# netgwas: An R Package for Network-Based Genome Wide Association Studies

P. Behrouzi
Wageningen University and Research
pariya.behrouzi@wur.nl

D. Arends
Humboldt-Universitt zu Berlin
danny.arends@gmail.com

E. C. Wit
University of Groningen
e.c.wit@rug.nl

#### Abstract

Graphical models are powerful tools for modeling and making statistical inferences regarding complex associations among variables in multivariate data. In this paper we introduce the R package **netgwas**, which is designed based on undirected graphical models to accomplish three important and interrelated goals in genetics: constructing linkage map, reconstructing linkage disequilibrium (LD) networks from multi–loci genotype data, and detecting high–dimensional genotype–phenotype networks.

The **netgwas** package deals with species with any chromosome copy number in an unified way, unlike other software. It implements recent improvements in both linkage map construction (Behrouzi and Wit, 2018), and reconstructing conditional independence network for non-Gaussian continuous data, discrete data, and mixed discrete-and-continuous data (Behrouzi and Wit, 2017). Such datasets routinely occur in genetics and genomics such as genotype data, and genotype-phenotype data.

We demonstrate the value of our package functionality by applying it to various multivariate example datasets taken from the literature. We show, in particular, that our package allows a more realistic analysis of data, as it adjusts for the effect of all other variables while performing pairwise associations. This feature controls for spurious associations between variables that can arise from classical multiple testing approach. This paper includes a brief overview of the statistical methods which have been implemented in the package. The main body of the paper explains how to use the package. The package uses a parallelization strategy on multi-core processors to speed-up computations for large datasets. In addition, it contains several functions for simulation

and visualization. The netgwas package is freely available at https://cran.r-project.org/web/packages/netgwas.

## 1 Introduction

Graphical models (Lauritzen, 1996) are commonly used, particularly in statistics and machine learning, to describe conditional independence relationships among variables in multivariate data. In graphical models, each random variable is associated with a node in a graph, and links represent conditional dependency between variables; the absence of a link implies that the variables are conditionally independent given the rest of the variables – the pairwise Markov property.

The **netgwas** package reconstructs undirected graphs for non-Gaussian, discrete, and mixed discrete-and-continuous datasets which arise routinely in genetics and genomics, particularly in systems genetics. The **netgwas** package includes three key functions for: (i) linkage map construction, (ii) linkage disequilibrium networks construction, and (iii) genotype—phenotype networks reconstruction. The package includes various functional modules, including ordinal data (e.g. genotype data) generation for simulation studies, several methods for reconstructing underlying undirected conditional independence graphs, and a visualization tool. Below we provide a brief introduction for each module.

The linkage map describes the linear order of genetic markers within linkage groups (chromosomes). It is the first requirement for estimating the genetic background of phenotypic traits in quantitative trait loci (QTL) studies. In practice, many software packages for performing QTL analysis require linkage maps (Taylor et al., 2011, Huang et al., 2012). Most organisms are categorized as diploid or polyploid by comparing its copy number of each chromosome. Diploids have two copies of each chromosome (like humans). Polyploid organisms have more than two copies of each chromosome (like most of crops). Polyploidy is common in plants and in different crops such as apple, potato, and wheat, which contain three (triploid), four (tetraploid), and six (hexaploid) copies from each of their chromosomes, respectively. So far, the linkage map construction tools that have been developed before are based on the ploidy level of the species, where different map construction methods have been proposed for different species. However, in Behrouzi and Wit (2018) we developed a method that is able to construct linkage map in a unified way for both diploid and polyploid species. And we have implemented the method in netgwas package. Tools like R/qtl (Broman et al., 2003), OneMap (Margarido et al., 2007), Pheno2Geno (Zych et al., 2015), and MSTMAP (Taylor and Butler, 2017) among others construct linkage maps only for diploid species. And tools like MAPMAKER (Lander et al., 1987), Tetraploid-SNPMap (Hackett et al., 2017), and polymapR (Bourke et al., 2018) construct linkage map for polyploid species. Despite the importance of polyploids especially in crop research, statistical tools for their map construction are underdeveloped. The existing tools for polyploid map construction are mainly focused on a specific type of polyploid species (mostly tetraploids), or they require manual interaction and visual inspection, which limit their usability. Existing tools for polyploid map construction are mainly based on estimation of recombination frequency and LOD (logarithm of the odds ratio) scores (Wang et al., 2016). Whereas, netgwas uses more information by using a multivariate approach. It implicitly uses the complete set of comparisons between all markers and combine this into a single map, whereas the other methods use pairwise testing to construct maps. This often leads to an underpowered approach and confounding of merely correlated genotypes by failing to correct for intermediate markers (Behrouzi and Wit, 2018).

The linkage map, which can be constructed using the first module of netgwas, provides the genetic basis for the second module of netgwas that detects the patterns of linkage disequilibrium and segregation distortion in a population. Segregation distortion (SD) refers to any deviation from expected segregation ratios based on Mendelian rules of inheritance. And linkage disequilibrium (LD) refers to non-random relations between loci (locations) on the same or on different chromosomes. Revealing the patterns of linkage disequilibrium is important for association mapping study as well as for studying the genomic architecture of a genome. Various methods have been published in the literature for measuring statistical association between alleles at different loci (see for instance Hedrick (1987), Mangin et al. (2012)). Most of these measures are based on an exhaustive genome scan for pairs of loci and  $r^2$ measure, the square of the loci correlation. The drawback of such approaches is that association testing in the genome-scale is underpowered, so that weak longrange LD will go undetected. Also, such methods cannot simultaneously take the information of more than two loci into account. Moreover, they do not make full and efficient use of modern multi-loci data. Here, we implemented the method proposed by Behrouzi and Wit (2017) to detect short- and long-range LD patterns in diploid and polyploid species. **netgwas** efficiently estimates pairwise interactions between different loci in a genome while adjusting for the effect of other loci. Technically, this requires estimating a sparse adjacency matrix from a multi-loci genotype data, which usually contains large number of markers (loci), where the number of markers can far exceed the number of individuals. The non-zero patterns of the adjacency matrix in **netgwas** shows the structures of short—and long—range LD of the genome. The strength of associations between distant loci can be calculated using partial correlations. Furthermore, netgwas already accounts for the correlation between markers, while associating them to each other and thereby avoids the problem of population structure (that is physically unlinked markers are correlated).

A major problem in genetics is the association between genetic markers and the status of a disease (trait or phenotype). Genome-wide association studies are among the most important approaches for further understanding of genetics underlying com-

plex traits (Welter et al., 2013). However, in genome-wide association methods genetic markers are often tested individually for association with the phenotype. Since genome-wide scans analyze thousands or even millions of markers, the issue of multiple testing is usually addressed by using a stringent significance threshold of  $5 \times 10^8$ (Panagiotou et al., 2011). Such methods work only if the associations are strong enough to pass the stringent threshold. However, even if that is the case, this type of analysis has several limitations, which have been discussed extensively in the literature (Hoggart et al., 2008, He and Lin, 2010, Rakitsch et al., 2012, Buzdugan et al., 2016). Particularly, the main issue of this type of analysis is when we test the association of the phenotype to each genetic marker individually, and ignore the effects of all other genetic markers. This leads to failures in the identification of causal loci. If we consider two correlated loci, of which only one is causal for the phenotype, both may show a marginal association, but only the causal locus will be detected by our method. The **netgwas** tackles this issue by using Gaussian copula graphical model, which accounts for the correlation between markers, while associating them to the phenotypes. As it is shown in Klasen et al. (2016), this key feature avoids the need to correct for population structure or any genetic background, as **netgwas** simultaneously associates all markers to the phenotype. In contrast, previous genome-wide association methods rely on population structure correction to avoid false genotypephenotype due to their single-loci approach. In the genotype-phenotype network construction module of **netgwas**, nodes in the graph are either genetic markers or phenotypes, and each phenotype is connected by an edge to a marker or a group of markers. Furthermore, in genotype-phenotype-environment networks, nodes in the graph are either genetic markers, phenotypes or environmental variables, and two nodes are connected if they are associated yet after adjusting for the effect of remaining nodes.

To make our method computationally faster for large datasets, the **netgwas** package uses multi-core computing capabilities based on the **parallel** package. To make it easy to use, **netgwas** uses S3 classes as return values of its functions. Our package is available under general public license (GPL  $\geq$  3) at the Comprehensive R Archive Network (CRAN) at http://cran.r-project.org/packages=netgwas.

