

Data Visualization Techniques of Single and Multiple Spike Trains

Wanxin Li

Joint work with Professor Reza Ramezan

Contents

1	Testing Data	3
2	Crosscorrelation Histogram	5
2.1	Description	5
2.2	Notes	5
2.3	References	5
2.4	Testing Code	6
2.4	Interpretation	6
3	Crosscorrelation Histogram with Brillinger Normalization	7
3.1	Description	7
3.2	Notes	7
3.3	References	7
3.4	Testing Code	7
3.5	Interpretation	9
4	Correlation Grid	10
4.1	Description	10
4.2	Notes	10
4.3	References	10
4.4	Testing Code	10
4.5	Interpretation	22
5	Gravity Transformation	25
5.1	Description	25
5.2	Notes	25
5.3	References	25
5.4	Testing Code	25
5.5	Interpretation	31
6	Interspike Interval Plot	32
6.1	Description	32
6.2	References	32
6.3	Testing Code	32
6.4	Interpretation	33
7	Joint Peristimulus Time Histogram	35
7.1	Description	35
7.2	References	35
7.3	Testing Code	35
7.4	Interpretation	40
8	Peristimulus Time Histogram	41
8.1	Description	41
8.2	References	41

8.3 Testing Code	41
8.4 Interpretation	42
9 Raster plot	43
9.1 Description	43
9.2 References	43
9.3 Testing Code	43
9.4 Interpretation	44
10 iRaster Plot	45
10.1 Description	45
10.2 Notes	45
10.3 References	45
10.4 Testing Code	45
10.5 Interpretation	53
11 Tunnel Plot	54
11.1 Description	54
11.2 References	54
11.3 Testing Code	54
11.4 Interpretation	55
12 Modified Tunnel Plot	56
12.1 Description	56
12.2 References	56
12.3 Testing Code	56
12.4 Interpretation	57
13 Snowflake Plot	58
13.1 Description	58
13.2 Notes	58
13.3 References	58
13.4 Testing Code	58
13.5 Interpretation	62
14 Spike Train Synchrony: Bivariate Distance	63
14.1 Description	63
14.2 References	63
14.3 Testing Code	63
14.4 Interpretation	65
15 Spike Train Synchrony: Multivariate Diversity	67
15.1 Description	67
15.2 Notes	67
15.3 References	67
15.4 Testing Code	67
15.5 Interpretation	68

1 Testing Data

Single Spike Train

mn120L15_1_LGN: the spike time over 5s from mn120L15_1_LGN
A: first trial of mn120L15_1_LGN
mn120L15_1_RGC: the spike time over 5s from mn120L15_1_RGC
B: first trial of mn120L15_1_RGC
mn121R7_1_LGN: the spike time over 5s from mn121R7_1_LGN
C: first trial of mn121R7_1_LGN
mn121R7_1_RGC: the spike time over 5s from mn121R7_1_RGC
D: first trial of mn121R7_1_RGC
mn122R4_2_LGN: the spike time over 5s from mn122R4_2_LGN
E: first trial of mn122R4_2_LGN
mn122R4_2_RGC: the spike time over 5s from mn122R4_2_RGC
F: first trial of mn122R4_2_RGC
mn122R4_3_LGN: the spike time over 5s from mn122R4_3_LGN
G: first trial of mn122R4_3_LGN
mn122R4_3_RGC: the spike time over 5s from mn122R4_3_RGC
H: first trial of mn122R4_3_RGC

Multiple Spike Trains

mn120L15_1_RGC.10: first 10 trials of mn120L15_1_RGC
mn121R7_1_LGN.10: first 10 trials of mn121R7_1_LGN

```
source("../Common/check.packages.R")
packages <- c("STAR")
check.packages(packages)

## STAR
## TRUE

# testing data
## single spike train
mn120L15_1_LGN <- dget("../Data/mn120L15_1_LGN.txt")
A <- mn120L15_1_LGN[[1]]
mn120L15_1_RGC <- dget("../Data/mn120L15_1_RGC.txt")
B <- mn120L15_1_RGC[[1]]
mn121R7_1_LGN <- dget("../Data/mn121R7_1_LGN.txt")
C <- mn121R7_1_LGN[[1]]
mn121R7_1_RGC <- dget("../Data/mn121R7_1_RGC.txt")
D <- mn121R7_1_RGC[[1]]
mn122R4_2_LGN <- dget("../Data/mn122R4_2_LGN.txt")
E <- mn122R4_2_LGN[[1]]
mn122R4_2_RGC <- dget("../Data/mn122R4_2_RGC.txt")
F <- mn122R4_2_RGC[[1]]
mn122R4_3_LGN <- dget("../Data/mn122R4_3_LGN.txt")
G <- mn122R4_3_LGN[[1]]
mn122R4_3_RGC <- dget("../Data/mn122R4_3_RGC.txt")
H <- mn122R4_3_RGC[[1]]

## multiple spike trains
mn120L15_1_LGN <- dget("../Data/mn120L15_1_LGN.txt")
mn120L15_1_RGC <- dget("../Data/mn120L15_1_RGC.txt")
mn121R7_1_LGN <- dget("../Data/mn121R7_1_LGN.txt")
```

```
## multiple spike trains
mn120L15_1_RGC.10 <- as.repeatedTrain(head(mn120L15_1_RGC, 10))
mn121R7_1_LGN.10 <- as.repeatedTrain(head(mn121R7_1_LGN, 10))
mn120L15_1_LGN.10 <- as.repeatedTrain(head(mn120L15_1_LGN,10))

## multiple spike trains
mn120L15_1_RGC.20 <- as.repeatedTrain(head(mn120L15_1_RGC, 20))
mn121R7_1_LGN.20 <- as.repeatedTrain(head(mn121R7_1_LGN, 20))
mn120L15_1_LGN.20 <- as.repeatedTrain(head(mn120L15_1_LGN,20))
```

2 Crosscorrelation Histogram

2.1 Description

- Indicate the firing rate of one neuron (the “target” neuron) versus another (the “reference” neuron)

2.2 References

Readings/Crosscorrelation Histogram/Crosscorrelation.pdf

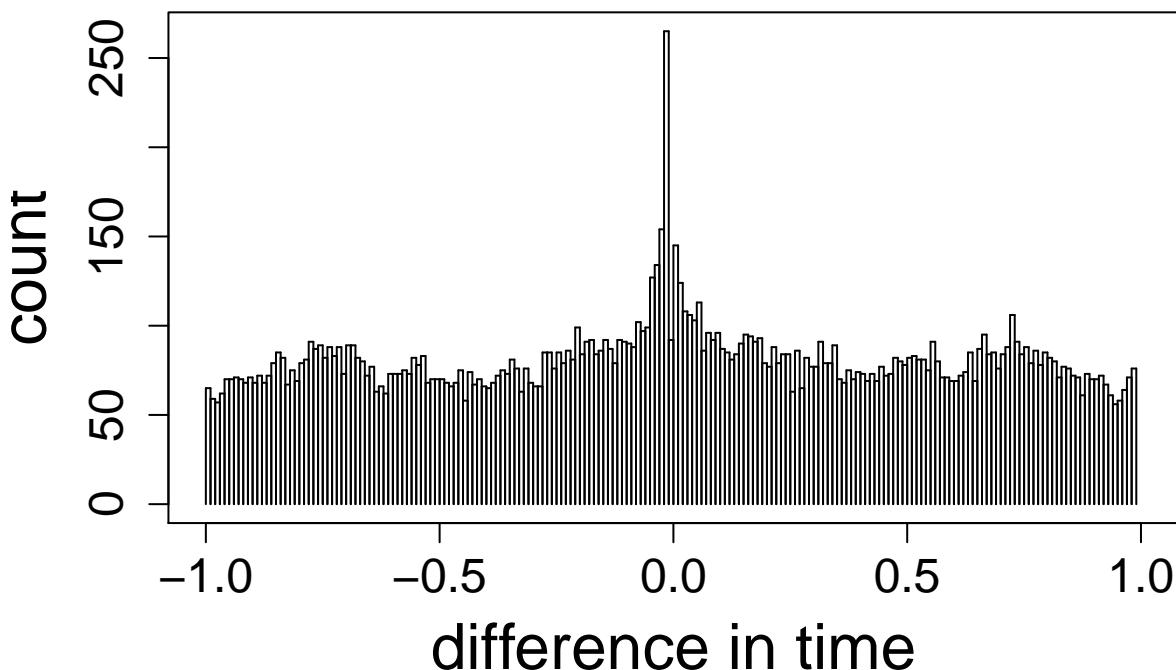
2.3 Testing Code

```
source("../Data/testing_data.R")
source("../CCH/CCH.R")

# testing set up
time.span <- 5
bin.width <- 0.01
bin.num <- 200

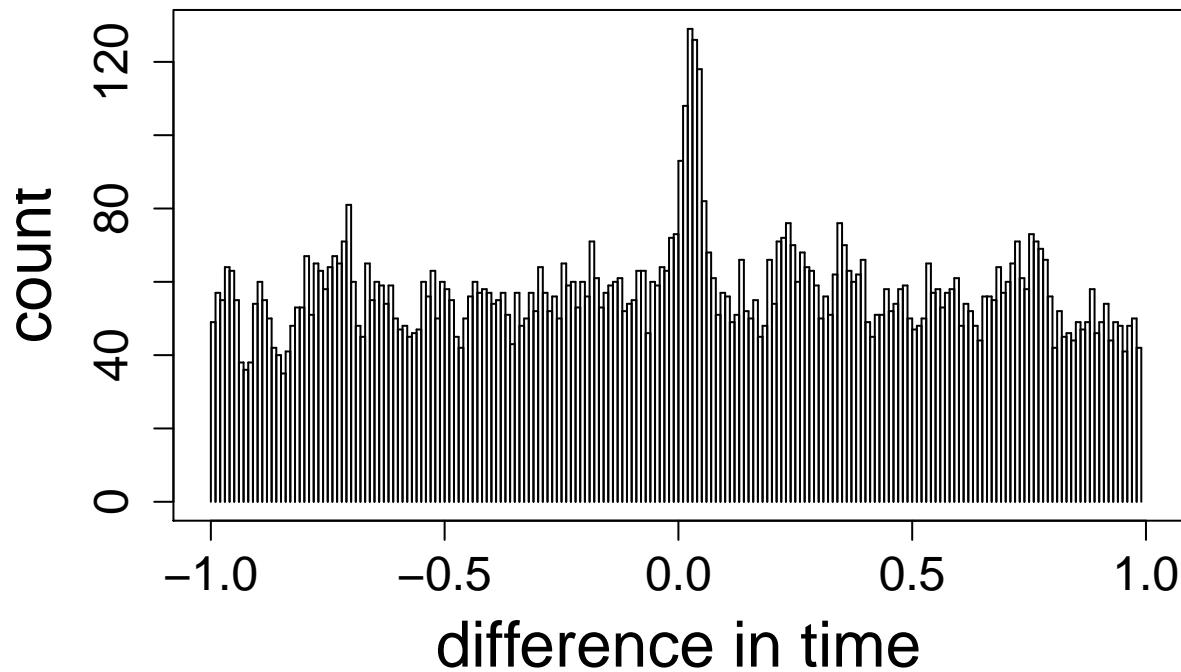
## CCH of reference spike train A and target spike train B
## in a time span of 5 seconds and with the bin being 0.01 second and 200 bins
# CCH_AB.pdf
CCH.plot(A, B, time.span, bin.width, bin.num)

## [1] 200
```



```
## CCH of reference spike train A and target spike train B
## in a time span of 5 seconds and with the bin being 0.01 second and 200 bins
# CCH_AC.pdf
CCH.plot(A, C, time.span, bin.width, bin.num)
```

```
## [1] 200
```



2.4 Interpretation

- General
 - CCH of neurons that are connected tends to have a peak around 0.
 - CCH of neurons that are not connected tend to have a flat plot.
 - We can find the time lapse between the firing time of two neurons by comparing the x-coordinate of the main peak and x = 0.
- CCH_AB.pdf
 - Main peak is 200 counts.
 - A, B are highly correlated.
 - The time lapse of the main peak is 0, which indicates A and B are strongly connected.
- CCH_AC.pdf
 - Main peak is above 100 counts.
 - The connection between A and C is not as strong as A and B.
 - The time lapse of the main is slightly greater than 0, which indicates whenever C fires, A usually has a spike before C.

3 Crosscorrelation Histogram with Brillinger Normalization

3.1 Description

- Apply Brillinger Normalization on Crosscorrelation Histogram
- Brillinger normalization “standardizes” CCH to make different CCHs comparable

3.2 Notes

- Since data is not stationary, Brillinger normalization might not be valid.

3.3 References

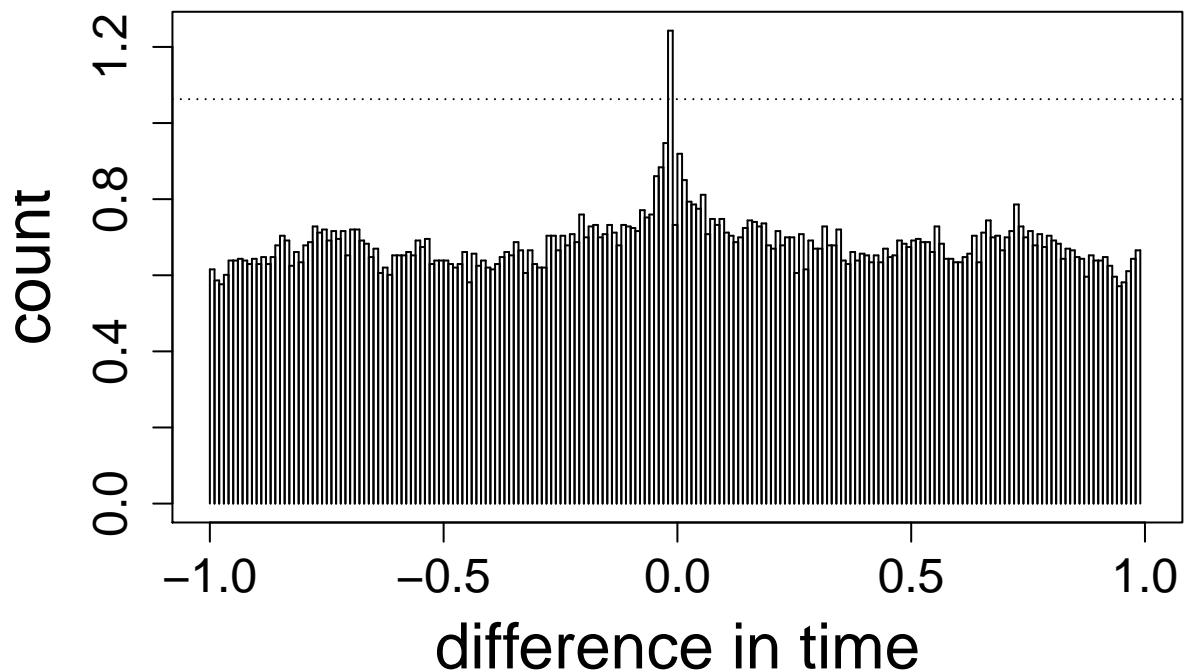
Readings/Crosscorrelation Histogram/ACG Paper _May 2017.pdf
Readings/Crosscorrelation Histogram/selecta.pdf

3.4 Testing Code

```
source("../Data/testing_data.R")
source("../CCH/bn.R")

# testing set up
time.span <- 5
bin.width <- 0.01
bin.num <- 200

## CCH of reference spike train A and target spike train B
## in a time span of 5 seconds and with the bin width being 0.01 second and 200 bins
## and apply Brillinger Normalisation with confidence level 0.95
## AB_CCH_bn_95.pdf
bn.calc(A, B, time.span, bin.width, bin.num, plot.bn = TRUE)
```

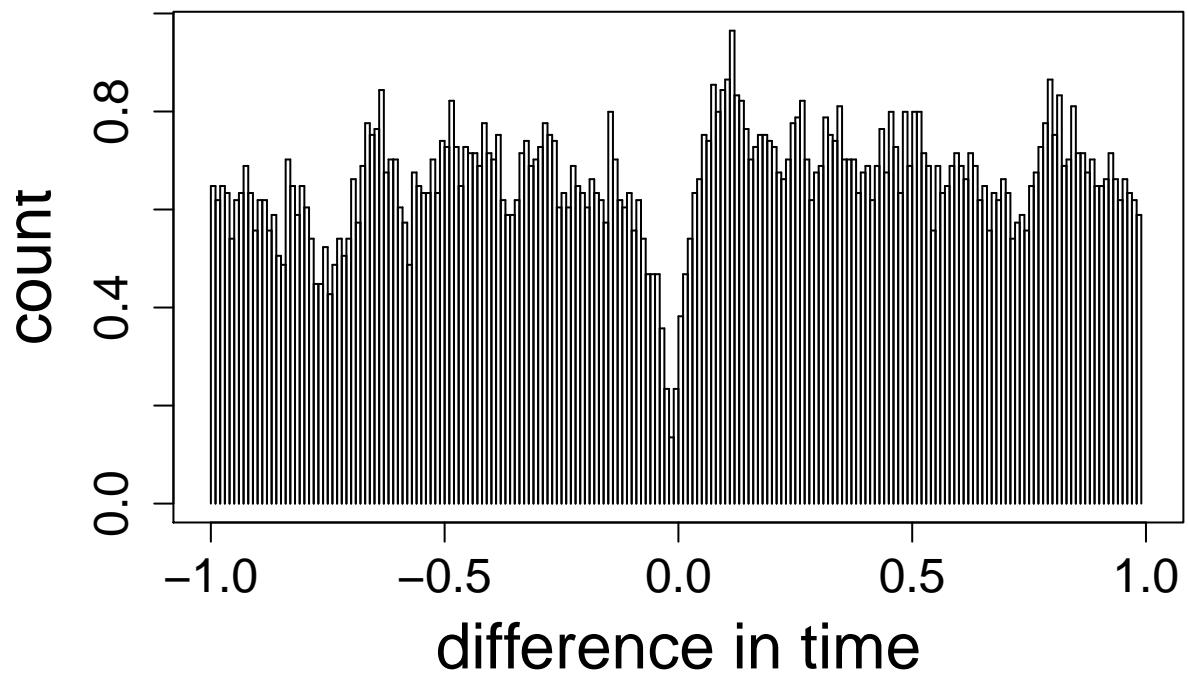


```

## [1] 1.242694
## get the time lapse of the main peak of
## CCH of reference spike train A and target spike train B
## in a time span of 5 seconds and with the bin width being 0.01 second and 200 bins
## and apply Brillinger Normalisation with confidence level 0.95
bn.lapse(A, B, time.span, bin.width, bin.num)

## [1] 0
## CCH of reference spike train A and target spike train F
## in a time span of 5 seconds and with the bin width being 0.01 second and 200 bins
## and apply Brillinger Normalisation with confidence level 0.95
## AB_CCH_bn_95.pdf
bn.calc(A, F, time.span, bin.width, bin.num, plot.bn = TRUE)

```



```
## [1] 0
```

3.5 Interpretation

- AB_CCH_bn_95.pdf
 - Main peak after Brilinger Normalization is 1.242684.
 - Main peak exceeds 95% confidence interval (dotted line).
- AF_CCH_bn_95.pdf
 - Dotted line is not plotted since main peak is way below 95% confidence interval.
 - The trough in t = 0 suggests a inhibitory effects between A and F.

4 Correlation Grid

4.1 Description

- Visualize the connection strength between neurons pairwise
- Reorder neurons by performing hierarchical clustering using complete linkage method on the strength of correlation
- Six measures are used in measuring the similarity bewteen spike trains: coincidence (coin.calc), correlation index (cor.index.calc), correlation tiling (cor.tiling.calc), simple (simple.calc), CCH with Brillinger normalizaton (bn.calc) and firing rate (fr.calc). Larger return value means more similar.
- (Advanced correlation grid) Classify neurons by functional connection: directly connected, common source connected or indirect connected.

4.2 Notes

- The classification of functional connection has not been implemented. There is no data to test common-source and indirect connection. And I do not know how to choose the boundaries of time lapses and main peaks for dividing groups. The modified z-score (in the examples below) can be used in future implementation of this functionality (Section 5.2.2 of the reference paper).

