kangar00: Kernel Approaches for Nonlinear Genetic Association Regression

Stefanie Friedrichs and Juliane Manitz

2018-10-23

Introduction

Methodological Background

The genetic information collected in genome-wide association studies (GWAS) is represented by the genotypes of various single-nucleotid polymorphisms (SNPs). Testing biological meaningful SNP sets is a successful strategy for the evaluation of GWAS data, as it may increase power as well as interpretation of results. Via mapping of SNPs to genes forming a network, association between pathways and disease risk can be investigated.
Kernel methods are particularly well suited to cope with the challenges connected to the analysis of large SNP sets from GWAS data. They do not require to model a direct functional relationship between SNPs and effects, while at the same time can deal with high-dimensional data and allow for straightforward incorporation of covariates. The model for a logistic kernel machine regression of a pathway on a binary outcome is given by

$$\text{logit}(P(y_i = 1|x_i, z_i)) = x_i^t\beta + h(z_i)(1)$$

where y_i denotes the case or control status of individual i, x_i is the vector including informative covariates (such as age, sex, etc.) and z_i represents the genotypes of individual i. β is the regression coefficients for the parametric part of the model, while $h(.)$ denotes an unknown function, non-parametrically incorporating the pathway's influence. The intercept is assumed to be included in x_i. For more details see Liu et al (2008).

Different kernels have been proposed that convert the genomic information of two individuals into a quantitative value reflecting their genetic similarity. This package includes the linear kernel as well as two more advanced kernels, adjusting for size bias in the number of SNPs and genes in a pathway or incorporating the network structure of genes within the pathway, respectively. The kernel functions are described in more detail in the instructions below.

A variance component test, constructed around the similarity matrix, can be used to evaluate a pathway's influence on disease risk. In `kangar00` p-values can be calculated with the Satterthwaite approximation or Davies method as described in Schaid (2010) and Davies (1980), respectively.

Package Overview and S4 Object Classes

In order to make the described functionalities of `kangar00` available load the package

```r
library(kangar00)
```

The package `kangar00` makes use of four S4 object classes:

- `pathway`,
- `GWASdata`
- `kernel` respresents a kernel matrix calculated for a pathway.
- `lkmt` captures the resuls from the variance component test.

In the following, methods for data extraction and preparation, calculation of kernel matrices and the implementation of the variance component test are described in more detail.

Linear Kernel Machine Test

- `show()` displays basic information on lkmt object
- `summary()` generates a lkmt object summary including the used kernel, pathway and the test result

Data Extraction and Preparation

Pathways

The `kangar00` package offers several functions for data extraction from internet databases. In the following they will be explained using the Circadian rhythm pathway as an example. In the KEGG database (Kanehisa et al 2014) this pathway is identified with the id *hsa04710*.

The function `pathway_info()` can use this id to create a table listing all genes included in Circadian rhythm. For each gene the startpoint, endpoint and the chromosome are listed.

```
pathway_info('hsa04710')
```

```
## An object of class pathway_info
## Number of pathways: 1
## Number of genes: 30
## First six rows:
##    pathway gene_start  gene_end chr    gene
## 1 hsa04710   13276652  13387266  11   ARNTL
## 2 hsa04710    4979116   4985323   3 BHLHE40
## 3 hsa04710   26120026  26125127  12 BHLHE41
## 4 hsa04710  101354033 101557321  10    BTRC
## 5 hsa04710   55427903  55547138   4   CLOCK
## 6 hsa04710  207529737 207603431   2   CREB1
```

Gene membership is obtained directly from KEGG, while startpoints, endpoints and chromosome information is extracted from Ensembl (Cunningham et al 2015). The database is accessed via the function `getBM()` in the biomaRt package. This means that the gene boundaries given will equal the current build used in Ensemble. An internet connection is required for this step.

