# Package ‘CovCombR’

October 12, 2022

**Type**  Package

**Title**  Combine Partial Covariance / Relationship Matrices

**Version**  1.0

**Date**  2020-01-08

**Author**  Deniz Akdemir, Mohamed Somo, Julio Isidro Sanchez

**Maintainer**  Deniz Akdemir <deniz.akdemir.work@gmail.com>

**Description**  Combine partial covariance matrices using a Wishart-EM algorithm. Methods are described in the November 2019 article by Akdemir et al. [https://www.biorxiv.org/content/10.1101/857425v1](https://www.biorxiv.org/content/10.1101/857425v1).

It can be used to combine partially overlapping covariance matrices from independent trials, partially overlapping multi-view relationship data from genomic experiments, partially overlapping Gaussian graphs described by their covariance structures.

High dimensional covariance estimation, multi-view data integration, high dimensional covariance graph estimation.

**License**  GPL

**Imports**  Matrix, nlme, CholWishart

**Suggests**  knitr, plyr, spcov, qgraph, igraph

**VignetteBuilder**  knitr

**NeedsCompilation**  yes

**Depends**  R (>= 3.5.0)

**Repository**  CRAN

**Date/Publication**  2020-01-18 11:30:08 UTC

---

## R topics documented:

- CovCombR-package ........................................... 2
- BarleyPheno .................................................. 2
- CovComb ..................................................... 3
- GetVarCov ................................................... 6

**Index** 8
**CovCombR-package**  
Combine Partial Covariance / Relationship Matrices

**Description**

Combine partial covariance matrices using a Wishart-EM algorithm. Methods are described in the November 2019 article by Akdemir et al. [https://www.biorxiv.org/content/10.1101/857425v1]. It can be used to combine partially overlapping covariance matrices from independent trials, partially overlapping multi-view relationship data from genomic experiments, partially overlapping Gaussian graphs described by their covariance structures. High dimensional covariance estimation, multi-view data integration. high dimensional covariance graph estimation.

**Details**

The input to the main program CovComb is a list of partial covariance matrices. The output is an estimated combined (high dimensional) covariance matrix. The output of the algorithm, the completed covariance matrix, can be used to make inferences about unobserved covariances, as an input to sparse covariance estimation algorithms, in covariance graph estimation, in discriminant analysis.

**Author(s)**

Deniz Akdemir, Mohamed Somo, Julio Isidro Sanchez  
Maintainer: Deniz Akdemir <deniz.akdemir.work@gmail.com>

**References**

Adventures in Multi-Omics I: Combining heterogeneous data sets via relationships matrices Deniz Akdemir, Julio Isidro Sanchez. [https://www.biorxiv.org/content/10.1101/857425v1].

---

**BarleyPheno**  
Phenotype data from 194 phenotype trials

**Description**

The data was downloaded from [https://triticeaetoolbox.org/barley/](https://triticeaetoolbox.org/barley/) and curated. The data is in list format. Each element in the list is data from one phenotypic trial concerning a sample of traits.

**Usage**

```r
data(BarleyPheno)
```
**CovComb**

*Programs to combine partially observed (high dimensional) covariance matrices. Combining datasets this way, using relationships, is an alternative to imputation.*

---

**Description**

Use for combining partially observed covariance matrices. This function can be used for combining data from independent experiments by combining the estimated covariance or relationship matrices learned from each of the experiments.

**Usage**

```r
CovComb(Klist = NULL, Kinvlist = NULL,
lambda = 1, w = 1, nu = 1000,
maxiter = 500, miniter = 100, Kinit = NULL,
tolparconv = 1e-04,
loglik=FALSE, plotll=FALSE)
```

