Uncover viral sharing through global srtucure of host-virus meta-network

Mathis Gheno [1,2,‡]   Timothée Poisot [2,3,‡]

[1] Univeristé de Rennes   [2] Université de Montréal   [3] Québec Centre for Biodiversity Sciences

[‡] Equal contributions

Correspondance to:

Timothée Poisot — `timothee.poisot@umontreal.ca`

Purpose: This template provides a series of scripts to render a markdown document into an interactive website and a series of PDFs.

Internals: GitHub actions and a series of python scritpts. The markdown is handled with `pandoc`.

Motivation: It makes collaborating on text with GitHub easier, and means that we never need to think about the output.

Last revision: *July 24, 2023*

1 Introduction

2 Methods

Communicability quantify how well information transit between two nodes by considering all possible path in a network and penalizing longer ones. It is compute with the exponent of the adjacency matrix of the network.

$$G = \sum_{k=0}^{\infty} \frac{(\mathbf{A}^k)}{k!} = e^{\mathbf{A}}$$

where G is the communicability matrix, A the adjacency matrix and k is used as a penalizing term. It is possible to compute the exponential of a matrix with the graph spectrum :

$$G = \sum_{j=1}^{n} \varphi_j \varphi_j^T e^{\lambda_j}$$

where φ_j and λ_j are respectively the jth eigenvectors and eigenvalues of the matrix A. Obtain cluster with Communicability is done only by removing the fist dimension in the computation of the previous equation.

$$\Delta G = \sum_{j=1}^{n} \varphi_j \varphi_j^T e^{\lambda_j} - \varphi_1 \varphi_1^T e^{\lambda_1}$$

In this equation we are removing to global sum the dot production of the first eigenvector(and eigenvalue) to find the clustering matrix ΔG. Let's keep in mind this final result and explain why this work for clustering.

We start with the spectral form of G that can be decompose by the following way :

$$G = \varphi_1 \varphi_1^T e^{\lambda_1} + \sum_{j=2}^{n} \varphi_j^+ \varphi_j^{+T} e^{\lambda_j} + \sum_{j=2}^{n} \varphi_j^- \varphi_j^{-T} e^{\lambda_j} + \sum_{j=2}^{n} \varphi_j^- \varphi_j^{+T} e^{\lambda_j} \tag{1}$$

where φ_j^+ or φ_j^- indicate respectively all the positives or negatives values of the jth eigenvector. A way to think about it would be that when φ_j^+ all negative values are set to 0 and when when φ_j^- all positive values are set to 0. Estrada & Hatano (2008) explain that "two nodes have the same sign in an eigenvector if they can be

considered as being in the same partition of the network, while those pairs having different signs correspond to nodes in different partitions.".

$$\sum_{j=2}^{intracluster} \varphi_j \varphi_j^T e^{\lambda_j} = \sum_{j=2}^{n} \varphi_j^+ \varphi_j^{+T} e^{\lambda_j} + \sum_{j=2}^{n} \varphi_j^- \varphi_j^{-T} e^{\lambda_j}$$

and

$$\sum_{j=2}^{intercluster} \varphi_j \varphi_j^T e^{\lambda_j} = \sum_{j=2}^{n} \varphi_j^- \varphi_j^{+T} e^{\lambda_j}$$

so the clustering matrix is obtain with

$$\Delta G = \sum_{j=2}^{intracluster} \varphi_j \varphi_j^T e^{\lambda_j} - \left| \sum_{j=2}^{intercluster} \varphi_j \varphi_j^T e^{\lambda_j} \right|$$

The absolute operator is not useful, it is just here to remind that all inter cluster values are negatives. The dot product of $\varphi_j \varphi_j^T$ produce a matrix with positive and negative sign depending on the sign of $\varphi_j(p)$ and $\varphi_j(q)$. The first is not include because all the values the eigenvector the same sign, so it is not really informative (In fact it consider the hole network as one cluster thus it does not bring interesting information on the clustering).