4.3 References

Readings/Correlation Grid/Stuart et al_2005.pdf

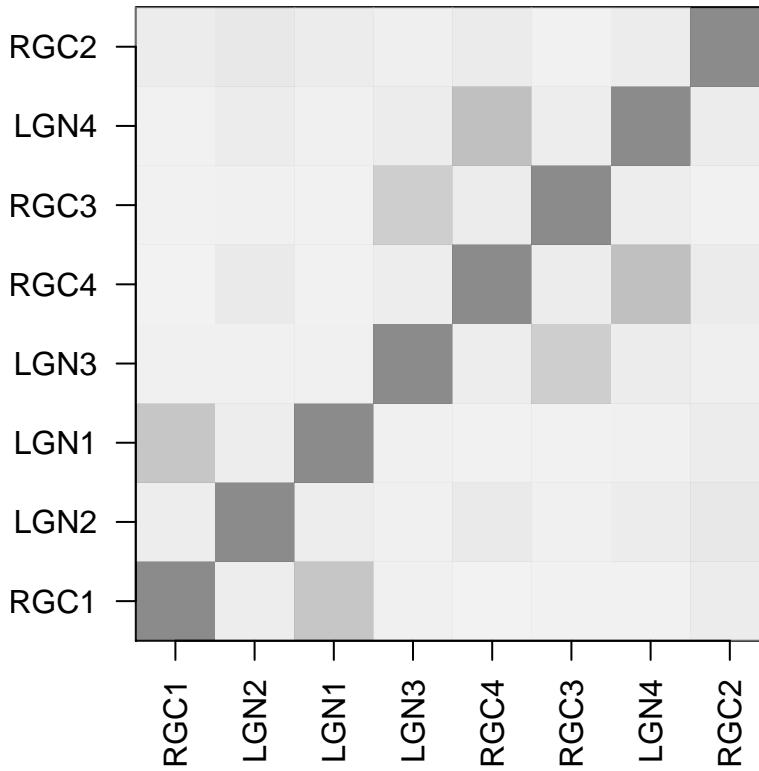
Readings/Correlation Grid/ACG Paper _May 2017.pdf

4.4 Testing Code

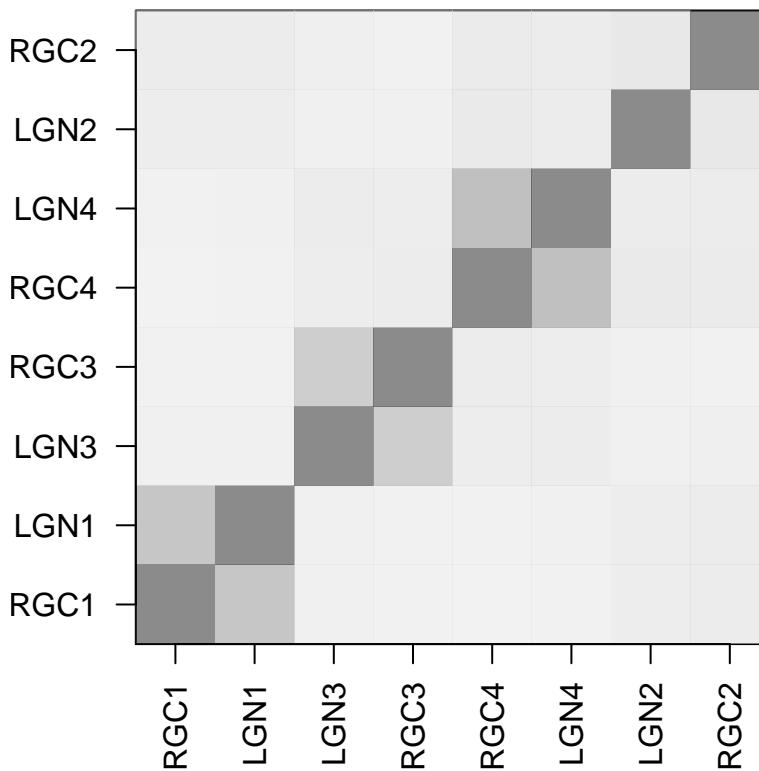
```
# library(rstudioapi)
# wd <- dirname(rstudioapi::getActiveDocumentContext()$path)
# setwd(wd)
source("../Correlation Grid/cg.R")
source("../Data/testing_data.R")

##### sorted correlation grid by different similarity measures #####
spike.trains <- list(B, C, A, E, H, F, G, D)
labels <- list("RGC1", "LGN2", "LGN1", "LGN3", "RGC4", "RGC3", "LGN4", "RGC2")

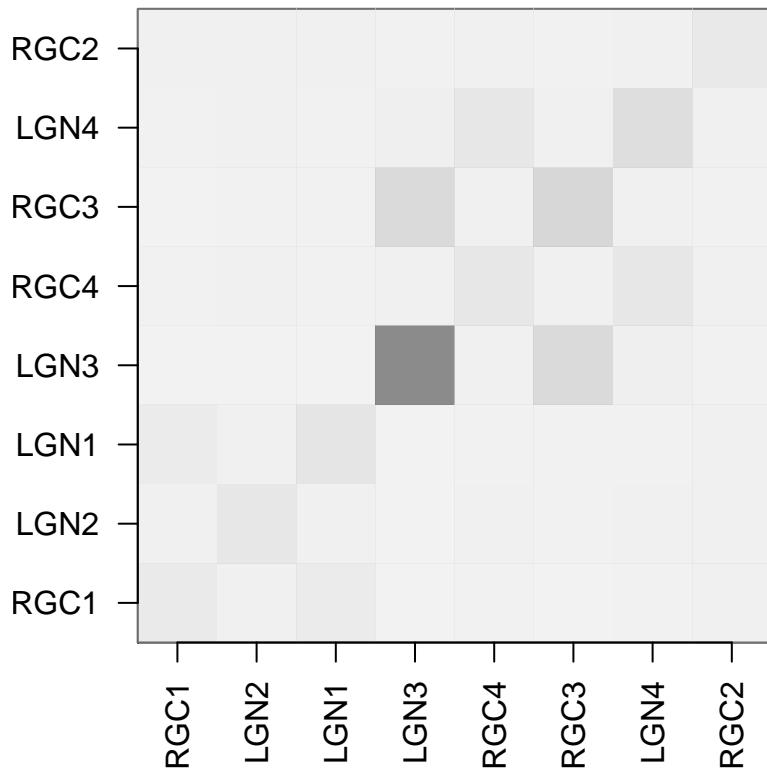
par(mar=c(4,5,2,2))
# unsorted
# coin_unsorted.pdf
cg(spike.trains, labels = labels)
```



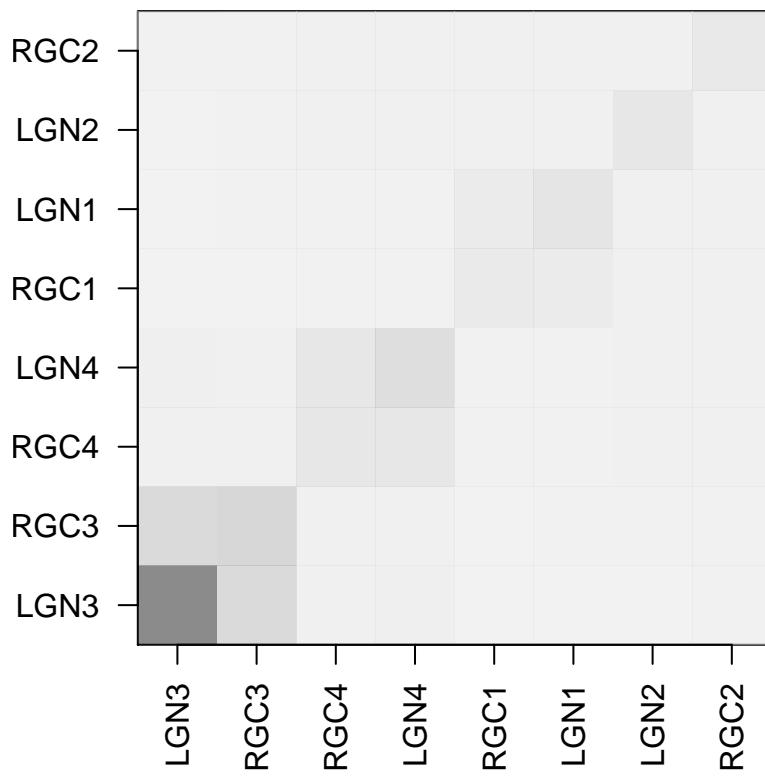
```
# sort by coin.calc
# coin_sorted.pdf
sorted.trains <- sort.trains(spike.trains)
sorted.labels <- sort.labels(spike.trains, labels)
cg(sorted.trains, labels = sorted.labels)
```



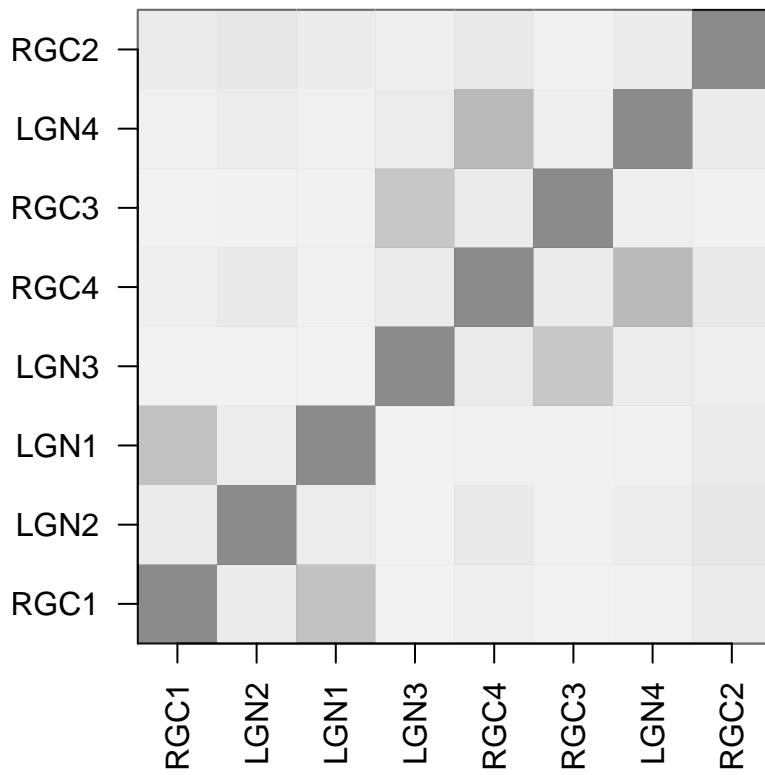
```
# unsorted
# cor_index_unsorted.pdf
cg(spike.trains, labels = labels, "cor.index.calc")
```



```
# sort by cor.index.calc
# cor_index_sorted.pdf
sorted.trains <- sort.trains(spike.trains, "cor.index.calc")
sorted.labels <- sort.labels(spike.trains, labels, "cor.index.calc")
cg(sorted.trains, labels = sorted.labels, sort.func = "cor.index.calc")
```



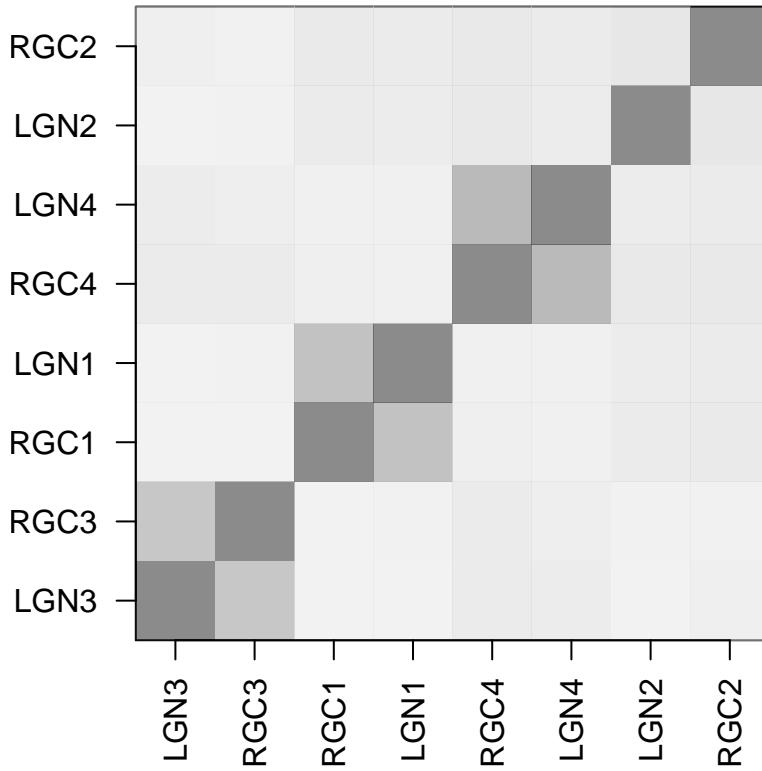
```
# unsorted
# cor_tiling_unsorted.pdf
cg(spike.trains, labels = labels, "cor.tiling.calc")
```



```

# sort by cor.tiling.calc
# cor_tiling_sorted.pdf
sorted.trains <- sort.trains(spike.trains, "cor.tiling.calc")
sorted.labels <- sort.labels(spike.trains, labels, "cor.tiling.calc")
cg(sorted.trains, labels = sorted.labels, sort.func = "cor.tiling.calc")

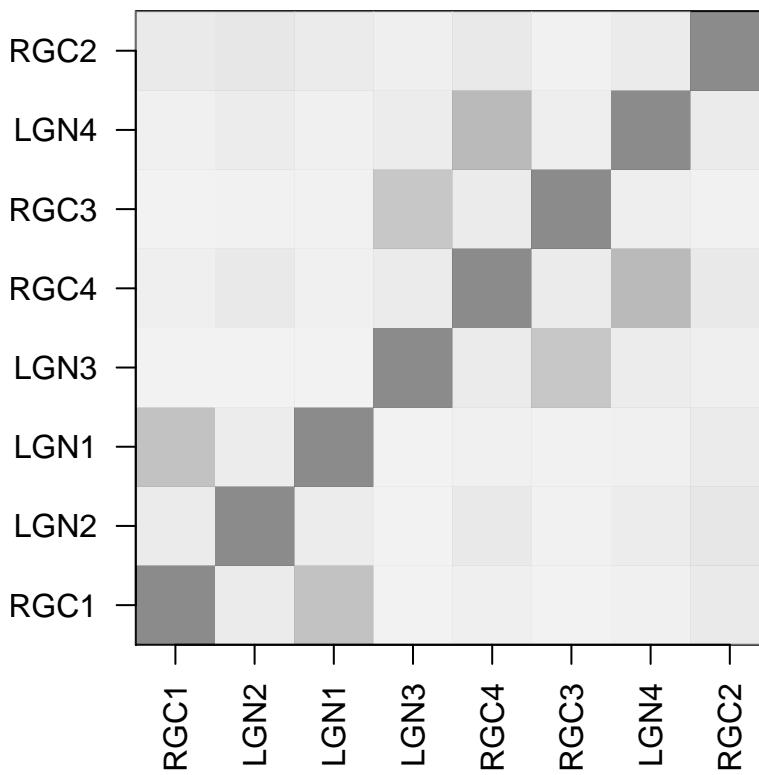
```



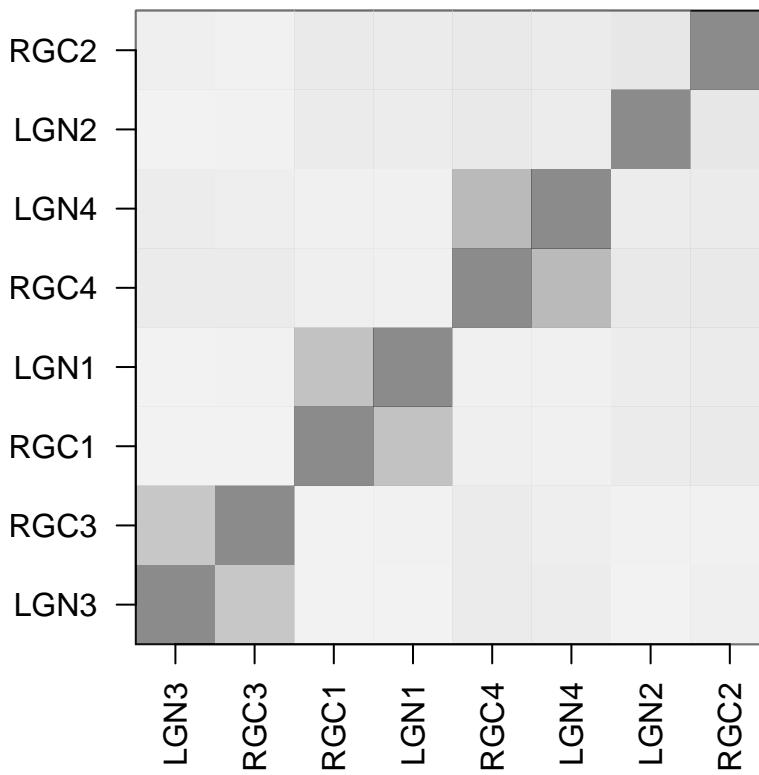
```

# unsorted
# simple_unsorted.pdf
cg(spike.trains, labels = labels, "simple.calc")

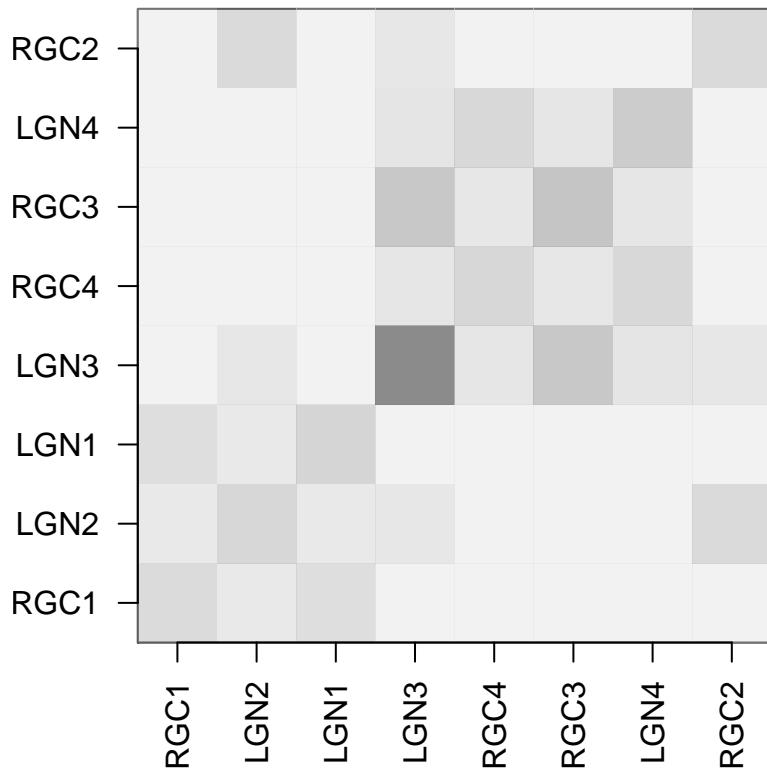
```



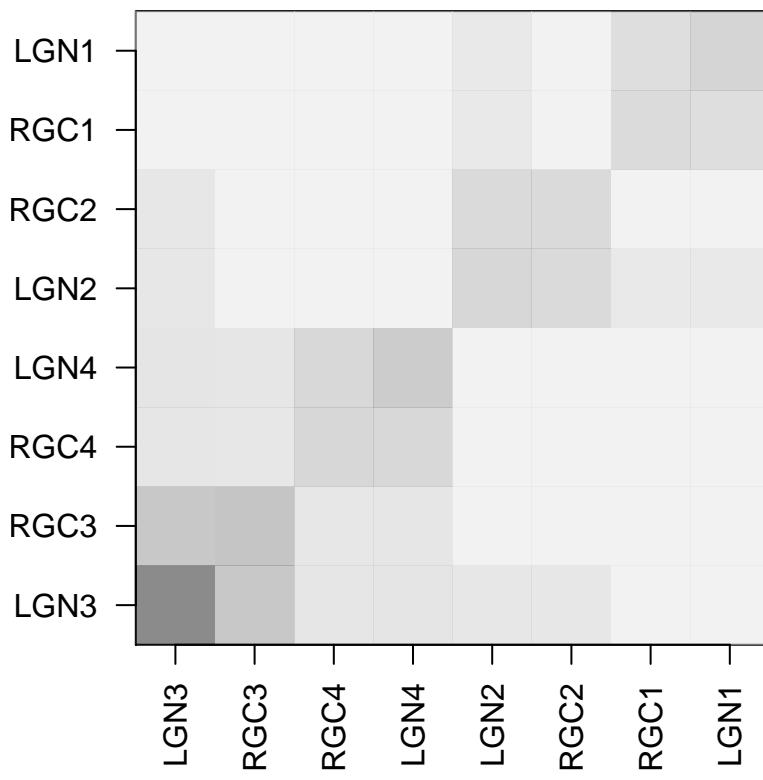
```
# sort by cor.tiling.calc
# simple_sorted.pdf
sorted.trains <- sort.trains(spike.trains, "simple.calc")
sorted.labels <- sort.labels(spike.trains, labels, "simple.calc")
cg(sorted.trains, labels = sorted.labels, sort.func = "simple.calc")
```



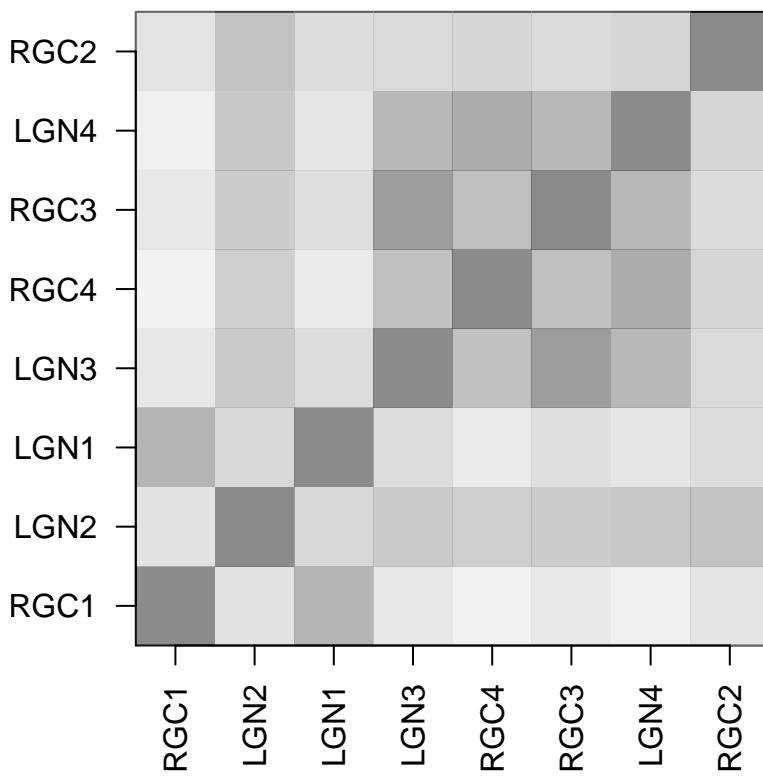
```
# unsorted
# bn_unsorted.pdf
cg(spike.trains, labels = labels, "bn.calc")
```



```
# sort by CCH after Brillinger Normalisation
# bn_sorted.pdf
sorted.trains <- sort.trains(spike.trains, "bn.calc")
sorted.labels <- sort.labels(spike.trains, labels, "bn.calc")
cg(sorted.trains, labels = sorted.labels, sort.func = "bn.calc")
```



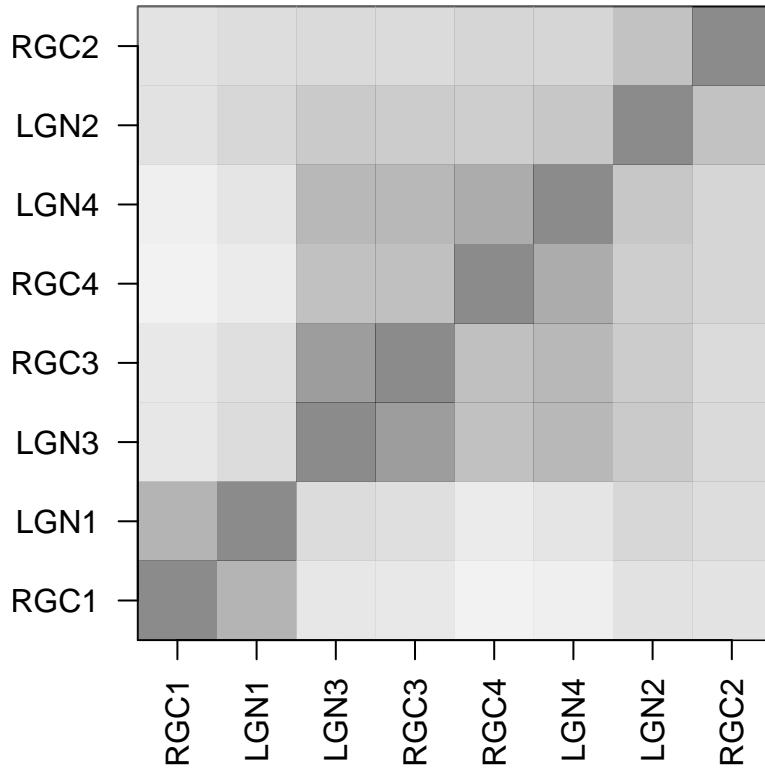
```
# unsorted
# fr_unsorted.pdf
cg(spike.trains, labels = labels, "fr.calc")
```



```

# sort by fr.calc
# fr_sorted.pdf
sorted.trains <- sort.trains(spike.trains, "fr.calc")
sorted.labels <- sort.labels(spike.trains, labels, "fr.calc")
cg(sorted.trains, labels = sorted.labels, sort.func = "fr.calc")

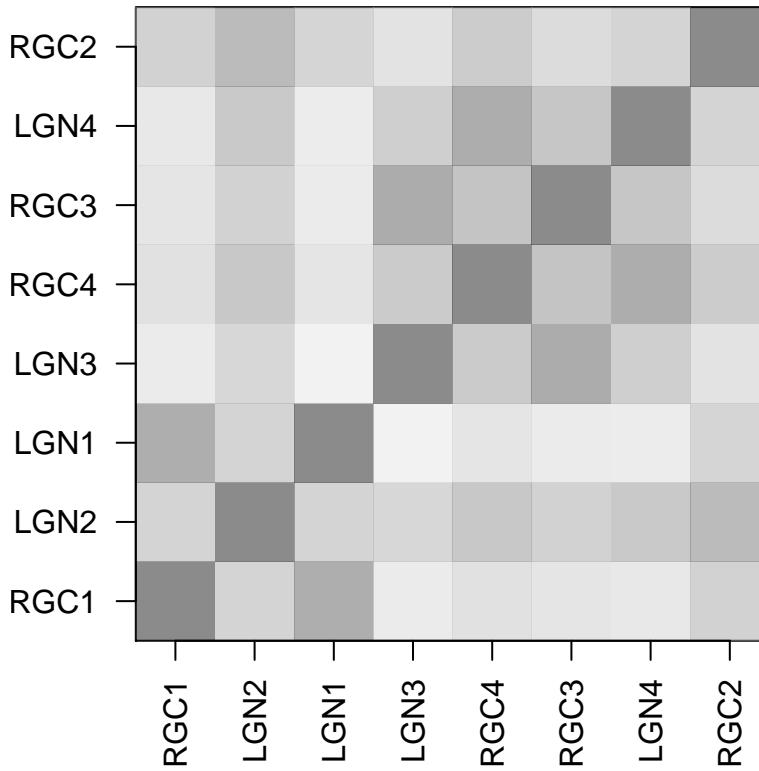
```