In Section 2 we illustrate the user interface of the package. In Section 3 we explain the methodological background of the package. In Section 4 we describe the main functions implemented in the **netgwas** package. Furthermore, we present illustrative examples that involves map construction process for a diploid A.thaliana and a tetraploid potato populations. In addition, given the usual size of GWAS data, performing the analysis on a real dataset with  $> 10^5$  SNPs is computationally expensive. For this reason, in order to explain the two GWAS modules of the **netgwas** package, we use small real datasets in mice and plants.

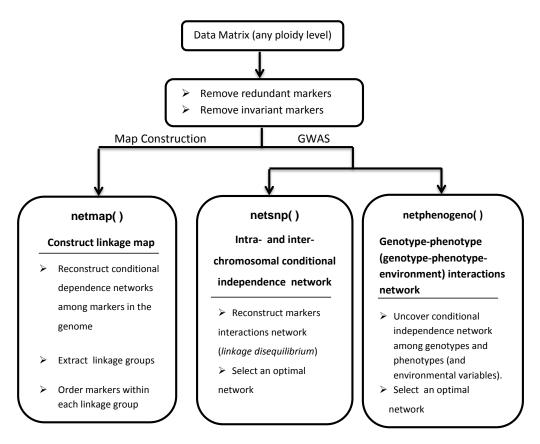


Figure 1: The main functions in **netgwas** package.

## 2 User interface

In the R environment, the **netgwas** package can be loaded using the following commands:

```
R> install.packages( "netgwas" )
R> library( "netgwas" )
```

The **netgwas** package consists of three modules:

Module 1. Data simulation: this simulates data in two different ways:

1. Based on a Gaussian copula graphical model it simulates ordinal variables with a genome-like network structure. An inbred genotype data can be generated for p number of SNP markers, for n number of individuals, for k genotype states in a q-ploid species where q represents chromosome copy number ( or ploidy level of chromosomes).

The simulated data mimic a genome-like graph structure: First, there are g linkage groups (each of which represents a chromosome); then within each linkage

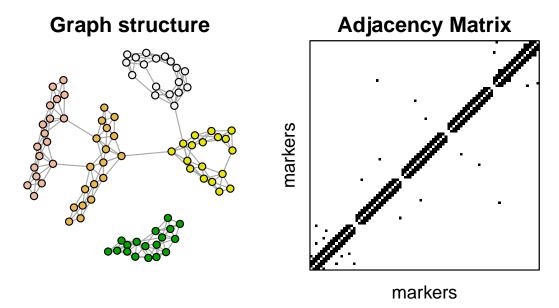


Figure 2: A model-based simulation. (left) Each color corresponds to a chromosome, (right) the correspondent adjacency matrix. The five chromosomes are shown in the diagonal of the matrix.

group adjacent markers, adjacent, are linked via an edge as a result of genetic linkage. Also, with probability alpha a pair of non-adjacent markers in the same chromosome are given an edge. Inter-chromosomal edges are simulated with probability beta. These links represent long-range linkage disequilibriums. The corresponding positive definite precision matrix  $\Theta$  has a zero pattern corresponding to the non-present edges. The underlying variable vector Z is simulated from either a multivariate normal distribution,  $N_p(0, \Theta^{-1})$ , or a multivariate t-distribution with degrees of freedom d and covariance matrix  $\Theta^{-1}$ . We generate the genotype marginals using random cutoff-points from a uniform distribution, and partition the latent space into k states. The function can be called with the following arguments

The output of the example is shown in Figure 2.

2. It generates diploid recombinant inbred lines (RILs) using recombination fraction and a CentiMorgan position of markers across the chromosomes. The function can be called with the following arguments

```
R> set.seed(2)
R > ril < sim RIL(g = 5, d = 25, n = 200, cM = 100, selfing = 2)
R> ril$data[1:3, ]
     M1.1 M2.1 M3.1 M4.1 M5.1 M6.1 M7.1 M8.1
                                                          M23.5 M24.5 M25.5
                       0
                             0
                                   0
                                              0
                                                             0
                                                                   0
ind1
                                        0
                                                                          0
ind2
      2
            1
                  1
                       1
                             1
                                   1
                                        2
                                              2
                                                             1
                                                                   1
                                                                          1
            2
                  2
                       2
                             2
                                   2
                                        2
                                              1
                                                             0
                                                                   0
                                                                          0
ind3
      1
R> ril$map
     chr
           marker
                    cМ
1
       1
           M1.1
                   0.00000
2
       1
           M2.1
                   4.166667
3
       1
           M3.1
                   8.333333
123
           M23.5
                   91.666667
      5
124
       5
           M24.5
                   95.833333
125
      5
```

where g and d represent the number of chromosomes and the number of markers in each chromosome, respectively. The number of sample size can be specified by n. The arguments cM and selfing show the length of chromosome based on centiMorgan position and the number of selfing in the RIL population, respectively.

Module 2. Inference Method: The functions netmap(), netsnp(), and netphenogeno() provide three methods to estimate undirected graphs as follows: a Gaussian copula graphical model using (i) the Gibbs sampling algorithm described in Behrouzi and Wit (2017); and (ii) the approximation algorithm described in Behrouzi and Wit (2017); (iii) the nonparanormal skeptic method Liu et al. (2012) as alternative, along with the Gaussian copula models.

**Module 3. Output:** This module includes two types of functions:

M25.5

100.00000

- Graph selection: The function selectnet tunes the penalty parameter, based on an information criterion, and provides the selected graph.
- Visualization: The plotting function plot.netgwas provides a visualization plot to monitor the path of estimated networks for a range of penalty terms; the functions plot.netgwasmap, plot.select and plot.simgeno visualize the corresponding network, the selected graph and the conditional dependence structures of the model-based simulated data.

## 3 Methodological background

In graphical models, each random variable is associated with a node on a graph. The conditional dependence relationships among the random variables are presented as a graph G = (V, E) in which  $V = \{1, 2, ..., p\}$  specifies a set of nodes and a set of existing links  $E \subset V \times V$  Lauritzen (1996). Our focus here is on undirected graphs, in which  $(i, j) \in E \Leftrightarrow (j, i) \in E$ . The absence of a link between two nodes specifies the pairwise conditional independence of those two associated random variables given the remaining variables, while a link between two variables indicates their conditional dependence. In Gaussian graphical models, the observed data follow a multivariate Gaussian distribution  $\mathcal{N}_p(\mu, \Theta^{-1})$ . Here, conditional independence is implied by the zero structure of the precision matrix  $\Theta$ . Based on the pairwise Markov property, variables i and j are conditionally independent given the remaining variables, if and only if  $\Theta_{ij} = 0$ . This property implies that the links in graph G = (V, E) correspond with the nonzero elements of the precision matrix  $\Theta$ , i.e.  $E = \{(i, j) | \Theta_{ij} \neq 0\}$ .

## Sparse latent graphical model

A p-dimensional copula  $\mathcal{C}$  is a multivariate distribution with uniform margins on [0,1]. Any joint distribution function can be written in terms of its marginals and a copula which encodes the dependence structure. Here we consider a subclass of joint distributions encoded by the Gaussian copula,

$$F(y_1, \dots, y_p) = \Phi_p \Big( \Phi^{-1}(F_1(y_1)), \dots, \Phi^{-1}(F_p(y_p)) \mid \mathcal{C} \Big)$$

### Algorithm 1 Monte Carlo Gibbs sampling of latent covariance

**Input:** A data set containing the variables  $Y_i$ , i = 1, ..., n.

**Output:** Mean of the conditional expectation.

- 1: For each  $j \in V$  generate the latent data from  $Y_j = F_j^{-1}(\Phi(Z_j))$ , where  $\Phi$  defines the CDF of the standard normal distribution.
- 2: Calculates vectors of lower and upper truncation points.
- 3: **for** i = 1, ..., n **do**
- 4: Sample from a p-variate truncated normal distribution.
- 5: **for** N iteration **do**
- 6: Estimate  $R_i = E(Z^{(i)}Z^{(i)t}|y^{(i)}, \widehat{\Theta}^{(m)}, \widehat{\mathcal{D}}).$
- 7: end for
- 8: Update  $\widehat{R}_i = \frac{1}{N} Z_{\star}^{(i)t} Z_{\star}^{(i)}$ .
- 9: end for
- 10: Calculate  $\hat{\bar{R}} = \frac{1}{n} \sum_{i=1}^{n} \hat{R}_{i}$ .

where  $\Phi_p(. \mid \mathcal{C})$  is the cumulative distribution function (CDF) of p-variate Gaussian distribution with correlation matrix  $\mathcal{C}$ ;  $\Phi$  is the univariate standard normal CDF; and  $F_i$  is the CDF of  $Y_i$ . Note that  $y_i$  and  $y_{i'}$  are independent if and only if  $\mathcal{C}_{ii'} = 0$ .