Pathway object

In `kangar00` all information on a specific pathway is combined in a S4 object of class `pathway`. It includes the slots

Slot	Description	Values
`id`	A pathway ID as used in KEGG.	
`adj`	A network adjacency matrix of dimension equal to the number of genes	1 interaction, 0 otherwise
`sign`	A numeric vector indicating the interaction type for each link in the interaction network for the pathway	1 activation, -1 inhibition

The following example creates a new pathway object, to which gene-interaction information has yet been added

```
pathw_shell <- pathway(id='hsa04710', adj=matrix(0), sign=as.vector(matrix(0)[matrix(0)!=0]))
str(pathw_shell)
```

```
## Formal class 'pathway' [package "kangar00"] with 3 slots
##   ..@ id  : chr "hsa04710"
##   ..@ adj : num [1, 1] 0
##   ..@ sign: num(0)
```

Network Matrix

The gene-gene interactions within pathways are represented by a network matrix. This quadratic matrix is of dimension equal to the number of genes in the corresponding pathway. It includes entries equal to 1 (representing an activation interaction), -1 (denoting an inhibiting interaction) or 0 (no interaction).

A network matrix can be created using the function `get_network_matrix()`. Gene interaction information for a specific pathway is extracted from the KEGG database, which can be accessed via the function `retrieveKGML()` from the `KEGGgraph` package. An internet connection is required for this step.

```
pathw <- get_network_matrix(pathw_shell, directed=FALSE)
```

```
## [1] "hsa04710 has both: Activation/Inhibition edges and edges without type!"
```

```
str(pathw)
```

```
## Formal class 'pathway' [package "kangar00"] with 3 slots
##   ..@ id  : chr "hsa04710"
##   ..@ adj : num [1:31, 1:31] 0 0 0 0 0 0 0 0 0 1 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:31] "ARNTL" "NPAS2" "CLOCK" "RORA" ...
##   .. .. ..$ : chr [1:31] "ARNTL" "NPAS2" "CLOCK" "RORA" ...
##   ..@ sign: num [1:116] -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
```

A KEGG XML file for the pathway with ID *hsa04710* is downloaded and saved in the working directory. Furthermore, the function converts the data into a network matrix and adds it to the given pathway object. The updated pathway object is returned. The user can specify whether the gene-interaction matrix should be given directed (`directed=TRUE`) or undirected (`directed=FALSE`).

Pathway Object Helper Functions

Several helper functions are available to handle objects of S4 class `pathway`. The functions `show()` displays the pathway object briefly and `summary()` generates a pathway object summary including basic network properties.

```
summary(pathw)
```

```
## An object of class pathway
##
## 31 nodes and 58 links; 35 activations and  inhibitions.
##
## Density:      0.1247312
## Average degree:       3.741935
## Inhibition degree:    2.258065
## Diameter:         4
## Transitivity:          0.08083832
## Signed transitivity:   0.06887755
```

A more comprehensive list of pathway network properties can be generated using `analyze()`.

