}$, we have a new candidate path $\mathrm{CP}_1 = \{A, s_3, g_1, p_1, T\}$.



[Step 3] It picks a different spur node g_2 and removes the edge $\{g_2, p_1\}$ from the network. Concatenating the root path $\{A, s_3, g_2\}$ and the spur path $\{g_2, p_3, T\}$, we have another candidate path $\mathrm{CP}_2 = \{A, s_3, g_2, p_3, T\}$.

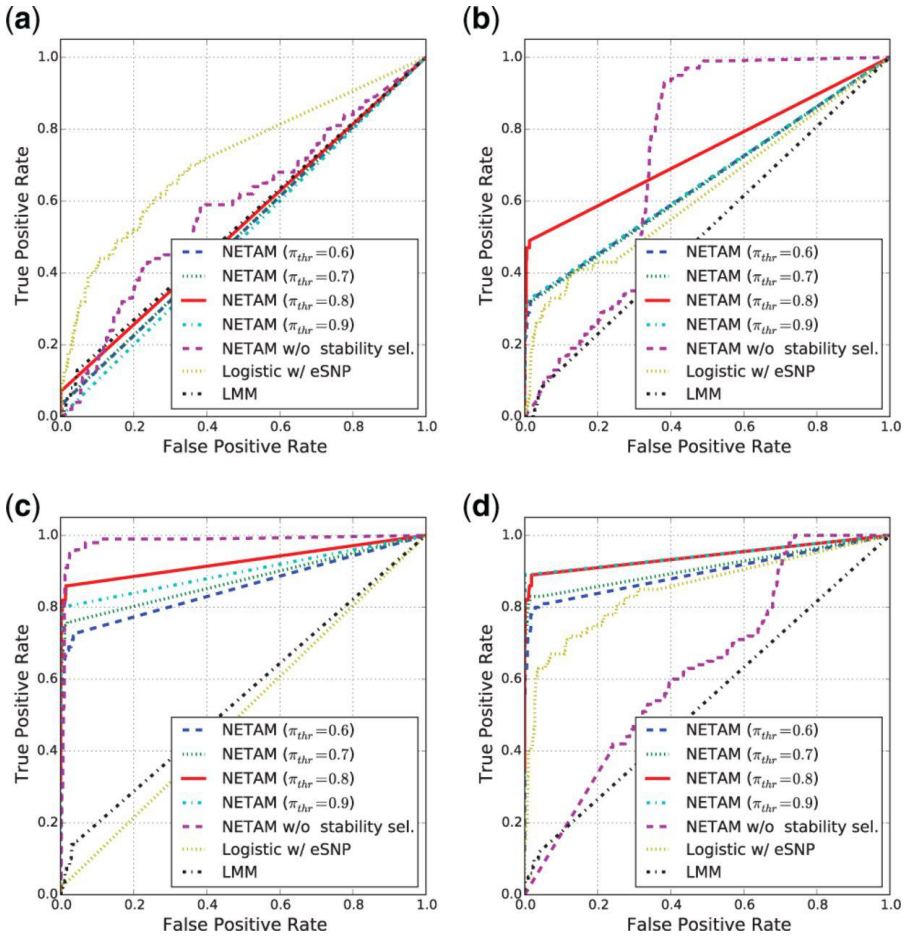


[Step 4] After generating the candidate paths from P, we have a set of candidate paths $C = \{\mathrm{CP}_1, \mathrm{CP}_2\}$. If the path score of CP_2 is smaller than that of CP_1, we select $\mathrm{CP}_2 = \{A, s_3, g_2, p_3, T\}$ as the next shortest path.

Results

ROC curves to compare the performance of NETAM with association mapping methods such as linear mixed model (LMM) and L1-regularized logistic regression with eSNPs (Logistic w/ eSNP) with different sample sizes: (a) N = 200, (b) N = 500, (c) N = 800, and (d) N = 1100.

- **Simulation data**
- **NETAM > traditional methods**
- **Stability selection works**



Results

- Late-onset Alzheimer's disease data (Zhang et al., 2013)

- 270 AD cases and 270 controls with 511 997 SNPs
- Expression levels of 40638 DNA probes including known and predicted genes, miRNAs, and non-coding RNAs in the cerebellum in the brain.

✓ ran Algorithm 1 with lasso, 10-fold cross-validation,

$$\pi_{thr} = 0.5, \quad and \quad T = 100$$

found 477 path associations in the AD data, focus on analyzing top 36 paths involving ZNF720, RHOBTB3 or VAMP1

Summary

- Use genotypes, transcriptome, phenotypes data
- Association network construction
  - Sparse regression method to do association mapping
  - Stability selection for false positive control
- Finding significant paths using K-shortest paths algorithm

✓ Extend association networks to capture complex association mechanisms by introducing additional edges between different SNPs, between different gene traits or between different phenotypes