A Gaussian copula can be written in terms of latent variables Z: Let  $F_j^{-1}(y) = \inf\{y: F_j(x) \geq y, x \in \mathcal{R}\}$  be the pseudo-inverse of the marginals and  $\Omega$  be the covariance matrix whose diagonal has normalized with  $\mathcal{C}$  as its correlation matrix. Then a Gaussian copula is defined as:

$$Y_{ij} = F_j^{-1}(\Phi(Z_{ij}))$$
$$Z \sim \mathcal{N}_p(0, \Omega)$$

where  $Y = (Y_1, \ldots, Y_p)$  and  $Z = (Z_1, \ldots, Z_p)$  represent the non-Gaussian observed variables and Gaussian latent variables, respectively. We denote the associated latent data as  $z^{(1:i)} = [z^{(1)}, \ldots, z^{(n)}]$ , where  $z^{(i)} = (z_1^{(i)}, \ldots, z_p^{(i)})$ .

In order to learn the graphical model, our objective is to estimate precision, the inverse of covariance, matrix  $\Omega^{-1} = \Theta$  from n independent observations  $y^{(1:i)} = [y^{(1)}, \ldots, y^{(n)}]$ , where  $y^{(i)} = (y_1^{(i)}, \ldots, y_p^{(i)})$ . It is well known that the conditional independence between two variables, given other variables, is equivalent to the corresponding element in the precision matrix being zero, i.e.  $\theta_{ij} = 0$ ; or put another way, a missing edge between two variables in a graph G represents conditional independence between the two variables given all other variables. In other words, conditional independence is quantified in terms of partial correlations.

### **Algorithm 2** Approximation of the conditional expectation

```
Input: A data set containing the variables Y_i, i = 1, ..., n.
```

**Output:** Mean of the conditional expectation.

```
1: Initialize r_{j,j'} for 1 \le j, j' \le p using:
        E(z_{j}^{(i)} \mid y^{(i)}; \widehat{\mathcal{D}}, \widehat{\Theta}) \approx E(z_{j}^{(i)} \mid y_{j}^{(i)}; \widehat{\mathcal{D}}), E((z_{j}^{(i)})^{2} \mid y^{(i)}; \widehat{\mathcal{D}}, \widehat{\Theta}) \approx E((z_{j}^{(i)})^{2} \mid y_{j}^{(i)}; \widehat{\mathcal{D}}),
        and E(z_j^{(i)} z_{j'}^{(i)} \mid y^{(i)}; \widehat{\mathcal{D}}, \widehat{\Theta}) \approx E(z_j^{(i)} \mid y_j^{(i)}; \widehat{\mathcal{D}}) E(z_{j'}^{(i)} \mid y_{j'}^{(i)}; \widehat{\mathcal{D}}) for i = 1, \dots, n
  2: Estimate \widehat{\Theta}
  3: for i = 1, ..., n do
              if j = j' then
  4:
                    Calculate E((z_j^{(i)})^2 \mid y_i^{(i)}; \widehat{\mathcal{D}}, \widehat{\Theta})
  5:
              else
  6:
                    Calculate E(z_i^{(i)} \mid y^{(i)}; \widehat{\mathcal{D}}, \widehat{\Theta})
  7:
                    update E(z_{i}^{(i)}z_{i'}^{(i)} | y^{(i)}; \widehat{\mathcal{D}}, \widehat{\Theta}) \approx E(z_{i}^{(i)} | y^{(i)}; \widehat{\mathcal{D}}) E(z_{i'}^{(i)} | y^{(i)}; \widehat{\mathcal{D}})
              end if
  9: end for
10: Calculate r_{j,j'} = \frac{1}{n} \sum_{i=1}^{n} E(z_j^{(i)} z_j^{(i)t} \mid y^{(i)}, \widehat{\Theta}^{(m)}, \widehat{\mathcal{D}})
```

In the classical low-dimensional setting, in which p is smaller than n, it is natural to implement a maximum likelihood approach to obtain the inverse of the sample covariance matrix. However, in modern applications like genetic networks, including linkage map construction, intra— and inter— chromosomal interactions and network—based QTL analysis, the dimension p is routinely far larger than n, meaning that the inverse sample covariance matrix does not exist. Motivated by the sparseness assumption of the graph, i.e., most  $\theta_{ij}$  are zero, we tackle the high-dimensional inference problem by using the penalized log-likelihood estimation procedure. We consider the penalized likelihood,

$$\ell_Y^p(\Theta) = \frac{n}{2} \log |\Theta| - \frac{1}{2} \sum_{i=1}^n \int_{z^{(i)}} z^{(i)^T} \Theta z^{(i)} dz^{(i)} - \sum_{j \neq j'}^p P_{\lambda}(|\theta_{jj'}|)$$
 (1)

where we use a sparsity penalty function such as the  $L_1$  norm penalty or smoothly clipped absolute deviation (SCAD) penalty on the precision matrix. The  $L_1$  norm is defined as

$$P_{\lambda}(\theta) = \lambda |\theta|$$

which leads to a desirable optimization problem. Alternatively, we define the SCAD penalty in terms of its first order derivative, given by

$$P'_{\lambda,a}(\theta) = \lambda \left\{ I(|\theta| \le \lambda) + \frac{(a\lambda - |\theta|)_+}{(a-1)\lambda} I(|\theta| > \lambda) \right\}$$

for  $\theta \ge 0$ , where  $\lambda > 0$  and a > 0 are two tuning parameters. This penalty function produces sparse solution and approximately unbiased coefficient estimates for large coefficients. For the numerical studies we use a = 3.7 as recommended by Fan and Li (2001).

Since Y includes discrete variables, those integrals in (1) are intractable, and instead we solve (1) by a penalized expectation maximization (EM) algorithm.

$$\Theta_{\lambda}^{(m)} = \arg\max_{\Theta} Q(\Theta|\Theta^{\star}) - \sum_{j \neq j'}^{p} P_{\lambda}(|\theta_{jj'}|)$$
(2)

where

$$Q(\Theta \mid \Theta^{\star}) = \frac{n}{2} \left[ \log |\Theta| - \operatorname{tr}\left(\frac{1}{n} \sum_{i=1}^{n} E\left(Z^{(i)} Z^{(i)^{T}} \mid y^{(i)}, \Theta^{\star}\right)\Theta\right) \right], \tag{3}$$

and m is the iteration number within the EM algorithm. We compute the conditional expectation inside (3) using two different approaches: numerically through a Monte Carlo (MC) sampling method as explained in algorithm 1, and through a first order approximation based on algorithm 2. The most flexible and generally applicable

approach for obtaining a sample in each iteration of an MCEM algorithm is through a Markov chain Monte Carlo (MCMC) routine like Gibbs and MetropolisHastings samplers (more details in Behrouzi and Wit (2017)). Alternatively, the conditional expectation in equation (3) can be computed by using an efficient approximation approach which calculates elements of the empirical covariance matrix using the first and second moments of a truncated normal distribution with mean and variance as follows (see Behrouzi and Wit (2017) for more details):

$$\mu_{ij} = \widehat{\Omega}_{j,-j} \widehat{\Omega}_{-j,-j}^{(-1)} z_{-j}^{(i)^T},$$

$$\sigma_{i,j}^2 = 1 - \widehat{\Sigma}_{j,-j} \widehat{\Sigma}_{-j,-j}^{-1} \widehat{\Sigma}_{-j,-j}.$$

The proposed method is practical when some observations are missing. If genotype information for genotype j is missing, it is still possible to draw Gibbs samples for  $Z_j$  or approximate the empirical covariance matrix, as the corresponding conditional distribution is Gaussian.

The optimization problem in (2) can be solved efficiently in various ways by using glasso or CLIME approaches Friedman et al. (2008), Hsieh et al. (2011). Convergence of the EM algorithm for penalized likelihood problems has been proved in Green (1990). Our experimental study shows that the algorithm usually converges after several iterations (< 10). Note that in both cases the penalty parameter  $\lambda$  needs to be selected appropriately in the last EM iteration to recover the precision matrix. Thus, in line with Behrouzi and Wit (2017) we perform model selection using eBIC or AIC to choose a suitable regularization parameter  $\lambda^*$  in equation (1) to produce a sparse undirected graph with a sparsity pattern corresponding to  $\widehat{\Theta}_{\lambda^*}$ . Alternatively, instead of using the EM algorithm, a nonparanormal skeptic approach can be used to estimate graph structure through Spearman's rho and Kendall's tau statistics; details can be found in Liu et al. (2012) and Behrouzi and Wit (2018).

