

Package ‘kro.inv.test’

July 30, 2026

Title Kronecker-Invariant Tests for High-Dimensional Separability Testing

Version 0.1.2

Description

Kronecker-invariant tests for high-dimensional separability testing of matrix-variate data, focusing on Gaussian populations as benchmark cases. Tests whether the population covariance matrix is represented as a Kronecker product of row and column covariance matrices. Implements the tests based on the eigenvalues of the sample core whose test statistics are invariant to the separable component of the population covariance matrix, referred to as Kronecker-invariance. Tests constructed using the largest eigenvalue and the separable expansion of the sample core and applying the extended likelihood ratio test for sphericity testing to the sample core. For details, see Sung and Hoff (2025) <[doi:10.48550/arXiv.2506.17463](https://doi.org/10.48550/arXiv.2506.17463)>.

License GPL-3

Encoding UTF-8

RoxygenNote 8.0.0

Imports RMTstat, RSpectra, covKCD, pracma, stats

URL <https://github.com/Seungbongjung/kro.inv.test>

BugReports <https://github.com/Seungbongjung/kro.inv.test/issues>

NeedsCompilation no

Author Bongjung Sung [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-2464-9977>>)

Maintainer Bongjung Sung <bongjung.sung@duke.edu>

Repository CRAN

Date/Publication 2026-07-30 17:30:02 UTC

Contents

dat2cov	2
elrt.alt	3
elrt.null	4
elrt.para	5

elrt.power	6
elrt.power.dat	8
large.eig.alt	9
large.eig.null	11
large.eig.power	12
large.eig.power.dat	14
pi.core	15
pi.rank1.core	16
pi.rank2.core	17
pi.rank_r.core	19
sep.exp.alt	20
sep.exp.null	21
sep.exp.power	23
sep.exp.power.dat	24
sym.inv.root	26
sym.root	27
tw.para	27
Index	29

dat2cov	<i>Sample covariance matrix of the data tensor.</i>
---------	---

Description

Compute the sample covariance matrix of a given data tensor.

Usage

dat2cov(dat, center)

Arguments

dat the $n \times p1 \times p2$ data tensor.
center logical, whether to center the data or not.

Value

the sample covariance matrix of a given data tensor dat.

Author(s)

Bongjung Sung

Examples

```
set.seed(100)
X=array(rnorm(36),dim=c(3,3,4))
dat2cov(X,center=TRUE)
```

elrt.alt	<i>Test statistic based on the extended LRT for sphericity testing under the alternative regime</i>
----------	---

Description

Implement the Monte-Carlo simulations of the test statistic based on the the extended LRT, applied to the sample core $\hat{C}$, for Gaussian populations as benchmark cases under the alternative regime. Here the sample core $\hat{C}$ refers to the core component of the sample covariance matrix S . The test statistic may be transformed following Theorem 1–2 of Wang and Xu (2021) if `trans=TRUE`. This transformation is based on the quantities depending only on (n, p_1, p_2) . For the details, see Section 3.2 and 5.1 of Sung and Hoff (2025). The population covariance matrix should be specified in `sigma`. Also, if the mean is assumed to be known, you may not center the data (`center=FALSE`). Otherwise, the data should be centered.

Usage

```
elrt.alt(
  n,
  p1,
  p2,
  sigma,
  center = TRUE,
  trans = TRUE,
  samp.num = 1000,
  iter = 10
)
```

Arguments

<code>n</code>	the sample size.
<code>p1</code>	the row dimension.
<code>p2</code>	the column dimension.
<code>sigma</code>	the population covariance matrix.
<code>center</code>	logical, whether to center the data or not; TRUE by default.
<code>trans</code>	logical, whether to transform the test statistic; TRUE by default.
<code>samp.num</code>	the number of iterations for the Monte-Carlo simulation; 1000 by default.
<code>iter</code>	the unit number at which to print the number of current iterations; 10 by default.

Value

the vector of `samp.num` Monte-Carlo simulated test statistics based on the extended LRT, applied to the sample core, under the alternative.

Author(s)

Bongjung Sung

References

Sung, B. and Hoff, P. (2025). Testing Separability of High-Dimensional Covariance Matrices. arXiv preprint arXiv:2506.17463.

Wang, Z. and Xu, X. (2021). High-dimensional sphericity test by extended likelihood ratio. *Metrika* 84:1169—1212.

Examples

```
p1=12; p2=10; r=4; n=200

set.seed(100)
para.list=pi.rank_r.core(p1,p2,r,lambda.gen=FALSE)
Sigma=pi.core(para.list,lambda0=0.95)

elrt.alt(n,p1,p2,Sigma,center=FALSE,samp.num=20)
```

elrt.null	<i>Test statistic based on the extended LRT for sphericity testing under the null</i>
-----------	---

Description

Implement the Monte-Carlo simulations of the test statistic based on the the extended LRT, applied to the sample core $\hat{C}$, for Gaussian populations as benchmark cases under the null hypothesis of separability. Here the sample core $\hat{C}$ refers to the core component of the sample covariance matrix S . The extended LRT by Wang and Xu (2021) aims to test the sphericity of the population covariance matrix. The test statistic may be transformed following Theorem 1–2 of Wang and Xu (2021) if `trans=TRUE`. This transformation is based on the quantities depending only on (n, p_1, p_2) . For the details, see Section 3.2 and 5.1 of Sung and Hoff (2025). Also, if the mean is assumed to be known, you may not center the data (`center=FALSE`). Otherwise, the data should be centered.

