

# Package: rwc (via r-universe)

September 1, 2024

**Type** Package

**Title** Random Walk Covariance Models

**Version** 1.11

**Date** 2018-06-04

**Author** Ephraim M. Hanks

**Depends** R (>= 2.10), raster, Matrix, mvtnorm, MASS

**Maintainer** Ephraim M. Hanks <hanks@psu.edu>

**Description** Code to facilitate simulation and inference when connectivity is defined by underlying random walks. Methods for spatially-correlated pairwise distance data are especially considered. This provides core code to conduct analyses similar to that in Hanks and Hooten (2013) <doi:10.1080/01621459.2012.724647>.

**License** GPL-2

**NeedsCompilation** no

**Date/Publication** 2018-06-04 18:23:43 UTC

**Repository** <https://ehanks.r-universe.dev>

**RemoteUrl** <https://github.com/cran/rwc>

**RemoteRef** HEAD

**RemoteSha** 461a722a94cc26e90ced3777b4f367d2a70f82fd

## Contents

rwc-package . . . . .	2
cov.from.dist . . . . .	5
dGenWish . . . . .	7
dist.from.cov . . . . .	8
get.Phi . . . . .	9
get.Q . . . . .	10
get.TL . . . . .	11
mcmc.wish.icar . . . . .	12
rGenWish . . . . .	15
rnorm.Q . . . . .	17

---

rwc-package

*Random Walk Covariance Models*

---

## Description

Code to facilitate simulation and inference of connectivity defined by random walks.

## Details

This package contains code to simulate (rGenWish) and evaluate the likelihood of (dGenWish) distance matrices from the Generalized Wishart distribution. It also contains helper functions to create and manage spatial covariance models created from landscape grids with resistance or conductance defined by landscape features.

## Author(s)

Ephraim M. Hanks

Maintainer: Ephraim M. Hanks

## References

McCullagh 2009. Marginal likelihood for distance matrices. *Statistica Sinica* 19: 631-649.

Hanks and Hooten 2013. Circuit theory and model-based inference for landscape connectivity. *Journal of the American Statistical Association*. 108(501), 22-33.

Hanks 2017. Modeling spatial covariance using the limiting distribution of spatio-temporal random walks. *Journal of the American Statistical Association*.

## Examples

```
## Not run:  
## The following code conducts a simulation example by  
## first simulating a heterogeneous landscape, then  
## simulating pairwise distance data on the landscape  
## and finally making inference on model parameters.
```

```
library(rwc)  
library(MASS)
```

```
## source("GenWishFunctions_20170901.r")
```

```
##  
## specify 2-d raster  
##
```

```
ras=raster(nrow=30,ncol=30)  
extent(ras) <- c(0,30,0,30)
```

```
values(ras) <- 1
plot(ras,asp=1)
ras

int=ras
cov.ras=ras

## simulate "resistance" raster
TL.int=get.TL(int)
Q.int=get.Q(TL.int,1)
set.seed(1248)
## values(cov.ras) <- as.numeric(rnorm.Q(Q.int
values(cov.ras) <- as.numeric(rnorm.Q(Q.int,zero.constraint=TRUE))
plot(cov.ras)

stack=stack(int,cov.ras)
plot(stack)
TL=get.TL(stack)

## Create "barrier" raster which has no effect, to test model selection

barrier=int
values(barrier) <- 0
barrier[,10:11] <- 1

plot(barrier)

TL.all=get.TL(stack(int,cov.ras,barrier))

##
## sampling locations
##

nsamps=150
set.seed(4567)
samp.xy=cbind(runif(nsamps,0,30),runif(nsamps,0,30))
## samp.xy=samp.xy[rep(1:nsamps,times=5),]
samp.locs=cellFromXY(int,samp.xy)
samp.cells=unique(samp.locs)
nsamps=nrow(samp.xy)

plot(cov.ras)
points(samp.xy)

K=matrix(0,nrow=nsamps,ncol=length(samp.cells))
for(i in 1:nsamps){
  K[i,which(samp.cells==samp.locs[i])]=1
}
```

```

image(K)

##
## beta values
##

betas=c(-2,-1)
tau=.01

Q=get.Q(TL,betas)
Phi=get.Phi(Q,samp.cells)

## simulate from ibr model
D.rand.ibr=rGenWish(df=20,Sigma=K%%ginv(as.matrix(Phi))%%t(K) + diag(nsamps)*tau)
image(D.rand.ibr)

## crude plot of geographic distance vs genetic distance

plot(as.numeric(as.matrix(dist(xyFromCell(ras,samp.locs)))),as.numeric(D.rand.ibr))

#####
##
##
## fitting using optim
##
##

nll.gen.wish.icar <- function(theta,D,df,TL,obs.idx){
  ## get K
  cells.idx=unique(obs.idx)
  n.cells=length(cells.idx)
  n.obs=length(obs.idx)
  K <- matrix(0, nrow = n.obs, ncol = n.cells)
  for (i in 1:n.obs){
    K[i, which(cells.idx == obs.idx[i])] <- 1
  }
  ## get precision matrix of whole graph
  tau=exp(theta[1])
  beta=theta[-1]
  Q=get.Q(TL,beta)
  ## get precision matrix of observed nodes
  max.diag=max(diag(Q))
  Q=Q/max.diag
  Phi=get.Phi(Q,cells.idx)
  ## get covariance matrix of observations
  Sigma.nodes=ginv(as.matrix(Phi))
  Sigma.nodes=Sigma.nodes/max.diag