## Extension to map construction

Here we convert a p-dimensional network to a one-dimensional map using two different approaches. Let  $G(V^{(d)}, E^{(d)})$  be a sub-graph on the set of unordered d markers, where  $V^{(d)} = \{1, \ldots, d\}$ ,  $d \leq p$  and the edge set  $E^{(d)}$  represents all the links among d markers. Depending on the type of experimental population we introduce two methods to order markers, one based on dimensionality reduction and another based on bandwidth reduction. Both methods result in a one-dimensional map.

In inbred populations, loci in the genome of the progenies can be assigned to their parental homologues, resulting in a simpler conditional independence relationship between neighboring markers. Here, we use multidimensional scaling (MDS) to represent the original p-dimensional space in a one-dimensional map while attempting to maintain pairwise distances. We calculate the distance matrix  $\mathcal{D}$  as follow

$$\mathcal{D}_{ij} = \begin{cases} -\log(\rho_{ij}) & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}$$

$$\rho_{ij} = -\frac{\theta_{ij}}{\sqrt{\theta_{ii}}\sqrt{\theta_{jj}}},$$

where  $\theta_{ij}$  is the ij-th element of the precision matrix  $\Theta$ . We aim to construct a configuration of d data points in a one-dimensional Euclidean space by using information about the distances between the d nodes. In this regards, we define a linear ordering L of d elements such that the distance  $\widehat{\mathcal{D}}$  between them is similar to  $\mathcal{D}$ . We consider a metric multi-dimensional scaling

$$\widehat{L} = \arg\min_{L} \sum_{i=1}^{d} \sum_{j=1}^{d} (\mathcal{D}_{ij} - \widehat{\mathcal{D}}_{ij})^{2}$$

that minimizes  $\widehat{L}$  across all linear orderings.

We propose a different ordering algorithm for outbred populations. In these populations, mating of two non-homozygous parents result in markers in the genome of progenies that cannot easily be mapped into their parental homologues. To order markers in outbred populations, we use the reverse Cuthill-McKee (RCM) algorithm (Cuthill and McKee, 1969) to permute the sparse matrix  $\widehat{\Theta}_{\lambda}^{(d)}$  that has a symmetric sparsity pattern into a band matrix form with a small bandwidth. The bandwidth of the associated adjacency matrix A is defined as  $\beta = \max_{A_{ij} \neq 0} |i-j|$ . The algorithm produces a permutation matrix P such that  $PAP^T$  has a smaller bandwidth than matrix A does. The bandwidth decreases by moving the non-zero elements of the matrix A closer to the main diagonal. The way to move the non-zero elements is determined by relabeling the nodes in graph  $G(V_d, E_d)$  in consecutive order. Moreover, all of the nonzero elements are clustered near the main diagonal.

## 4 The netgwas environment

The **netgwas** package implements the Gaussian copula graphical models (Behrouzi and Wit, 2017) to (i) construct linkage maps for bi-parental species with any ploidy level, namely diploid (2 sets), triploid (3 sets), tetraploid (4 sets) and so on; (ii) explore high-dimensional short—and long—range linkage disequilibrium (LD) networks among SNP markers while controlling for the effect of other SNPs. The inferred LD networks reveal epistatic interactions across a genome when viability of the particular genetic recombination of the parental lines is considered as phenotype; (iii) infer genotype-phenotype networks from multi-loci multi-trait data, where it measures the pairwise

associations with adjusting for the effect of other markers and phenotypes. Moreover, it detects markers that directly are responsible for that phenotype (disease), and reports the strength of their associations in terms of partial correlations. In addition, the package is able to reconstruct conditional dependence networks among SNPs, phenotypes, and environmental variables. We describe below the user interface and the three main functions (see Figure 1) of the package.

## 4.1 Linkage Map Construction

#### netmap

The netmap() function reconstructs linkage maps for diploid and polyploid organisms. Diploid organisms contain two copies of each chromosomes, one from each parent, whereas polyploids contain more than two copies of each chromosome. In polyploids the number of chromosome sets reflects their level of ploidy: triploids have three copies, tetraploids have four, pentaploids have five, and so forth.

Typically, mating is between two parental lines that have recent common biological ancestors; this is called inbreeding. If they have no common ancestors up to roughly 4-6 generations, then this is called outcrossing. In both cases the genomes of the derived progenies are random mosaics of the genome of the parents. However, in the case of inbreeding parental alleles are distinguishable in the genome of the progeny; in outcrossing this does not hold.

Some *inbreeding* designs, such as Doubled haploid (DH), lead to a homozygous population where the derived genotype data include only homozygous genotypes of the parents namely AA and aa (conveniently coded as 0 and 1) for diploid species. Other inbreeding designs, such as F2, lead to a heterozygous population where the derived genotype data contain heterozygous genotypes as well as homozygous ones, namely AA, Aa, and aa (conveniently coded as 0, 1 and 2) for diploid species. We remark that the Gaussian copula graphical models help us to keep heterozygous markers in the linkage map construction, rather than turn them to missing values as most other methods do in map construction for RIL populations.

Outcrossing or outbred experimental designs, such as full-sib families, derive from two non-homozygous parents. Thus, the genome of the progenies includes a mixed set of many different marker types containing fully informative markers and partially informative markers. Markers are called fully informative when all of the resulting gamete types can be phenotypically distinguished on the basis of their genotypes; markers are called partially informative when they have identical phenotypes (Wu et al., 2002).

The netmap() function handles various inbred and outbred mapping populations, including recombinant inbred lines (RILs), F2, backcross, doubled haploid, and full-sib families, among others. Not all existing methods for linkage mapping support all

inbreeding and outbreeding experimental designs. However, our proposed algorithm constructs a linkage map for any type of experimental design of biparental inbreeding and outbreeding. Also, it covers a wide range of possible population type.

The function can be called with the following arguments

The netmap returns an object of the S3 class type "netgwasmap". The plot.netgwasmap and print.netgwasmap functions work with this object type. The main task in constructing a linkage map using graphical models is to explore the conditional dependence relationships between markers. The argument method is used to specify which method is to be performed. The estimation procedure relies on maximum penalized log-likelihood, where the argument rho controls the sparsity level. To give an example, we show the steps to construct a linkage map for the example data set TetraPotato. This example regards the tetraploid potato. The data are derived from a cross between Jacqueline Lee and MSG227-2, where 156 F1 plants were genotyped across 1972 genetic markers Massa et al. (2015). Five allele dosages are possible in this full-sib autotetraploid mapping population (AAAA, AAAB, AABB, ABBB, BBBB), where the genotypes are coded as  $\{0,1,2,3,4\}$ . This dataset includes 0.07% missing observations. In the following code we estimate the linkage map and plot the results.

```
R> data(tetraPotato)
# Shuffle the order of markers
R> dat <- tetraPotato[ , sample(1:ncol(tetraPotato), ncol(tetraPotato))]</pre>
R> potato.map <- netmap(dat, cross = "outbred"); potato.map</pre>
Number of linkage groups: 12
Number of markers per linkage group: 165 152 183 173 148 129 187 196 153 161 146 157
Total number of markers in the linkage map: 1950.
(22 markers removed from the input genotype data)
Number of sample size: n = 156
Number of categories in data: 5 (01234)
The estimated linkage map is inserted in <YOUR OUTPUT NAME>$map
To visualize the associated network consider plot(<YOUR OUTPUT NAME>)
To visualize the other associated networks consider plot(<YOUR OUTPUT NAME>$allres)
To build a linkage map for your desired network consider buildMap() function
R> plot(potato.map, vis = "unordered markers")
R> plot(potato.map, vis = "ordered markers")
R> potatoMap <- potato.map$map</pre>
```

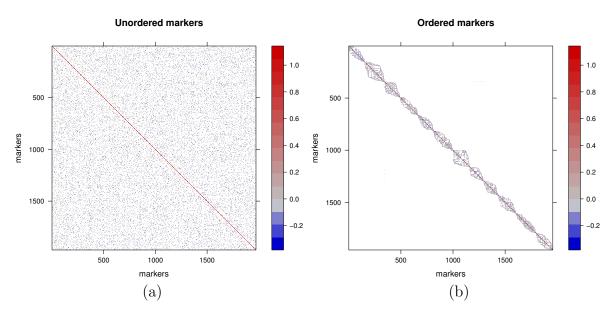


Figure 3: Linkage map construction in potato. (a) Estimated precision matrix for unordered genotype data of tetraploid potato. (b) Estimated precision matrix after ordering markers. All 12 potato chromosomes detected correctly.