Usage

```
elrt.null(n, p1, p2, center = TRUE, trans = TRUE, samp.num = 1000, iter = 10)
```

Arguments

n	the sample size.
p1	the row dimension.
p2	the column dimension.
center	logical, whether to center the data or not; TRUE by default.

trans	logical,whether to transform the test statistic; TRUE by default.
samp.num	the number of iterations for the Monte-Carlo simulation; 1000 by default.
iter	the unit number at which to print the number of current iterations; 10 by default.

Value

the vector of samp.num Monte-Carlo simulated null test statistics based on the extended LRT, applied to the sample core.

Author(s)

Bongjung Sung

References

Sung, B. and Hoff, P. (2025). Testing Separability of High-Dimensional Covariance Matrices. arXiv preprint arXiv:2506.17463.

Wang, Z. and Xu, X. (2021). High-dimensional sphericity test by extended likelihood ratio. *Metrika* 84:1169—1212.

Examples

```
p1=10; p2=8; n=100

set.seed(100)
elrt.null(n,p1,p2,center=FALSE,samp.num=20)
```

elrt.para	<i>Parameters of the transformation for the extended LRT statistic</i>
-----------	--

Description

Compute the parameters associated with transforming the extended LRT statistic based on the sample core.

Usage

```
elrt.para(p1, p2, n)
```

Arguments

p1	the row dimension
p2	the column dimension
n	the sample size

Value

a parameter that is associated with transforming the extended LRT statistic based on the sample core.

Author(s)

Bongjung Sung

Examples

```
p1=20; p2=10; n=200
elrt.power(p1,p2,n)
```

elrt.power

Empirical power of the test based on the extended LRT to the sample core under Gaussian populations

Description

Evaluate the empirical power of the test based on the extended LRT to the sample core $\hat{C}$ with n i.i.d. random matrices generated according to $N_{p_1 \times p_2}(0, \Sigma)$ with given Σ (specified in sigma). The power is evaluated with the transformed test statistic following Theorem 1–2 of Wang and Xu (2021). Given a level $\alpha \in (0, 1)$ (specified in alpha), the parametric power is evaluated. To this end, the p-value (para.pval) is first evaluated for each test statistic by comparing it to the Monte-Carlo approximated null distribution (null.stat). Then the power (para.power) is evaluated as the proportion of these p-values smaller than α . For details, see Section 5.2 of Sung and Hoff (2025). If the mean is assumed to be known, you may not center the data (center=FALSE). Otherwise, you should center the data (center=TRUE).

Usage

```
elrt.power(
  n,
  p1,
  p2,
  sigma,
  alpha = 0.05,
  center = TRUE,
  null.samp.num = 1000,
  alt.samp.num = 1000,
  iter = 10
)
```

Arguments

n	the sample size.
p1	the row dimension.
p2	the column dimension.
sigma	the population covariance matrix.
alpha	the level of the test.
center	logical, whether to center the data or not; TRUE by default.
null.samp.num	the number of iterations for the Monte-Carlo simulation of the test statistics under the null; 1000 by default.
alt.samp.num	the number of iterations for the Monte-Carlo simulation of the test statistics under the alternative; 1000 by default.
iter	the unit number at which to print the number of current iterations; 10 by default.

Value

elrt.power returns a list of the following elements:

alt.stat the vector of alt.samp.num Monte-Carlo simulated test statistics under $N_{p_1 \times p_2}(0, \text{sigma})$ after some transformation;

null.stat the vector of null.samp.num Monte-Carlo simulated test statistics under the null after some transformation;

para.pval the vector of p-values for each test statistic in alt.stat evaluated based on Monte-Carlo approximated empirical null distribution (null.stat);

para.power the proportion of the p-values in para.pval smaller than alpha;

Author(s)

Bongjung Sung

References

Sung, B. and Hoff, P. (2025). Testing Separability of High-Dimensional Covariance Matrices. arXiv preprint arXiv:2506.17463.

Wang, Z. and Xu, X. (2021). High-dimensional sphericity test by extended likelihood ratio. *Metrika* 84:1169—1212.

Examples

```
p1=12; p2=8; r=4; n=120

set.seed(100)
para.list=pi.rank_r.core(p1,p2,r,lambda.gen=FALSE)
Sigma=pi.core(para.list,lambda0=0.98)

elrt.power(n,p1,p2,Sigma,center=FALSE,null.samp.num=20,alt.samp.num=20)
```

elrt.power.dat	<i>Empirical power of the test based on the extended LRT to the sample core with the given data</i>
----------------	---

Description

Evaluate the empirical power of the test based on the extended LRT to the sample core $\hat{C}$ with the tensor data, $Y \in \mathbb{R}^{n \times p_1 \times p_2}$. The power is evaluated with the transformed test statistic following Theorem 1–2 of Wang and Xu (2021). Given a level $\alpha \in (0, 1)$ (specified in alpha), the parametric power is evaluated. To this end, the p-value (para.pval) is first evaluated for each test statistic by comparing it to the Monte-Carlo approximated null distribution (null.stat). Then the power (para.power) is evaluated as the proportion of these p-values smaller than α . For details, see Section 5.2 of Sung and Hoff (2025). If the mean is assumed to be known, you may not center the data (center=FALSE). Otherwise, you should center the data (center=TRUE).

Usage

```
elrt.power.dat(dat, center = TRUE, samp.num = 1000, iter = 10)
```

Arguments

dat	the $n \times p_1 \times p_2$ data tensor (sample x row x column).
center	logical, whether to center the data or not; TRUE by default.
samp.num	the number of iterations for simulating the transformed null test statistic; 1000 by default.
iter	the unit number at which to print the number of current iterations; 10 by default.