```

```

Psi=K%*%Sigma.nodes%*%t(K)+tau*diag(nrow(K))
## get nll
nll=-dGenWish(D,Psi,df,log=TRUE)
nll
}

fit=optim(c(log(tau),betas),nll.gen.wish.icar,D=D.rand.ibr,df=20,TL=TL,
obs.idx=samp.locs,control=list(trace=10),hessian=TRUE)

fit.all=optim(c(log(tau),betas,0),nll.gen.wish.icar,D=D.rand.ibr,df=20,TL=TL.all,
obs.idx=samp.locs,control=list(trace=10),hessian=FALSE)

fit.ibd=optim(c(log(tau),0),nll.gen.wish.icar,D=D.rand.ibr,df=20,TL=TL.int,
obs.idx=samp.locs,control=list(trace=10),hessian=FALSE)

## model selection using AIC

aic.ibr=2*fit$value + 2*length(fit$par)
aic.all=2*fit.all$value + 2*length(fit.all$par)
aic.ibd=2*fit.ibd$value + 2*length(fit.ibd$par)

aic.ibr
aic.all
aic.ibd

## se's for best fit

str(fit)
## get standard errors from optim
H=fit$hessian
S=solve(H)
se=sqrt(diag(S))
se

## CI's for best fit

CImat=matrix(NA,3,4)
rownames(CImat) <- c("log(tau)","intercept","resistance")
colnames(CImat) <- c("truth","estimate","lower95CI","upper95CI")
CImat[,1]=c(log(tau),betas)
CImat[,2]=fit$par
CImat[,3]=fit$par-1.96*se
CImat[,4]=fit$par+1.96*se

CImat

## End(Not run)

```

---

cov.from.dist

*Create covariance matrix from a distance matrix*

---

**Description**

This computes a covariance matrix from a squared-distance matrix using the centering method of Gower (1996). When the squared-distance matrix is a resistance distance matrix, or a variogram matrix from a spatial model, the resulting covariance matrix is the spatial covariance matrix corresponding to a random walk model for connectivity as in Hanks and Hooten (2013).

**Usage**

```
cov.from.dist(R)
```

**Arguments**

R                    A negative semi-definite matrix of squared differences.

**Value**

A positive semi-definite covariance matrix, for which the variogram (or resistance distance) is equal to the input R matrix.

**Author(s)**

Ephraim M. Hanks

**References**

Hanks and Hooten 2013. Circuit theory and model-based inference for landscape connectivity. *Journal of the American Statistical Association*. 108(501), 22-33.

Gower 1996. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika* 53(3), 325-338.