The argument cross needs to be specified for ordering markers because, as it discussed before, we introduce different ordering methods in inbred and outbred populations. In inbred populations, markers in the genome of the progenies can be assigned to their parental homologues, resulting in a simpler conditional independence pattern between neighboring markers. In the case of inbreeding, we use multidimensional scaling (MDS). A metric MDS is a classical approach that maps the original high-dimensional space to a lower dimensional space, while attempting to maintain pairwise distances. An outbred population derived from mating two non-homozygous parents results in markers in the genome of progenies that cannot be easily assigned to their parental homologues. Neighboring markers that vary only on different haploids will appear as independent, therefore requiring a different ordering algorithm. In that case, we use the reverse Cuthill-McKee (RCM) algorithm Cuthill and McKee (1969) to order markers. The RCM algorithm is based on graph models. It reduces the bandwidth of the associated adjacency matrix,  $A_{d\times d}$ , for the sparse matrix  $\widehat{\Theta}_{d\times d}$ , where  $d \leq p$ .

Figure 3 visualizes a summary of mapping process. The argument  ${\tt vis}$  in the above plot function can be fixed to "interactive", which it gives a better network resolution particularly for a large number of markers. Figure 3a shows the conditional dependence pattern between unordered SNP markers in the Jacqueline Lee  $\times$  MSG227-2 population. Figure 3b shows the structure of the selected graph after ordering markers. All 12 potato chromosomes were detected correctly. The tetraploid

potato map construction was computed in about 7 minutes on an Intel i7 laptop with 16 GB RAM.

The buildMap() function is designed to construct a linkage map for the "netgwasmap" object format. Its return object is also of class "netgwasmap". This function allows the user to manually interact with the map construction procedure, where the argument opt.index in the below function

```
R> buildMap( res, opt.index, min.m = NULL, use.comu = FALSE)
```

allows to manually select the index of a regularization parameter to build a linkage map on the related network. Default range for opt.index is a value between 1 and 6. The argument min.m is an optional argument that helps the user to keep linkage groups (LGs) that have at least a minimum number of markers of size min.m. This option helps to have a clear appearance of linkage map, where it removes very small group of markers that created "linkage groups". Default value for this argument is 1. The use.comu argument is an alternative approach to find LGs. This option uses fast-greedy algorithm to detect LGs. Below we provide an example of using buildMap function to construct linkage map for A.thaliana.

```
R> data(CviCol)
R> set.seed(1)
R> cvicol <- CviCol[ ,sample(1:ncol(CviCol), ncol(CviCol), replace= FALSE)]
R> out <- netmap(cvicol, cross= "inbred", ncores= 1)
R> out$opt.index
[1] 6
```

The last line of the above code provides the index of the selected graph using information criteria within the map construction procedure. If one is interested in building linkage map, for instance, on 4th network then the related code is

R> plot(bm.thaliana, vis= "summary")

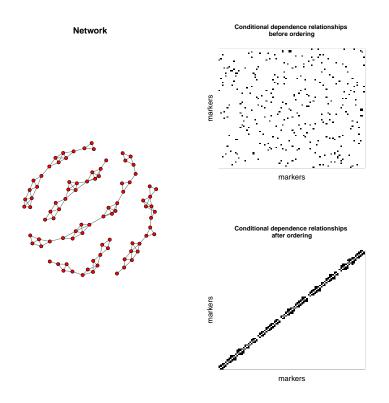


Figure 4: Linkage map construction in *A.thaliana*. All five chromosomes were detected correctly.

Figure 4 shows the output of plotting bm.thaliana. The estimated linkage map is consistent with the existing linkage map for *A.thaliana* (Behrouzi and Wit, 2018). The map construction for CviCol genotype data was computed in 0.6 seconds on an Intel i7 laptop with 16 GB RAM. The plot.netgwasmap and print.netgwasmap functions also work with output object of buildMap function (see the last line of the above code).

If required, the netgwas algorithm detects genotyping errors using the detect.err function. This function calculates the error LOD score for each individual at each marker using Lincoln and Lander (1992) approach; large scores show likely genotyping errors. For identification of genotyping errors, netgwas uses the R/qtl package (Broman et al., 2003), where it gives a list of genotypes that might be in error, when the error LOD scores are smaller than 4 they can probably be ignored (Broman, 2009). The function can be called with the following arguments

```
R> detect.err(netgwas.map, err.prob = 0.01, cutoff = 4,
    pop.type = NULL, map.func = "haldane")
```

As an input argument it requires an object of class netgwasmap. This function supports doubled haploid (DH), backcross (BC), non-advanced recombinant inbred line population with n generations of selfing (RILn) and advanced RIL (ARIL) population types.

The cal.pos() function calculates the genetic distance for an object from netgwasmap class. This function is applicable for diploid populations. It uses the R/qtl package to calculate genetic distance using different distance functions. Default is the Haldane genetic distance function. The function can be called with the following arguments

```
R> cal.pos (netgwasmap, pop.type = NULL , map.func = "haldane", chr )
```

The netgwas2cross() function converts netgwasmap object from netgwas package to cross object from R/qtl package, and vice versa using cross2netgwas() function converts cross object to netgwasmap object. These two functions make netgwas flexible with respect to further genetic investigation using R/qtl package. Also, cross objects from R/qtl package can be analyzed using netgwas package.

### 4.2 Genome Wide Association Studies

#### netsnp

The function netsnp() reconstructs conditional independence relationships simultaneously among all genetic markers in a genome. In other words, it constructs intranand inter-chromosomal conditional interaction networks. The function is called via

The input data can be any biparental genotype data containing at least two genotype states. Genotype data from the **netmap** function can also be inserted here. This function can be used to reveal the high-dimensional linkage disequilibrium interactions network for polyploid genotype data. We note that this function also handles missing observations.

As an example we implement this function to the *Arabidopsis thaliana* genotype data that are derived from a RIL cross between Columbia-0 (Col-0) and Cape Verde Island (Cvi-0), where 367 individual plants were genotyped across 90 genetic markers (Simon et al., 2008). The data contain 3 possible genotype states: A (homozygous) denoted by 0, H (heterozygous) denoted by 1, and B (homozygous) denoted by 2.

```
R> data(CviCol)
R> out <- netsnp(CviCol)
R> sel <- selectnet(out)</pre>
```

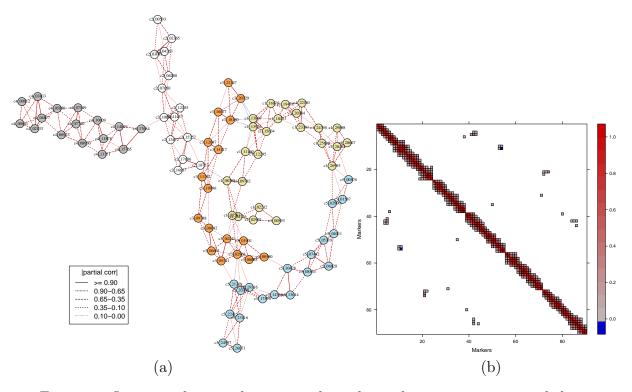


Figure 5: Intra— and inter—chromosomal conditional interactions network between 90 markers across the A. thaliana genome. (a) Each color corresponds to a different chromosome: yellow, white, orange, gray, and blue represent chromosomes 1 to 5, respectively. Different edge colors show positive — and negative — values of partial correlations. (b) Represents the related partial correlation matrix.

Figure 5 shows that in the  $Cvi \times Col$  population our method finds some transchromosomal regions that do interact. In particular, the bottom of chromosome 1 and the top of chromosome 5 do not segregate independently of each other. Besides this, interactions between the tops of chromosomes 1 and 3 involve pairs of loci that also do not segregate independently. Bikard et al. (2009) studied this genotype extensively. They reported that the first interaction (chr 1 and 5) causes arrested embryo development, resulting in seed abortion, whereas the latter interaction causes root growth impairment. In addition to these two regions, we have discovered a few other trans-chromosomal interactions in the A.thaliana genome. In particular, two adjacent markers, c1-13869 and c1-13926 in the middle of the chromosome 1, interact epistatically with the adjacent markers, c3-18180 and c3-20729, at the bottom of chromosome 3. The sign of their conditional correlation score is negative, indicating strong negative epistatic selection in  $F_2$  population. These markers therefore seem evolutionarily favored to come from the two different  $F_0$  grandparents. This suggests some positive effect of the interbreeding of the two parental lines: it could be that the paternal-maternal combination at these two loci protects against some underlying disorder, or that it actively enhances the fitness of the resulting progeny. Regarding the computational time, this example was run in 4 minutes on an Intel i7 laptop with 16 GB RAM.