In short we can rewrite ΔG as follow :

$$\Delta G = G - \varphi_1 \varphi_1^T e^{\lambda_1}$$

which was the form in which we introduce it.

Remark Each dimension of the spectra form the 2nd to the last one is a clustering configuration of the network. Cluster identified by dim 3 are not necessarily independent form those in dim 2 or 4 (or all others). Althoug the cluster form by dim 2 are "stronger" then 3, 4, 5 … and so on till the last one. So the dim 2 is the one that contibute the most for ΔG (we can see it in the following exemple). #### Example

```
library(tidyverse)
library(lattice)
library(igraph)
```

```
library(colorRamps)

A   = matrix(c(0,1,0,1,1,0,0,0,0,0,0,

               1,0,1,1,1,0,0,0,0,0,0,

               0,1,0,1,1,0,0,0,0,0,0,

               1,1,1,0,1,0,0,0,0,0,0,

               1,1,1,1,0,1,0,0,0,0,0,

               0,0,0,0,1,0,1,0,0,0,0,

               0,0,0,0,0,1,0,1,1,0,1,

               0,0,0,0,0,0,1,0,1,1,1,

               0,0,0,0,0,0,1,1,0,1,1,

               0,0,0,0,0,0,0,1,1,0,1,

               0,0,0,0,0,0,1,1,1,1,0), nrow =11, ncol =11)
grap = graph_from_adjacency_matrix(A, mode = "undirected")
plot(grap)
```

32 A graph with 11 nodes and 2 distinct group. First we need to compute the graph spectrum

```
spectra = eigen(A)
levelplot(spectra$vectors, ylab ="j th position" , xlab ="eigenvectors")
```

33 The above plot just represent the 11 eigenvectors. We can see that the fisrt one is the only full of same sign value.

34 Now let's take the 2nd dimension as an example.

```
##
G_dim2 = spectra$vectors[,2]%*%t(spectra$vectors[,2])*exp(spectra$values[2])
levelplot(G_dim2, ylab = "node", xlab ="node",col.regions = rev(matlab.like(16)))
```

35 And that it ! The second dimension of the graph communicability identify 2 cluster (in blue).

36 We can compute for the third dimension

```
G_dim3 = spectra$vectors[,3]%*%t(spectra$vectors[,3])*exp(spectra$values[3])

levelplot(G_dim3, ylab = "node", xlab ="node",col.regions = rev(matlab.like(16)))
```

Which identify clusters between 5:6 and 6:7. The cluster of the third dimension are less "obvious" than those from the second dimension

Now if we want to use both dimension in the clustering, we just have to add

```
levelplot(G_dim2+G_dim3, ylab = "node", xlab ="node",col.regions = rev(matlab.like(16)))
```

We could continue like that till the last dimension (11th), but it was for the explanation. So now we can compute directly ΔG by adding dimension from 2 to 11 (or subtraction of the first dim which is exactly the same)

```
delta_G = matrix(0, nrow =11, ncol =11)

for(dim in 2:11){

  delta_G = delta_G + spectra$vectors[,dim]%*%t(spectra$vectors[,dim])*exp(spectra$values[dim])

}

levelplot(delta_G, ylab = "node", xlab ="node",col.regions = rev(matlab.like(16)))
```

Results

[Figure 1 about here.]

[Figure 2 about here.]

[Figure 3 about here.]

[Figure 4 about here.]

[Figure 5 about here.]

[Figure 6 about here.]

[Figure 7 about here.]

[Figure 8 about here.]

[Figure 9 about here.]

[Figure 10 about here.]

