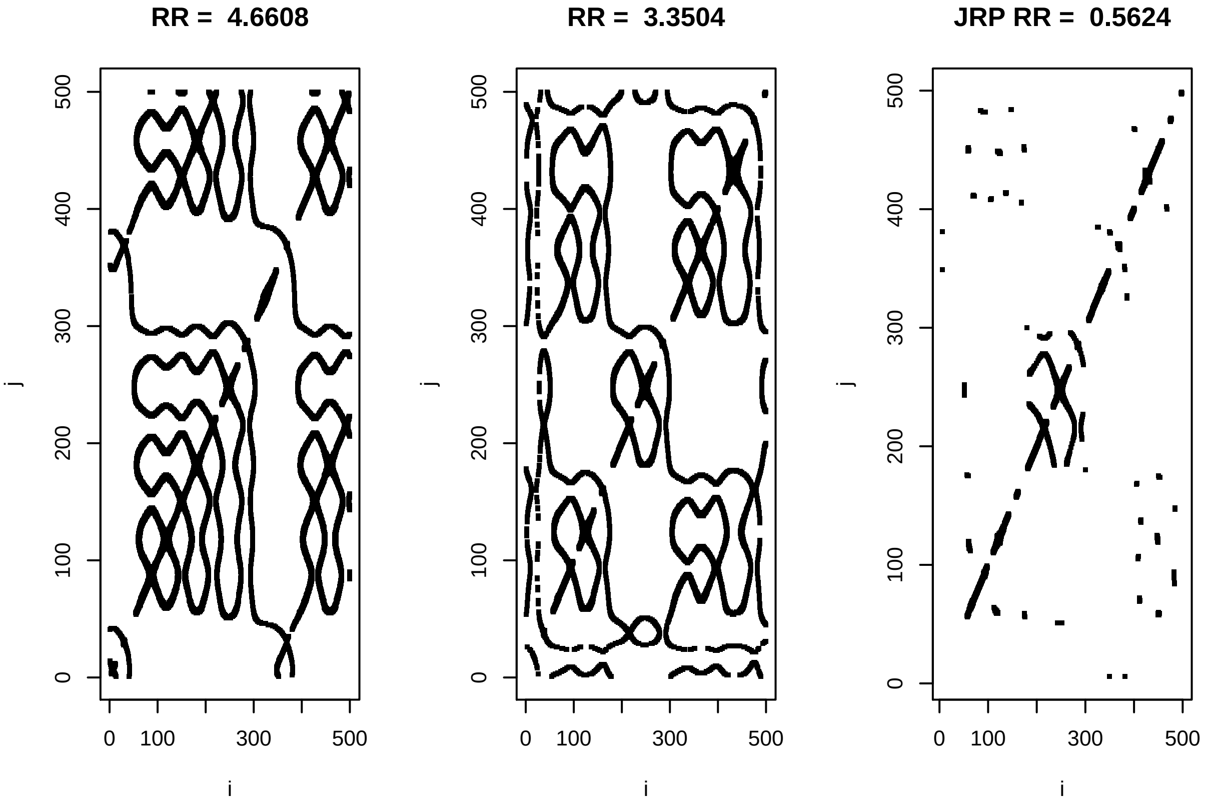
```
# for details on using crqa, see:
# Coco, M. I., & Dale, R. (2014). Cross-recurrence quantification analysis of
# categorical and continuous time series: an R package. Frontiers in
# Quantitative Psychology and Measurement, 5
# http://cognaction.org/rdmaterials/php.cv/pdfs/article/coco_dale_2014.pdf
# please cite if you use crqa -- thanks!

library(crqa)
```

```
## Loading required package: Matrix
## Loading required package: tseriesChaos
## Loading required package: deSolve
## Loading required package: fields
## Loading required package: spam
## Loading required package: grid
## Spam version 1.0-1 (2014-09-09) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
##
## Attaching package: 'spam'
##
## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve
##
## Loading required package: maps
## Loading required package: pracma
##
## Attaching package: 'pracma'
##
## The following object is masked from 'package:deSolve':
##
##     rk4
##
## The following objects are masked from 'package:Matrix':
##
##     expm, lu, tril, triu
```

```
setwd('~/Downloads/')
source('lorenzattractor.R')

par(mfrow=c(1,3)) # let's plot the RPs in columns to compare

# build some lorenz time series with subtly different parameters
dt = 0.01
sigma = 10
b = 8/3
numsteps = 500
plots = F
lorenz_1 = lorenzattractor(numsteps, dt, sigma, r = 28, b, plots)
```

```
## Loading required package: plot3D
```

```
lorenz_2 = lorenzattractor(numsteps, dt, sigma, r = 35, b, plots)

# do crqa for first lorenz
res_1 = crqa(lorenz_1[, 1], lorenz_1[, 1], delay = 1, embed = 1,
             rescale = 1, radius = 5, normalize = 0,
           minvertline = 2, mindiagline = 2, tw = 1, whiteline = F,
           recpt = F)

# build first lorenz RP
RP_1 = as.matrix(res_1$RP) # convert into numeric non-sparse matrix
ij = which(RP_1==1,arr.ind=T) # get coordinates
plot(ij[,1],ij[,2],
     xlab='i',ylab='j',pch=15,cex=.5,main=paste('RR = ',res_1$RR))

# do the same for second lorenz
res_2 = crqa(lorenz_2[, 1], lorenz_2[, 1], delay = 1, embed = 1,
             rescale = 1, radius = 5, normalize = 0,
             minvertline = 2, mindiagline = 2, tw = 1, whiteline = F,
             recpt = F)

RP_2 = as.matrix(res_2$RP) # convert into numeric non-sparse matrix
ij = which(RP_2==1,arr.ind=T) # get coordinates
plot(ij[,1],ij[,2],
     xlab='i',ylab='j',pch=15,cex=.5,main=paste('RR = ',res_2$RR))

# jrp = simply the multiplication of the two
JRP = (RP_1*RP_2)

# how we feed in JRP matrix into crqa function
JRP_res = crqa(JRP, delay = 1, embed = 1, rescale = 1, radius = 5, normalize = 0,
             minvertline = 2, mindiagline = 2, tw = 1, whiteline = F, recpt = T)
             # note recpt parameter = T

# let's plot the JRP, too
ij = which(JRP==1,arr.ind=T) # get coordinates
plot(ij[,1],ij[,2],
     xlab='i',ylab='j',pch=15,cex=.5,main=paste('JRP RR = ',JRP_res$RR))
```


RR = 4.6608
RR = 3.3504
JRP RR = 0.5624
j
i
0
100
200
300
400
500
0
100
300
500