### netphenogeno

Complex genetic traits are influenced by multiple interacting loci, each with a possibly small effect. Thus, to overcome the limitations of traditional analysis, such as single-locus association analysis (looking for main effects of single marker loci), multiple testing, and QTL analysis, we have developed a method based on discrete graphical models to investigate the simultaneous associations between phenotypes and SNPs. Our method is different and allows for a more powerful interpretation of the findings than the traditional methods, which only analyze few SNPs at a time. This is because we adjust for the effect of all other SNPs and phenotypes while measuring the pairwise associations between them. Statistically speaking this implies inferring conditional dependence relationships between variables in the data.

Networks or graphs are used to model interactions. In a genotype-phenotype network, nodes are either phenotypes or SNPs and edges are direct associations after adjustments. In our modeling framework, a genotype-phenotype network is a complex network made up of interactions among: (i) genetic markers, (ii) phenotypes (e.g. disease), and (iii) between genetic markers and phenotypes. It may happen that some phenotypes are associated with a SNP marker, or with multiple SNP markers. We remark that due to the conditional dependence feature often we reduce the number of possible SNPs from hundreds of SNPs to fewer SNPs.

It is of great interest to geneticists and biologists to discover such graph structure. The first problem with this is that such data consist of mixed ordinal-and-continuous variables, where the markers have ordinal values, and phenotypes (disease) can be measured in continuous or discrete scales. We deal with mixed variables by means of copula. A second issue relates to the high-dimensional setting of the data, where thousands of genetic markers are measured across a few samples; we are dealing with inferring potentially large networks with only few biological samples. Fortunately, biological networks are sparse, in the sense that only few elements interact with each other. This sparsity assumption is incorporated into our statistical methods based

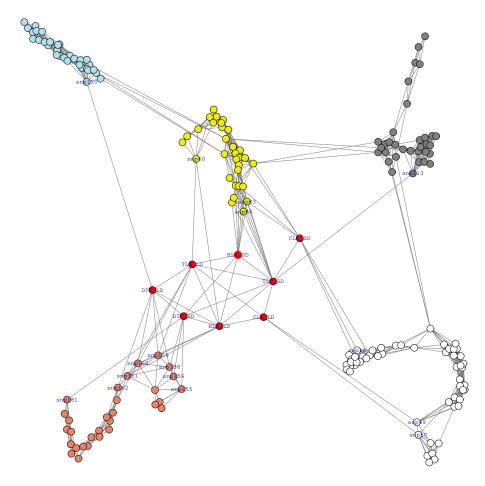


Figure 6: Genotype–phenotype conditional interaction networks in A. thaliana. Red nodes show phenotypes; white, yellow, gray, blue, and brown colors stand for chromosomes 1 to 5, respectively. Phenotypes measured in long days (TLN-LD, RLN-LD, DTF-LD) conditionally dependent on a region on top of chromosome 5 given the other locations in the genome. CLN-LD is correlated to a region in chromosome 1. Phenotypes measured in short days are linked mostly to chromosomes 1, 2, and 5.

on penalized graphical models.

The proposed method is implemented in the netphenogeno() function. The function can be called with the following arguments:

```
R> netphenogeno(data, method = "gibbs", rho = NULL,
+ n.rho = NULL, rho.ratio = NULL, ncores = "all",
+ em.iter = 5, em.tol = .001, verbose = TRUE)
```

The netphenogeno returns an object of S3 class type "netgwas". The functions plot, print and summary work with the object "netgwas". The input data can be

an  $(n \times p)$  matrix or a data.frame where n is the sample size and p is the dimension that includes marker data and phenotype measurements. One may consider including more columns, like environmental variables.

The argument method determines the type of methods, "gibbs", "approx", or "npn". Option "gibbs" is based on the Gibbs sampler within Gaussian copula graphical models. It is designed for small data (p < 1500). Option "approx" is based on the Gaussian copula graphical model with the approximation approach, and "npn" is based on semi-parametric Gaussian copula, nonparanormal. The last two methods are faster compare with "gibbs". In particular "npn" is designed for very large datasets. All the three methods are designed for exploring the conditional independence network for ordinal data, non-Gaussian continuous data, and mixed discrete-and-continuous data.

In the argument rho a sequence of decreasing positive numbers can be provided to control the regularization. Typical usage is to leave the input rho = NULL and have the program compute its own rho sequence based on n.rho and rho.ratio. The program automatically sets up a sequence of n.rho regularization parameters and estimates the graph path. Option ncores determines the number of cores to use for the calculations. Using ncores = "all" automatically detects the number of available cores and runs the computations in parallel on the available cores minus one. The code is memory-optimized, using the sparse matrix data structure when estimating and storing full regularization paths for large data sets.

## Genotype-phenotype network in A.thaliana

We have applied our algorithm to a public Arabidopsis thaliana dataset, where the accession Kend-L (Kendalville-Lehle; Lehle-WT-16-03) is crossed with the common lab strain Col (Columbia) (Balasubramanian et al., 2009). The resulting lines were taken through six rounds of selfing without any intentional selection. The resulting 282 KendC (Kend-L × Col) lines were genotyped at 181 markers. Flowering time was measured for 197 lines of this population both in long days, which promote rapid flowering in many A. thaliana strains, and in short days. Flowering time was measured using days to flowering (DTF) as well as the total number of leaves (TLN), partitioned into rosette and cauline leaves. In total, eight phenotypes were measured, namely days to flowering (DTF), cauline leaf number (CLN), rosette leaf number (RLN), and total leaf number (TLN) in long days (LD), and DTF, CLN, RLN, and TLN in short days (SD). Thus, the final dataset consists of 197 observations for 189 variables (8 phenotypes and 181 genotypes - SNP markers).

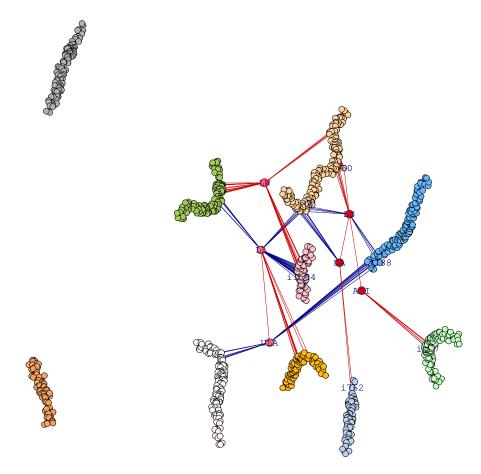


Figure 7: Genotype—phenotype networks for 1106 SNP markers and 6 phenotypes in mazie NAM population, where flowering and leaf traits are shown in • and •. SNPs are shown on chromosome 1 as •, on chromosome 2 as •, on chromosome 3 as •, on chromosome 4 as •, on chromosome 5 as •, on chromosome 6 as •, on chromosome 7 as •, on chromosome 8 as •, on chromosome 9 as •, and on chromosome 10 as •. Different edge colors show positive — and negative — values of partial correlations.

```
R> data(thaliana)
R > head(thaliana, n = 3)
    DTF_LD CLN_LD RLN_LD TLN_LD DTF_SD CLN_SD RLN_SD TLN_SD snp1 snp2
[1,] 17.58
             3.42
                   12.17
                           15.58
                                  56.92
                                          12.42
                                                  50.92
                                                         63.33
                                                                   2
                                                                        2
                                                                   0
[2,] 17.00
             2.58
                   11.33
                           13.92
                                  53.33
                                           8.42
                                                 41.58
                                                         50.00
                                                                        0
[3,] 27.50
            8.08
                   26.92
                                  69.17
                                          15.17
                                                  66.92
                                                         82.08
                                                                   2
                                                                       NA
                           35.00
```

... snp181
[1,] 2
[2,] 2
[3,] 0

```
R> set.seed(12)
R> out <- netphenogeno(thaliana)
R> sel <- selectnet(out)
# Steps to visualize the network
R> cl <- c(rep("red", 8), rep("white",56), rep("yellow2",31),
            rep("gray", 33), rep("lightblue2", 31), rep("salmon2", 30))
R> id <- c("DTF_LD", "CLN_LD", "RLN_LD", "TLN_LD", "DTF_SD", "CLN_SD",
          "RLN_SD", "TLN_SD", "snp16", "snp49", "snp50", "snp60", "snp83",
          ","snp86", "snp113", "snp150", "snp155", "snp159", "snp156",
          "snp161", "snp158", "snp160", "snp162", "snp181")
R> set.seed(1)
R> plot(sel, vis= "interactive", n.mem= c(8,56,31,33,31,30),
        vertex.color= cl, label.vertex= "some", sel.nod.label= id,
        edge.color= "gray", w.btw= 200, w.within= 20,
+
        tk.width = 900, tk.height = 900)
```