Conclusion

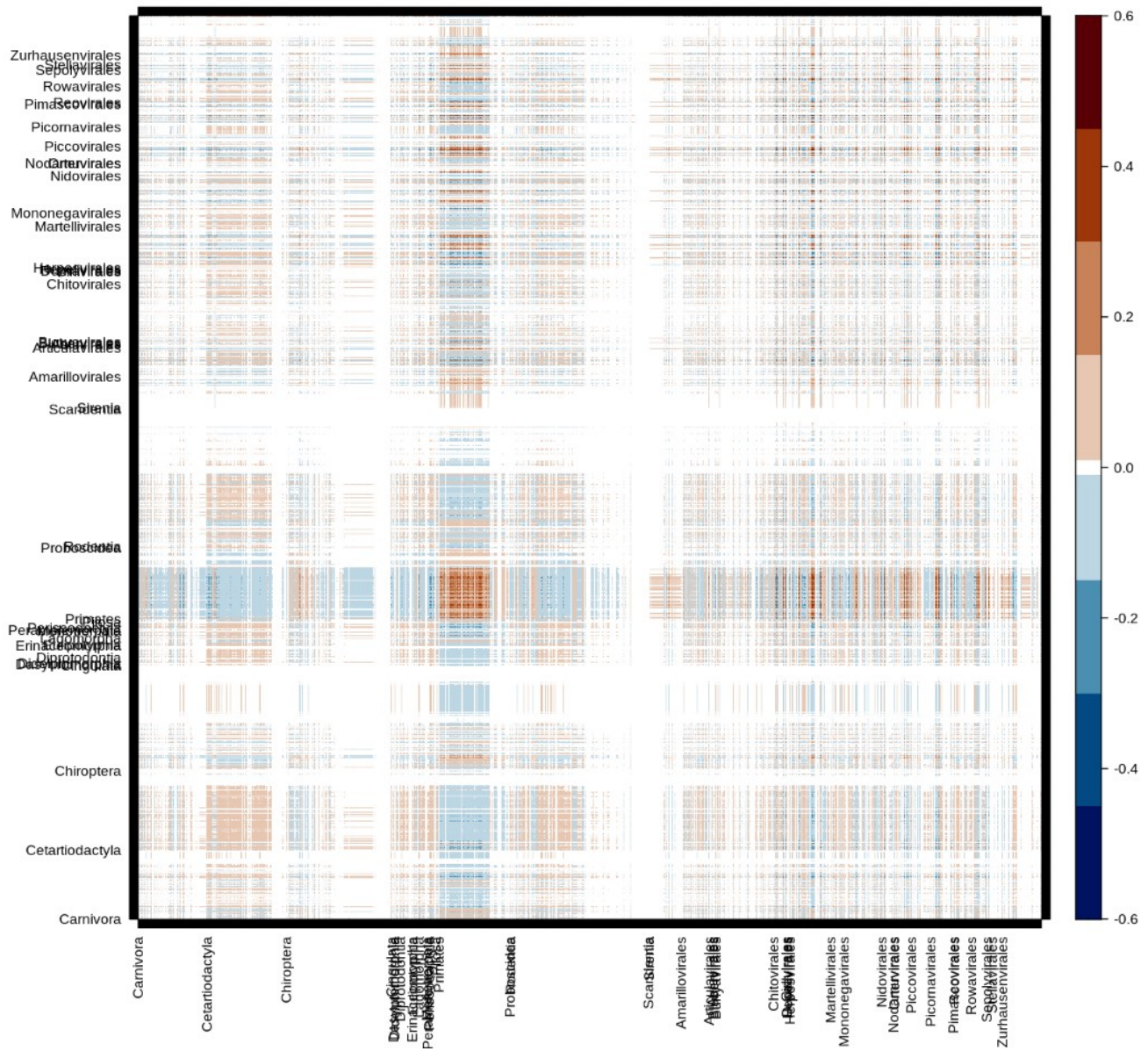Estrada, E. & Hatano, N. (2008). Communicability in complex networks. *Physical Review E*, 77, 036111.