Value

elrt.power.dat returns a list with the following elements:

test.stat the computed test statistic based on dat after some transformation;
null.stat the vector of samp.num Monte-Carlo simulated null test statistics after some transformation;
para.pval the p-value evaluated based on Monte-Carlo approximated empirical null distribution (null.stat);

Author(s)

Bongjung Sung

References

Sung, B. and Hoff, P. (2025). Testing Separability of High-Dimensional Covariance Matrices. arXiv preprint arXiv:2506.17463.
Wang, Z. and Xu, X. (2021). High-dimensional sphericity test by extended likelihood ratio. *Metrika* 84:1169—1212.

Examples

```
p1=14; p2=10; r=4; n=200
p=p1*p2

set.seed(100)
para.list=pi.rank_r.core(p1,p2,r,lambda.gen=FALSE)
Sigma=pi.core(para.list,lambda0=0.98)
Sigma.root=sym.root(Sigma)

dat=crossprod(Sigma.root,matrix(rnorm(n*p),ncol=n))
dat=array(dat,dim=c(p1,p2,n))
dat=aperm(dat,perm=c(3,1,2))
elrt.power.dat(dat,center=FALSE,samp.num=20)
```

large.eig.alt

Test statistic based on the largest eigenvalue of the sample core under the alternative regime

Description

Implement the Monte-Carlo simulations of the test statistic based on the largest eigenvalue of the sample core $\hat{C}$ for Gaussian populations as benchmark cases under the alternative regime. Here the sample core $\hat{C}$ refers to the core component of the sample covariance matrix S . The test statistic may be transformed to approximate the Tracy-Widom (TW) law if `trans=TRUE`, particularly under the local alternative regime (see Theorem 2 of Sung and Hoff (2025)). This transformation is based on the quantities depending only on (n, p_1, p_2) . If the population covariance matrix of the mean is assumed to be known (`sigma.known=TRUE`), the transformation is done regardless of `trans` based on the Kronecker MLE (the separable component of S). For the details, see (20) of Sung and Hoff (2025). Unless transformed, the function will return the values of $\lambda_1(\hat{C})$. Also, if the mean is assumed to be known, you may not center the data (`center=FALSE`). Otherwise, the data should be centered.

Usage

```
large.eig.alt(
  n,
  p1,
  p2,
  sigma,
  center = TRUE,
  trans = TRUE,
  sigma.known = FALSE,
  samp.num = 1000,
  iter = 10
)
```

Arguments

<code>n</code>	the sample size.
<code>p1</code>	the row dimension.
<code>p2</code>	the column dimension.
<code>sigma</code>	the population covariance matrix.
<code>center</code>	logical, whether to center the data or not; TRUE by default.
<code>trans</code>	logical, whether to transform the test statistic to approximate TW-law or not; TRUE by default.
<code>sigma.known</code>	logical, whether to assume the known population covariance matrix or not; FALSE by default. If TRUE, the transformed test statistic is returned, regardless of <code>trans</code> .
<code>samp.num</code>	the number of iterations for the Monte-Carlo simulation; 1000 by default.
<code>iter</code>	the unit number at which to print the number of current iterations; 10 by default.

Value

the vector of `samp.num` Monte-Carlo simulated test statistics based on the largest eigenvalue of the sample core under the alternative.

Author(s)

Bongjung Sung

References

Sung, B. and Hoff, P. (2025). Testing Separability of High-Dimensional Covariance Matrices. arXiv preprint arXiv:2506.17463.

Examples

```
p1=12; p2=8; n=120

set.seed(100)
para.list=pi.rank2.core(p1,p2,lambda.gen=FALSE)

lambda=1-(1/(96/2+1))
Sigma=pi.core(para.list,lambda0=lambda)
large.eig.alt(n,p1,p2,sigma=Sigma,center=FALSE,samp.num=20)
```

large.eig.null	<i>Test statistic based on the largest eigenvalue of the sample core under the null</i>
----------------	---

Description

Implement the Monte-Carlo simulations of the test statistic based on the largest eigenvalue of the sample core $\hat{C}$ for Gaussian populations as benchmark cases under the null hypothesis of separability. Here the sample core $\hat{C}$ refers to the core component of the sample covariance matrix S . The test statistic may be transformed to approximate the Tracy-Widom (TW) law if `trans=TRUE`. This transformation is based on the quantities depending only on (n, p_1, p_2) . If the population covariance matrix under the null is assumed to be known (`sigma.known=TRUE`), the transformation is done regardless of `trans` based on the Kronecker MLE (the separable component of S). For the details, see (20) of Sung and Hoff (2025). Unless transformed, the function will return the values of $\lambda_1(\hat{C})$. Also, if the mean is assumed to be known, you may not center the data (`center=FALSE`). Otherwise, the data should be centered.

Usage

```
large.eig.null(
  n,
  p1,
  p2,
  center = TRUE,
  trans = TRUE,
  sigma.known = FALSE,
  samp.num = 1000,
  iter = 10
)
```

Arguments

<code>n</code>	the sample size.
<code>p1</code>	the row dimension.
<code>p2</code>	the column dimension.
<code>center</code>	logical, whether to center the data or not; TRUE by default.
<code>trans</code>	logical, whether to transform the test statistic to approximate TW-law or not; TRUE by default.
<code>sigma.known</code>	logical, whether to assume the known population covariance matrix or not; FALSE by default. If TRUE, the transformed test statistic is returned, regardless of <code>trans</code> .
<code>samp.num</code>	the number of iterations for the Monte-Carlo simulation; 1000 by default.
<code>iter</code>	the unit number at which to print the number of current iterations; 10 by default.