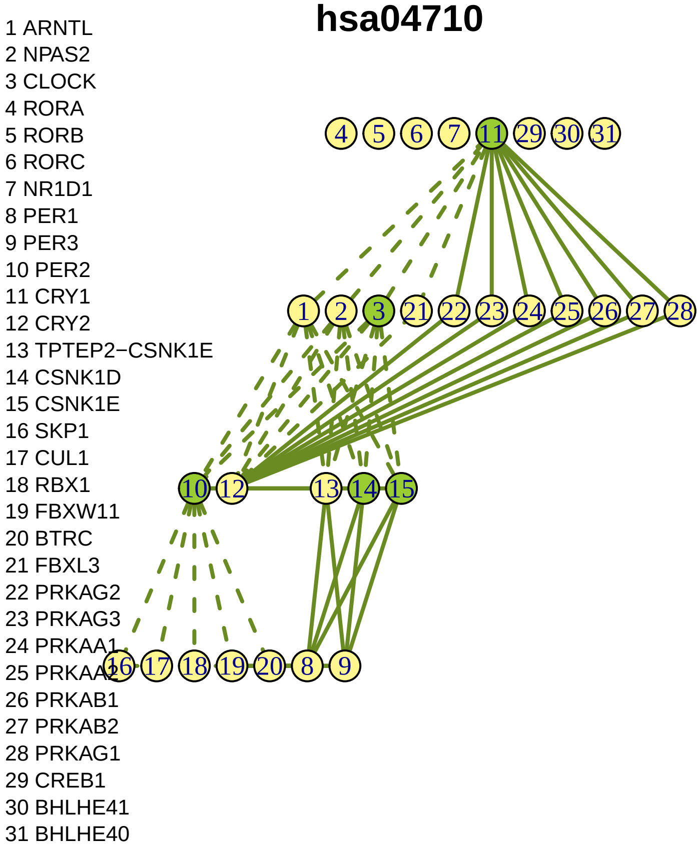
The helper function `get_genes()` extracts the gene names in a pathway and returns a vector containing character elements of gene names

```
get_genes(pathw)
```

```
##  [1] "ARNTL"         "NPAS2"         "CLOCK"         "RORA"
##  [5] "RORB"          "RORC"          "NR1D1"         "PER1"
##  [9] "PER3"          "PER2"          "CRY1"          "CRY2"
## [13] "TPTEP2-CSNK1E" "CSNK1D"        "CSNK1E"        "SKP1"
## [17] "CUL1"          "RBX1"          "FBXW11"        "BTRC"
## [21] "FBXL3"         "PRKAG2"        "PRKAG3"        "PRKAA1"
## [25] "PRKAA2"        "PRKAB1"        "PRKAB2"        "PRKAG1"
## [29] "CREB1"         "BHLHE41"       "BHLHE40"
```

Furthermore, `sample_genes()` randomly selects effect genes in a pathway and returns a vector of length no with vertex id's of sampled genes. In addition, the function `plot()` can be used to visualize the pathway as igraph object. In this context `pathway2igraph()` converts a pathway object into an igraph object with edge attribute sign

```
sample5 <- sample_genes(pathw, no = 5)
plot(pathw, highlight.genes = sample5)
```



SNP Positions

The package kangar00 also offers a function to download current positions of the SNPs available in your GWAS dataset from the Ensembl database.

The function snp_info() will take a vector of rs-numbers and returns a data frame with the corresponding base pair positions.

```
snp_info("rs234")
```

```
## An object of class snp_info
## Number of SNPs: 1
##   chr  position   snp
## 1   7 105920689 rs234
```

The SNP positions are extracted from the Ensembl database and thus equal the current build used on the website. The database is accessed via the function getBM() from the package biomaRt. This requires an internet connection.

Pathway Annotation

In order to define SNP sets associated to a specific pathway, the function get_anno() can be used. Input arguments are a pathway_info as well as a snp_info object. The package sqldf is used.

```
rs10243170_info <- snp_info("rs10243170")
hsa04022_info <- pathway_info('hsa04022')
get_anno(rs10243170_info, hsa04022_info)
```

```
##    pathway  gene chr        snp position
## 1 hsa04022 ADCY1   7 rs10243170 45676168
```

The return argument is a data frame listing all SNPs that lie inside the boundaries of one or more genes in the pathway. That means that genes can appear several times, depending on the number of SNPs mapped to them. A SNP can and will be mapped to multiple genes if they overlap.

GWAS data

Genome-wide association study data from a case control study is basis for a test of a pathway's influence on a disease risk with the logistic kernel machine test.

GWASdata Object

In kangar00, all needed information is captured by the S4 GWASdata object, which can be constructed by

```
my_gwas <- GWASdata(pheno=pheno, geno=geno, anno=anno, desc="study xy")
```

```
## Warning: closing unused connection 5 (http://togows.dbcls.jp/entry/pathway/
## hsa04022/genes)
```

It includes the slots which have to fullfill certain requirements

- geno: Genotype data for each individual.
  - Genotype data needs to be a matrix with one line per individual and one column for each SNP.
  - Rownames give ID numbers for the individuals while columnames give the rs-numbers corresponding to the SNPs genotyped in the study.

  - Note that missing values are not allowed and SNPs with missing genotypes have to be imputed or excluded from the sample prior to creation of the GWASdata object.