Figure 1: Figure 1 : Global matrix of clustering communicability. Positive values indicate species in same cluster, negative value species in "opposite" cluster. Only host order and virus order names are display on x and y.
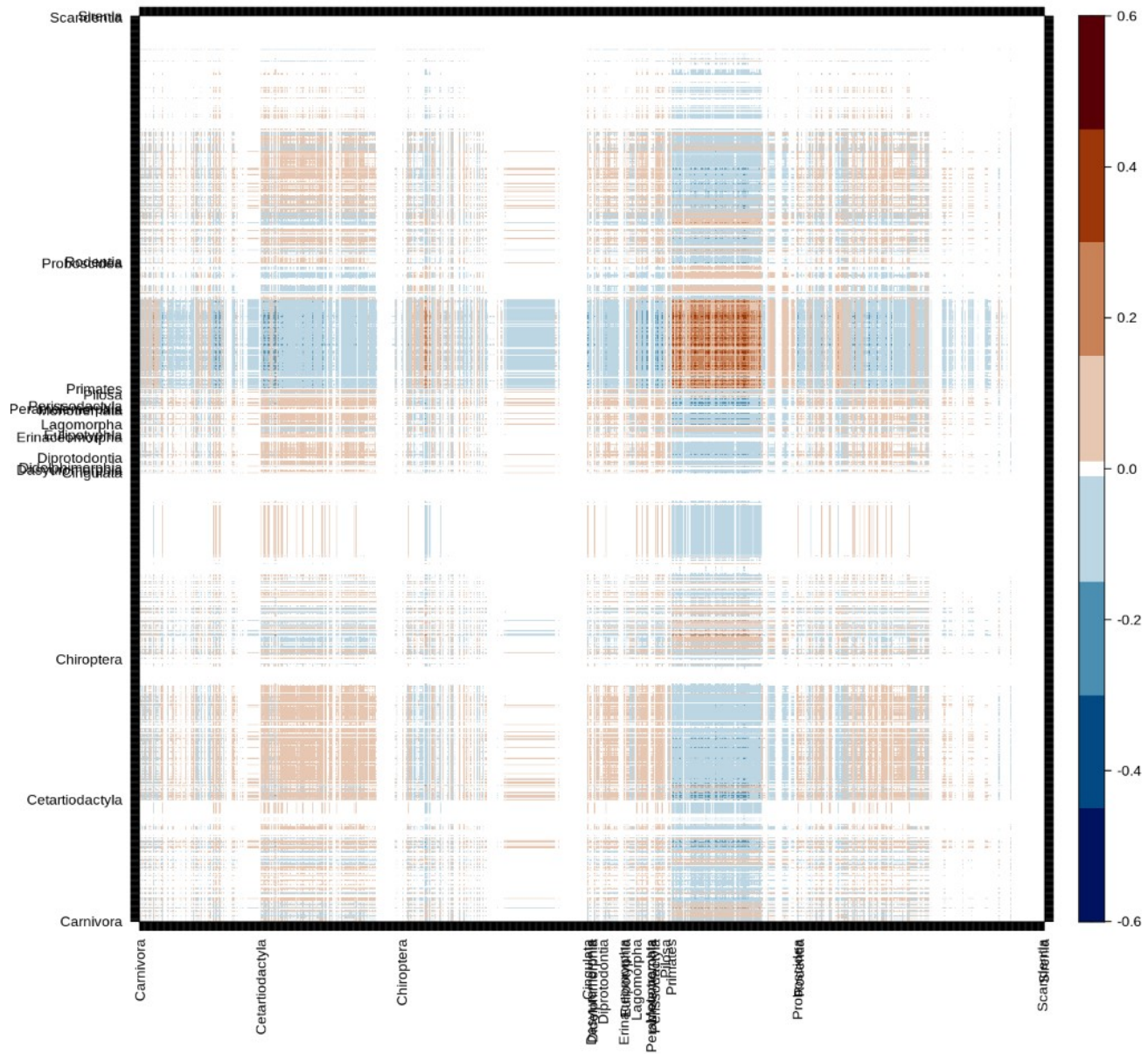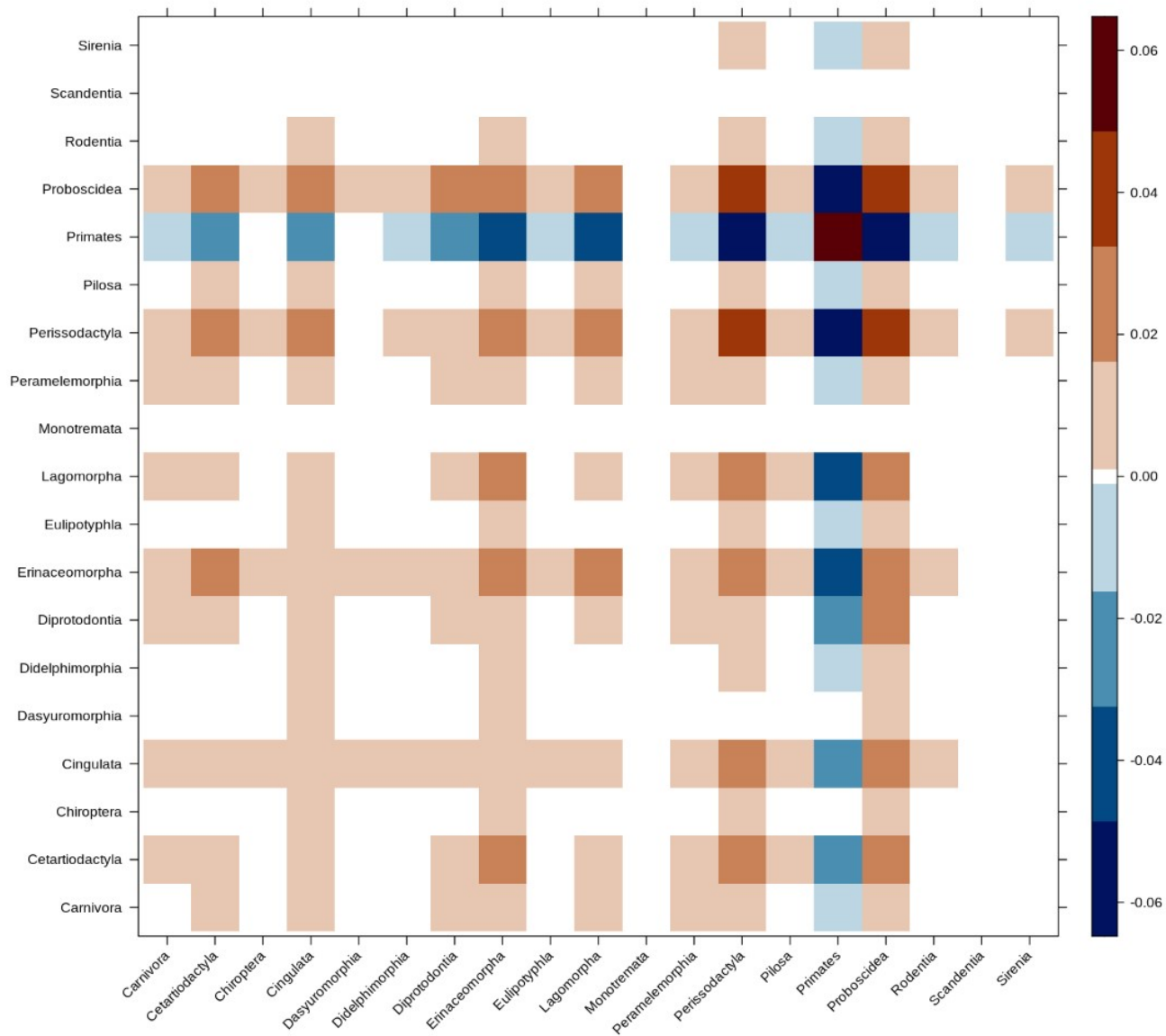
Figure 2: Figure 2 : Host-Host cluster matrix