Value

the vector of `samp.num` Monte-Carlo simulated null test statistics based on the largest eigenvalue of the sample core.

Author(s)

Bongjung Sung

References

Sung, B. and Hoff, P. (2025). Testing Separability of High-Dimensional Covariance Matrices. arXiv preprint arXiv:2506.17463.

Examples

```
p1=10; p2=10; n=400
set.seed(100)
large.eig.null(n,p1,p2,center=FALSE,samp.num=20)
```

large.eig.power

Empirical power of the test based on the largest eigenvalue of the sample core under Gaussian populations

Description

Evaluate the empirical power of the test based on $\lambda_1(\hat{C})$ with n i.i.d. random matrices generated according to $N_{p_1 \times p_2}(0, \Sigma)$ for the sample core $\hat{C}$ with given Σ (specified in `sigma`). The power is evaluated with the transformed $\lambda_1(\hat{C})$ so that it may follow the Tracy-Widom law under the null and some local alternative regimes (see Theorem 2 of Sung and Hoff (2025)). Given a level $\alpha \in (0, 1)$ (specified in `alpha`), both parametric and nonparametric power are evaluated. For the parametric power, the p-value (`para.pval`) is first evaluated for each test statistic by comparing it to the Monte-Carlo approximated null distribution (`null.stat`). Then the power (`para.power`) is evaluated as the proportion of these p-values smaller than α . Similarly, to compute the nonparametric power, the p-value (`nonpara.pval`) is first evaluated for each test statistic by comparing it to the Tracy-Widom law. Finally, the power (`nonpara.power`) is evaluated as an analogy to `para.power`. For details, see Section 5.2 of Sung and Hoff (2025). If the mean is assumed to be known, you may not center the data (`center=FALSE`). Otherwise, you should center the data (`center=TRUE`).

Usage

```
large.eig.power(
  n,
  p1,
  p2,
  sigma,
  alpha = 0.05,
```

```

    center = TRUE,
    null.samp.num = 1000,
    alt.samp.num = 1000,
    iter = 10
  )

```

Arguments

<code>n</code>	the sample size.
<code>p1</code>	the row dimension.
<code>p2</code>	the column dimension.
<code>sigma</code>	the population covariance matrix.
<code>alpha</code>	the level of the test.
<code>center</code>	logical, whether to center the data or not; TRUE by default.
<code>null.samp.num</code>	the number of iterations for the Monte-Carlo simulation of the test statistics under the null; 1000 by default.
<code>alt.samp.num</code>	the number of iterations for the Monte-Carlo simulation of the test statistics under the alternative; 1000 by default.
<code>iter</code>	the unit number at which to print the number of current iterations; 10 by default.

Value

`large.eig.power` returns a list of the following elements:

- alt.stat** the vector of `alt.samp.num` Monte-Carlo simulated test statistics under $N_{p_1 \times p_2}(0, \text{sigma})$ after some transformation;
- null.stat** the vector of `null.samp.num` Monte-Carlo simulated test statistics under the null after some transformation;
- para.pval** the vector of p-values for each test statistic in `alt.stat` evaluated based on Monte-Carlo approximated empirical null distribution (`null.stat`);
- para.power** the proportion of the p-values in `para.pval` smaller than `alpha`;
- nonpara.pval** the vector of p-values for each test statistic in evaluated by comparing the test statistic to the Tracy-Widom law;
- nonpara.power** the proportion of the p-values in `nonpara.pval` smaller than `alpha`.

Author(s)

Bongjung Sung

References

Sung, B. and Hoff, P. (2025). Testing Separability of High-Dimensional Covariance Matrices. arXiv preprint arXiv:2506.17463.

Examples

```
p1=10; p2=12; n=200

set.seed(100)
para.list=pi.rank2.core(p1,p2,lambda.gen=FALSE)

lambda=1-(0.84/(120/2+0.84))
Sigma=pi.core(para.list,lambda0=lambda)
large.eig.power(n,p1,p2,Sigma,center=FALSE,null.samp.num=20,alt.samp.num=20)
```

large.eig.power.dat	<i>Empirical power of the test based on the largest eigenvalue of the sample core with the given data</i>
---------------------	---

Description

Evaluate the empirical power of the test based on $\lambda_1(\hat{C})$ with the tensor data, $Y \in \mathbb{R}^{n \times p_1 \times p_2}$. The power is evaluated with the transformed $\lambda_1(\hat{C})$ so that it may follow the Tracy-Widom law under the null and some local alternative regimes (see Theorem 2 of Sung and Hoff (2025)). The parametric p-value (para.pval) is evaluated for each test statistic by comparing it to the Monte-Carlo approximated null distribution (null.stat). On the other hand, the nonparametric p-value (nonpara.pval) is evaluated for each test statistic by comparing it to the Tracy-Widom law. For details, see Section 5.2 of Sung and Hoff (2025). If the mean is assumed to be known, you may not center the data (center=FALSE). Otherwise, you should center the data (center=TRUE).

Usage

```
large.eig.power.dat(dat, center = TRUE, samp.num = 1000, iter = 10)
```

Arguments

dat	the $n \times p_1 \times p_2$ data tensor (sample x row x column).
center	logical, whether to center the data or not; TRUE by default.
samp.num	the number of iterations for simulating the transformed null test statistic; 1000 by default.
iter	the unit number at which to print the number of current iterations; 10 by default.