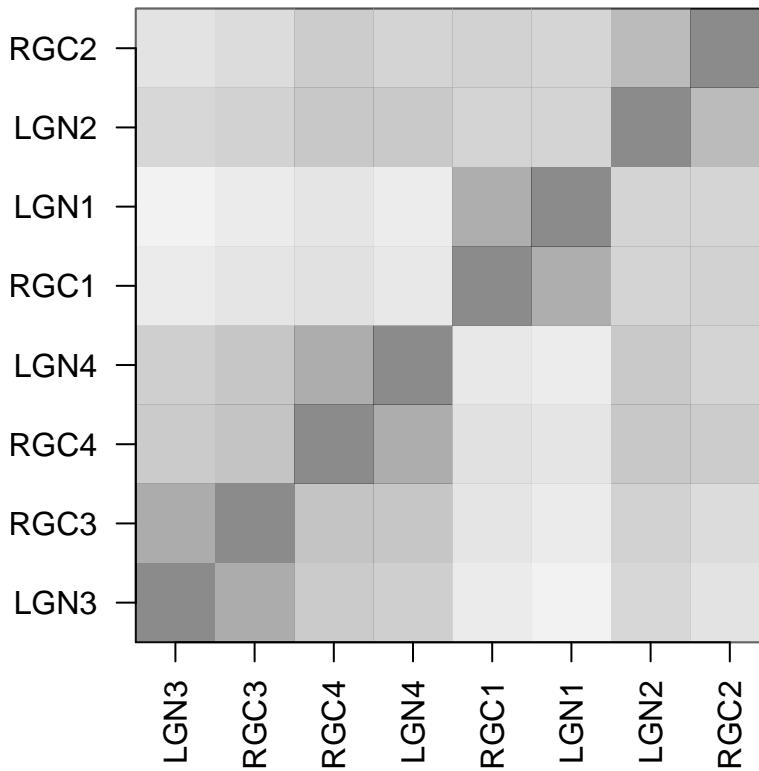
```

# unsorted
# fr_adj_unsorted.pdf
cg(spike.trains, labels = labels, "fr.adj.calc")

```



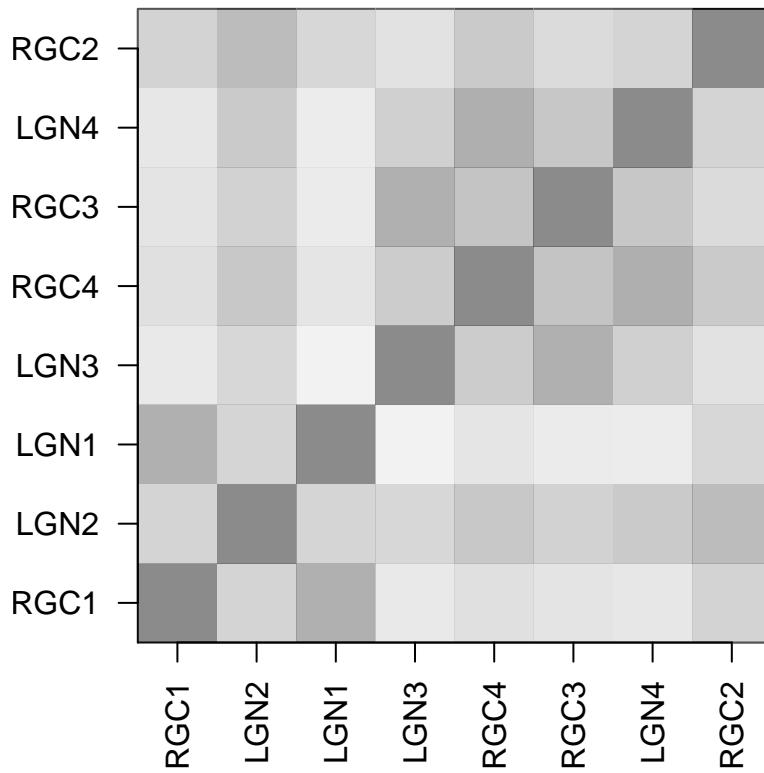
```
# sort by fr.adj.calc
# fr_adj_sorted.pdf
sorted.trains <- sort.trains(spike.trains, "fr.adj.calc")
sorted.labels <- sort.labels(spike.trains, labels, "fr.adj.calc")
cg(sorted.trains, labels = sorted.labels, sort.func = "fr.adj.calc")
```



```

# unsorted
# fr_adj_unsorted.pdf
cg(spike.trains, labels = labels, "fr.spline.adj.calc")

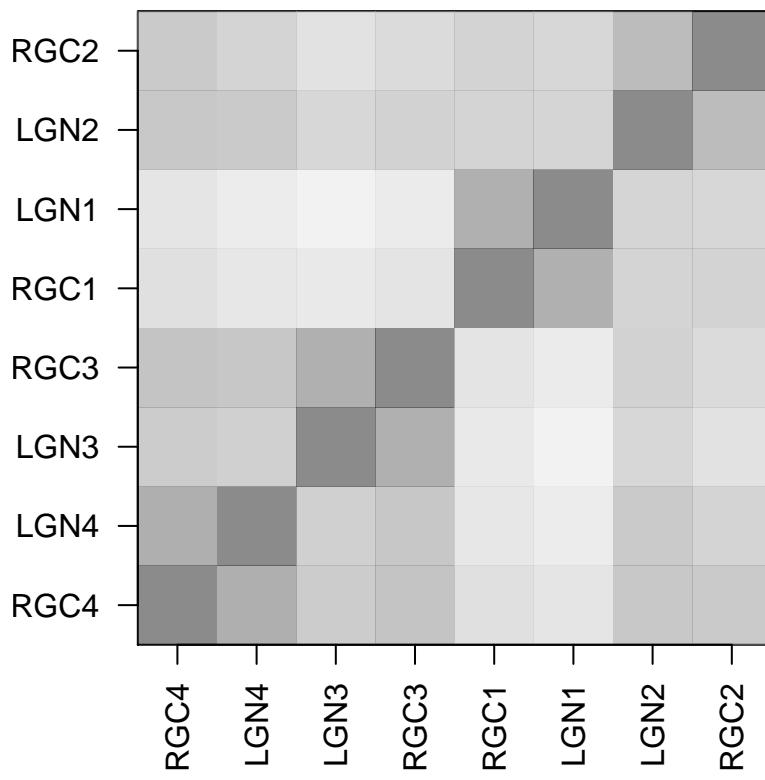
```



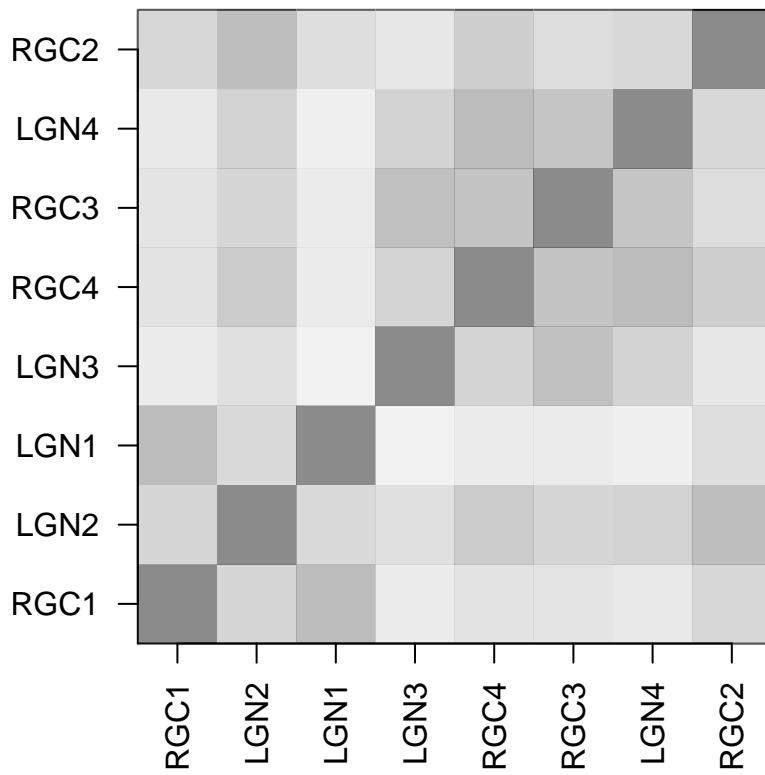
```

# sort by fr.adj.calc
# fr_adj_sorted.pdf
sorted.trains <- sort.trains(spike.trains, "fr.spline.adj.calc")
sorted.labels <- sort.labels(spike.trains, labels, "fr.spline.adj.calc")
cg(sorted.trains, labels = sorted.labels, sort.func = "fr.spline.adj.calc")

```



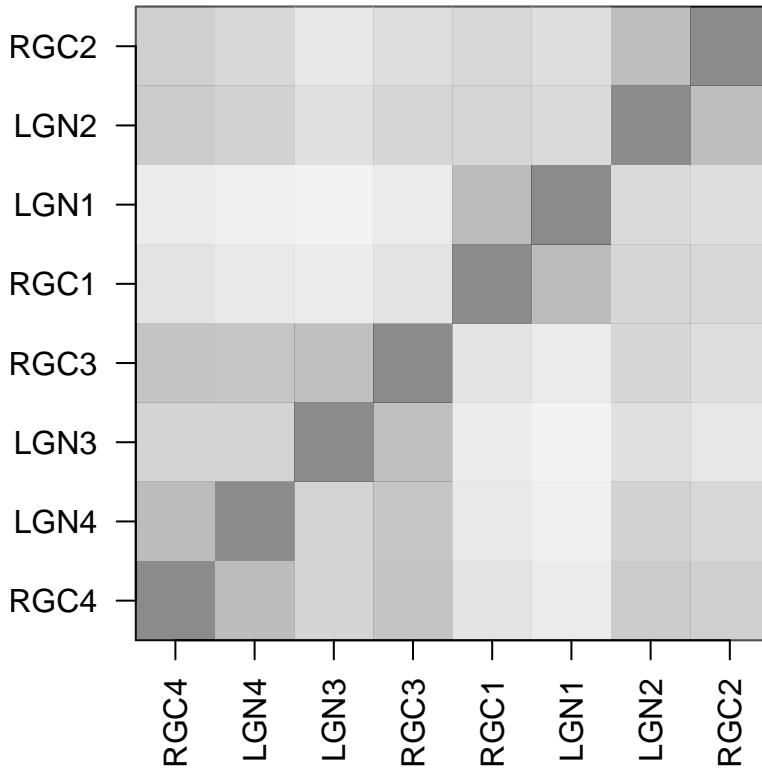
```
# unsorted
# fr_avg_spline_unsorted.pdf
cg(spike.trains, labels = labels, "fr.avg.spline.calc")
```



```

# sort by fr.avg.spline.calc
# fr_avg_spline_sorted.pdf
sorted.trains <- sort.trains(spike.trains, "fr.avg.spline.calc")
sorted.labels <- sort.labels(spike.trains, labels, "fr.avg.spline.calc")
cg(sorted.trains, labels = sorted.labels, sort.func = "fr.avg.spline.calc")

```



```
#####
Detecting outliers using modified z-score #####

```

```

time.span <- 5
bin.width <- 0.01
bin.num <- 200
# modified z-score of A and B
z.score(A, B, time.span, bin.width, bin.num)

```

```
## [1] 11.53455
```

```
# modified z-score of A and C
z.score(A, C, time.span, bin.width, bin.num)
```

```
## [1] 0
```

```
# modified z-score of E and F
z.score(E, F, time.span, bin.width, bin.num)
```

```
## [1] 7.275485
```

4.5 Interpretation

- General
 - Darker color of the cell means larger value returned by the corresponding measure.
- coin_unsorted.pdf

- coin_sorted.pdf
 - sort coin_unsorted.pdf by coin.calc.
 - We can tell from the plot from left to right that B and A, E and F, H and G, C and D are highly correlated, which is consistent with the source of data.
- cor_index_unsorted.pdf
- cor_index_sorted.pdf
 - sort coin_unsorted.pdf by cor.index.calc.
 - We can tell from the plot from left to right that E and F, H and G, B and A, C and D are highly correlated, which is consistent with the source of data.
 - From the darkness of this plot, we can tell the return values of cor.index.calc are not as large as coin.calc. Note that only E vs. E has a dark value.
- cor_tiling_unsorted.pdf
- cor_tiling_sorted.pdf
 - sort coin_unsorted.pdf by cor.tiling.calc.
 - We can tell from the plot from left to right that E and F, H and G, B and A, C and D are highly correlated, which is consistent with the source of data.
- simple_unsorted.pdf
- simple_sorted.pdf
 - sort coin_unsorted.pdf by simple.calc.
 - We can tell from the plot from left to right that E and F, B and A, H and G, C and D are highly correlated, which is consistent with the source of data.
- bn_unsorted.pdf
- bn_sorted.pdf
 - sort coin_unsorted.pdf by bn.calc.
 - We can tell from the plot from left to right that E and F, H and G, C and D, B and A are highly correlated, which is consistent with the source of data.
 - This one has similar darkness as cor_tiling_sorted.pdf.
- fr_unsorted.pdf
- fr_sorted.pdf
 - sort coin_unsorted.pdf by fr.calc
 - We can tell from the plot from left to right that B and A, E and F, H and G, C and D are highly correlated, which is consistent with the source of data.
 - We can also see a dark square around E, F, H and G, which indicates there might be some connection bewteen these four neurons.
- fr_adj_unsorted.pdf
- fr_adj_sorted.pdf
 - sort coin_unsorted.pdf by fr.adj.calc
 - We can tell from the plot from left to right that E and F, H and G, B and A, C and D are highly correlated, which is consistent with the source of data.
 - We can also see a dark square around E, F, H and G, which indicates there might be some connection bewteen these four neurons.
- fr_adj_unsorted.pdf
- fr_adj_sorted.pdf
 - sort coin_unsorted.pdf by fr.spline.adj.calc
 - We can tell from the plot from left to right that H and G, E and F, B and A, C and D are highly correlated, which is consistent with the source of data.
 - We can also see a dark square around E, F, H and G, which indicates there might be some connection bewteen these four neurons.
- fr_avg_spline_unsorted.pdf
- fr_avg_spline_sorted.pdf
 - sort coin_unsorted.pdf by fr.avg.spline.calc
 - We can tell from the plot from left to right that H and G, E and F, B and A, C and D are highly correlated, which is consistent with the source of data.
 - We can also see a dark square around H, G, E and F, which indicates there might be some

connection bewteen these four neurons.

5 Gravity Transformation

5.1 Description

- Apply Gravity Transformation on multiple spike trains to measure the similarity bewteen pairs
- Measure the synchrony bewteen pairs of spike trains dynamically over time
- Visualize the result by pairwise distance plots over time and iterative parallel coordinates

5.2 Notes

- We have not figured out how to code this procedure to n neurons. Coding n as parameter requires an unfixed number of variables and an unfixed number of differential equations. The current impmlementation hardcodes 3 neurons and 4 neurons.
- When $n > 4$, the computational cost becomes extremely large due to ode solver.
- The result of the distance plot of 4 neurons is not consistent with the source of data. Possible reasons are noted in interpretation.
- The gif in R Code/Gravity Transformation/gif is a better representation of parallel coordinates.
- We can adjust the value for a , b and τ for different inputs.

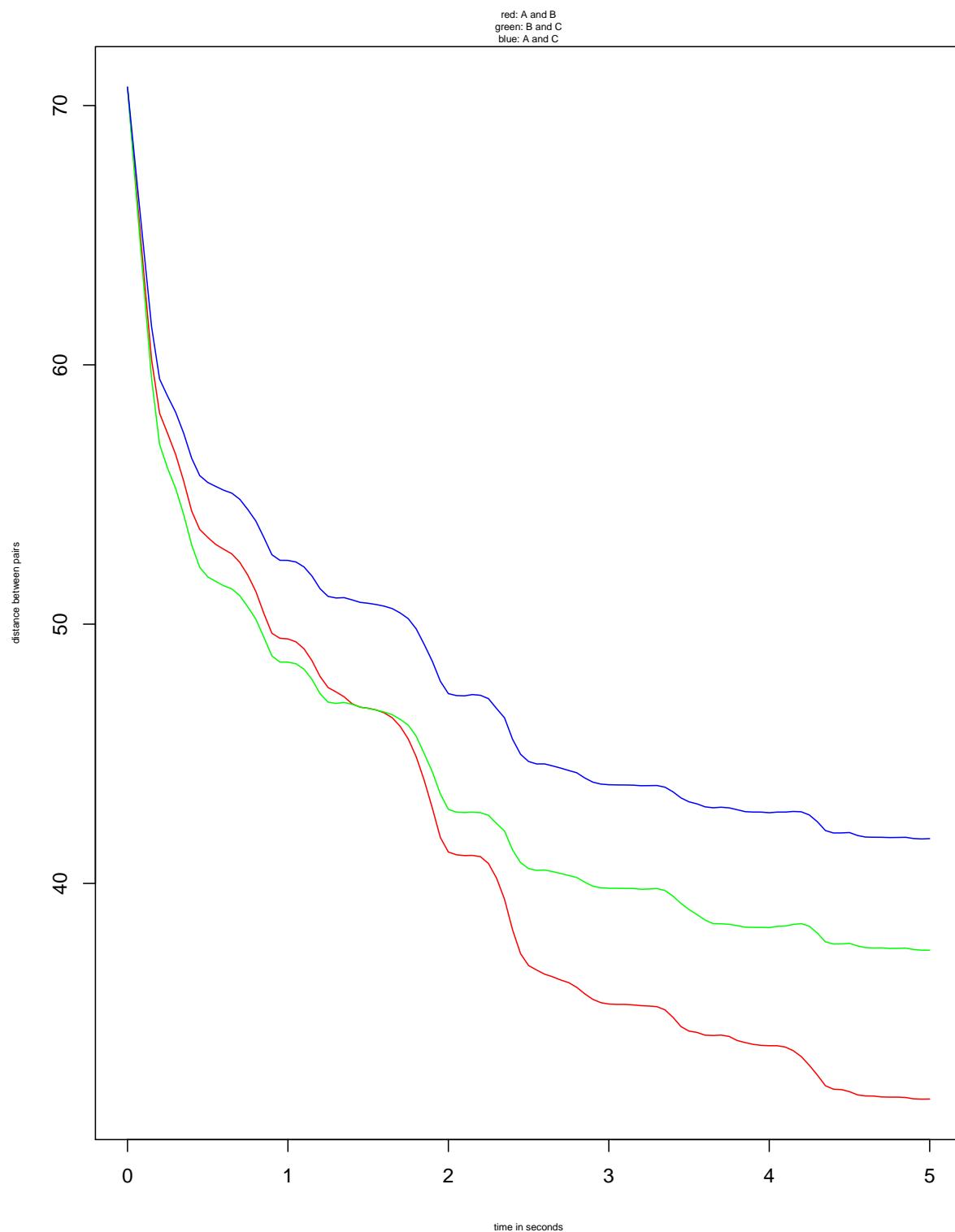
5.3 References

Readings/Gravity Transformation/Paper_Biosystems_2002.pdf
Readings/Gravity Transformation/Gravity Transformation.pdf