- pheno: Phenotype data for each individual.
  - Phenotypes need to be given in a data.frame with the first column including the individual IDs as in the genotype sample.
  - Further columns can contain informative covariates (such as age, sex, ...) to be used in the logistic regression model.

- anno: Annotation of study SNPs to pathways.
  - This data.frame defines the SNP set representing a specific pathway. It can be created using the function get_anno().

- desc: A character value describing the data can be added to the GWASdata object. This could for instance be the name of the study.

GWASdata Helper Functions

Available methods for objects of class GWASdata are show(), summary() and GeneSNPsize(). The function GeneSNPsize() results in a data frame of pathway names with numbers of snps and genes in each pathway, which is also displayed by summary().

```
summary(my_gwas)
```

```
## An object of class GWASdata from study xy
##
## Phenotypes for 50 individuals:
##
##      pheno           sex           age
##  Min.   :0.00   Min.   :0.0   Min.   :20.00
##  1st Qu.:0.00   1st Qu.:0.0   1st Qu.:24.25
##  Median :1.00   Median :1.0   Median :36.00
##  Mean   :0.52   Mean   :0.6   Mean   :35.82
##  3rd Qu.:1.00   3rd Qu.:1.0   3rd Qu.:45.00
##  Max.   :1.00   Max.   :1.0   Max.   :55.00
##
## Genotypes:
##
## Total number of genes and SNPs per pathway:
##          genes SNPs
## hsa04020   180 1335
## hsa04022   167 1264
## hsa04024   200 1457
```

Calculation of Kernel Matrices

Once a GWASdata object is created, we can start to calculate kernel matrices to test a pathways influence on disease risk. The package kangar00 offers three different kernel functions to compute a similarity matrix for the individuals in analysis. They will be explained in the following.

Linear Kernel (Lin)

The linear kernel assumes additive SNP effects. It is calculated as

$$ZZ^t(2)$$

where Z denotes the genotype matrix (See also Liu et al, 2010). In `kangar00` a linear kernel can be created using the function `kernel_lin()`. It requires as arguments

- A `GWASdata` object containing the genotype information.
- A `pathway` object specifying the pathway to be tested.
- A value for argument `calculation` to decide how the kernel should be calculated. Options are `cpu` for calculation on cpu and `gpu` for gpu calculation.

```
K_lin <- lin_kernel(my_gwas, pathw, calculation='cpu')
summary(K_lin)
# Error: could not find function "lin_kernel"

K_lin <- calc_kernel(gwas, pathw, type='lin')
plot(K_lin)
```

will return a quadratic `matrix` of dimension equal to the number of individuals in the `GWASdata` object.

Size-adjusted Kernel (Sia)

The size-adjusted kernel takes into consideration the numbers of SNPs and genes in a pathway to correct for size bias. It is calculated as

$$K_{i,j} = exp(-\sqrt{\frac{1}{r_p}}\sum_g(\frac{||z_i^g - z_j^g||}{\mu_g k_g^{eff}})_g^\delta)(3)$$

Here z_i^g is the vector of individual i 's genotypes in gene g and r_p the number of genes in pathway p. Scaling parameters k_g^{eff}, μ^g and δ_g adjust for the number of genes in the pathway and the number of SNPs within these genes (for more details refer to Freytag et al. 2012).

A kernel of this type can be calculated using the function `kernel_sia()` with
the following arguments

- A `GWASdata` object containing the genotype information.
- A `pathway` object specifying the pathway to be tested.
- A value for argument `calculation` to decide how the kernel should be calculated. Currently only `cpu` for cpu calculation is available.

```
K_sia <- sia_kernel(gwas, p, calculation='cpu')
# Error: could not find function "sia_kernel"

#K_sia <- calc_kernel(gwas, pathw, type='sia')
```

will return a quadratic `matrix` of dimension equal to the number of individuals in the `GWASdata` object.