Value

large.eig.power.dat returns a list with the following elements:

- test.stat** the computed test statistic based on dat after some transformation;
- null.stat** the vector of samp.num Monte-Carlo simulated null test statistics after some transformation;
- para.pval** the p-value evaluated based on Monte-Carlo approximated empirical null distribution (null.stat);
- nonpara.pval** the p-value evaluated by comparing the test statistic to the Tracy-Widom law.

Author(s)

Bongjung Sung

References

Sung, B. and Hoff, P. (2025). Testing Separability of High-Dimensional Covariance Matrices. arXiv preprint arXiv:2506.17463.

Examples

```
p1=10; p2=12; r=4; n=150
p=p1*p2

set.seed(100)
para.list=pi.rank_r.core(p1,p2,r,lambda.gen=FALSE)

# local alternative
Sigma=pi.core(para.list,lambda0=0.8)
Sigma.root=sym.root(Sigma)
dat=crossprod(Sigma.root,matrix(rnorm(n*p),ncol=n))
dat=array(dat,dim=c(p1,p2,n))
dat=aperm(dat,perm=c(3,1,2))
large.eig.power.dat(dat,center=FALSE,samp.num=20)
```

pi.core*A covariance matrix with a partial-isotropy rank-r core.*

Description

Given a list of parameters constituting the covariance matrix with a partial-isotropy rank- r core, the function assembles the covariance matrix using these parameters.

Usage

```
pi.core(para.list, lambda0 = 0.5, root = "sym")
```

Arguments

para.list	the list of parameters constituting the covariance matrix with a partial-isotropy rank- r core.
lambda0	the pre-specified value of the non-spiked eigenvalue λ in a partial-isotropy core. If it is not specified in para.list or does not belong to (0,1), it will be randomly generated; 0.5 by default.
root	the choice of the square root of positive definite matrices; must be either "sym" (symmetric square root) or "chol" (Cholesky factor), "sym" by default.

Value

a $p_1 p_2 \times p_1 p_2$ covariance matrix Σ with a partial-isotropy rank- r core. The attribute λ of Σ denotes the value of the non-spiked eigenvalue.

Author(s)

Bongjung Sung

Examples

```
set.seed(100)
# generate a list of parameters for a covariance matrix of a partial-isotropy core.
p1=14; p2=10; r=3
para.list=pi.rank_r.core(p1,p2,r)
# assembles the covariance matrix using the above list of parameters.
pi.core(para.list)
```

pi.rank1.core	<i>Partial-isotropy rank-1 core</i>
---------------	-------------------------------------

Description

Randomly generate a list of parameters for the covariance matrix with a partial-isotropy rank-1 core for square matrix-variate data.

Usage

```
pi.rank1.core(p1, lambda.gen = TRUE)
```

Arguments

p1	the row and column dimensions.
lambda.gen	logical, whether to generate a non-spiked eigenvalue or not.; TRUE by default.

Details

If a core covariance matrix is of rank-1, the data should be a square matrix. Thus, the row and column dimensions must coincide. The covariance matrix Σ with a partial-isotropy rank-1 core takes the form of

$$\Sigma = (K_2 \otimes K_1)^{1/2} ((1 - \lambda) \text{vec}(A) \text{vec}(A)^\top + \lambda I_{p_1^2}) (K_2 \otimes K_1)^{1/2, \top}$$

for $\lambda \in (0, 1)$, positive definite matrices K_1, K_2 of the same dimensions $p_1 \times p_1$, and $A \in \mathbb{R}^{p_1 \times p_1}$ such that $\text{vec}(A) \text{vec}(A)^\top$ is a rank-1 core. Here $M^{1/2}$ denotes either symmetric square root or the Cholesky factor of a positive definite matrix M . The separable and core components of Σ , denoted K and C , are

$$K = K_2 \otimes K_1, \quad C = (1 - \lambda) \text{vec}(A) \text{vec}(A)^\top + \lambda I_{p_1^2}.$$

For the exact formula, see Theorem 1 of Sung and Hoff (2025).

Value

- pi.rank1.core returns a list with the following elements:
- K1** the row covariance matrix of dimension $p1 \times p1$;
 - K2** the column covariance matrix of dimension $p1 \times p1$;
 - A** the factor matrix of a rank-1 core of dimension $p1 \times p1$;
 - lambda** If lambda.gen=TRUE, the non-spiked eigenvalue $\lambda \in (0, 1)$. Otherwise, NULL.

Author(s)

Bongjung Sung

References

Sung, B. and Hoff, P. (2025). Testing Separability of High-Dimensional Covariance Matrices. arXiv preprint arXiv:2506.17463.

Examples

```
set.seed(100)
p1=10
pi.rank1.core(p1)
```

pi.rank2.core	<i>Partial-isotropy rank-2 core</i>
---------------	-------------------------------------

Description

Randomly generate a list of parameters for the covariance matrix with a partial-isotropy rank-2 core for matrix-variate data.

Usage

```
pi.rank2.core(p1, p2, lambda.gen = TRUE)
```

Arguments

- p1 the row dimension.
- p2 the column dimension.
- lambda.gen logical, whether to generate a non-spiked eigenvalue or not; TRUE by default.