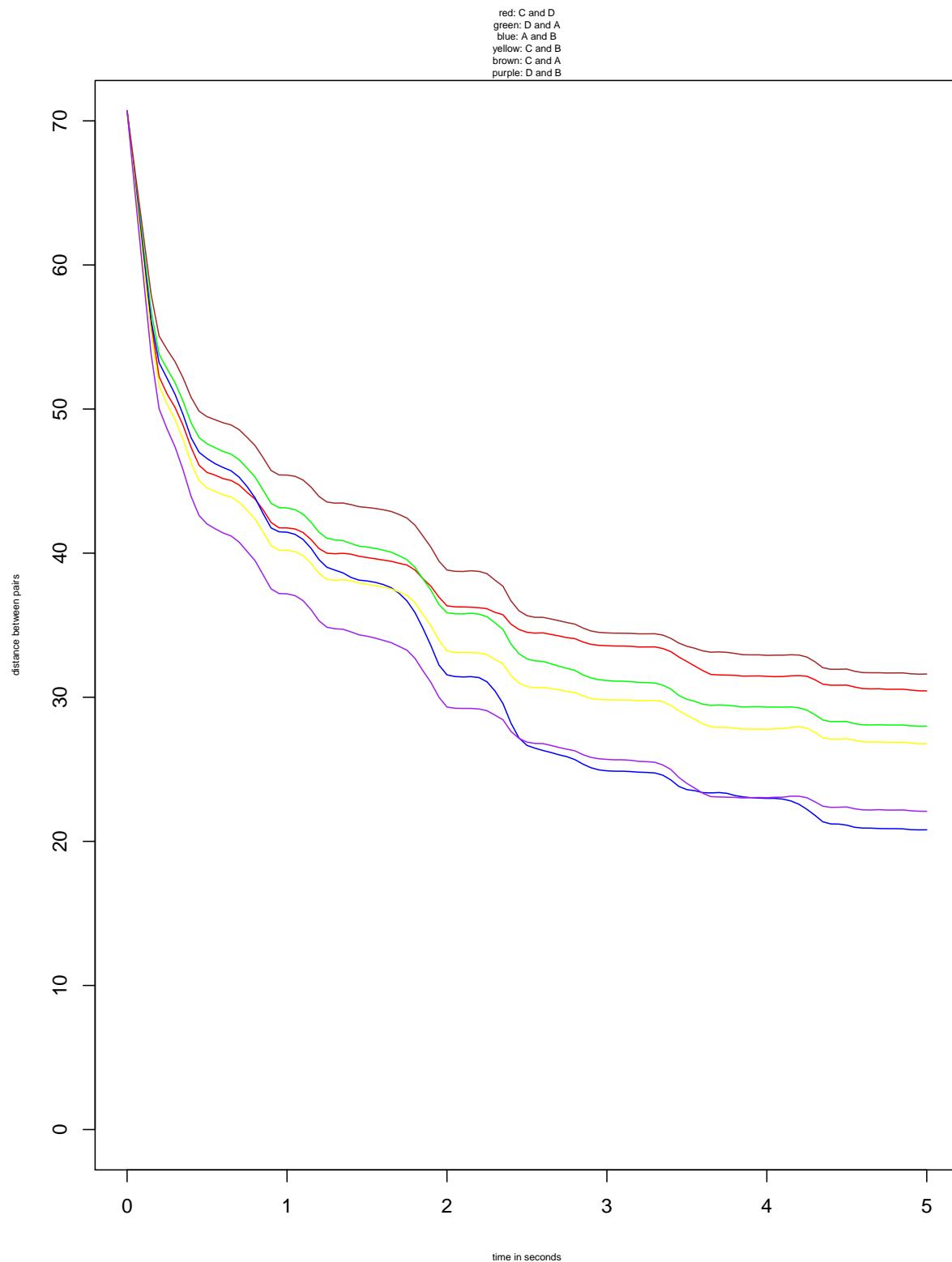
5.4 Testing Code

Note: pdf cannot be generated. Refer to code in R Code/Gravity Transformation/

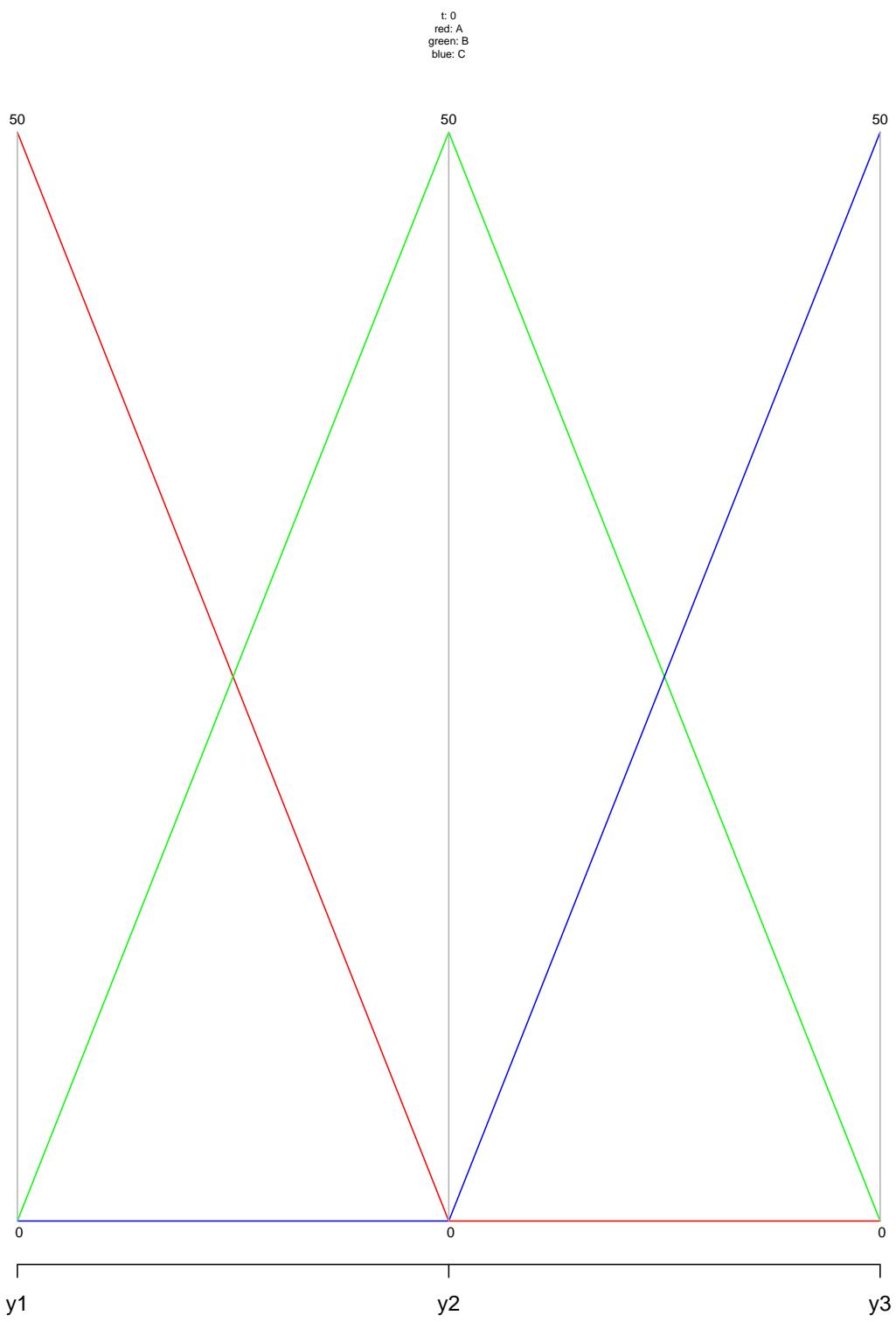
```
# manually include plots  
knitr:::include_graphics('..../Gravity Transformation/Plots/gt_ABC.pdf')
```



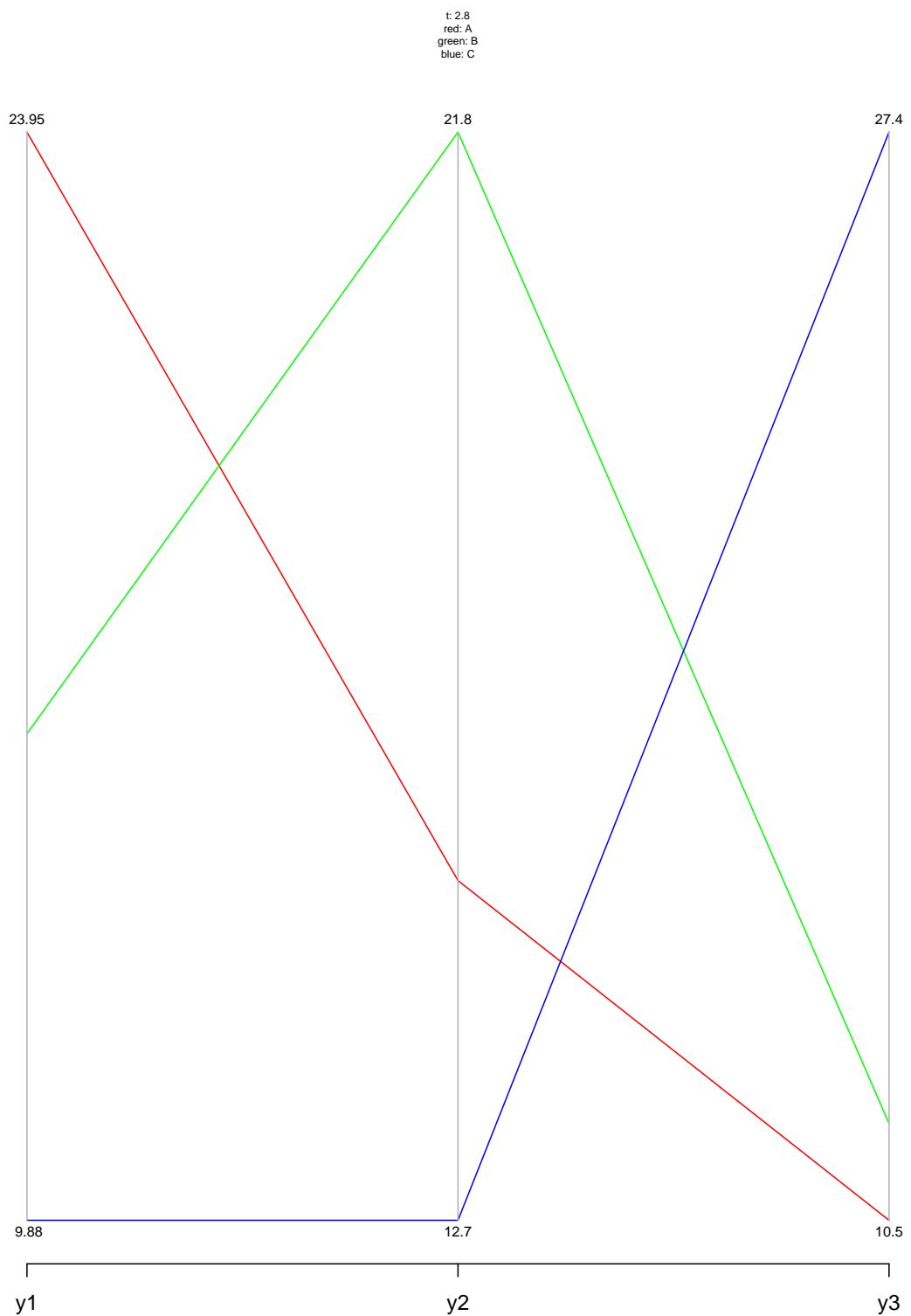
```
knitr::include_graphics('..../Gravity Transformation/Plots/gt_ABCD.pdf')
```



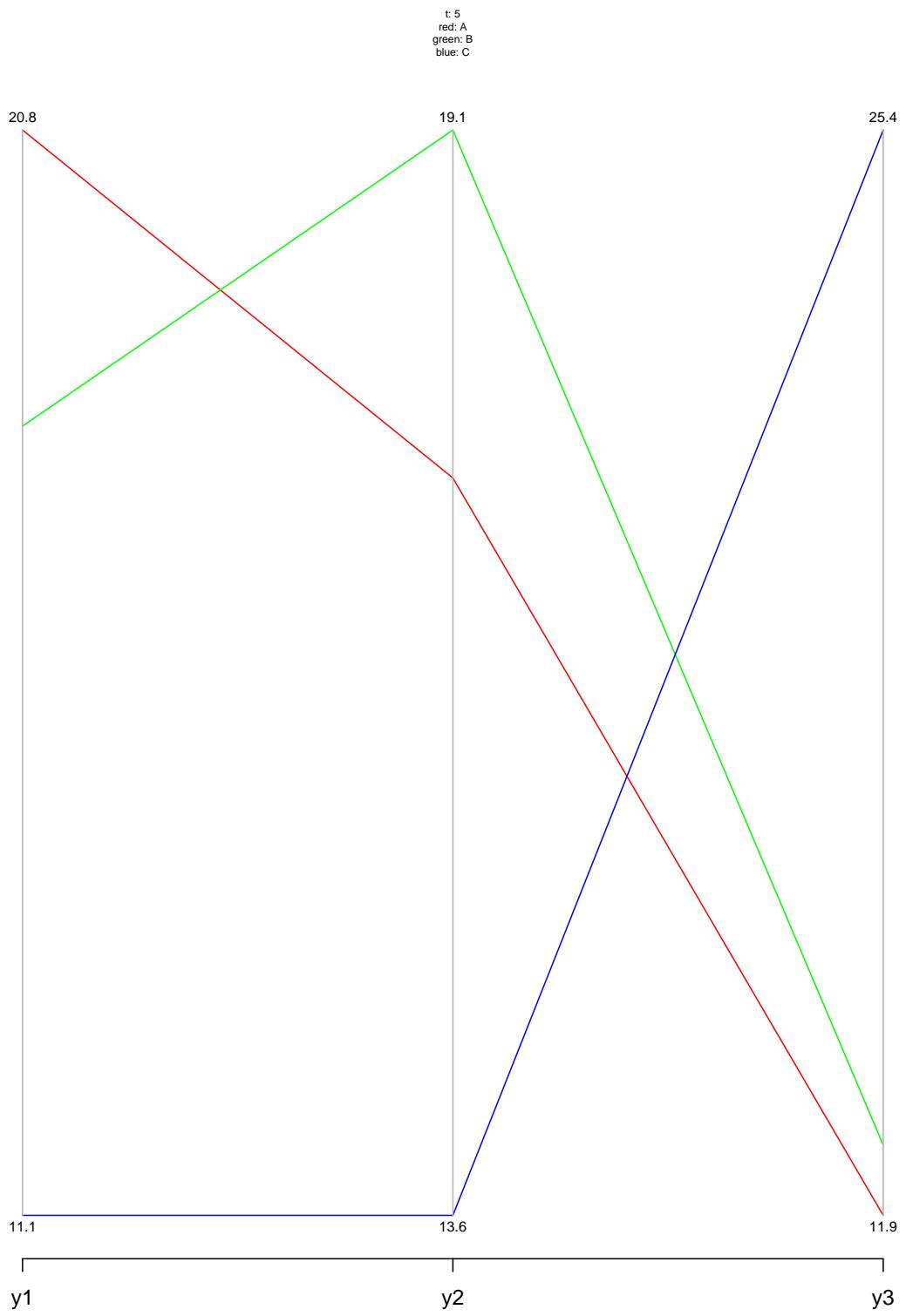
```
knitr:::include_graphics('..../Gravity Transformation/Plots/parcoord_3_0.pdf')
```



```
knitr::include_graphics('..../Gravity Transformation/Plots/parcoord_3_2.8.pdf')
```



```
knitr::include_graphics('..../Gravity Transformation/Plots/parcoord_3_5.pdf')
```



5.5 Interpretation

- General: distance plots
 - All points start with equidistance, so all curves start at the same point.
 - Pairs of more synchronous neurons tend to have their distance curves move faster to lower values than other pairs.
- General: parallel coordinates
 - All points start with equidistance; that is, have only one coordinate at top, and others are at 0.
 - As time goes on, the attractive forces make all pairs of neurons move closer; furthermore, the pairs of more synchronous neurons tend to have their coordinates in all dimensions move closer.
- gt_ABC.pdf
 - Distance plot of A, B and C. From the plot, after 2s, we can tell A and B move closer than other pairs, which is consistent with the source of data.
- gt_ABCD.pdf
 - Distance plot of A, B, C and D. From the plot, at 5s, we can tell D and B, A and B move closer than other pairs, which is inconsistent with the source of data: D and B should not be among the 2 most similar pairs. Potential reason could be
 - * We only take the first trial of data into account. However, taking average distance requires more computation cost.
 - * The values of a , b and τ are not properly chosen.
- parcoord_3_0.pdf
 - Parallel coordinates of neuron A, B and C at time 0. We can tell all neurons start from equidistance.
- parcoord_3_2.8.pdf
 - Parallel coordinates of neuron A, B and C at time 2.8. We can tell y_1 and y_2 coordinates of A and B move closer than others.
- parcoord_3_5.pdf
 - Parallel coordinates of neuron A, B and C at time 5. We can tell A and B moves closer than other pairs, which is consistent with gt_ABC.pdf and the source of data.

6 Interspike Interval Plot

6.1 Description

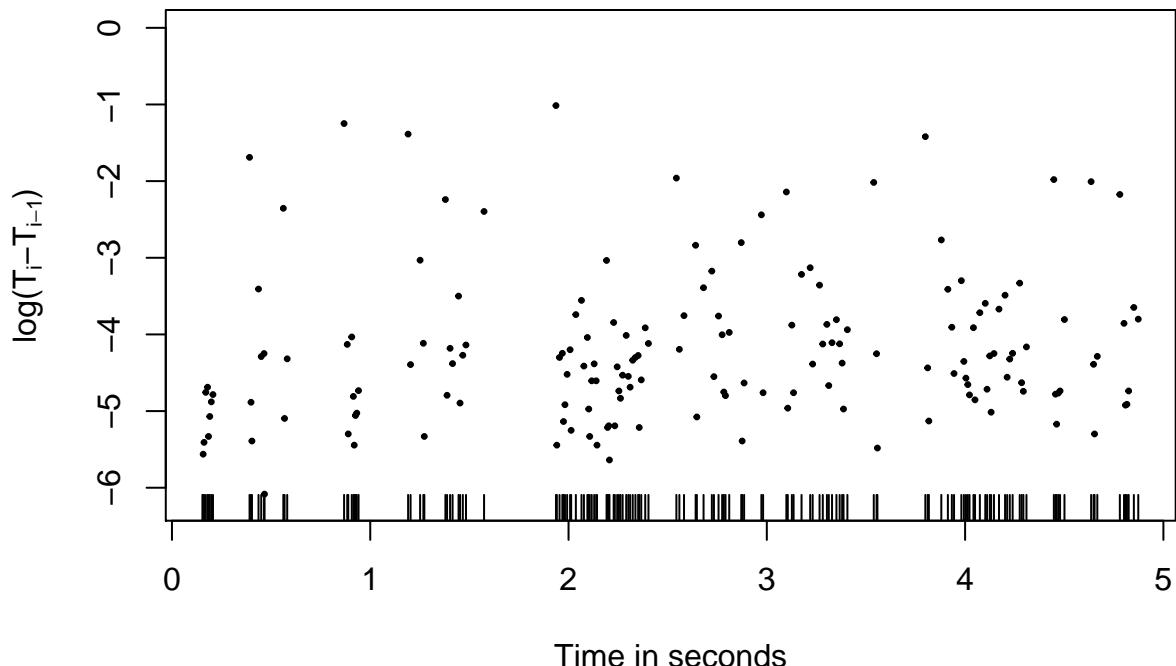
- Visualize spike activities from large and small time scales
- Visualize the departure from time homogeneity in the spiking activity

6.2 References

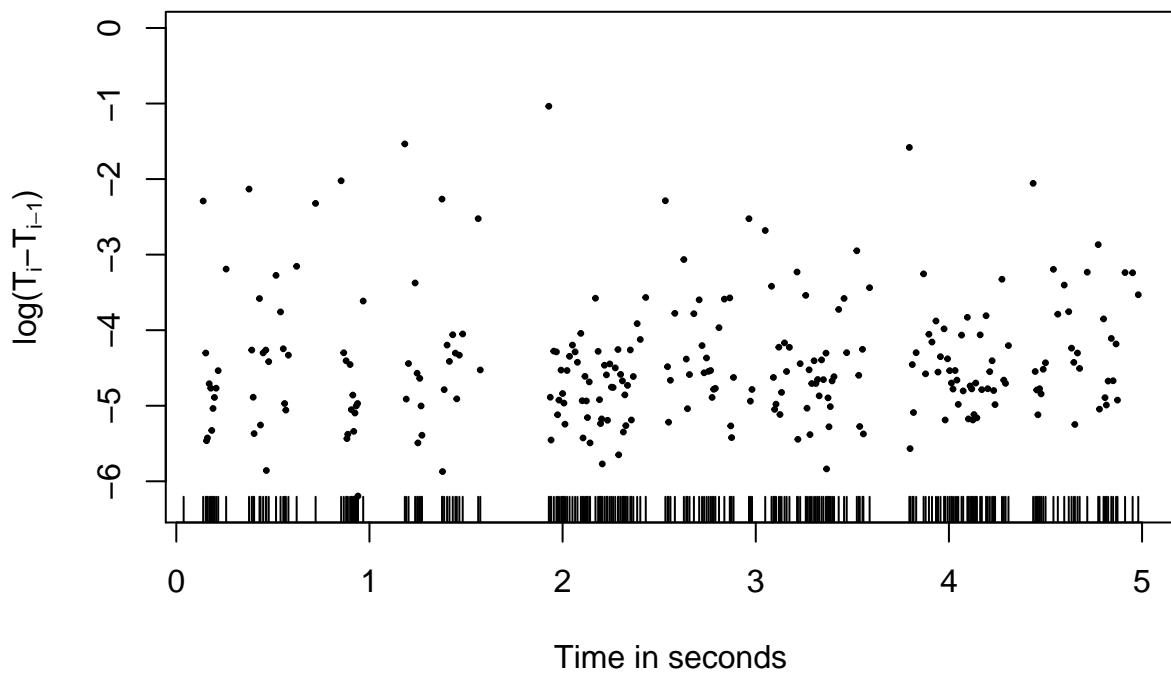
Readings/Interspike Interval/Ramezan et al (2014).pdf

6.3 Testing Code

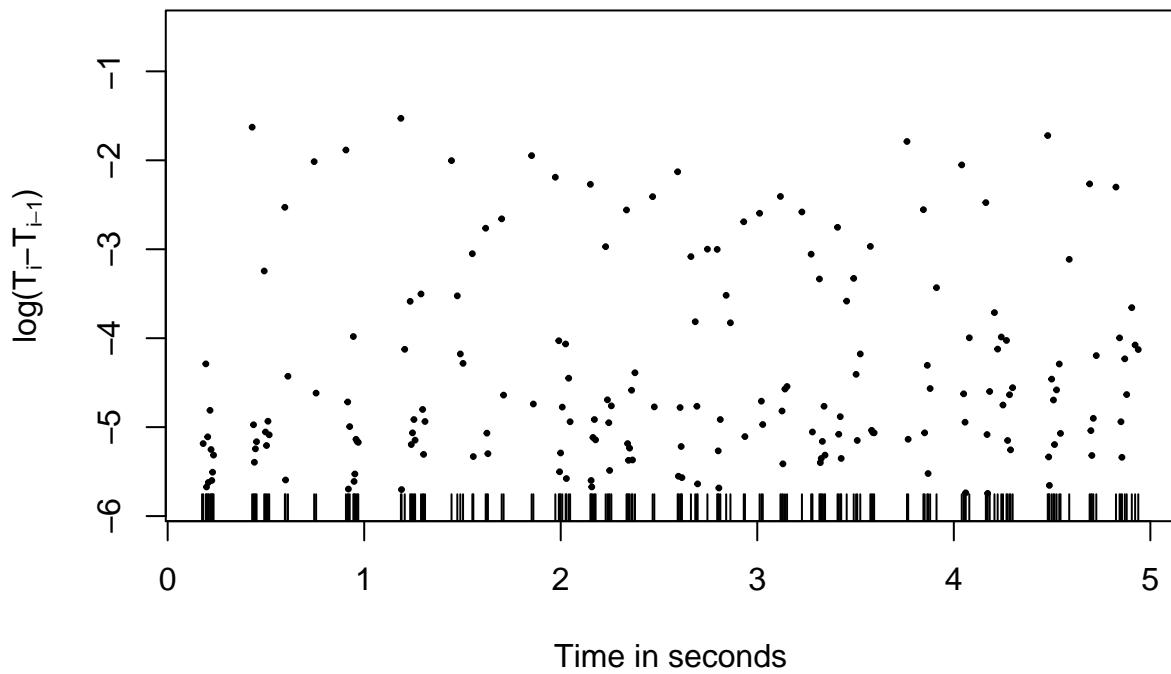
```
source("../ISI/ISI.R")
## testing
## ISI plot of spike train A
## A.pdf
ISI(A)
```



```
## ISI plot of spike train B
## B.pdf
ISI(B)
```



```
## ISI plot of spike train C
## C.pdf
ISI(C)
```



6.4 Interpretation

- General
 - Dense points in a vertical strip indicate firing activity.
 - Vertical empty strip shows temporal inactivity.
 - Horizontal band shows periodic spiking activity at specific frequency ranges.

- A.pdf
 - Between 1.5 and 1.9s, there is a vertical white strip, which indicates temporal inactivity.
 - Between 1.9 and 2.5s, there is a vertical dark strip, which indicates firing activities.
- B.pdf
 - B has the same pattern as A, which indicates B and A are synchronous to some degree.