**Examples**

```
## create a Wishart covariance matrix with independent structure
Z=matrix(rnorm(10*20),ncol=20,nrow=10)
W=Z%*%t(Z)

## convert to resistance distance matrix
D=dist.from.cov(W)

## convert back to covariance matrix
C=cov.from.dist(D)

## compare C and W
max(abs(C-W))
```

dGenWish

*Density of the (singular) Generalized Wishart distribution***Description**

Computes the density of the (possibly singular) Generalized Wishart distribution with null-space equal to the space spanned by the "one" vector. This corresponds to the case considered by McCullagh (2009) and Hanks and Hooten (2013).

**Usage**

```
dGenWish(Dobs, Sigma, df, log=FALSE)
```

**Arguments**

Dobs	An observed squared-distance matrix.
Sigma	The covariance parameter of the Generalized Wishart.
df	An integer specifying the degrees of freedom.
log	Logical. If True, then the log-likelihood is computed.

**Details**

Following McCullagh (2009), the likelihood can be computed by considering any contrast matrix  $L$  of full rank, and with  $n-1$  rows and  $n$  columns, where  $n$  is the number of columns of 'Dobs'. If

$$Dobs \sim GenWish(Sigma, df, 1)$$

is distributed as a generalized Wishart distribution with kernel (null space) equal to the one vector, and  $df$  degrees of freedom, then the likelihood can be computed by computing the likelihood of

$$L(-Dobs)L' \sim Wishart(L(2*Sigma)L', df)$$

Additionally, following Srivastava (2003), this likelihood holds (up to a proportionality constant) in the singular case where  $df < n$ .

Following this formulation, the log-likelihood computed here (up to an additive constant) is

$$-df/2 * \log |L(2*Sigma)L'| - 1/2 * \text{tr} (L(2*Sigma)L')^{-1} L(-D)L'$$
**Value**

A numeric likelihood or log-likelihood

**Author(s)**

Ephraim M. Hanks

## References

- McCullagh 2009. Marginal likelihood for distance matrices. *Statistica Sinica* 19: 631-649.
- Srivastava 2003. Singular Wishart and multivariate beta distributions. *The Annals of Statistics*. 31(5), 1537-1560.
- Hanks and Hooten 2013. Circuit theory and model-based inference for landscape connectivity. *Journal of the American Statistical Association*. 108(501), 22-33.

## Examples

```
ras=raster(nrow=30,ncol=30)
extent(ras) <- c(0,30,0,30)
values(ras) <- 1
int=ras
cov.ras=ras

## get precision matrix of entire graph
B.int=get.TL(int)
Q.int=get.Q(B.int,1)

## get precision at a few nodes
Phi=get.Phi(Q.int,obs.idx=1:20)
S=ginv(as.matrix(Phi))

## simulate distance matrix
Dsim=rGenWish(df=20,Sigma=S)
image(Dsim)

## calculate log-likelihood
ll=dGenWish(Dsim,S,df=20,log=TRUE)
ll
```

---

dist.from.cov

*Compute a squared distance matrix from a covariance matrix.*

---

## Description

This computes a squared distance matrix from a covariance matrix, or other positive semi-definite matrix. The resulting squared distance matrix is the variogram matrix or the resistance distance matrix under a random walk model for connectivity as in Hanks and Hooten (2013).

## Usage

```
dist.from.cov(Sigma)
```

## Arguments

**Sigma** A symmetric positive definite matrix.



**Value**

A negative definite matrix of the same dimensions as Sigma.

**Author(s)**

Ephraim M. Hanks

**References**

Hanks and Hooten 2013. Circuit theory and model-based inference for landscape connectivity. Journal of the American Statistical Association. 108(501), 22-33.

**Examples**

```
## create a Wishart covariance matrix with independent structure

Z=matrix(rnorm(10*20),ncol=20,nrow=10)
W=Z %*% t(Z)

## convert to resistance distance matrix
D=dist.from.cov(W)

## convert back to covariance matrix
C=cov.from.dist(D)

## compare C and W
max(abs(C-W))
```

---

get.Phi

*Compute the precision matrix Phi of observed nodes on a graph.*

---

**Description**

Given a Gaussian Markov random field defined by a precision matrix Q, this returns Phi, which is the precision matrix of the nodes indexed by 'obs.idx', computed using the Schur complement.