Details

If a core covariance matrix is of rank-2, the dimension (p_1, p_2) should satisfy one of the followings: $p_1 = p_2$ or $p_1 - p_2 | p_1$ ($p_1 > p_2$), or $p_2 - p_1 | p_2$ ($p_2 > p_1$). The covariance matrix Σ with a partial-isotropy rank-2 core takes the form of

$$\Sigma = (K_2 \otimes K_1)^{1/2} ((1 - \lambda)AA^\top + \lambda I_{p_1 p_2}) (K_2 \otimes K_1)^{1/2, \top}$$

for $\lambda \in (0, 1)$, positive definite matrices K_1 and K_2 of the dimensions $p_1 \times p_1$ and $p_2 \times p_2$, respectively, and $A \in \mathbb{R}^{p_1 p_2 \times 2}$ whose i th column is a vectorization of $p_1 \times p_2$ matrix A_i such that $AA^\top$ is a rank-2 core. Here $M^{1/2}$ denotes either symmetric square root or the Cholesky factor of a positive definite matrix M . The separable and core components of Σ , denoted K and C , are

$$K = K_2 \otimes K_1, \quad C = (1 - \lambda)AA^\top + \lambda I_{p_1 p_2}.$$

For the exact formula, see Theorem 1 of Sung and Hoff (2025).

Value

`pi.rank2.core` returns a list with the following elements:

K1 the row covariance matrix of dimension $p1 \times p1$;

K2 the column covariance matrix of dimension $p2 \times p2$;

A an array of factor matrices of a rank-2 core of dimension $p1 \times p2 \times 2$;

lambda If `lambda.gen=TRUE`, the non-spiked eigenvalue $\lambda \in (0, 1)$. Otherwise, `NULL`.

Author(s)

Bongjung Sung

References

Sung, B. and Hoff, P. (2025). Testing Separability of High-Dimensional Covariance Matrices. arXiv preprint arXiv:2506.17463.

Examples

```
set.seed(100)
# It must be that p1=p2 or p1-p2|p1 (p1>p2) or p2-p1|p2 (p2>p1).
p1=12; p2=10
pi.rank2.core(p1,p2)
```

pi.rank_r.core	<i>Partial-isotropy rank-r core.</i>
----------------	--------------------------------------

Description

Randomly generate a list of parameters for the covariance matrix with a partial-isotropy rank- r core for matrix-variate data.

Usage

```
pi.rank_r.core(p1, p2, r, lambda.gen = TRUE)
```

Arguments

p1	the row dimension.
p2	the column dimension.
r	the partial-isotropy rank.
lambda.gen	logical, whether to generate a non-spiked eigenvalue or not; TRUE by default.

Details

If a core covariance matrix is of rank- r for general r , the dimension (p_1, p_2) should satisfy one of the followings: $p_1/p_2 + p_2/p_1 < r$ or $p_1 = p_2 r$ ($p_1 \geq p_2$), or $p_2 = p_1 r$ ($p_2 \geq p_1$). The covariance matrix Σ with a partial-isotropy rank- r core takes the form of

$$\Sigma = (K_2 \otimes K_1)^{1/2}((1 - \lambda)AA^\top + \lambda I_{p_1 p_2})(K_2 \otimes K_1)^{1/2, \top}$$

for $\lambda \in (0, 1)$, positive definite matrices K_1 and K_2 of the dimensions $p_1 \times p_1$ and $p_2 \times p_2$, respectively, and $A \in \mathbb{R}^{p_1 p_2 \times r}$ whose i th column is a vectorization of $p_1 \times p_2$ matrix A_i such that $AA^\top$ is a rank- r core. Here $M^{1/2}$ denotes either symmetric square root or the Cholesky factor of a positive definite matrix M . The separable and core components of Σ , denoted K and C , are

$$K = K_2 \otimes K_1, \quad C = (1 - \lambda)AA^\top + \lambda I_{p_1 p_2}.$$

For the exact formula when $p_1 = p_2 r$ or $p_2 = p_1 r$, see Theorem 1 of Sung and Hoff (2025).

Value

pi.rank_r.core returns a list with the following elements:

- K1** the row covariance matrix of dimension $p_1 \times p_1$;
- K2** the column covariance matrix of dimension $p_2 \times p_2$;
- A** an array of factor matrices of a rank- r core of dimension $p_1 \times p_2 \times r$;
- lambda** If lambda.gen=TRUE, the non-spiked eigenvalue $\lambda \in (0, 1)$. Otherwise, NULL.

Author(s)

Bongjung Sung

References

Sung, B. and Hoff, P. (2025). Testing Separability of High-Dimensional Covariance Matrices. arXiv preprint arXiv:2506.17463.

Examples

```
set.seed(100)
# It must be that p1/p2+p2/p1<r or p1=p2r or p2=p1r
p1=12; p2=8; r=3
pi.rank_r.core(p1,p2,r)
```

sep.exp.alt

Test statistic based on the separable expansion test under the alternative regime.

Description

Implement the Monte-Carlo simulations of the test statistic based on the separable expansion of the sample core $\hat{C}$ for Gaussian populations as benchmark cases under the alternative regime. Here the sample core $\hat{C}$ refers to the core component of the sample covariance matrix S . The test statistic is given by

$$T(Y) = \|\hat{C}\|_F^2/p - 1,$$

where $p = p_1 p_2$ for row and column dimensions, p_1 and p_2 , respectively. The test statistic may be transformed as $nT(Y) - p - 1$ if `trans=TRUE`. For the details, see Section 3.3 and 5.1 of Sung and Hoff (2025). The population covariance matrix should be specified in `sigma`. Also, if the mean is assumed to be known, you may not center the data (`center=FALSE`). Otherwise, the data should be centered.