7 Joint Peristimulus Time Histogram

7.1 Description

- Investigate the dynamics of correlation between two neurons over the course of the stimulus

7.2 References

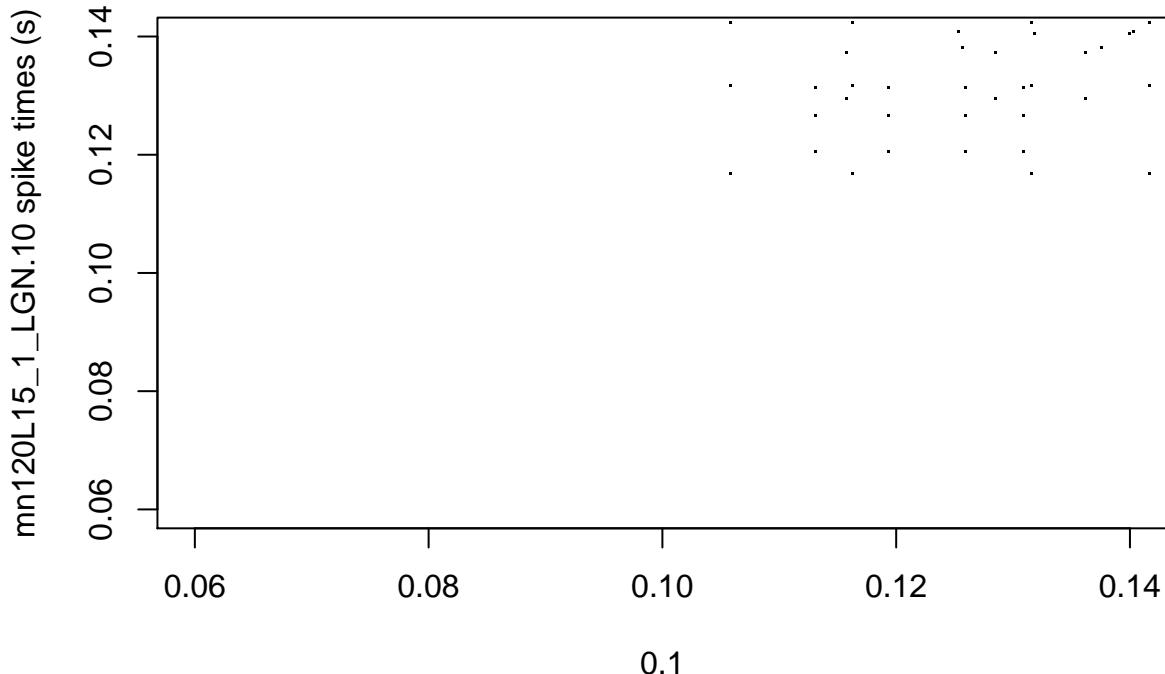
Readings/Joint Peristimulus Time Histogram/JPSTH.pdf

7.3 Testing Code

```
source("../Common/check.packages.R")
source("../Common/check.packages.R")
source("../Data/testing_data.R")
packages <- c("STAR")
check.packages(packages)

## STAR
## TRUE
## plot a jsd with neuron mn120L15_1_RGC.10 on X and neuron mn121R7_1_LGN.10 on Y
jsd(mn120L15_1_RGC.10, mn120L15_1_LGN.10, 0.1, 0.1)
```

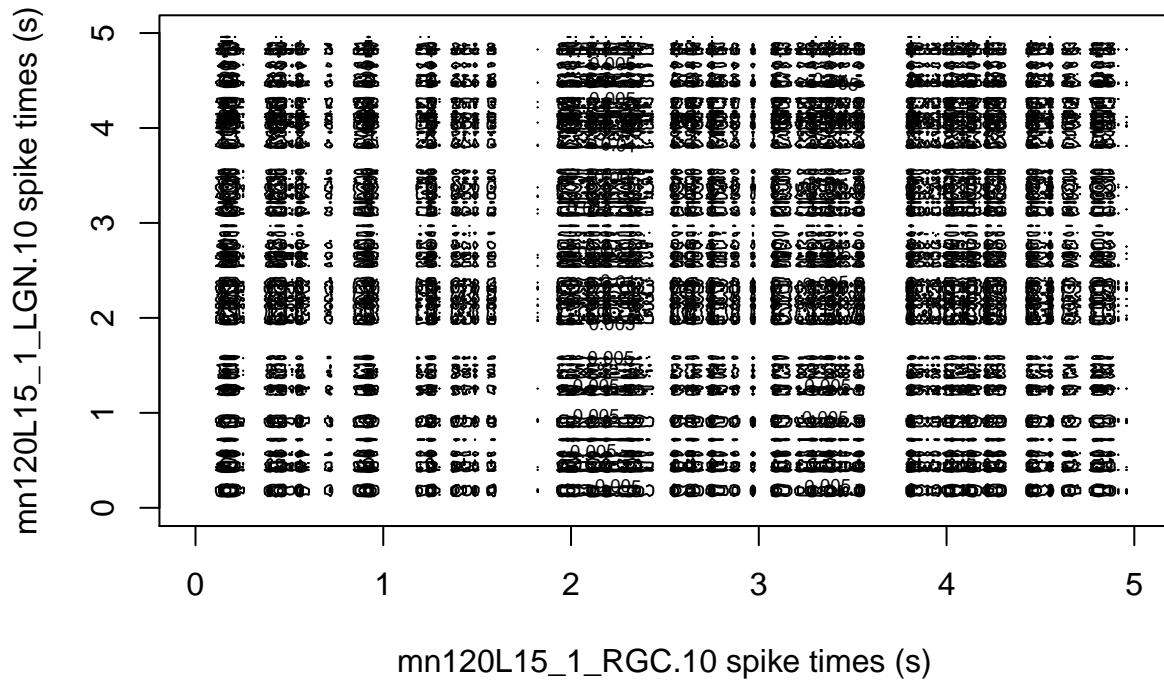
peri-stimulus scatter diagram of: mn120L15_1_RGC.10 and mn120L15_1_LGN.10



```
## now make the jpsth
j1.2 <- jpsth(mn120L15_1_RGC.10, mn120L15_1_LGN.10)
## make a contour plot
```

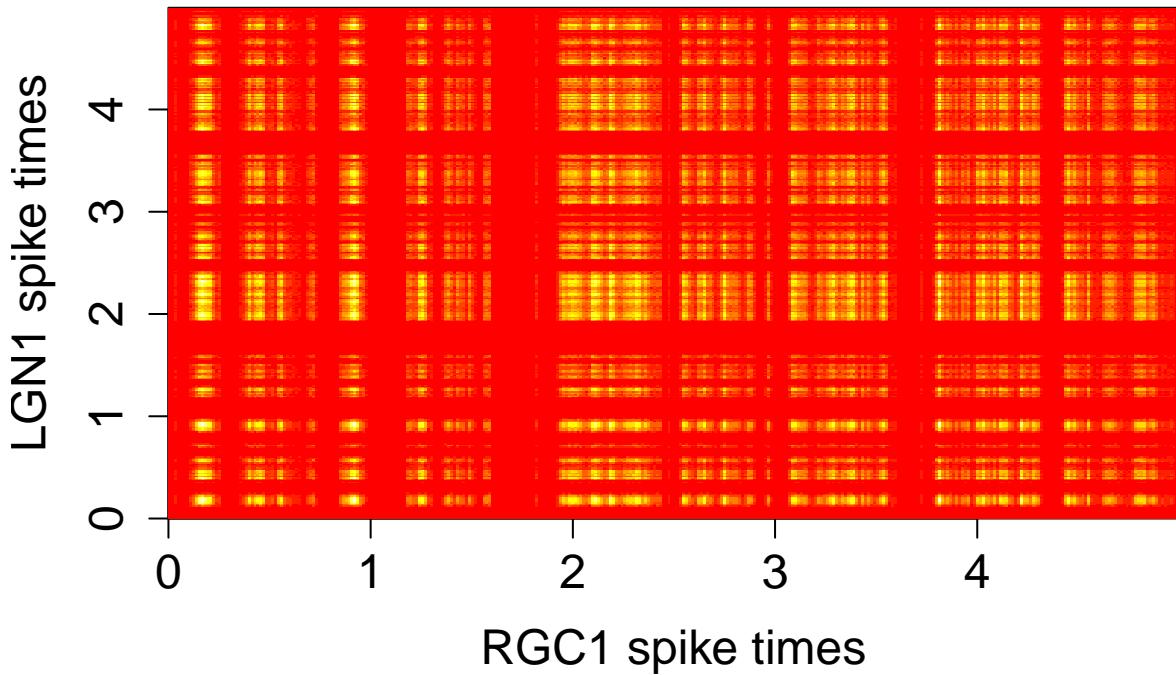
```
# contour.pdf  
contour(j1.2)
```

JPSTH of mn120L15_1_RGC.10 and mn120L15_1_LGN.10



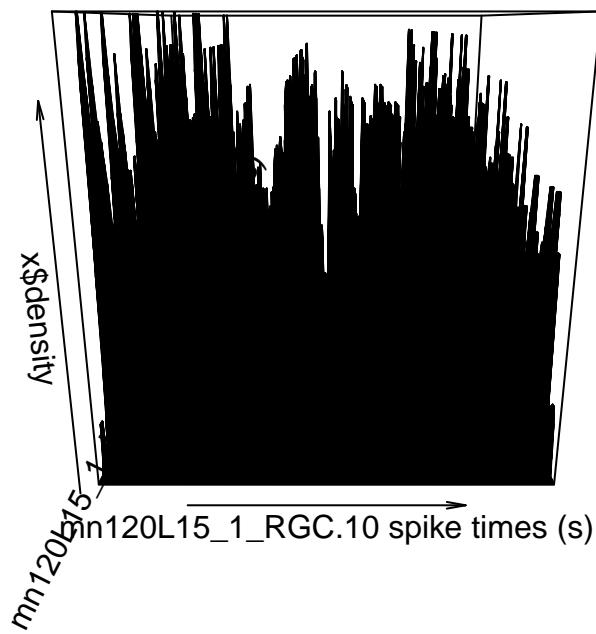
```
## make an image plot  
# image.pdf  
image(j1.2, main="JPSTH of RGC1 and LGN1", xlab = "RGC1 spike times", ylab="LGN1 spike times", cex.axis
```

JPSTH of RGC1 and LGN1



```
## make a persp plot  
# persp.pdf  
persp(j1.2)
```

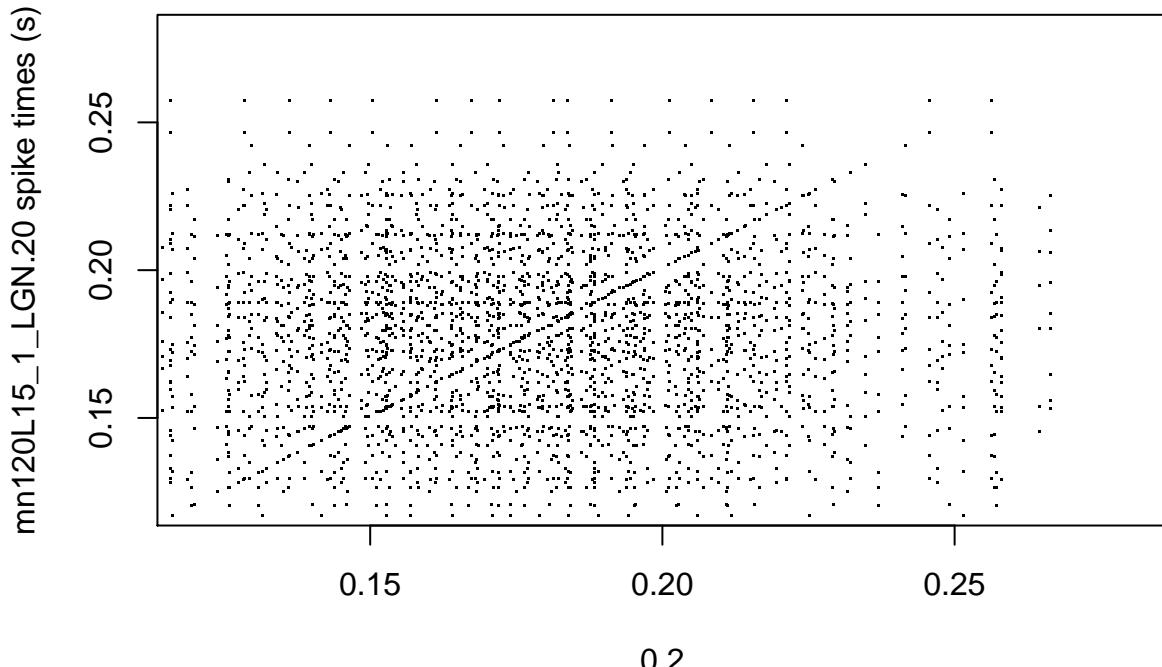
JPSTH of mn120L15_1_RGC.10 and mn120L15_1_LGN.10



```
## 20 trials  
## plot a jsd with neuron mn120L15_1_RGC.10 on X and neuron mn121R7_1_LGN.10 on Y
```

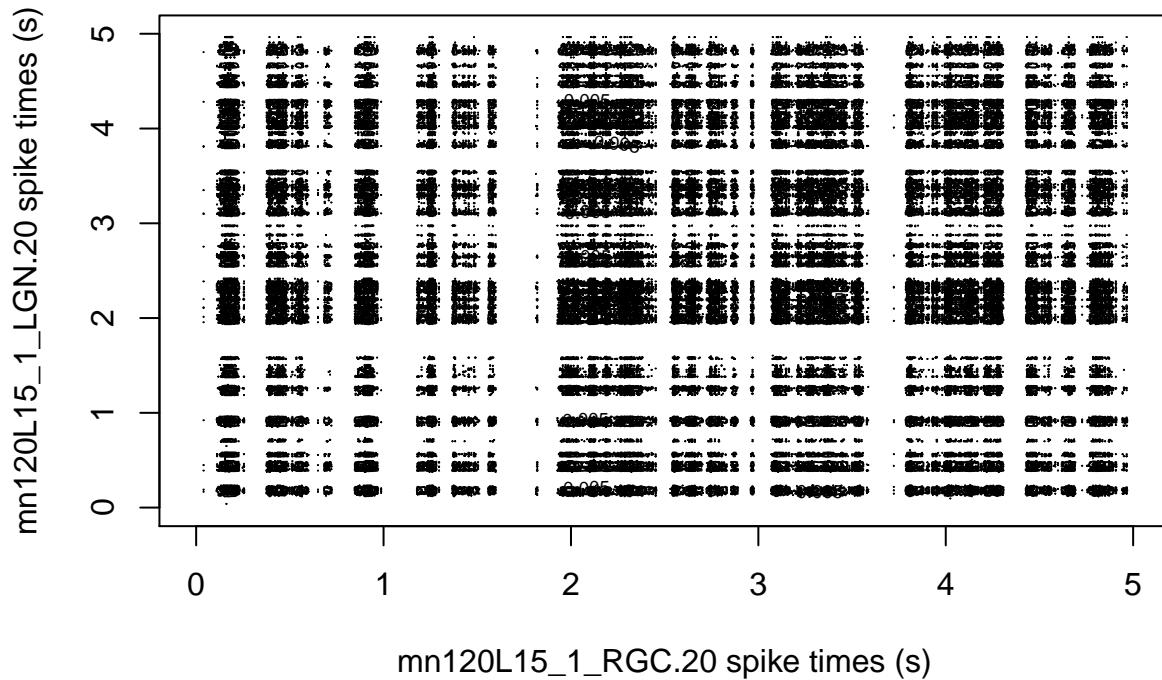
```
jsd(mn120L15_1_RGC.20, mn120L15_1_LGN.20, 0.2, 0.2)
```

peri-stimulus scatter diagram of: mn120L15_1_RGC.20 and mn120L15_



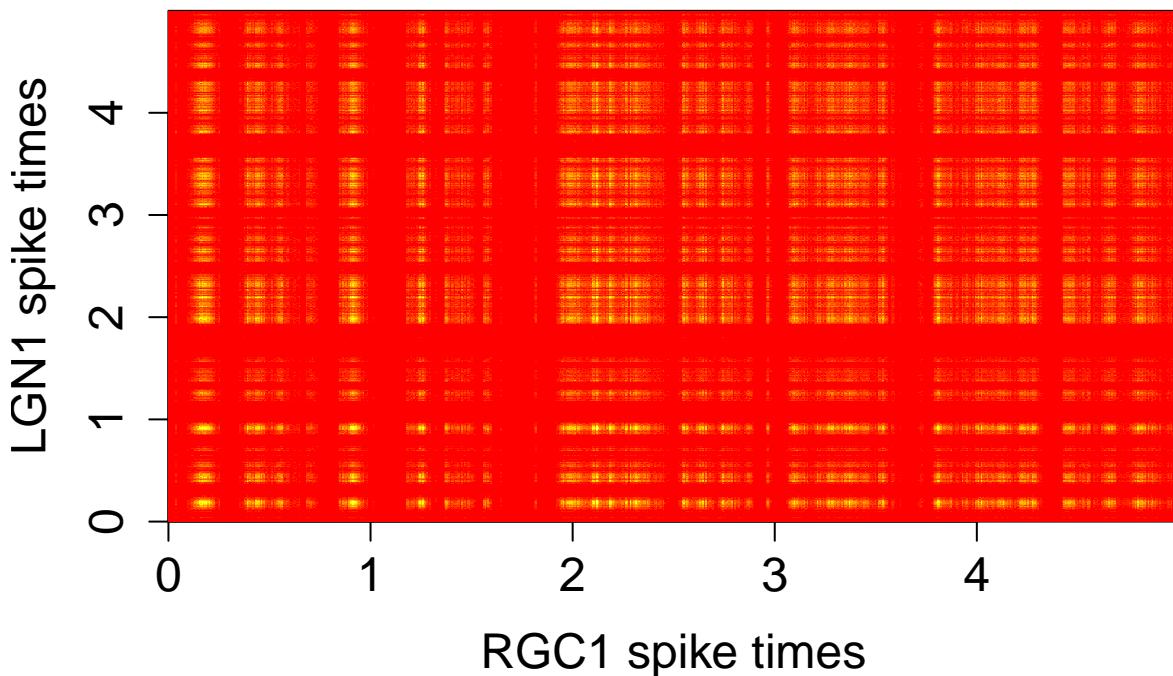
```
## now make the jpsth
j1.2 <- jpsth(mn120L15_1_RGC.20, mn120L15_1_LGN.20)
## make a contour plot
# contour.pdf
contour(j1.2)
```

JPSTH of mn120L15_1_RGC.20 and mn120L15_1_LGN.20



```
## make an image plot
# image.pdf
image(j1.2, main="JPSTH of RGC1 and LGN1", xlab = "RGC1 spike times", ylab="LGN1 spike times", cex.axis=
```

JPSTH of RGC1 and LGN1

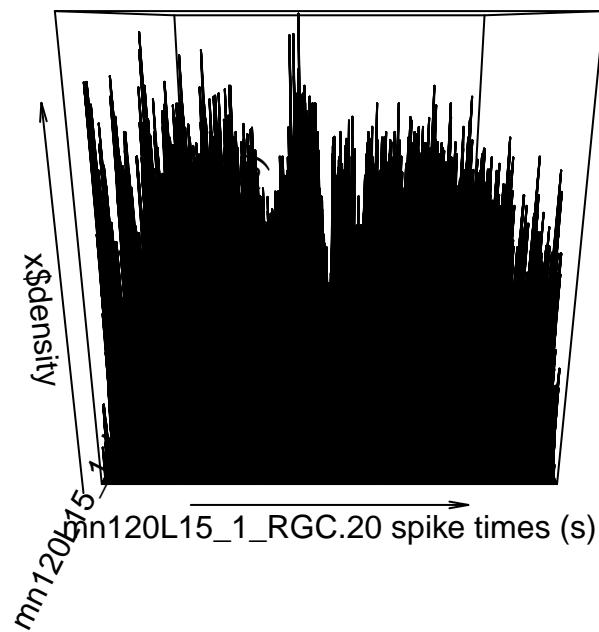


```

## make a persp plot
# persp.pdf
persp(j1.2)

```

JPSTH of mn120L15_1_RGC.20 and mn120L15_1_LGN.20



7.4 Interpretation

- General (heat plot)
 - Each bin corresponds to a specific time of the spike train on x-axis and a specific time of the spike train on y-axis.
 - Lighter color means higher coincidence of spike trains at the time.
- image.pdf
 - Yellow means higher coincidence (more spikes from two different spike trains fall into the bin).

8 Peristimulus Time Histogram

8.1 Description

- Plot histograms of the times at which neurons fire

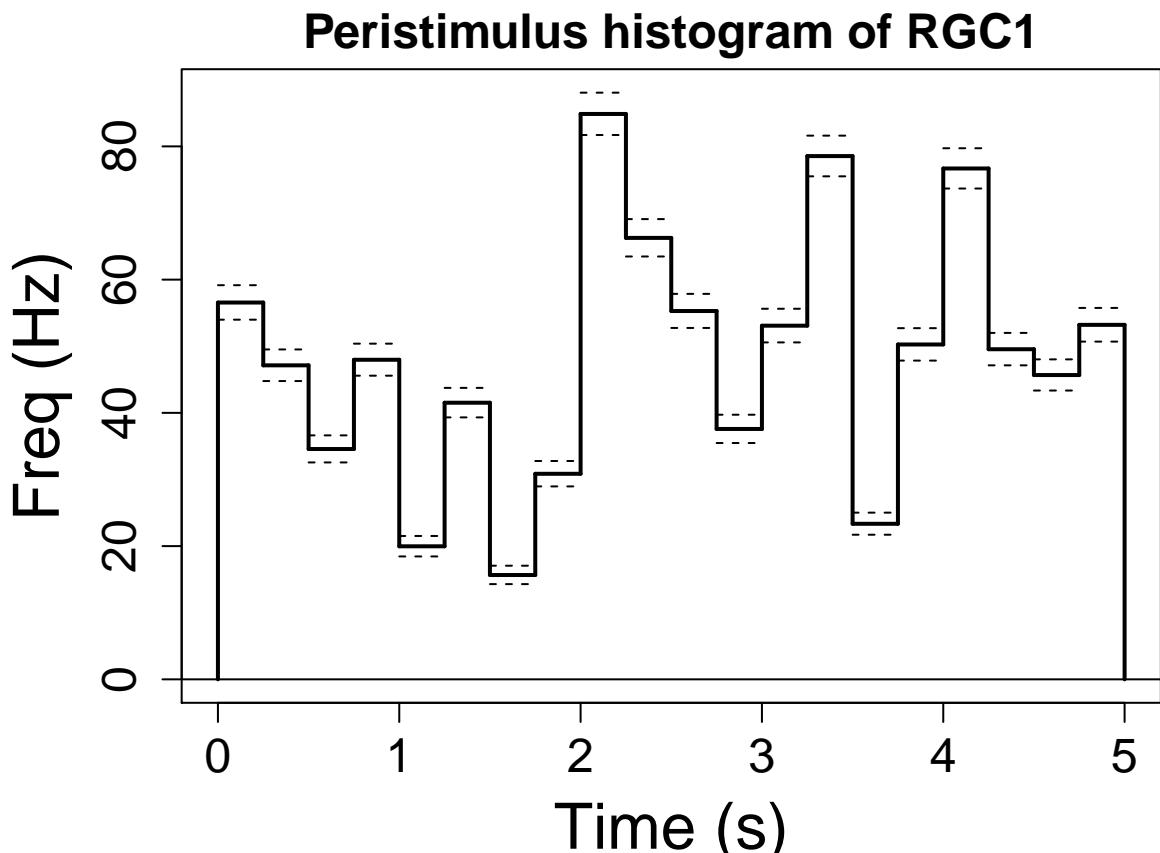
8.2 References

Readings/Peristimulus Time Histogram/T16_PSTH.doc

8.3 Testing Code

```
source("../Common/check.packages.R")
source("../Data/testing_data.R")
packages <- c("STAR")
check.packages(packages)

## STAR
## TRUE
## testing
## peristimulus histogram of neuron mn120L15_1_RGC
# psth.pdf
par(mar=c(4,5,2,2))
psth(as.repeatedTrain(mn120L15_1_RGC), main = "Peristimulus histogram of RGC1", cex.axis = 1.5, cex.lab
```



8.4 Interpretation

- General
 - Higher frequency means more spikes are repeated trials fall into the bin.
 - Dotted lines are the confidence bands obtained using Poisson approximation.
- psth.pdf
 - mn120L15_1_RGC has the highest firing rate somewhere between 2 and 2.5s over time.