Network Kernel (Net)

The network kernel incorporates information about gene-gene interactions into the model. It is defined as

$$K = ZANA^tZ^t(4)$$

where matrix A maps SNPs to genes, N represents the underlying network structure, and Z is the genotype matrix. The network based kernel matrix for a pathway can be calculated with the function `kernel_net()`. Following arguments are needed

- A `GWASdata` object containing the genotype information.
- A `pathway` object specifying the pathway to be tested.
- A value for argument 'calculation' to decide how the kernel should be calculated.

```
K_net <- net_kernel(gwas, p, calculation='cpu')
# Error: could not find function "net_kernel"

#K_net <- calc_kernel(gwas, pathw, type='net')
# Error in A.star %*% N : non-conformable arguments
```

will return a quadratic `matrix` of dimension equal to the number of individuals in the `GWASdata` object.

Alternatively, kernel matrices can be calculated using the function `calc_kernel()`. Here the kernel type is specified via an additional argument `type`. It can be set to lin, sia or net.

```
K <- calc_kernel(my_gwas, pathw, type='lin', parallel='none')
```

This function will simply call the suitable kernel function as described above and therefore has the same output.

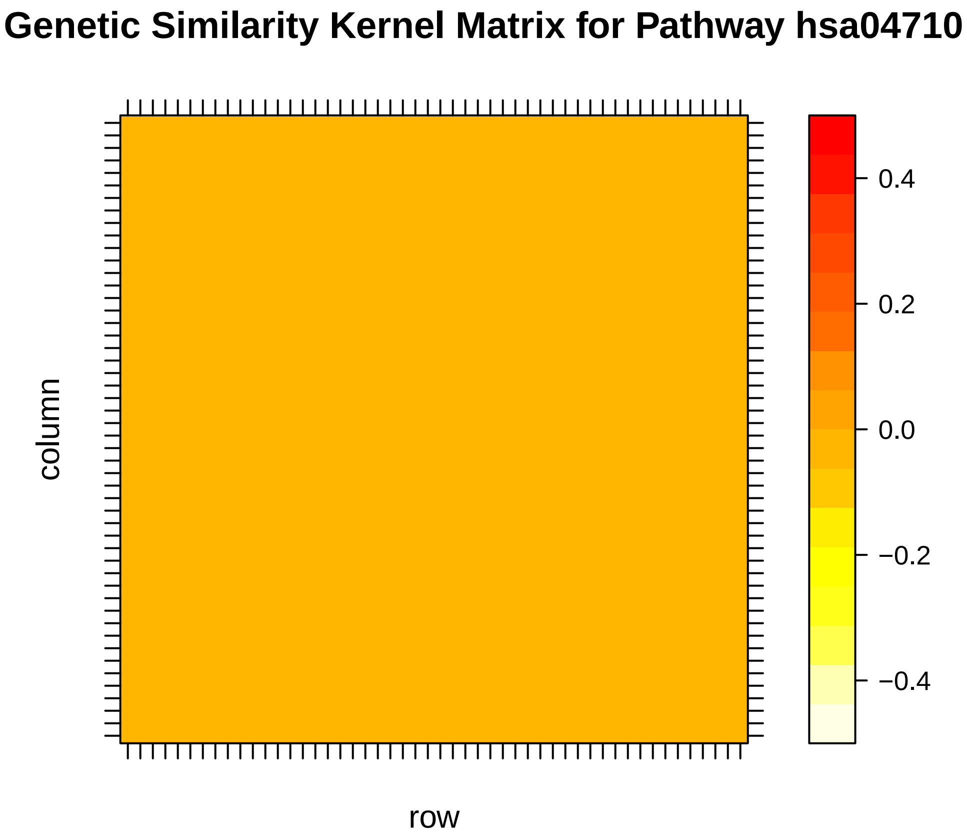
Standard methods are available for objects of S4 class `kernel`:

- `show()` displays the kernel object briefly.
- `summary()` generates a kernel object summary including the number of individuals and genes for the pathway.
- `plot()` creates an image plot of a kernel object.

```
summary(K)
```

```
## An object of class kernel of type lin for pathway hsa04710 with values:
##
##
## Number of Individuals: 50
## The pathway contains 31 genes.
```

```
plot(K)
```



```
#data(hsa04020)
#net_kernel <- calc_kernel(my_gwas, hsa04020, knots=NULL, type='net',
```