**Usage**

```
get.Phi(Q, obs.idx)
```

**Arguments**

Q                    A precision matrix  
obs.idx             A vector of unique indices of observed nodes in the graph defined by Q.

**Value**

A square matrix of dimension equal to the length of 'obs.idx'.

**Author(s)**

Ephraim M. Hanks

**References**

Hanks and Hooten 2013. Circuit theory and model-based inference for landscape connectivity. Journal of the American Statistical Association. 108(501), 22-33.

**Examples**

```
int=raster(nrow=30,ncol=30)
values(int)=1
TL.int=get.TL(int)
Q.int=get.Q(TL.int,1)
## get precision matrix of only the first 5 nodes
get.Phi(Q.int,1:5)
```

---

get.Q	<i>Create a precision matrix from a transition list and a set of log-conductance rates.</i>
-------	---

---

**Description**

Creates a precision matrix  $Q$ , with off diagonal entries equal to  $\exp(b_1 x_{1ij} + \dots + b_p x_{pij})$ , where  $\beta = (b_1, b_2, \dots, b_p)$  is a vector of log-conductance values of the covariates  $x_{kij}$ . Each  $x_{kij}$  is equal to  $(x_{ki} + x_{kj})/2$ .

**Usage**

```
get.Q(TL, beta)
```

**Arguments**

TL	A transition list from TL.from.stack
beta	A vector of log-conductance rates with length equal to the length of TL.

**Value**

A precision matrix, as a sparse matrix of class 'dgCMatrix', with dimension equal to  $n^2$  by  $n^2$ , where  $n$  is the number of nodes in the raster stack used to compute TL.

**Author(s)**

Ephraim M. Hanks

**References**

Hanks and Hooten 2013. Circuit theory and model-based inference for landscape connectivity. Journal of the American Statistical Association. 108(501), 22-33.

**Examples**

```
int=raster(nrow=30,ncol=30)
values(int)=1
B.int=get.TL(int)
Q.int=get.Q(B.int,1)
```

---

get.TL

*Construct a transition list from a raster or raster stack*

---

**Description**

This computes a list of log-transition matrices as a preliminary step to creating a precision matrix from covariate rasters.

**Usage**

```
get.TL(rast.stack)
```

**Arguments**

`rast.stack`      A raster layer or raster stack object.

**Value**

A list of length equal to the number of raster layers in `rast.stack`. Each element in the list is a sparse Matrix of class 'dgCMatrix'.

**Author(s)**

Ephraim M. Hanks

**References**

Hanks and Hooten 2013. Circuit theory and model-based inference for landscape connectivity. Journal of the American Statistical Association. 108(501), 22-33.

**Examples**

```
int=raster(nrow=30,ncol=30)
values(int)=1
TL.int=get.TL(int)
Q.int=get.Q(TL.int,1)
image(Q.int)
```

---

mcmc.wish.icar	<i>Markov chain Monte Carlo sampler for Generalized Wishart distance matrix data arising from an ICAR spatial model.</i>
----------------	--

---

### Description

Constructs and runs an MCMC sampler to estimate resistance parameters of different landscape features.

### Usage

```
mcmc.wish.icar(Dobs, TL, obs.idx, df=1,
               beta.start = rep(0, length(TL)),
               beta.prior.mean = rep(0, length(TL)),
               beta.prior.cov = diag(10, length(TL)),
               tau.start = 0.1, tau.prior.var = 1,
               theta.tune = diag(10^-4, length(TL)+1),
               n.mcmc=100, adapt.max=10000, adapt.int=100,
               print.iter=FALSE, output.trace.plot=FALSE)
```