9 Raster plot

9.1 Description

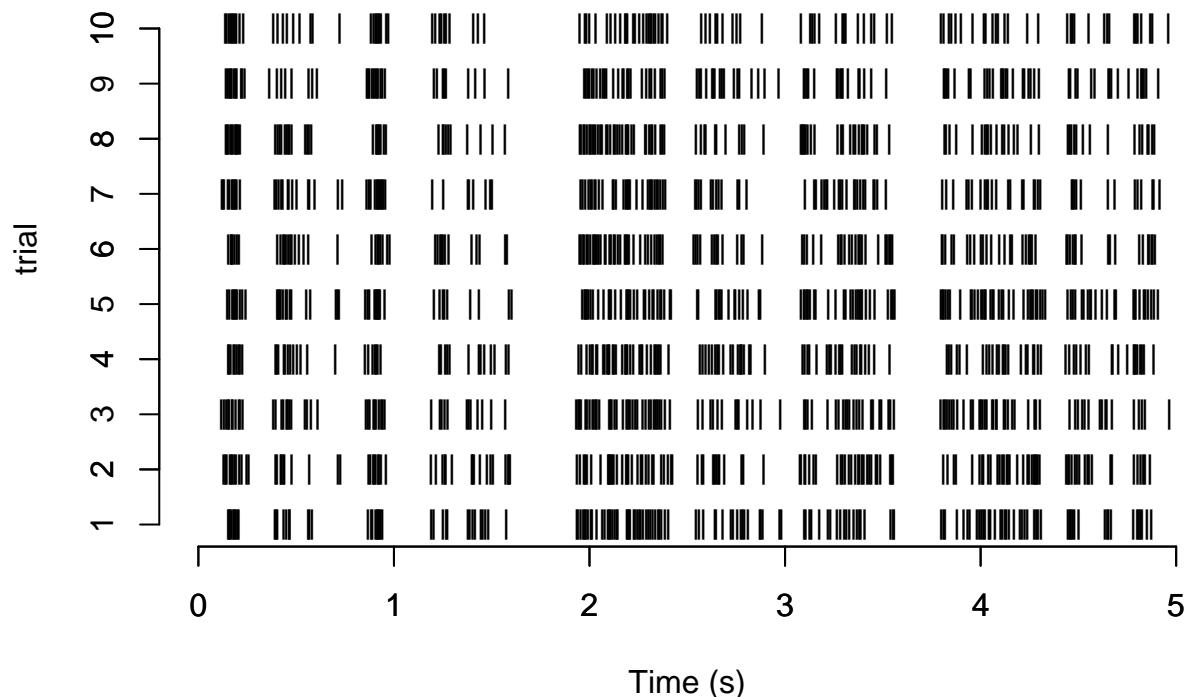
- Mark neural activities (spikes or action potentials) at specified positions

9.2 References

9.3 Testing Code

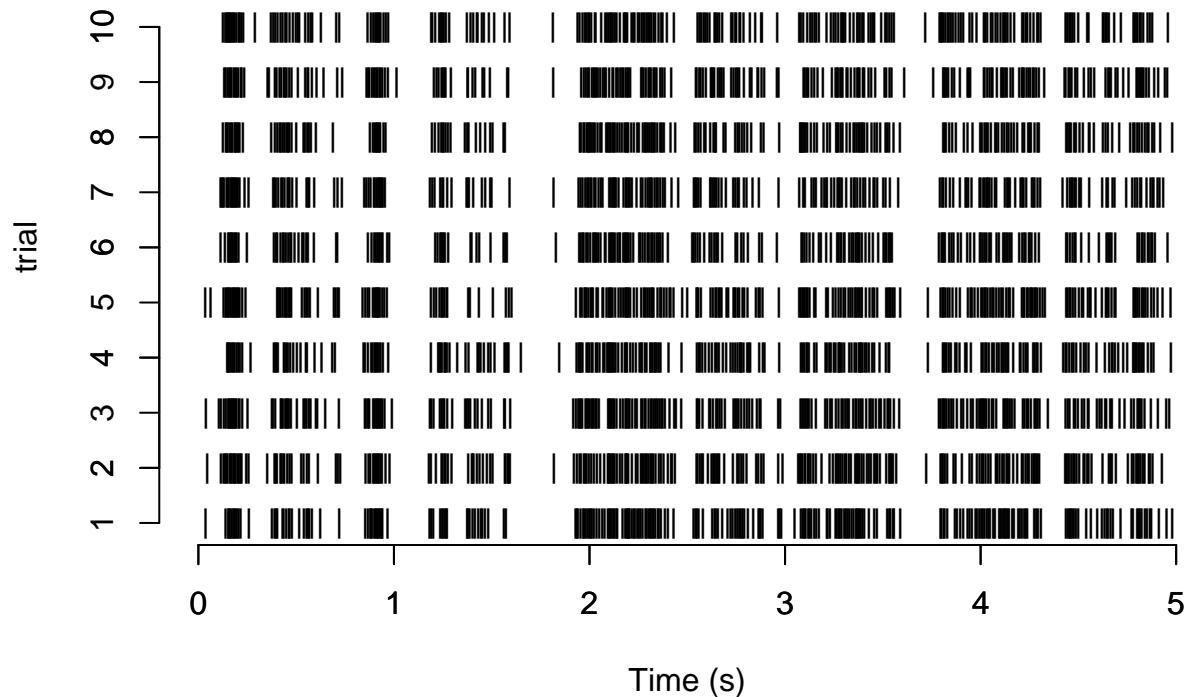
```
source("../Data/testing_data.R")
source("../Raster/Raster_builtin.R")
## plotting raster of the first 10 trials of A
## A.10.pdf
raster(head(mn120L15_1_LGN, 10))
```

Raster plot



```
## plotting raster of the first 10 trials of B
## B.10.pdf
raster(head(mn120L15_1_RGC, 10))
```

Raster plot



9.4 Interpretation

- General
 - Not very interpretative for multiple spike trains. Hence, iRaster is introduced.
- A.10.pdf, B.10.pdf
 - Trials from the same neuron have similar patterns.
 - These two plots have similar patterns of black and white segments, which indicates that A and B have firing activities and temporal inactivities at the same time.

10 iRaster Plot

10.1 Description

- Reorder Raster plot by 9 procedures (first 7 can be found in the reference paper)
 - procedure 0: no reordering
 - procedure 1: shortest ISI interval
 - procedure 2: longest ISI interval
 - procedure 3: mean ISI interval
 - procedure 4: variance of ISI interval
 - procedure 5: coefficient of variation of ISI interval
 - procedure 6: Welch's test statistic
 - procedure 7: coin.calc
 - procedure 8: cor.index.calc
 - procedure 9: cor.tiling.calc
- Superimpose firing rate onto Raster plot
- Investiage the independence of multiple spike trains. If independent, there should not been a structure/pattern of iRaster plots after reordering by different measures. See R Code/Onset/ and R Code/Response/

10.2 Notes

- As discussed, procedure 6 may not be reliable. We cannot use different t-values to compare.

10.3 References

Readings/iRaster/iRaster.pdf

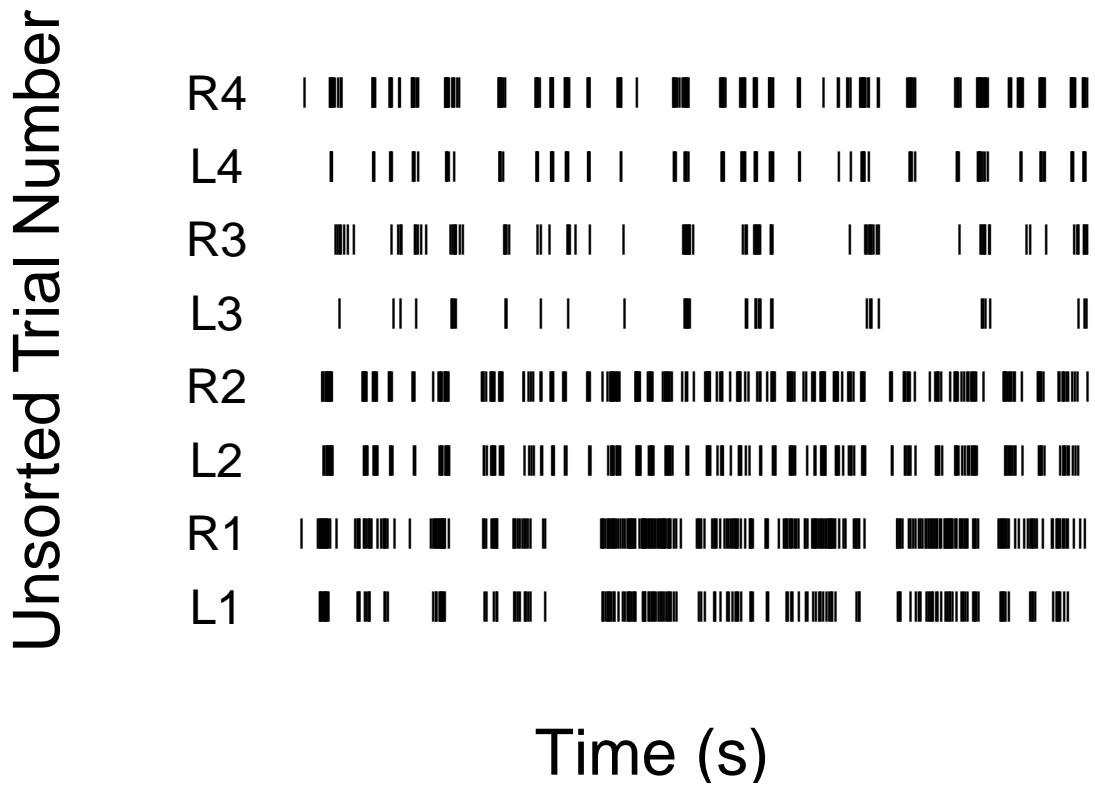
10.4 Testing Code

```
source("../Data/testing_data.R")
source("../iRaster/iRaster.R")

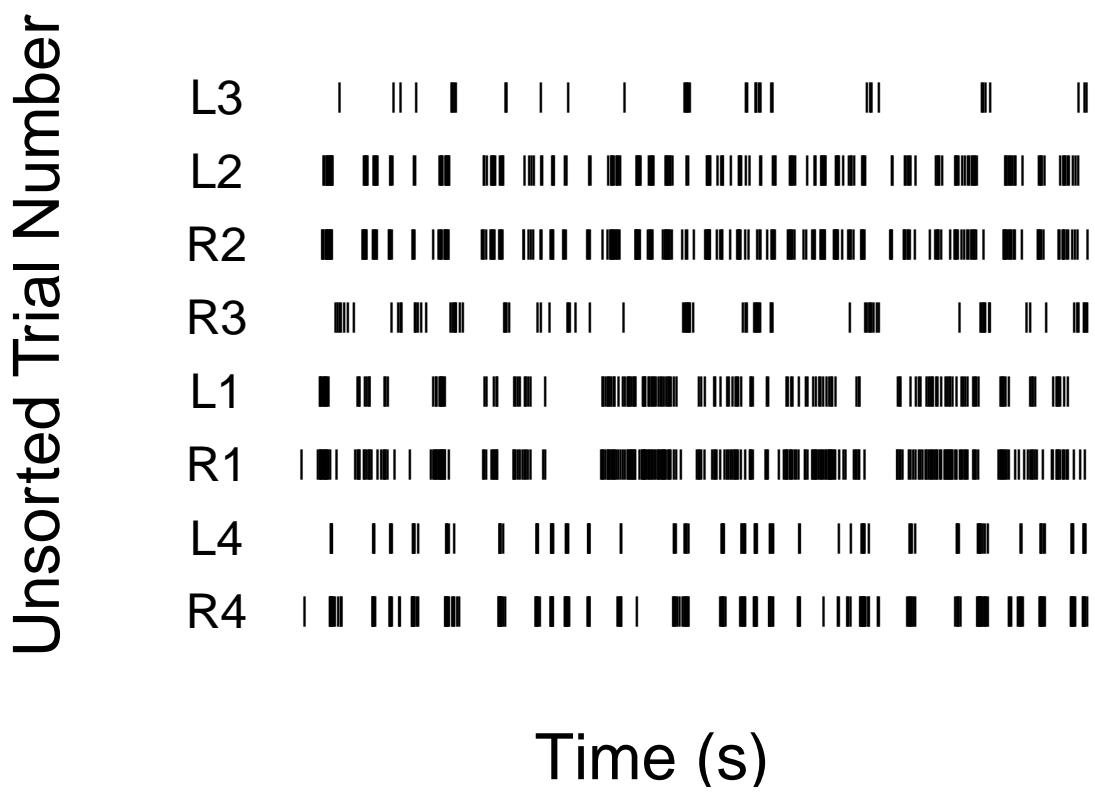
labels.to.show = c("L1" , "R1" , "L2" , "R2" , "L3" , "R3" , "L4" , "R4")

par(mar=c(4,6,3,2))

spike.trains<- list(A, B, C, D, E, F, G, H)
# # testing
# r0.pdf
r0 <- iRaster.reordering(spike.trains, 0)
```



```
# r1.pdf  
r1 <- iRaster.reordering(spike.trains, 1)
```



```
# r2.pdf  
r2 <- iRaster.reordering(spike.trains, 2)
```

Unsorted Trial Number



Time (s)

```
# r3.pdf  
r3 <- iRaster.reordering(spike.trains, 3)
```

Unsorted Trial Number



Time (s)

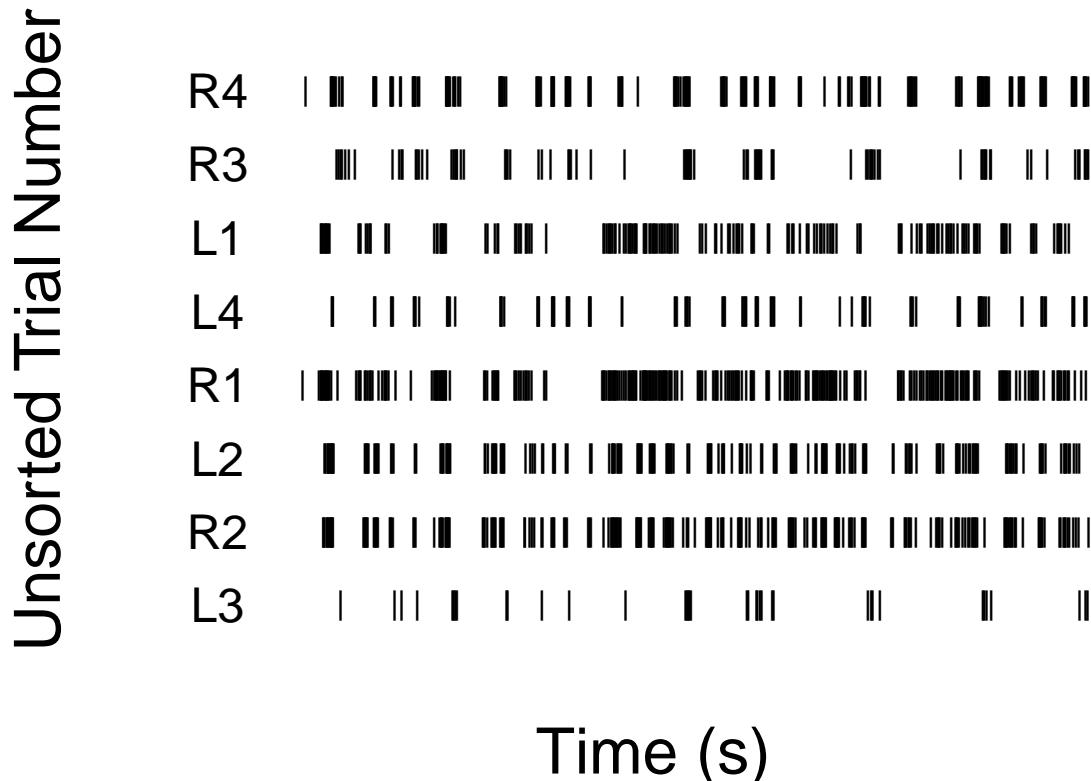
```
# r4.pdf  
r4 <- iRaster.reordering(spike.trains, 4)
```

Unsorted Trial Number

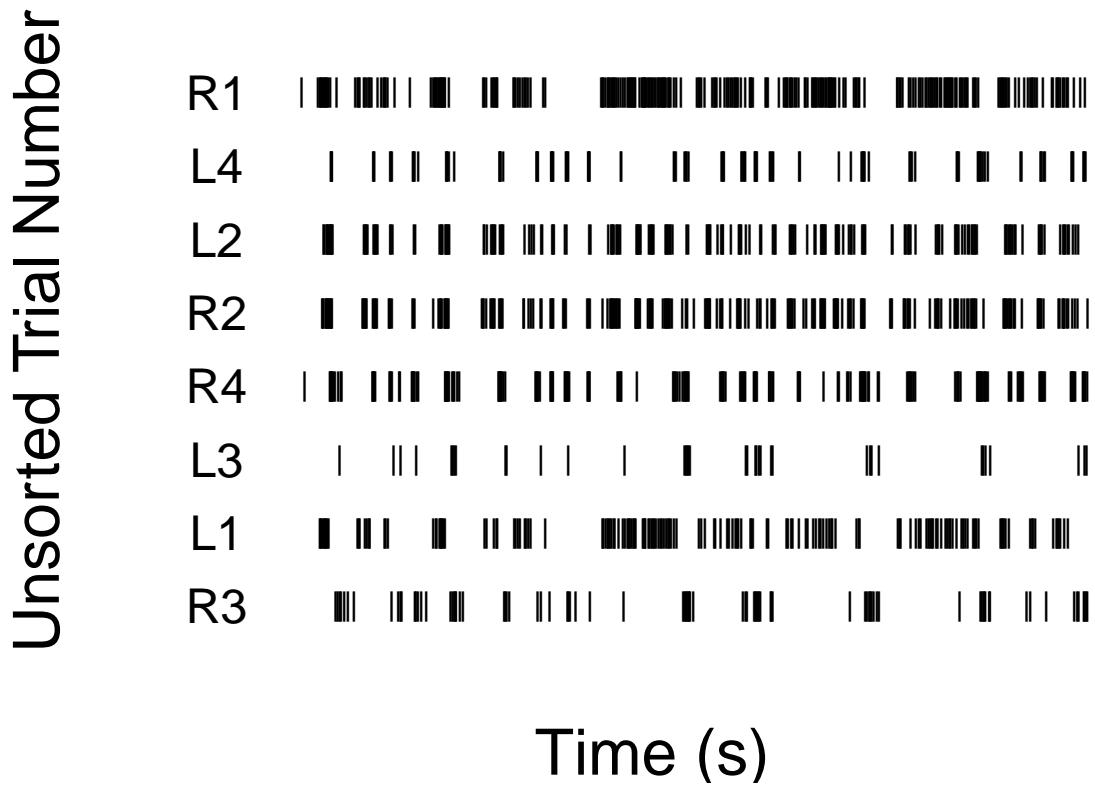


Time (s)

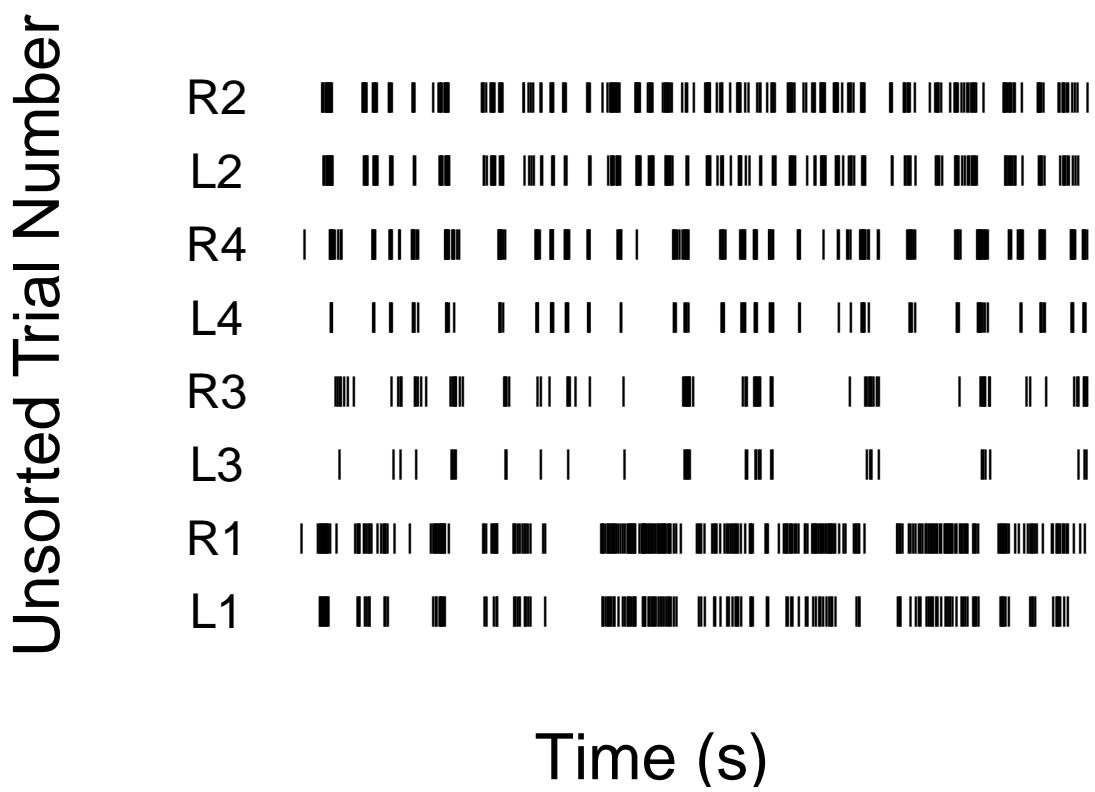
```
# r5.pdf  
r5 <- iRaster.reordering(spike.trains, 5)
```