Figure 6 shows the genotype-phenotype network for this population. The network reveals those SNP markers that are directly correlated with the flowering phenotypes. For example in long days, the phenotype days to flowering (DTF-LD) is directly associated with markers snp158, snp159, snp160, and snp162 on chromosome 5 which have assay IDs 44607857, 44606159, 44607242, and 44607209. Balasubramanian et al. (2009) have reported that the phenotypes DTF-LD is associated with markers from snp158 to snp162 with assay ID 44607857 to 44607209. Our finding regarding DTF-LD phenotype is consistent with their finding; however, our result has a stronger interpretation compared to original findings, because we control for all possible effects. We find that snp161 does not show any association with DTF-LD after adjustment, but snp159, snp160 and snp162 on chromosome 5 do show an association with DTF-LD, even after taking into account the effect of all other SNPs and phenotypes. We remark that our method reduces the number of candidate SNPs and gives a small set of much more plausible ones. Balasubramanian et al. (2009) have reported that snp158- snp162 are associated with TLN-LD phenotype, But our method reduces this set to the smaller set of snp159-snp161 after we control the effect of all other SNPs. It avoids false positives that can occur when using traditional QTL analysis. Furthermore, the association between phenotype CLN-LD and markers snp49 and snp50 has remained undetected by the use of traditional QTL analysis, potentially the result of a lack of power. Our method goes beyond the bivariate testing of individual SNPs that looks only marginal association. Instead we use a multivariate approach which includes

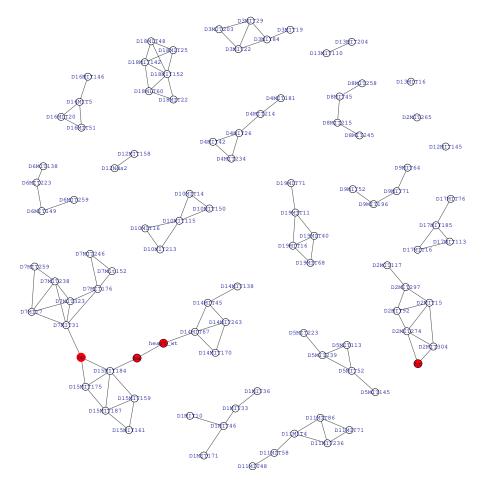


Figure 8: Conditional independence network between phenotypes blood pressure (bp), heart rate (hr), body weight (bw), and heart weight (heart-wt) and genetic map in mice.

all the SNPs and phenotypes. In this regards, Balasubramanian et al. (2009) have reported that the TLN-SD phenotype is associated with a region in chromosome 5, whereas our proposed method shows that there is no direct link between the TLN-SD phenotype and a region in chromosome 5; TLN-SD is connected to chromosome 5 through the DTF-SD phenotype. This example was run in about 4 minutes on an Intel i7 laptop with 16 GB RAM.

#### Genotype-phenotype network in maize

The genotypic and phenotypic maize data used in this paper were downloaded from www.panzea.org. The data comprised three datasets: a genotype data, and two

phenotypes datasets from the flowering time (Buckler et al., 2009) and the leaf architecture (Tian et al., 2011). The SNP data included 1106 genetic markers for 194 diverse maize recombinant inbred lines, which were derived from a cross between B73 and B97 from the maize Nested Association Mapping (NAM) populations. The 194 maize lines were scored for their flowering time using days to silking (DS), days to anthesis (DA), and the anthesis-silking interval (ASI) phenotypes. The leaf related traits such as upper leaf angle (ULA), leaf length (LL) and leaf width (LW) were also measured for all 194 maize lines.

Figure 7 constructs genotype—phenotype networks between the 6 phenotypes and 1106 SNPs. Regarding flowering time: five SNPs on chromosome 1 (from i140 until i144) directly affect both DA and DS traits after removing the effect of other variables. In addition, SNPs from i60 until i64 on chromosome 1 and SNPs on the beginning of chromosome 2 (i188 until i191) regulates DS after adjusting for the effect of remaining variables. DA is associated with two SNPs (i762 and i763) of chromosome 7 and ASI is connected to chromosome 8 (i877 until i883) given the remaining variables. Two SNPs i188 and i189 on chromosome 2 control both ULA and DS traits. The two leaf related traits, ULA and LL, are linked together, but not to the LW. Three SNPs i1064, i1062, and i1080 are yet associated to both LL and LW traits after adjustments. ULA is directly connected to six locations (from i569 until i574) on chromosome 5. The LL is connected to different locations on chromosomes 1, 3, 9, and 10. Chromosomes 4 and 6 are isolated with respect to the studied flowering time and leaf architecture traits.

### Genotype-phenotype network in mice

To better understand the genetic basis of essential hypertension, we reconstruct a conditional independence network between genotypes and phenotypes on an available data in mice. The data are from an intercross between BALB/cJ and CBA/CaJ mouse strains (Sugiyama et al., 2002). Only male offspring were considered. The data consist of 93 SNP markers across the genome, and four phenotypes: blood pressure (bp), heart rate (hr), body weight (bw), and heart weight (heart-weight), as measured for 163 individuals. Data are shown as follows:

```
R> data(bp)
R> head(bp, n = 3)
```

	bp	hr	bw	heart_wt	D1MIT171	D1MIT46	D1MIT10	 D19MIT11	D19MIT71
1	104	517	37.0	133	0	0	0	 2	2
2	108	690	38.9	135	0	1	1	 0	0
3	115	653	43.8	159	0	2	2	 0	0

There are 3 possible genotype states: CC (homozygous) denoted by 0, CB (heterozygous) denoted by 1, and BB (homozygous) denoted by 2. In data frame bp, the genotypes are ordinal variables, whereas the phenotypes are continuous. The data also include some missing observations.

Figure 8 shows the conditional dependence network between the genetic markers across the mice genome and the phenotypes: blood pressure (bp), heart rate (hr), body weight (bw), and heart weight (heart-wt). The conditional independence network in Figure 8 explores genomic regions that regulate blood pressure, heart rate, and heart weight. We identified that two loci "D15MIT184" and "D15MIT175" on chromosome 15 and "D7MIT31" on chromosome 7 are yet associated with blood pressure (bp) after adjusting for the effects of other SNPs and phenotypes.

Our findings regarding blood pressure are consistent with Sugiyama et al. (2002), as we find it is associated with loci in chromosome 7 and 15. However, for the heart rate phenotype they reported association with loci on chromosomes 2 and 15, whereas in our findings only loci on chromosome 2 are associated with the heart rate phenotype. This example was run in about 2 minutes on an Intel i7 laptop with 16 GB RAM.

Computational timing. Figure 9 shows computational timing of netgwas for different number of variables p and different sample sizes n. In this Figure, we reported computational timing in minutes for the genetic map construction, which includes graph estimation plus the ordering algorithms. Note that the other two functions netsnp() and nethenogeno() include only the graph estimation, so we have only considered netmap function to cover the computational aspect of the netgwas package. For the simulated data, we generated p = 1000, 2000, 3500, 5000 markers using simRIL function, which evenly are distributed across 10 chromosomes, for different individuals n = 100, 200, 300. Figure 9 shows that computational time is not affected by sample size n and is roughly proportional to  $p^3$ , as long as  $p \times \max\{n, p\}$  elements can be stored in memory. The reported timing is based on the result from a computer with an Intel Core i7–6700 CPU and 32GB RAM.

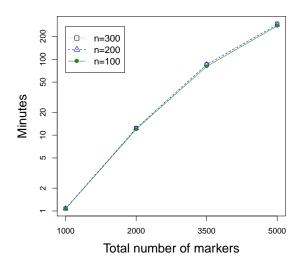


Figure 9: Computational time of map construction in **netgwas** for various simulated data with different combinations of individuals n and variables p, where markers were distributed evenly across 10 LGs.

## 5 Conclusion

We have presented the **netgwas** package, which is designed based on undirected graphical models, to accomplish three fundamental goals in genetics: linkage map construction, reconstruction of linkage disequilibrium networks, and exploration of high-dimensional genotype-phenotype (disease) networks. The novelty of the underlying methodology is the use of graphical model to accomplish these tasks in an unified way. Moreover, the netgwas package can deal with species of any ploidy level. Due to the fact that we adjust for the effect of all other variables while measuring the pairwise associations, this allows us for a more powerful interpretation of our findings than classical approaches, which tests only the marginal associations.