Variance Component Test

A pathways influence on the probability of being a case is evaluated in a variance component test. The test statistic is

$$Q = \frac{1}{2}(y - \mu)^t K(y - \mu)(5)$$

with μ the vector of null model estimators given by $\mu_i = logit^{-1}(x_i^t \beta)$ for an individual i and K a kernel matrix of the pathway to be tested. Q follows a mixture of X^2 distributions which can be approximated using the Satterthwaite procedure (Schaid 2012) or Davies method as implemented in the R package `QuadCompForm` (Davies 1980). More details on the test can be found in Wu et al (2010).

In `kangar00` the logistic kernel machine test can be applied to a SNP set defining a pathway with the function `lkmt`. It needs the following arguments

- A formula specifying the null model to be used in the test. The dependent variable is the case control status of the individual (in the example denoted as 'pheno') and is explained by an intercept and optional covariates.
- A linear, size-adjusted or network kernel matrix calculated by one of the kernel functions `kernel_lin()`, `kernel_sia()` or `kernel_net()`.
- A `GWASdata` object including the genotype based on which the test should be performed.
- A `character` specifying which method should be used to calculate the p-value. Available are options `'satt'` for the Satterthwaite approximation (Schaid, 2010) or `'davies'` for Davies method (Davies,

1980).

```
pval_net <- lkmt(pheno ~ 1+sex+age, K_mat, my_gwas, method='satt')

pval_net <- lkmt(pheno ~ 1+sex+age, K_lin, gwas, method='satt')
?lkmt
```

will return an object of type `lkmt` giving the test result for the pathway on which the kernel matrix 'K_mat' was calculated. The `GWASdata` object 'my_gwas' has to be the same as used to calculate the kernel matrix. The formula above would for example fit for a phenotype file of the following format (IDs in first column are always required in phenotype file)

ID	pheno	sex	age	smoker
ind1	1	1	41	1
ind2	0	0	38	0
ind3	1	1	56	1
...	...	...	...	...

note, that the columns to be used in the model are specified in the formula given to the `lkmt()` function and not all covariates have to be used.

References

- Liu D, Ghosh D, Lin X. Estimation and testing for the effect of a genetic pathway on a disease outcome using logistic kernel machine regression via logistic mixed models. BMC Bioinformatics 2008 9:292.
- Schaid DJ: Genomic similarity and kernel methods I: advancements by building on mathematical and statistical foundations. Hum Hered 2010, 70:109-131.
- Davies R: Algorithm as 155: the distribution of a linear combination of chi-2 random variables. J R Stat Soc Ser C 1980, 29:323-333.
- Wu MC, Kraft P, Epstein MP, Taylor DM, Chanock SJ, Hunter DJ, Lin X: Powerful SNP-Set Analysis for Case-Control Genome-Wide Association Studies. Am J Hum Genet 2010, 86:929-42
- Cunningham F, Amode MR, Barrell D et al. Ensembl 2015. Nucleic Acids Research 2015 43 Database issue:D662-D669
- Kanehisa, M., Goto, S., Sato, Y., Kawashima, M., Furumichi, M., and Tanabe, M.; Data, information, knowledge and principle: back to metabolism in KEGG. Nucleic Acids Res. 42, D199-D205 (2014).
- Freytag S, Bickeboeller H, Amos CI, Kneib T, Schlather M: A Novel Kernel for Correcting Size Bias in the Logistic Kernel Machine Test with an Application to Rheumatoid Arthritis. Hum Hered. 2012, 74(2):97-108.
- Freytag S, Manitz J, Schlather M, Kneib T, Amos CI, Risch A, Chang-Claude J, Heinrich J, Bickeboeller H: A network-based kernel machine test for the identification of risk pathways in genome-wide association studies. Hum Hered. 2013, 76(2):64-75.