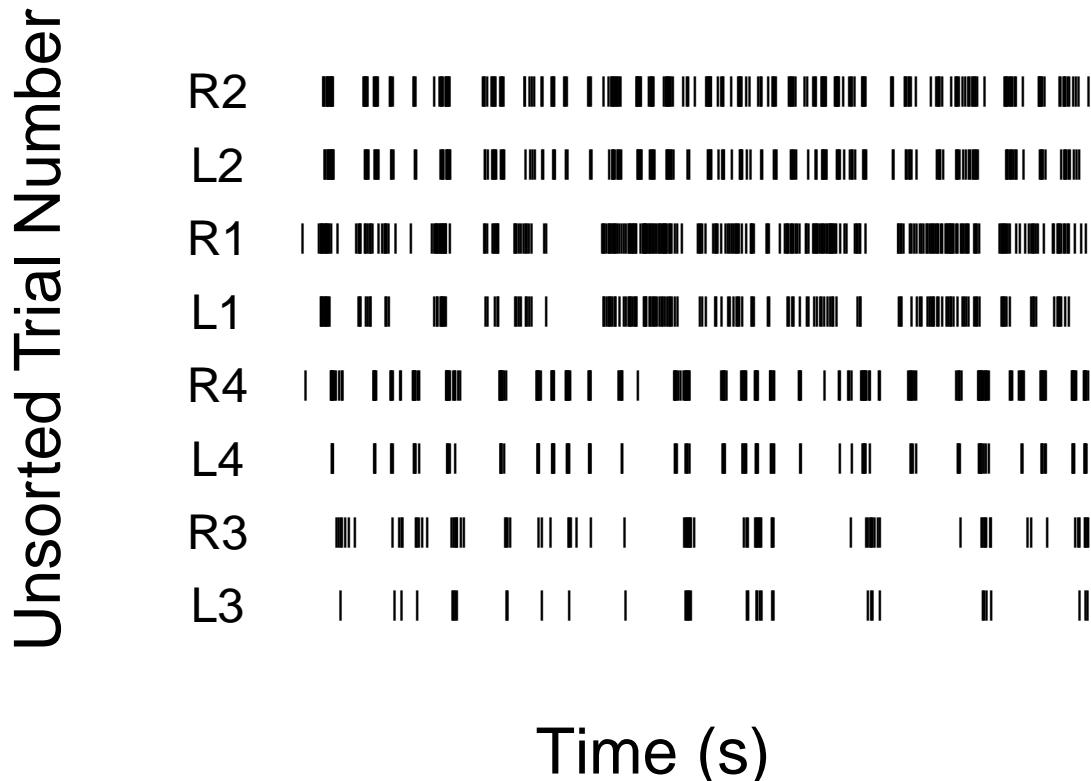
```
# r6.pdf  
r6 <- iRaster.reordering(spike.trains, 6)
```



```
# r7.pdf  
r7 <- iRaster.reordering(spike.trains, 7)
```



```
# r8.pdf  
r8 <- iRaster.reordering(spike.trains, 8)
```



```
# r9.pdf  
r9 <- iRaster.reordering(spike.trains, 9)
```

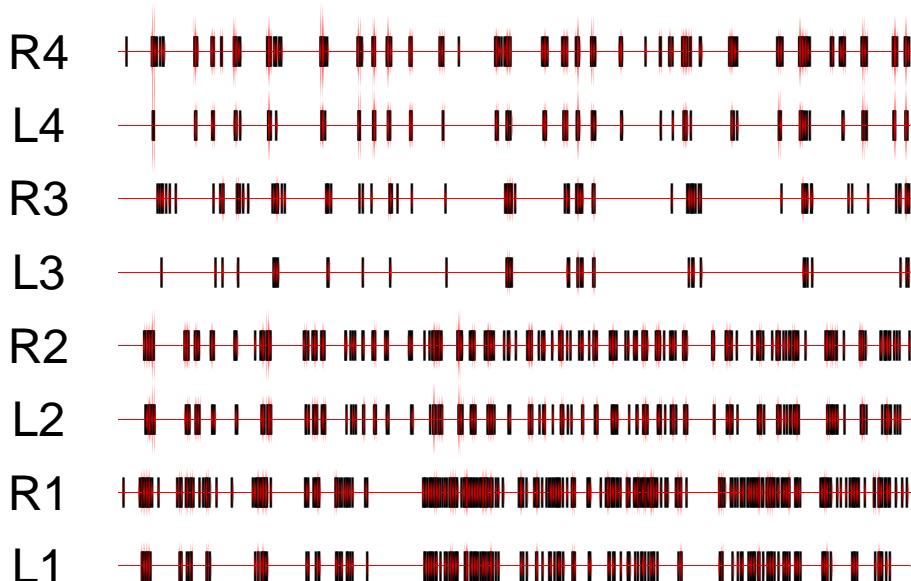
Unsorted Trial Number



Time (s)

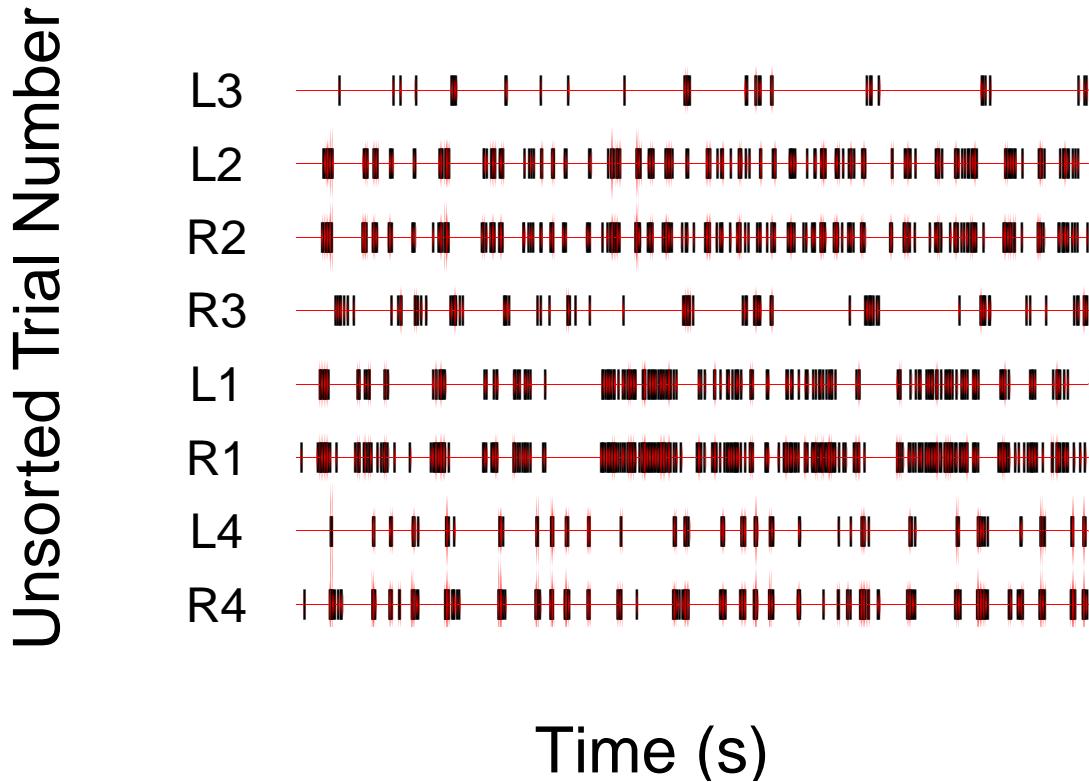
```
# testing  
# superimpose rate on spike trains without reordering (proc0)  
# p0.pdf  
iRaster.rate(spike.trains, 0)
```

Unsorted Trial Number



Time (s)

```
# superimpose rate on spike trains sorted by proc1
# p1.pdf
iRaster.rate(spike.trains, 1)
```



10.5 Interpretation

- General
 - iRaster plots are more interpretative than Raster plots as it sorts spike trains with similar properties adjacent.
 - Blue strips superimposed on Raster plots indicate firing rates. Longer blue strips mean higher firing rates.
- r1.pdf - r9.pdf
 - Raster plots sorted by different measures.
 - Spike trains are sorted in descending order by the magnitude of the measurement from top to bottom.
 - For example, by procedure 1, we can see the top one (neuron 5) has most sparsely populated spikes.
- p0.pdf
 - Firing rates are superimposed onto Raster plots.
 - Spike trains are unsorted (procedures 0).
- p1.pdf
 - Firing rates are superimposed onto Raster plots.
 - Spike trains are sorted in descending order from top to the bottom by procedure 1.
 - We can tell the top one (neuron 5) has fewer long blue strips than others; hence fewer firing activities than others.

11 Tunnel Plot

11.1 Description

- Discover the synchrony of neural activity over vast quantities of data
- Sorting functionality: sort the 2 spike trains that are most similar to the reference train adjacent to the reference train
- Highlighting functionality: enable focusing on specific spike trains
- Clustering functionality: perform hierarchical clustering using complete linkage method. The same 6 measures used in correlation grid can be used in Tunnel plot to measure the similarity distances between spike trains. In our examples, we only do clustering by coin.calc.

11.2 References

Readings/Tunnel/Walter et al (2004).pdf

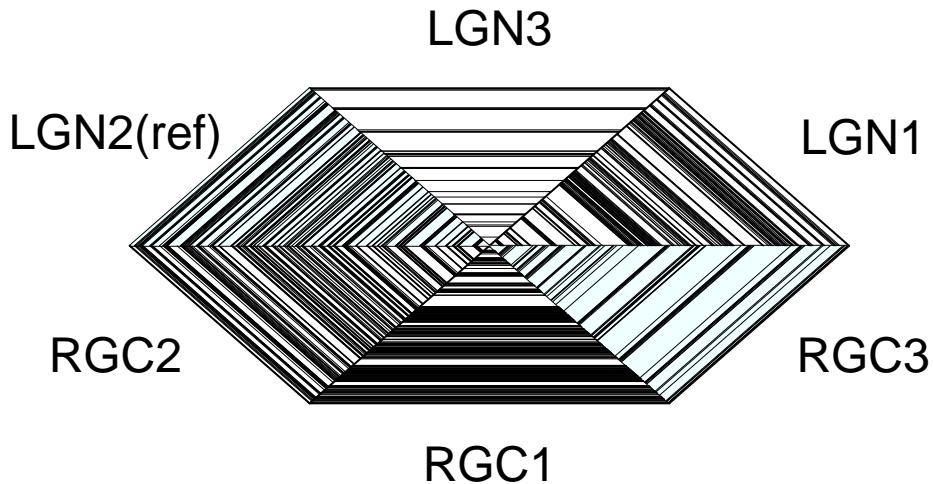
11.3 Testing Code

```
source("../Tunnel/tunnel.R")
source("../Clustering/cluster.R")

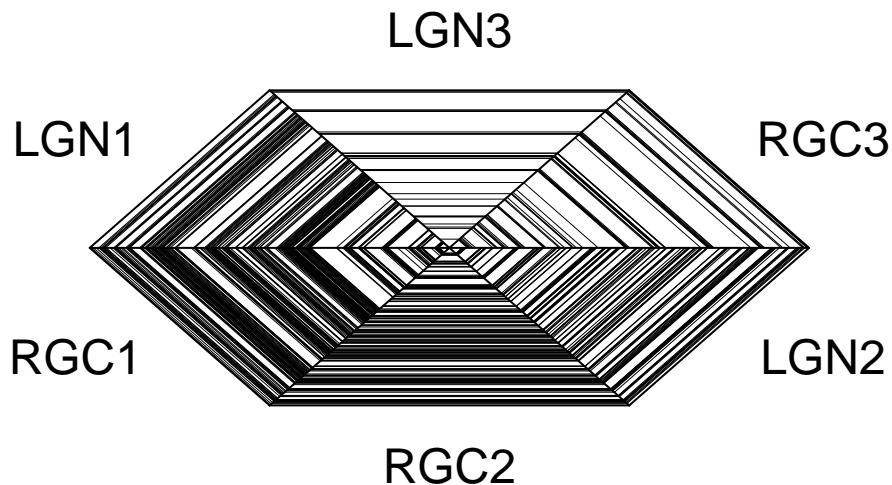
## testing
spike.trains <- list()
spike.trains[[1]] <- A
spike.trains[[2]] <- B
spike.trains[[3]] <- C
spike.trains[[4]] <- D
spike.trains[[5]] <- E

# tunning representation of 5 neurons with neuron 3 highlighted and with neuron 3 being the reference n
# 5n.pdf
labels <- c("LGN1", "RGC1", "LGN2", "RGC2", "LGN3")
# tunnel.rep(start=0, end=5, spike.trains=spike.trains, labels = labels, highlight = c("mn121R7_1_LGN"))

spike.trains[[6]] <- F
# tunning representation of 6 neurons with neuron 3 and 6 highlighted and with neuron 3 being the refer
# 6n.pdf
labels <- c( "LGN1", "RGC1", "LGN2", "RGC2", "LGN3", "RGC3")
tunnel.rep(start=0, end=5, spike.trains=spike.trains, labels = labels, highlight = c("LGN2", "RGC3"), r
```



```
##### testing reordering by clustering (coin) #####
spike.trains <- list()
spike.trains[[1]] <- A
spike.trains[[2]] <- D
spike.trains[[3]] <- F
spike.trains[[4]] <- B
spike.trains[[5]] <- E
spike.trains[[6]] <- C
labels <- c("LGN1", "RGC2", "RGC3", "RGC1", "LGN3", "LGN2")
sorted.trains <- sort.trains(spike.trains)
sorted.labels <- sort.labels(spike.trains, labels)
# 6n_ordering.pdf
tunnel.rep(start=0, end=5, spike.trains=sorted.trains, labels = sorted.labels, highlight = c("mn121R7_1",
"mn121R7_2", "mn121R7_3", "mn121R7_4", "mn121R7_5", "mn121R7_6"))
```



11.4 Interpretation

- General
 - When sorting is enabled, the spike trains with 2 most similar distances are sorted to be adjacent to the reference spike train.
 - Doing sorting after clustering is not useful.
- Notes
 - R Code/Shiny/Tunnel_interative.R is an UI for this technique.

- 5n.pdf
 - This plot is sorted but not clustered.
 - From the density of lines, we can tell mn120L15_1_RGC and mn120L15_1_LGN are highly correlated.
 - mn122R4_2_LGC has fewer spikes than others.
 - mn121R7_1_LGN is the reference neuron and is highlighted.
 - mn122R4_2_LGN and mn121R7_1_RGC are sorted to be adjacent to mn121R7_1_LGN since they have the top two coincidences with mn121R7_1_LGN among remaining neurons.
- 6n.pdf
 - This plot is sorted but not clustered.
 - From the density of lines, we can tell mn121R7_1_LGN and mn121R7_1_RGC are highly correlated; we cannot tell too much about other neurons since similar neurons are not sorted to be adjacent. Hence, we introduce the modified Tunnel plot.
 - mn121R7_1_LGN is the reference neuron and is highlighted; in addition, mn122R4_2_RGC is highlighted.
 - mn122R4_2_LGN and mn121R7_1_RGC are sorted to be adjacent to mn121R7_1_LGN since they have the top two coincidence with mn121R7_1_LGN among remaining neurons.
- 6n_ordering.pdf
 - This plot is clustered but not sorted.
 - 6 neurons are reordered by the leaves of hierarchical clustering; the distance measure is coin.calc.
 - From the density of lines, we can tell mn121R7_1_LGN and mn120L15_1_RGC, mn120L15_1_LGN and mn120L15_1_RGC, mn122R4_2_LGN and mn122R4_2_RGC are highly correlated, which is consistent with the source of data.

12 Modified Tunnel Plot

12.1 Description

- Modify the output of the sorting feature of Tunnel plot
- Insert reference neuron between every pair of sorted neurons
- The same 6 measures used in correlation grid can be used in measuring the similarity distance between spike trains. In our examples, we only do an example of coin.calc.

12.2 References

12.3 Testing Code

```

source("../Data/testing_data.R")
source("../Tunnel/modified_tunnel.R")

## testing
vec.data <- list()
vec.data[[1]] <- A
vec.data[[2]] <- B
vec.data[[3]] <- C
vec.data[[4]] <- D
vec.data[[5]] <- E

## tuning representation of 5 neurons with neuron 2, 3 and 4 highlighted and with neuron 5 being the ref
# tunnel.rep(5, vec.data, highlight = c(2,3,4), ref = 5)

```

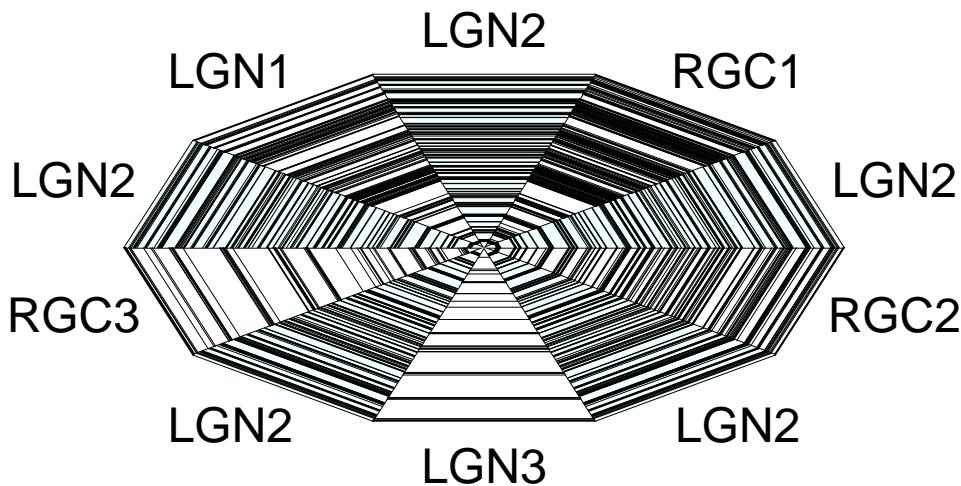
```

# 5n_sorting_display.pdf
tunnel.rep(start=0, end=5, vec.data=vec.data, label=c("mn120L15_1_LGN", "mn120L15_1_RGC", "mn121R7_1_LGN"))

mn121R7_1_LGN mn120L15_1_RGC
20L15_1_LGN mn121R7_1_I
21R7_1_LGN mn121R7_1_F
mn122R4_2_LGN mn121R7_1_LGN

vec.data[[6]] <- F
## tunning representation of 6 neurons measured in 5s
## with neuron 2 and 5 highlighted and with neuron 5 being the reference neuron
# 6n_sorting_display.pdf
tunnel.rep(start=0, end=5, vec.data=vec.data, label=c("LGN1", "RGC1", "LGN2", "RGC2", "LGN3", "RGC3"), hi

```



12.4 Interpretation

- General
 - The reference neuron is duplicated and highlighted.
 - The remaining neurons are reordered by the similarity distances with the reference neuron.
 - The similarities of remaining neurons in the plot are in a descending order, counter clock-wise, starting from one sector below positive x-axis.
- 5n_ordering.pdf and 6n_reordering.pdf
 - It is expected to be mn121R7_1_RGC to be the one on the sector below positive x-axis, as the reference neuron is mn121R7_1_LGN.

13 Snowflake Plot

13.1 Description

- Detect synchrony among three neurons
- Embed CCH and STJH
- Extend to higher dimensions > 3 . Lower dimension to $n - 1$

13.2 Notes

- The computation speed is slow due to 3 levels of nested loops. We have not figured out parallel programming. Alternatively, if a C++ compiler is installed, the C++ function computes much faster.
- R Code/Shiny/snowflake_interative.R (snowflake_interative_cpp.R) is an UI for this technique.

13.3 References

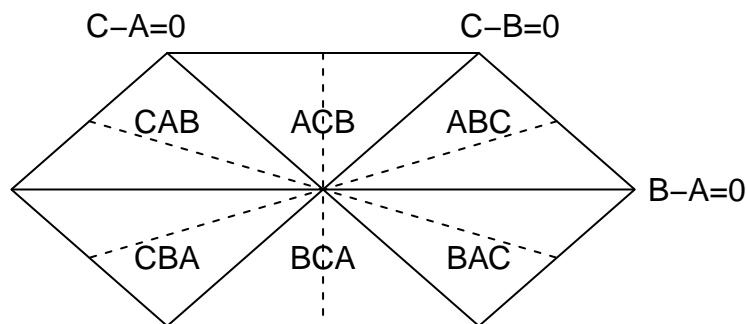
Readings/Snowflake/Czanner et al (2005).pdf

13.4 Testing Code

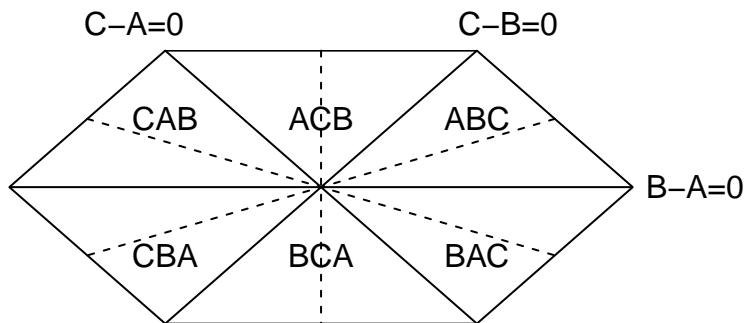
Note: plots cannot be generated completely. Refer to R Code/Snowflake/Plots

```
source("../Data/testing_data.R")
source("../Snowflake/snowflake_parallel.R")

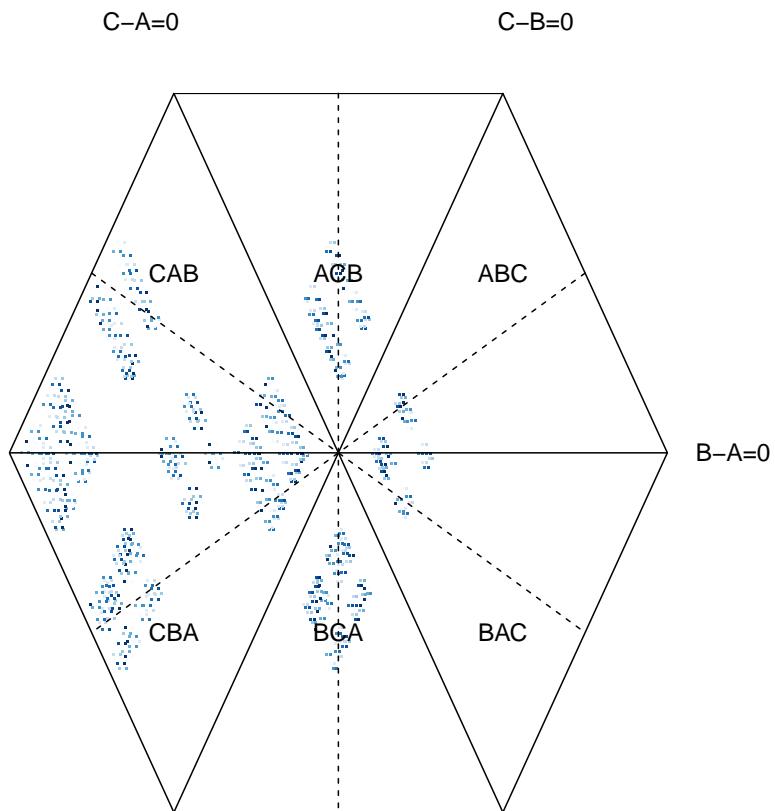
## testing
## snowflake plot of three spike trains A, B and E with span being 0.5 seconds
# 3n_1_1.5_par.pdf
span <- 0.5
snowflake(A, B, E, 1, 1.5, span)
```



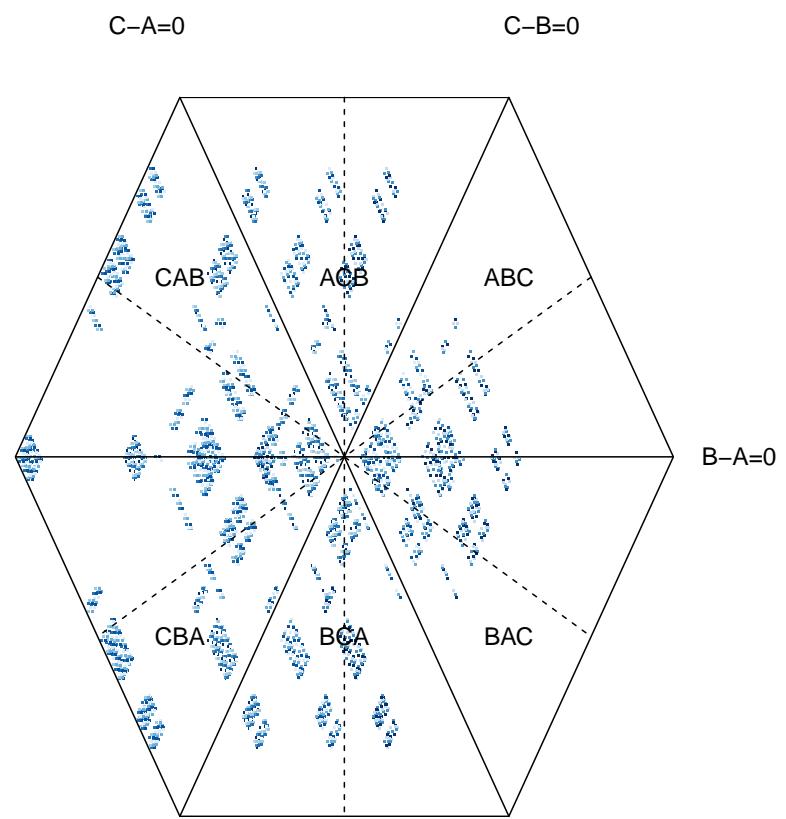
```
## snowflake plot of three spike trains A, B and C with span being 1 seconds
# 3n_1_2_par.pdf
span <- 1
snowflake(A, B, C, 1, 2, span)
```



```
# manually include plots  
knitr::include_graphics('..../Snowflake/Plots/3n_1_1.5_par.pdf')
```



```
knitr::include_graphics('..../Snowflake/Plots/3n_1_2_par.pdf')
```



13.5 Interpretation

- General
 - Points falling around blue lines mean two neurons firing at approximately the same time.
 - Referring to the paper, we can construct CCH and JPSTH from Snowflake plot by taking projections.
- 3n_1_1.5_par.pdf
 - More points fall around $B-A = 0$ means A and B are more synchronous than other pairs.
- 3n_1_2_par.pdf
 - This plot has similar patterns with 3n_1_1.5_par.pdf. However, it has more points since we take (2-1)s into account whereas 3n_1_1.5_par.pdf only takes (1.5-1)s second into account.