**Arguments**

- **Klist**: A list of covariance / relationship matrices with row and column names to be combined.
- **Kinvlist**: A list of inverse covariance / relationship matrices with row and column names to be combined, default NULL.
- **lambda**: A scalar learning rate parameter, between 0 and 1. 1 is the default value.
- **w**: Weight parameter, a vector of the same length as Klist, elements corresponding to weights assigned to each of the covariance matrices. Default is 1.
- **nu**: Degrees of freedom parameter. It is either a scalar (same degrees of freedom to each of the covariance component) or a vector of the same length as Klist elements of which correspond to each of the covariance matrices. Currently, only scalar nu is accepted. Default is 1000. the value of nu needs to be larger than the variables in the covariance matrix.
- **maxiter**: Maximum number of iterations before stop. Default value is 500.
- **miniter**: Minimum number of iterations before the convergence criterion is checked. Default value is 100.
- **Kinit**: Initial estimate of the combined covariance matrix. Default value is an identity matrix.
- **tolparconv**: The minimum change in convergence criteria before stopping the algorithm unless the maxiter is reached. This is not evaluated in the first miniter iterations. Default value is 1e-4.
- **loglik**: Logical with default FALSE. Return the path of the log-likelihood or not.
- **plotll**: Logical with default FALSE. Plot the path of the log-likelihood or not.
Let \( A = \{a_1, a_2, \ldots, a_m\} \) be the set of not necessarily disjoint subsets of genotypes covering a set of \( K \) (i.e., \( K = \bigcup_{i=1}^{m} a_i \)) with total \( n \) genotypes. Let \( G_{a_1}, G_{a_2}, \ldots, G_{a_m} \) be the corresponding sample of covariance matrices.

Starting from an initial estimate \( \Sigma^{(0)} = \nu \Psi^{(0)} \), the Wishart EM-Algorithm repeats updating the estimate of the covariance matrix until convergence:

\[
\Psi^{(t+1)} = \frac{1}{\nu m} \sum_{a \in A} P_a \left[ \begin{array}{c}
G_{aa} \\
B_{b|a}^{(t)} G_{aa} \\
\nu \Psi_{bb|a}^{(t)} + B_{b|a}^{(t)} G_{aa} (B_{b|a}^{(t)})' \\
\end{array} \right]
\]

where \( B_{b|a}^{(t)} = \Psi_{ab}^{(t)} (\Psi_{aa}^{(t)})^{-1} \), \( \Psi_{ab}^{(t)} = \Psi_{bb}^{(t)} - \Psi_{ab}^{(t)} (\Psi_{aa}^{(t)})^{-1} \Psi_{ba}^{(t)} \), \( a \) is the set of genotypes in the given partial covariance matrix and \( b \) is the set difference of \( K \) and \( a \). The matrices \( P_a \) are permutation matrices that put each matrix in the sum in the same order. The initial value, \( \Sigma^{(0)} \), is usually assumed to be an identity matrix of dimension \( n \). The estimate \( \Psi^{(T)} \) at the last iteration converts to the estimated covariance with \( \Sigma^{(T)} = \nu \Psi^{(T)} \).

A weighted version of this algorithm can be obtained replacing \( G_{aa} \) in above equations with \( C_{aa}^{(w_a)} = w_a G_{aa} + (1-w_a) \nu \Psi^{(T)} \) for a vector of weights \((w_1, w_2, \ldots, w_m)'\).

Value

Combined covariance matrix estimate. if loglik is TRUE, the this is a list with first element equal to the covariance estimate, second element in the list is the path of the log-likelihood.

Author(s)

Deniz Akdemir // Maintainer: Deniz Akdemir <deniz.akdemir.work@gmail.com>

References

- Adventures in Multi-Omics I: Combining heterogeneous data sets via relationships matrices. Deniz Akdemir, Julio Isidro Sanchez. bioRxiv, November 28, 2019

Examples

```r
###Using Iris data for a simple example
data(iris)
colnames(iris)<-c("S.L","S.W","P.L","P.W","Species")
iris$Species

#Setting seed for reproducability.
set.seed(1234)

###The input of the CovComb is a list of partial covariance matrices for the species 'virginica'.
CovList<-vector(mode="list", length=3)
CovList[[1]]<-cov(iris[sample(1:150,20),c(1,2)])
CovList[[2]]<-cov(iris[sample(1:150,25),c(1,3)])
CovList[[3]]<-cov(iris[sample(1:150,30),c(2,4)])

###Note that the covariances between the variables #1 and 2, 2 and 3, and 3 and 4 are not observed in
```
the above. We will use these covariance matrices
to obtain a 4 by 4 covariance matrix that estimates
these unobserved covariances.

library(CovCombR)
outCovComb<-CovComb(CovList, nu=40)

###
#### Compare the results with what we would get
# if we observed all data.
outCovComb
cov(iris[101:150,1:4])

#### Compare the same based on correlations.
cov2cor(outCovComb)
cov2cor(cov(iris[101:150,1:4]))

#### Here is a simple plot for visual comparison.
image(cov2cor(outCovComb),xlab="", ylab="", axes = FALSE, main="Combined")
axis(1, at = seq(0, 1, length=4),labels=rownames(outCovComb), las=2)
axis(2, at = seq(0, 1, length=4),labels=rownames(outCovComb), las=2)
image(cov2cor(cov(iris[101:150,1:4])),xlab="", ylab="", axes = FALSE, main="All Data")
axis(1, at = seq(0, 1, length=4),labels=colnames(iris[,1:4]), las=2)
axis(2, at = seq(0, 1, length=4),labels=colnames(iris[,1:4]), las=2)