### Arguments

Dobs	A square symmetric matrix of observed pairwise distances. For example, a genetic distance matrix.
TL	A list of transition matrices for different covariate raster layers, created by get.TL
obs.idx	A vector of unique indices of observed nodes in the graph defined by the raster grid.
df	An integer specifying the degrees of freedom of Dobs.
beta.start	Vector of initial values for conductance parameters beta. Default is a vector of zeros.
beta.prior.mean	Vector of prior mean values for conductance parameters beta. Default is a vector of zeros.
beta.prior.cov	Matrix of the prior covariance matrix for conductance parameters beta. Default is a diagonal matrix with diagonal entries equal to 10.
tau.start	Numeric starting value for the nugget variance tau. Default is 0.1
tau.prior.var	Variance of the half-normal prior for tau. Default is 1.
theta.tune	Covariance matrix for the random walk MH sampler for all parameters. Default is a diagonal matrix with variance $10^{-4}$ .
n.mcmc	Integer number of iterations of the MCMC sampler to run.
adapt.max	Integer number (or INF) specifying the last iteration at which the covariance matrix of the proposal distribution will be adapted. Default is $10^5$ .
adapt.int	Interval at which the covariance matrix of the proposal distribution is adapted. Default is every 100 iterations.

`print.iter` Logical. If TRUE, then the current state of the system will be printed to the console every 100 iterations.

`output.trace.plot` Logical. If TRUE, then the trace plots of the sampler will be saved to "trace-Out.pdf" every 100 iterations.

### Details

Runs an MCMC sampler to draw samples from the posterior distribution of model parameters ( $\beta, \tau$ ) from the following model for an observed squared distance matrix  $D_{obs}$ :

$-D_{obs} \sim \text{GenWish}(2 * \Sigma, df)$

$\Sigma = K(\Psi)K' + \tau * I$

where  $\Psi$  is the covariance matrix of the observed nodes of a graph described by the transition list TL. That is, the total graph has Laplacian (precision) matrix  $Q$ , with off-diagonal entries of  $Q$  given by

$Q_{ij} = \exp(b_0 + b_1 x_{1ij} + \dots + b_p x_{pij})$ , where  $\beta = (b_1, b_2, \dots, b_p)$  is a vector of log-conductance values of the covariates  $x_{kij}$ . Each  $x_{kij}$  is equal to  $(x_{ki} + x_{kj})/2$ .

The prior on  $\beta$  is  $N(\beta.prior.mean, \beta.prior.cov)$ , and the prior on  $\tau$  is  $\tau \sim \text{Half\_Normal}(0, \tau.prior.var)$ .

### Value

A list containing output from the MCMC sampler.

`beta` Posterior samples for conductance parameters  $\beta$ .

### Author(s)

Ephraim M. Hanks

### References

Hanks and Hooten 2013. Circuit theory and model-based inference for landscape connectivity. Journal of the American Statistical Association. 108(501), 22-33.

### Examples

```
## Not run:
## The following code conducts a simulation example by
## first simulating a heterogeneous landscape, then
## simulating pairwise distance data on the landscape
## and finally making inference on model parameters.

library(rwc)
library(MASS)

## source("GenWishFunctions_20170901.r")

##
```

```
## specify 2-d raster
##

ras=raster(nrow=30,ncol=30)
extent(ras) <- c(0,30,0,30)
values(ras) <- 1
plot(ras,asp=1)
ras

int=ras
cov.ras=ras

## simulate "resistance" raster
TL.int=get.TL(int)
Q.int=get.Q(TL.int,1)
set.seed(1248)
## values(cov.ras) <- as.numeric(rnorm.Q(Q.int
values(cov.ras) <- as.numeric(rnorm.Q(Q.int,zero.constraint=TRUE))
plot(cov.ras)

stack=stack(int,cov.ras)
plot(stack)
TL=get.TL(stack)

## Create "barrier" raster which has no effect, to test model selection

barrier=int
values(barrier) <- 0
barrier[,10:11] <- 1

plot(barrier)

TL.all=get.TL(stack(int,cov.ras,barrier))

##
## sampling locations
##

nsamps=150
set.seed(4567)
samp.xy=cbind(runif(nsamps,0,30),runif(nsamps,0,30))
## samp.xy=samp.xy[rep(1:nsamps,times=5),]
samp.locs=cellFromXY(int,samp.xy)
samp.cells=unique(samp.locs)
nsamps=nrow(samp.xy)

plot(cov.ras)
```

```

points(samp.xy)

K=matrix(0,nrow=nsamps,ncol=length(samp.cells))
for(i in 1:nsamps){
  K[i,which(samp.cells==samp.locs[i])]=1
}
image(K)

##
## beta values
##

betas=c(-2,-1)
tau=.01

Q=get.Q(TL,betas)
Phi=get.Phi(Q,samp.cells)

## simulate from ibr model
D.rand.ibr=rGenWish(df=20,Sigma=K%*%ginv(as.matrix(Phi))%*%t(K) + diag(nsamps)*tau)
image(D.rand.ibr)

## crude plot of geographic distance vs genetic distance

plot(as.numeric(as.matrix(dist(xyFromCell(ras,samp.locs)))),as.numeric(D.rand.ibr))

##
## fitting using MCMC
##

fit=mcmc.wish.icar(D.rand.ibr,TL,samp.locs,df=20,output.trace.plot=TRUE,
                  adapt.int=100,adapt.max=100000,n.mcmc=10000)

str(fit)

## End(Not run)