14 Spike Train Synchrony: Bivariate Distance

14.1 Description

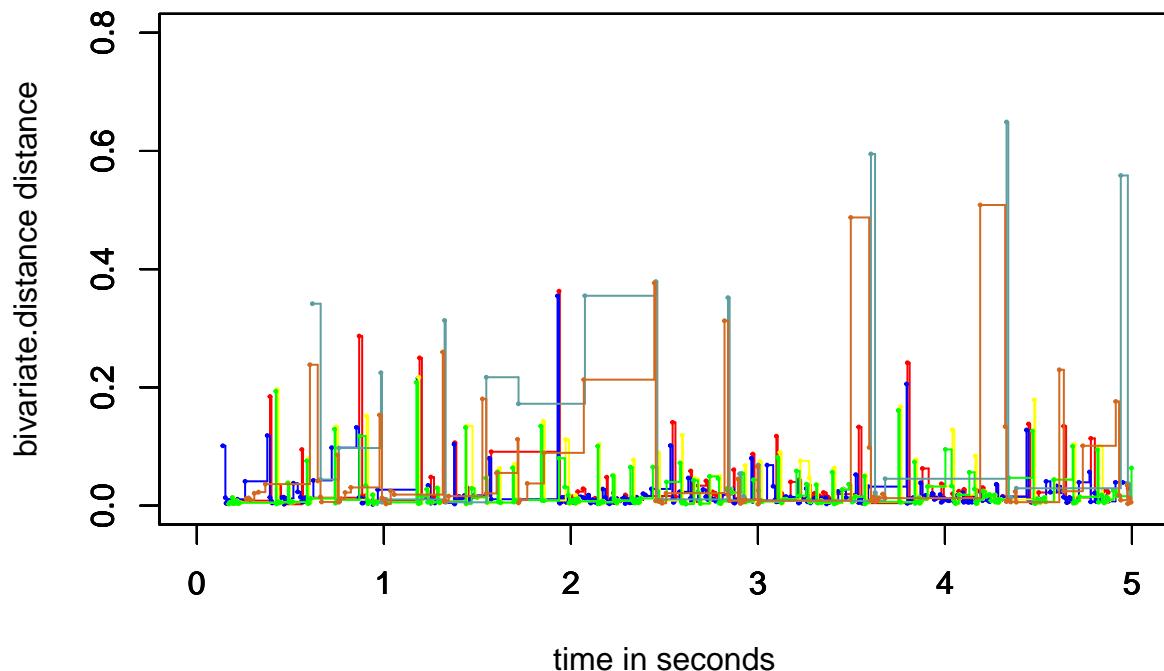
- Apply a scale-invariant measure of the relative firing rate pattern between two spike trains
- Distance plot: examine the firing rates in individual neurons
- Ratio plot: examine the firing rate ratios with respect to two neurons

14.2 References

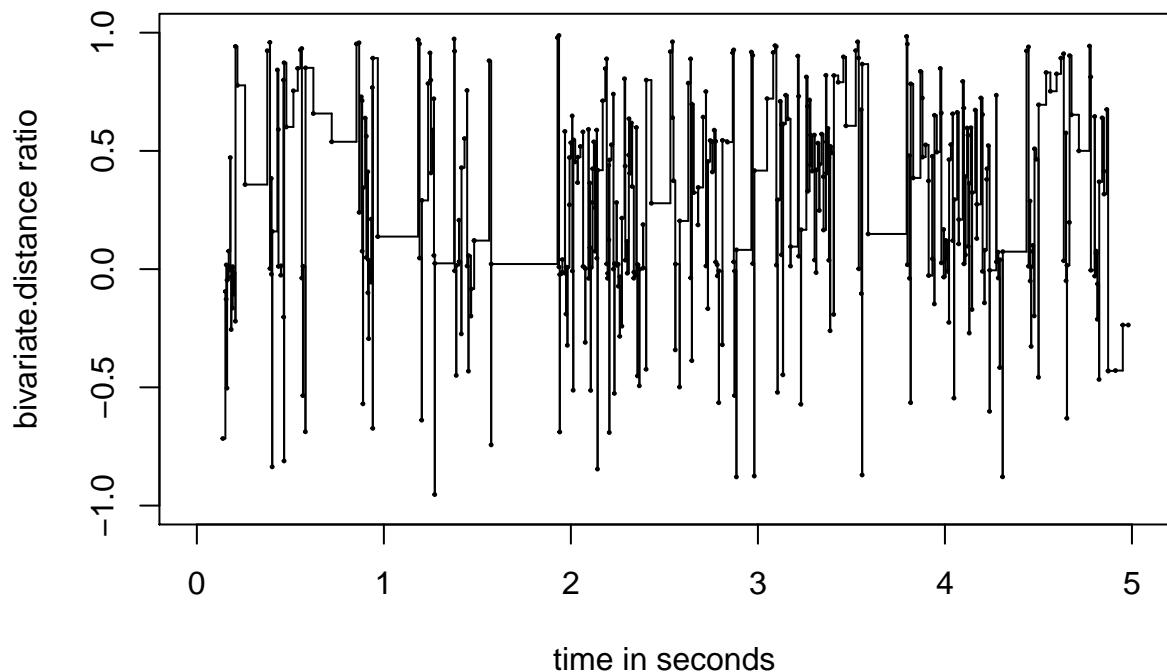
Readings/Spike Train Synchrony Measures/STS1.pdf

14.3 Testing Code

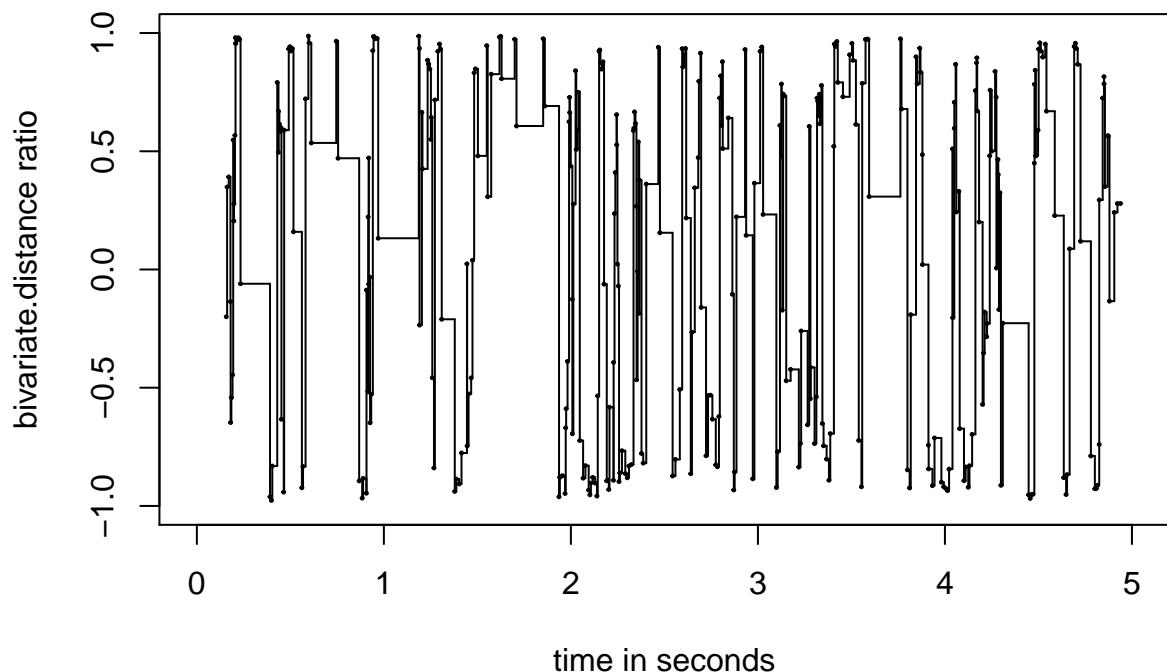
```
source("../STS/bivariate_distance.R")
source("../Data/testing_data.R")
#####
# bivariate.distance plot #####
# 6n_bivariate_distance.pdf
plot.bivariate.distance(A)
plot.bivariate.distance(B)
plot.bivariate.distance(C)
plot.bivariate.distance(D)
plot.bivariate.distance(E)
plot.bivariate.distance(F)
```



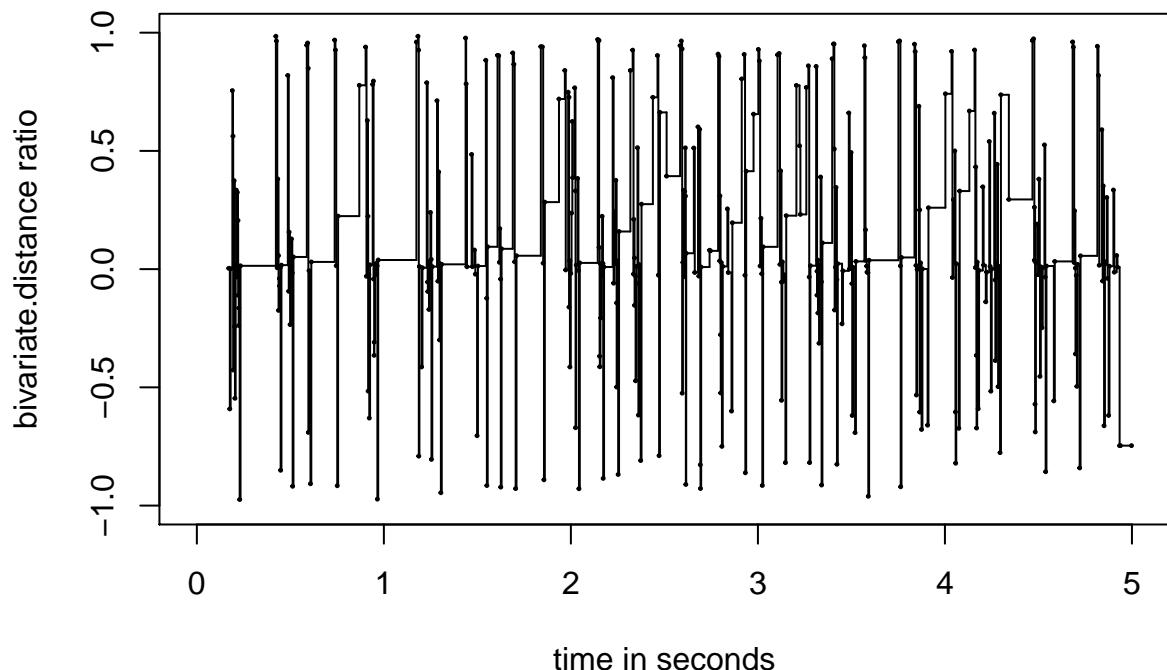
```
#####
# bivariate.distance ratio #####
# AB_bivariate_distance
plot.bivariate.distance.ratio(A, B)
```



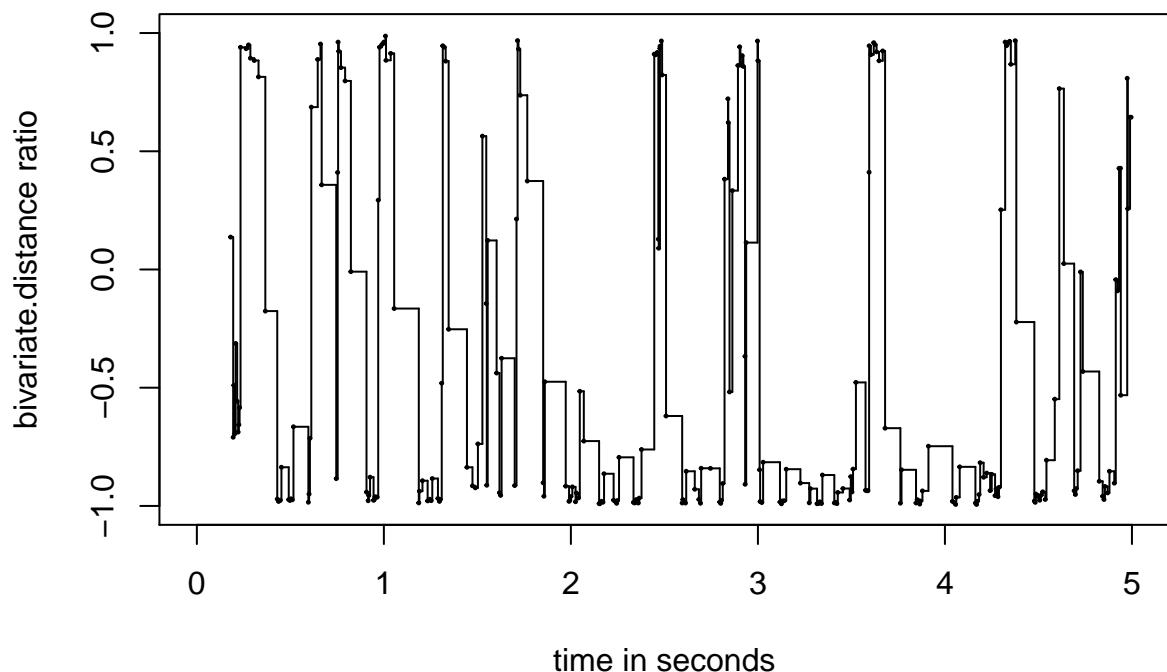
```
# AC_bivariate_distance  
plot.bivariate.distance.ratio(A, C)
```



```
# CD_bivariate_distance  
plot.bivariate.distance.ratio(C, D)
```



```
# CF_bivariate_distance
plot.bivariate.distance.ratio(C, F)
```



14.4 Interpretation

- General
 - In the distance ratio plots,
 - * points fall around $y=0$ show that two neurons at the time have similar firing rates.
 - * higher value on y -axis means greater time lapse since the last firing.

- In the distance ratio plots, repeating patterns over time can show coincidence, which are not presented in our examples.
- 6n_bivariate_distance.pdf
 - ISI distance of 6 neurons; each color corresponds to 1 neuron.
 - We can see E (cadetblue) and F(chocolate) have more larger values on y, which means they have fewer spikes on specific time segments than others.
- AB_bivariate_distance.pdf, AC_bivariate_distance.pdf, CF_bivariate_distance.pdf
 - In comparison, AB_bivariate_distance.pdf has more segments fall on y = 0, which indicates A and B are more synchronous; CF_bivariate_distance.pdf has fewest segments on y = 0, which indicates C and F are least synchronous.

15 Spike Train Synchrony: Multivariate Diversity

15.1 Description

- Apply the instantaneous coefficient of variation to visualize the synchrony bewtween multiple spike trains

15.2 Notes

- There is another plot in STS2: averaged bivariate ISI-distance. This one has not been completed due to $O(\prod_{i=1}^n a_i)$ computation cost, where n is the number of trains, and a_i is the number of spikes in train i.

15.3 References

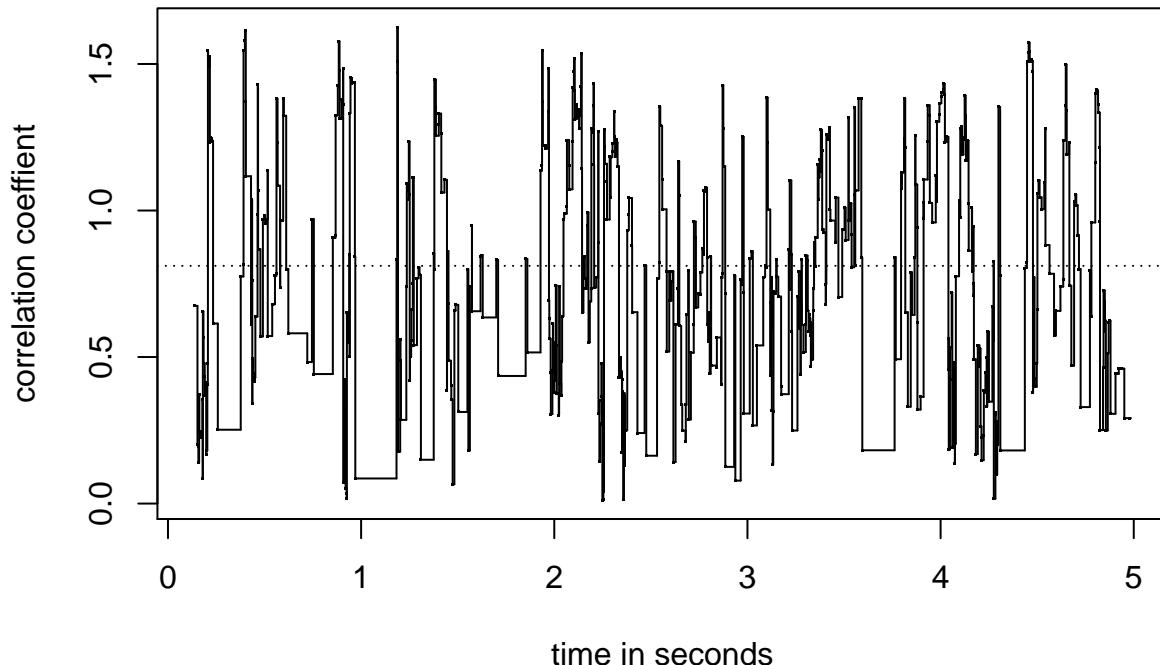
Readings/Spike Train Synchrony Measures/STS1.pdf
Readings/Spike Train Synchrony Measures/STS2.pdf

15.4 Testing Code

```
source("../STS/multivariate_diversity.R")
source("../Data/testing_data.R")

spike.trains <- list(A, B, C)
## ABC_multivariate_diversity.pdf
plot.multivariate.diversity(spike.trains)
```

multivariate diversity plot

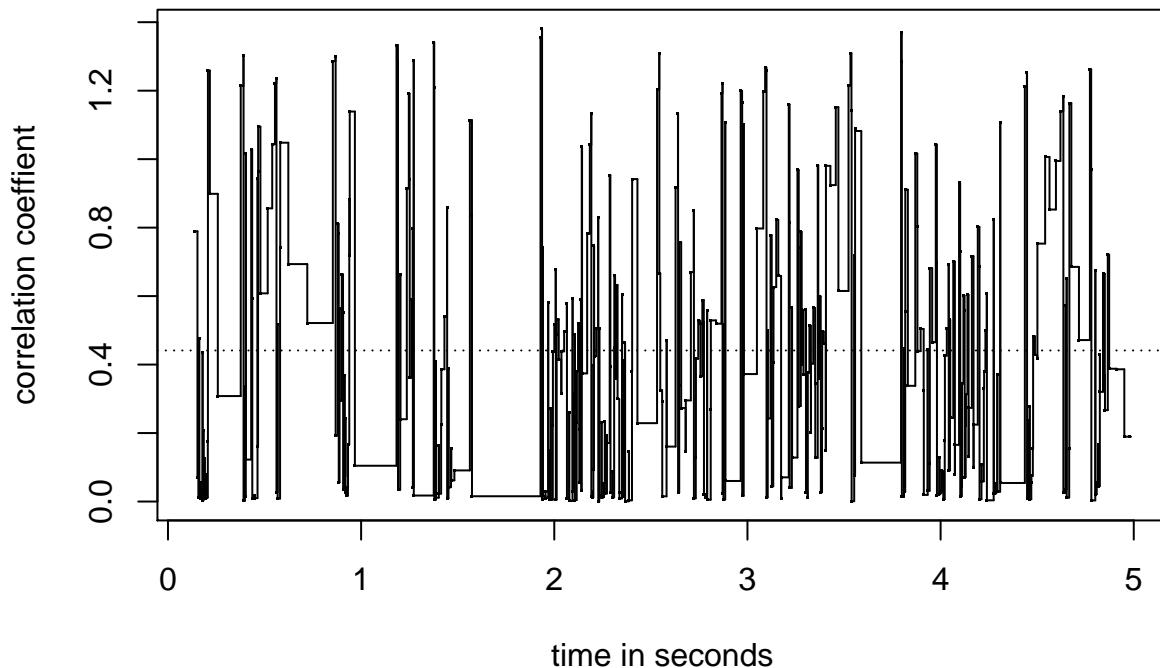


```

spike.trains <- list(A, B)
## ABC_multivariate_diversity.pdf
plot.multivariate.diversity(spike.trains)

```

multivariate diversity plot



15.5 Interpretation

- General
 - The coefficient of variation is very sensitive to outliers in ISI. That is, if one of the neurons is not in synchrony with others, it can boost the coefficient of variation.
- AB_multivariate_diversity.pdf, ABC_multivariate_diversity.pdf
 - Higher values indicate outliers in the ISI ratio exist.
 - Dotted line is the average correlation coefficient over time.
 - In comparison ABC_multivariate_diversity.pdf has more higher value than AB_multivariate_diversity.pdf, which is consistent with the fact C is an outlier to A and B.