#### Using Weights
outCovCombhtedwgt<-CovComb(CovList, nu=75,w=c(20/75,25/75,30/75))
cov2cor(outCovCombhtedwgt)

#### Refit and plot log-likelihood path
outCovCombhtedwgt<-CovComb(CovList, nu=75,w=c(20/75,25/75,30/75),
loglik=TRUE, plotll=TRUE)

#### For small problems (when the sample size
# moderate and the number of variables is small).
# we can try using optimization to estimate the degrees of freedom
# parameter nu. Nevertheless, this is not always satisfactory.
# The value of nu does not change the
# estimate of the covariance, but it is
# important for evaluating estimation errors.
negativellfornu<-function(nu){
  outCovComb<-CovComb(CovList, nu=ceiling(nu), loglik=TRUE, plotll=FALSE)
  return(-max(outCovComb[[2]]))
}
optout<-optimize(negativellfornu,interval=c(20,100),tol=1e-3)
est.df<-ceiling(optout$minimum)
est.df
#> est.df= 39

####### Estimated nu can be used as an input
## to other statistical procedures
## such as hypothesis testing about
## the covariance parameters, graphical modeling,
## sparse covariance estimation, etc,....

### GetVarCov
Asymptotic variance-covariance of the estimators

Description
Obtain the asymptotic covariance matrix for the combined covariance estimate. you need to run the CovComb first and then use the estimated covariance matrix as an input to this function.

Usage
GetVarCov(Hmat, Klist, nu = 100, w=1)

Arguments
- **Hmat**: The estimated covariance matrix. Output from CovComb.
- **Klist**: A list of covariance / relationship matrices with row and column names to be combined.
- **w**: Weight parameter, a vector of the same length as Klist, elements corresponding to weights assigned to each of the covariance matrices. Default is 1.
- **nu**: Degrees of freedom parameter. It is either a scalar (same degrees of freedom to each of the covariance component) or a vector of the same length as Klist elements of which correspond to each of the covariance matrices. Currently, only scalar nu is accepted. Default is 1000. the value of nu needs to be larger than the variables in the covariance matrix.

Value
Asymptotic sampling covariance matrix for the combined covariance estimate. The diagonals elements correspond to the sampling variances of the covariance estimates.

Author(s)
Deniz Akdemir // Maintainer: Deniz Akdemir <deniz.akdemir.work@gmail.com>
References

- Adventures in Multi-Omics I: Combining heterogeneous data sets via relationships matrices. Deniz Akdemir, Julio Isidro Sanchez. bioRxiv, November 28, 2019

Examples

data("mtcars")
my_data <- mtcars[, c(1,3,4,5)]

# print the first few rows
head(my_data)

# Artificially making 3 partial covariance matrices!
# These are the partial covariances obtained from
# independent multi-view experiments.
set.seed(123)
cov12<-cov(my_data[sample(nrow(my_data),20),1:2])
cov23<-cov(my_data[sample(nrow(my_data),20),2:3])
cov34<-cov(my_data[sample(nrow(my_data),20),3:4])

# Combine covariances using the package
Combined<-CovComb(Klist=list(cov12,cov23,cov34))

# Get asymptotic sampling variance-covariance matrix.
SEMAT<-GetVarCov(Hmat=Combined,
Klist=list(cov12,cov23,cov34),nu=20,w=1)

## Square root of the diagonal elements are
## the asymptotic standard errors.
round(sqrt(diag(SEMAT)),3)
Index

* covariance graph
  CovCombR-package, 2
* covariance
  CovCombR-package, 2
* genomics
  CovCombR-package, 2
* graphical models
  CovCombR-package, 2
* models
  CovComb, 3
  GetVarCov, 6
* multiomics
  CovCombR-package, 2
* multivariate
  CovComb, 3
  GetVarCov, 6
* multiview data
  CovCombR-package, 2
* phenomics
  CovCombR-package, 2
* regression
  CovComb, 3
  GetVarCov, 6
* wishart distribution
  CovCombR-package, 2

BarleyPheno, 2

CovComb, 3
CovCombR (CovCombR-package), 2
CovCombR-package, 2

GetVarCov, 6