```

**Description**

Simulates Wishart random variables, then computes the induced distance of the simulated Wishart random variables. The result is a random matrix distributed as a Generalized Wishart random variable.

**Usage**

```
rGenWish(Sigma, df)
```

**Arguments**

Sigma	The covariance parameter of the Generalized Wishart.
df	An integer specifying the degrees of freedom.

**Value**

A matrix of dimension equal to the dimension of Sigma.

**Author(s)**

Ephraim M. Hanks

**References**

McCullagh 2009. Marginal likelihood for distance matrices. *Statistica Sinica* 19: 631-649.

**Examples**

```
ras=raster(nrow=30,ncol=30)
extent(ras) <- c(0,30,0,30)
values(ras) <- 1
int=ras
cov.ras=ras

## get precision matrix of entire graph
B.int=get.TL(int)
Q.int=get.Q(B.int,1)

## get precision at a few nodes
Phi=get.Phi(Q.int,obs.idx=1:20)
S=ginv(as.matrix(Phi))

## simulate distance matrix
Dsim=rGenWish(df=20,Sigma=S)
image(Dsim)

## calculate log-likelihood
ll=dGenWish(Dsim,S,df=20,log=TRUE)
ll
```



---

rnorm.Q

*Sample random normal variables with precision matrix Q.*


---

**Description**

General function to make use of sparse matrix methods to efficiently simulate random multivariate normal random variables with sparse precision matrices.

**Usage**

```
rnorm.Q(Q, mu = rep(0, nrow(Q)), X = Matrix(1, nrow = nrow(Q), ncol =
1),
zero.constraint = FALSE, canon = FALSE, diag.adjust = .Machine$double.eps * 10)
```

**Arguments**

Q	Precision matrix, defined as a sparse Matrix object.
mu	Mean vector of dimension equal to the dimension of Q.
X	Matrix of vectors which should be orthogonal to the simulated random variable.
zero.constraint	If TRUE, then the simulated random variable is orthogonal to the columns of X.
canon	If TRUE, then draw from the 'canonical form'.
diag.adjust	Numeric value to be added to the diagonal of Q to make it positive definite.

**Details**

In the 'canonical form', the variable is drawn from:

$$v \sim N(Q^{-1} \mu, Q^{-1})$$

In the non-canonical form, the variable is drawn from

$$v \sim N(\mu, Q^{-1})$$

**Value**

A vector of the resulting random variable.

**Author(s)**

Ephraim M. Hanks

**References**

Rue and Held 2005. Gaussian Markov Random Fields: theory and applications. Chapman and Hall.

**Examples**

```
ras=raster(nrow=30,ncol=30)
extent(ras) <- c(0,30,0,30)
values(ras) <- 1

int=ras
cov.ras=ras

## simulate "resistance" raster
B.int=get.TL(int)
Q.int=get.Q(B.int,1)
values(cov.ras) <- as.numeric(rnorm.Q(Q.int,zero.constraint=TRUE))
plot(cov.ras)
```

# Index

`cov.from.dist`, [5](#)

`dGenWish`, [7](#)

`dist.from.cov`, [8](#)

`get.Phi`, [9](#)

`get.Q`, [10](#)

`get.TL`, [11](#)

`mcmc.wish.icar`, [12](#)

`rGenWish`, [15](#)

`rnorm.Q`, [17](#)

`rwc` (`rwc-package`), [2](#)

`rwc-package`, [2](#)