The package implements the methods developed by Behrouzi and Wit (2018) and Behrouzi and Wit (2017) for linkage map construction and inferring of conditional independence networks for non-Gaussian data, discrete data, and mixed discrete-and-continuous data. We note that reproducibility of our results and all the example data used to illustrate the package is supported by the open-source R package **netgwas**. We will in the future maintain and develop the package further, and extend our package to calculate the genetic distances in the linkage map construction for polyploid cases, and extend the package for multi-parental map construction.

## References

- Balasubramanian, S., C. Schwartz, A. Singh, N. Warthmann, M. C. Kim, J. N. Maloof, O. Loudet, G. T. Trainer, T. Dabi, J. O. Borevitz, et al. (2009). Qtl mapping in new arabidopsis thaliana advanced intercross-recombinant inbred lines. *PLoS One* 4(2), e4318.
- Behrouzi, P. and E. C. Wit (2017). Detecting epistatic selection with partially observed genotype data by using copula graphical models. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*.
- Behrouzi, P. and E. C. Wit (2018). De novo construction of polyploid linkage maps using discrete graphical models. *Bioinformatics*.
- Bikard, D., D. Patel, C. Le Metté, V. Giorgi, C. Camilleri, M. J. Bennett, and O. Loudet (2009). Divergent evolution of duplicate genes leads to genetic incompatibilities within a. thaliana. *Science* 323(5914), 623–626.
- Bourke, P. M., G. van Geest, R. E. Voorrips, J. Jansen, T. Kranenburg, A. Shahin, R. G. Visser, P. Arens, M. J. Smulders, and C. Maliepaard (2018). polymapr–linkage analysis and genetic map construction from f1 populations of outcrossing polyploids. *Bioinformatics* 1, 7.
- Broman, K. W. (2009). A brief tour of r/qtl. Disponivel em:; http://www. rqtl. org/tutorials/rqtltour. pdf.
- Broman, K. W., H. Wu, Ś. Sen, and G. A. Churchill (2003). R/qtl: Qtl mapping in experimental crosses. *Bioinformatics* 19(7), 889–890.
- Buckler, E. S., J. B. Holland, P. J. Bradbury, C. B. Acharya, P. J. Brown, C. Browne, E. Ersoz, S. Flint-Garcia, A. Garcia, J. C. Glaubitz, et al. (2009). The genetic architecture of maize flowering time. *Science* 325(5941), 714–718.
- Buzdugan, L., M. Kalisch, A. Navarro, D. Schunk, E. Fehr, and P. Bühlmann (2016). Assessing statistical significance in multivariable genome wide association analysis. *Bioinformatics* 32(13), 1990–2000.
- Cuthill, E. and J. McKee (1969). Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th national conference*, pp. 157–172. ACM.
- Fan, J. and R. Li (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association* 96 (456), 1348–1360.

- Friedman, J., T. Hastie, and R. Tibshirani (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* 9(3), 432–441.
- Green, P. J. (1990). On use of the em for penalized likelihood estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, 443–452.
- Hackett, C. A., B. Boskamp, A. Vogogias, K. F. Preedy, and I. Milne (2017). Tetraploidsnpmap: software for linkage analysis and qtl mapping in autotetraploid populations using snp dosage data. *Journal of Heredity* 108(4), 438–442.
- He, Q. and D.-Y. Lin (2010). A variable selection method for genome-wide association studies. *Bioinformatics* 27(1), 1–8.
- Hedrick, P. W. (1987). Gametic disequilibrium measures: proceed with caution. *Genetics* 117(2), 331–341.
- Hoggart, C. J., J. C. Whittaker, M. De Iorio, and D. J. Balding (2008). Simultaneous analysis of all snps in genome-wide and re-sequencing association studies. *PLoS* genetics 4(7), e1000130.
- Hsieh, C.-J., I. S. Dhillon, P. K. Ravikumar, and M. A. Sustik (2011). Sparse inverse covariance matrix estimation using quadratic approximation. In *Advances in neural information processing systems*, pp. 2330–2338.
- Huang, B. E., R. Shah, A. W. George, et al. (2012). dlmap: An r package for mixed model qtl and association analysis. *Journal of Statistical Software* 50(6), 1–22.
- Klasen, J. R., E. Barbez, L. Meier, N. Meinshausen, P. Bühlmann, M. Koornneef, W. Busch, and K. Schneeberger (2016). A multi-marker association method for genome-wide association studies without the need for population structure correction. *Nature communications* 7, 13299.
- Lander, E. S., P. Green, J. Abrahamson, A. Barlow, M. J. Daly, S. E. Lincoln, and L. Newburg (1987). Mapmaker: an interactive computer package for constructing primary genetic linkage maps of experimental and natural populations. *Genomics* 1(2), 174–181.
- Lauritzen, S. (1996). Graphical Models, Volume 17. Oxford University Press, USA.
- Lincoln, S. E. and E. S. Lander (1992). Systematic detection of errors in genetic linkage data. *Genomics* 14(3), 604–610.
- Liu, H., F. Han, M. Yuan, J. Lafferty, L. Wasserman, et al. (2012). High-dimensional semiparametric gaussian copula graphical models. *The Annals of Statistics* 40(4), 2293–2326.

- Mangin, B., A. Siberchicot, S. Nicolas, A. Doligez, P. This, and C. Cierco-Ayrolles (2012). Novel measures of linkage disequilibrium that correct the bias due to population structure and relatedness. *Heredity* 108(3), 285.
- Margarido, G., A. Souza, and A. Garcia (2007). Onemap: software for genetic mapping in outcrossing species. *Hereditas* 144(3), 78–79.
- Massa, A. N., N. C. Manrique-Carpintero, J. J. Coombs, D. G. Zarka, A. E. Boone, W. W. Kirk, C. A. Hackett, G. J. Bryan, and D. S. Douches (2015). Genetic linkage mapping of economically important traits in cultivated tetraploid potato (solanum tuberosum l.). *G3: Genes, Genomes, Genetics* 5(11), 2357–2364.
- Panagiotou, O. A., J. P. Ioannidis, and G.-W. S. Project (2011). What should the genome-wide significance threshold be? empirical replication of borderline genetic associations. *International journal of epidemiology* 41(1), 273–286.
- Rakitsch, B., C. Lippert, O. Stegle, and K. Borgwardt (2012). A lasso multi-marker mixed model for association mapping with population structure correction. *Bioinformatics* 29(2), 206–214.
- Simon, M., O. Loudet, S. Durand, A. Bérard, D. Brunel, F. Sennesal, M. Durand-Tardif, G. Pelletier, and C. Camilleri (2008). Qtl mapping in five new large ril populations of arabidopsis thaliana genotyped with consensus snp markers. *Genetics* 178, 2253–2264.
- Sugiyama, F., G. A. Churchill, R. Li, L. J. Libby, T. Carver, K.-i. Yagami, S. W. John, and B. Paigen (2002). Qtl associated with blood pressure, heart rate, and heart weight in cba/caj and balb/cj mice. *Physiological genomics* 10(1), 5–12.
- Taylor, J. and D. Butler (2017). R package asmap: efficient genetic linkage map construction and diagnosis. *Journal of Statistical Software* 79(6).
- Taylor, J., A. Verbyla, et al. (2011). R package wgaim: Qtl analysis in bi-parental populations using linear mixed models. *Journal of Statistical Software* 40(7), 1–18.
- Tian, F., P. J. Bradbury, P. J. Brown, H. Hung, Q. Sun, S. Flint-Garcia, T. R. Rocheford, M. D. McMullen, J. B. Holland, and E. S. Buckler (2011). Genome-wide association study of leaf architecture in the maize nested association mapping population. *Nature genetics* 43(2), 159.
- Wang, H., F. A. van Eeuwijk, and J. Jansen (2016). The potential of probabilistic graphical models in linkage map construction. *Theoretical and Applied Genetics*, 1–12.

- Welter, D., J. MacArthur, J. Morales, T. Burdett, P. Hall, H. Junkins, A. Klemm, P. Flicek, T. Manolio, L. Hindorff, et al. (2013). The nhgri gwas catalog, a curated resource of snp-trait associations. *Nucleic acids research* 42(D1), D1001–D1006.
- Wu, R., C.-X. Ma, I. Painter, and Z.-B. Zeng (2002). Simultaneous maximum likelihood estimation of linkage and linkage phases in outcrossing species. *Theoretical population biology* 61(3), 349–363.
- Zych, K., Y. Li, J. K. van der Velde, R. V. Joosen, W. Ligterink, R. C. Jansen, and D. Arends (2015). Pheno2geno-high-throughput generation of genetic markers and maps from molecular phenotypes for crosses between inbred strains. *BMC bioinformatics* 16(1), 51.