Usage

```
sep.exp.alt(
  n,
  p1,
  p2,
  sigma,
  center = TRUE,
  trans = TRUE,
  samp.num = 1000,
  iter = 10
)
```

Arguments

<code>n</code>	the sample size.
<code>p1</code>	the row dimension.
<code>p2</code>	the column dimension.
<code>sigma</code>	the population covariance matrix.
<code>center</code>	logical, whether to center the data or not; TRUE by default.
<code>trans</code>	logical, whether to transform the test statistic; TRUE by default.
<code>samp.num</code>	the number of iterations for the Monte-Carlo simulation; 1000 by default.
<code>iter</code>	the unit number at which to print the number of current iterations; 10 by default.

Value

the vector of `samp.num` Monte-Carlo simulated test statistics based on the separable expansion of the sample core under the alternative.

Author(s)

Bongjung Sung

References

Sung, B. and Hoff, P. (2025). Testing Separability of High-Dimensional Covariance Matrices. arXiv preprint arXiv:2506.17463.

Examples

```
p1=5; p2=3; r=4; n=100

set.seed(100)
para.list=pi.rank_r.core(p1,p2,r,lambda.gen=FALSE)
Sigma=pi.core(para.list,lambda0=0.95)

sep.exp.alt(n,p1,p2,Sigma,center=FALSE,samp.num=20)
```

Description

Implement the Monte-Carlo simulations of the test statistic based on the separable expansion of the sample core $\hat{C}$ for Gaussian populations as benchmark cases under the null hypothesis of separability. Here the sample core $\hat{C}$ refers to the core component of the sample covariance matrix S . The test statistic is given by

$$T(Y) = \|\hat{C}\|_F^2/p - 1,$$

where $p = p_1 p_2$ for row and column dimensions, p_1 and p_2 , respectively. The test statistic may be transformed as $nT(Y) - p - 1$ if `trans=TRUE`. For the details, see Section 3.3 and 5.1 of Sung and Hoff (2025). Also, if the mean is assumed to be known, you may not center the data (`center=FALSE`). Otherwise, the data should be centered.

Usage

```
sep.exp.null(  
  n,  
  p1,  
  p2,  
  center = TRUE,  
  trans = TRUE,  
  samp.num = 1000,  
  iter = 10  
)
```

Arguments

<code>n</code>	the sample size.
<code>p1</code>	the row dimension.
<code>p2</code>	the column dimension.
<code>center</code>	logical, whether to center the data or not; TRUE by default.
<code>trans</code>	logical, whether to transform the test statistic; TRUE by default.
<code>samp.num</code>	the number of iterations for the Monte-Carlo simulation; 1000 by default.
<code>iter</code>	the unit number at which to print the number of current iterations; 10 by default.

Value

the vector of `samp.num` Monte-Carlo simulated null test statistics based on the separable expansion of the sample core.

Author(s)

Bongjung Sung

References

Sung, B. and Hoff, P. (2025). Testing Separability of High-Dimensional Covariance Matrices. arXiv preprint arXiv:2506.17463.

Examples

```
p1=5; p2=3; n=60

set.seed(100)
sep.exp.null(n,p1,p2,center=FALSE,samp.num=20)
```

sep.exp.power	<i>Empirical power of the test based on the separable expansion the sample core under Gaussian populations</i>
---------------	--

Description

Evaluate the empirical power of the test based on the separable expansion of the sample core $\hat{C}$ with n i.i.d. random matrices generated according to $N_{p_1 \times p_2}(0, \Sigma)$ with given Σ (specified in sigma). The test statistic is given by $T(Y) = \|\hat{C}\|_F^2/p - 1$. The power is evaluated with respect to $nT(Y) - p - 1$. Given a level $\alpha \in (0, 1)$ (specified in alpha), the parametric power is evaluated. To this end, the p-value (para.pval) is first evaluated for each test statistic by comparing it to the Monte-Carlo approximated null distribution (null.stat). Then the power (para.power) is evaluated as the proportion of these p-values smaller than α . For details, see Section 5.2 of Sung and Hoff (2025). If the mean is assumed to be known, you may not center the data (center=FALSE). Otherwise, you should center the data (center=TRUE).

Usage

```
sep.exp.power(
  n,
  p1,
  p2,
  sigma,
  alpha = 0.05,
  center = TRUE,
  null.samp.num = 1000,
  alt.samp.num = 1000,
  iter = 10
)
```

Arguments

n	the sample size.
p1	the row dimension.
p2	the column dimension.
sigma	the population covariance matrix.
alpha	the level of the test.
center	logical, whether to center the data or not; TRUE by default.

<code>null.samp.num</code>	the number of iterations for the Monte-Carlo simulation of the test statistics under the null; 1000 by default.
<code>alt.samp.num</code>	the number of iterations for the Monte-Carlo simulation of the test statistics under the alternative; 1000 by default.
<code>iter</code>	the unit number at which to print the number of current iterations; 10 by default.

Value

`sep.exp.power` returns a list of the following elements:

alt.stat the vector of `alt.samp.num` Monte-Carlo simulated test statistics under $N_{p_1 \times p_2}(0, \text{sigma})$ after some transformation;

null.stat the vector of `null.samp.num` Monte-Carlo simulated test statistics under the null after some transformation;

para.pval the vector of p-values for each test statistic in `alt.stat` evaluated based on Monte-Carlo approximated empirical null distribution (`null.stat`);

para.power the proportion of the p-values in `para.pval` smaller than `alpha`;

Author(s)

Bongjung Sung

References

Sung, B. and Hoff, P. (2025). Testing Separability of High-Dimensional Covariance Matrices. arXiv preprint arXiv:2506.17463.

Examples

```
p1=12; p2=10; r=4; n=100

set.seed(100)
para.list=pi.rank_r.core(p1,p2,r,lambda.gen=FALSE)
Sigma=pi.core(para.list,lambda0=0.98)

sep.exp.power(n,p1,p2,Sigma,center=FALSE,null.samp.num=20,alt.samp.num=20)
```

<code>sep.exp.power.dat</code>	<i>Empirical power of the test based on the separable expansion of the sample core with the given data</i>
--------------------------------	--

Description

Evaluate the empirical power of the test based on the separable expansion of the sample core $\hat{C}$ with the tensor data, $Y \in \mathbb{R}^{n \times p_1 \times p_2}$. The test statistic is given by $T(Y) = \|\hat{C}\|_F^2/p - 1$. The power is evaluated with respect to $nT(Y) - p - 1$. Given a level $\alpha \in (0, 1)$ (specified in `alpha`), the parametric power is evaluated. To this end, the p-value (`para.pval`) is first evaluated for each test statistic by comparing it to the Monte-Carlo approximated null distribution (`null.stat`). Then the power (`para.power`) is evaluated as the proportion of these p-values smaller than α . For details, see Section 5.2 of Sung and Hoff (2025). If the mean is assumed to be known, you may not center the data (`center=FALSE`). Otherwise, you should center the data (`center=TRUE`).

Usage

```
sep.exp.power.dat(dat, center = TRUE, samp.num = 1000, iter = 10)
```

Arguments

<code>dat</code>	the $n \times p_1 \times p_2$ data tensor (sample x row x column).
<code>center</code>	logical, whether to center the data or not; TRUE by default.
<code>samp.num</code>	the number of iterations for simulating the transformed null test statistic; 1000 by default.
<code>iter</code>	the unit number at which to print the number of current iterations; 10 by default.

Value

`sep.exp.power.dat` returns a list with the following elements:

- test.stat** the computed test statistic based on `dat` after some transformation;
- null.stat** the vector of `samp.num` Monte-Carlo simulated null test statistics after some transformation;
- para.pval** the p-value evaluated based on Monte-Carlo approximated empirical null distribution (`null.stat`);

Author(s)

Bongjung Sung

References

Sung, B. and Hoff, P. (2025). Testing Separability of High-Dimensional Covariance Matrices. arXiv preprint arXiv:2506.17463.

Examples

```
p1=10; p2=12; r=4; n=100
p=p1*p2

set.seed(100)
para.list=pi.rank_r.core(p1,p2,r,lambda.gen=FALSE)
Sigma=pi.core(para.list,lambda0=0.99)
```

```

Sigma.root=sym.root(Sigma)

dat=crossprod(Sigma.root,matrix(rnorm(n*p),ncol=n))
dat=array(dat,dim=c(p1,p2,n))
dat=aperm(dat,perm=c(3,1,2))
sep.exp.power.dat(dat,center=FALSE,samp.num=20)

```

sym.inv.root

Inverse symmetric square root

Description

Compute the inverse of the symmetric square root of a positive definite matrix.

Usage

```
sym.inv.root(cov)
```

Arguments

cov a positive definite matrix.

Value

the inverse of the symmetric square root of given a positive definite matrix cov.

Author(s)

Bongjung Sung

Examples

```

# generate a positive definite matrix
set.seed(100)
X=matrix(rnorm(4*10),ncol=4)
S=crossprod(X,X)/10
sym.inv.root(S)

```

sym.root	<i>Symmetric square root</i>
----------	------------------------------

Description

Compute the symmetric square root of a positive definite matrix.

Usage

```
sym.root(cov)
```

Arguments

cov a positive definite matrix.

Value

the symmetric square root of given a positive definite matrix cov.

Author(s)

Bongjung Sung

Examples

```
# generate a positive definite matrix
set.seed(100)
X=matrix(rnorm(4*10),ncol=4)
S=crossprod(X,X)/10
sym.root(S)
```

tw.param	<i>Parameters of the transformation to obtain the Tracy-Widom law</i>
----------	---

Description

Compute parameters associated with transforming the largest eigenvalue of the sample core to obtain TW-law.

Usage

```
tw.param(K1, K2, n)
```

Arguments

K1	the $p_1 \times p_1$ covariance matrix.
K2	the $p_2 \times p_2$ covariance matrix.
n	the sample size.

Value

a vector of parameters that are associated with transforming the largest eigenvalue of the sample core to obtain TW-law.

Author(s)

Bongjung Sung

Examples

```
set.seed(100)
p1=3; p2=5; n=60; p=p1*p2
X=matrix(rnorm(p*n),ncol=p)
S=crossprod(X,X)/n
S.kcd=covKCD::covKCD(S,p1,p2)
K1=S.kcd$K1
K2=S.kcd$K2
tw.param(K1,K2,n)
```

Index

`dat2cov`, [2](#)

`elrt.alt`, [3](#)
`elrt.null`, [4](#)
`elrt.para`, [5](#)
`elrt.power`, [6](#)
`elrt.power.dat`, [8](#)

`large.eig.alt`, [9](#)
`large.eig.null`, [11](#)
`large.eig.power`, [12](#)
`large.eig.power.dat`, [14](#)

`pi.core`, [15](#)
`pi.rank1.core`, [16](#)
`pi.rank2.core`, [17](#)
`pi.rank_r.core`, [19](#)

`sep.exp.alt`, [20](#)
`sep.exp.null`, [21](#)
`sep.exp.power`, [23](#)
`sep.exp.power.dat`, [24](#)
`sym.inv.root`, [26](#)
`sym.root`, [27](#)

`tw.para`, [27](#)