Figure 3: Figure 3 : Host Host recap (with median of clustering value for each order)
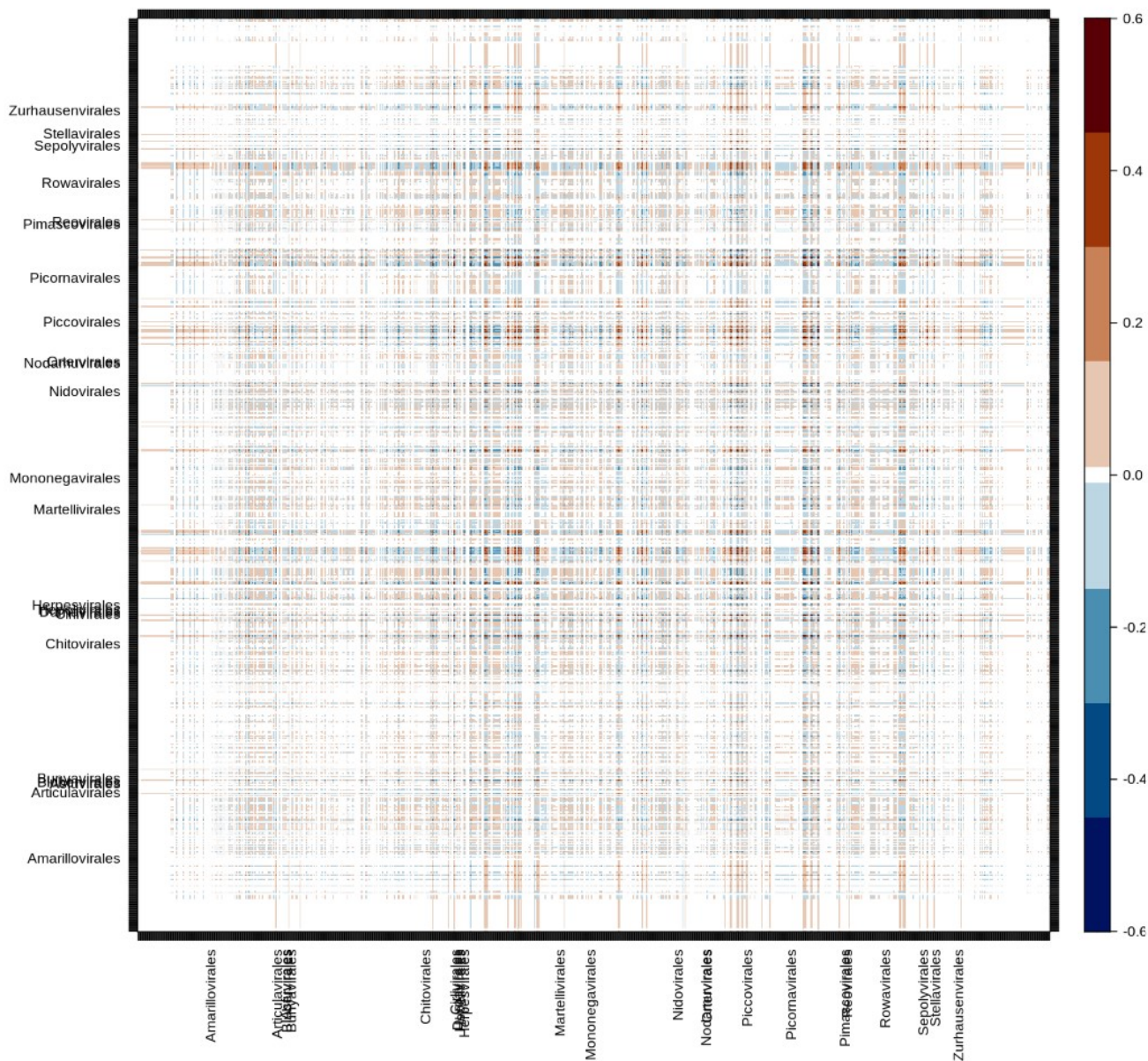
Figure 4: Figure 4 : Virus-Virus cluster matrix
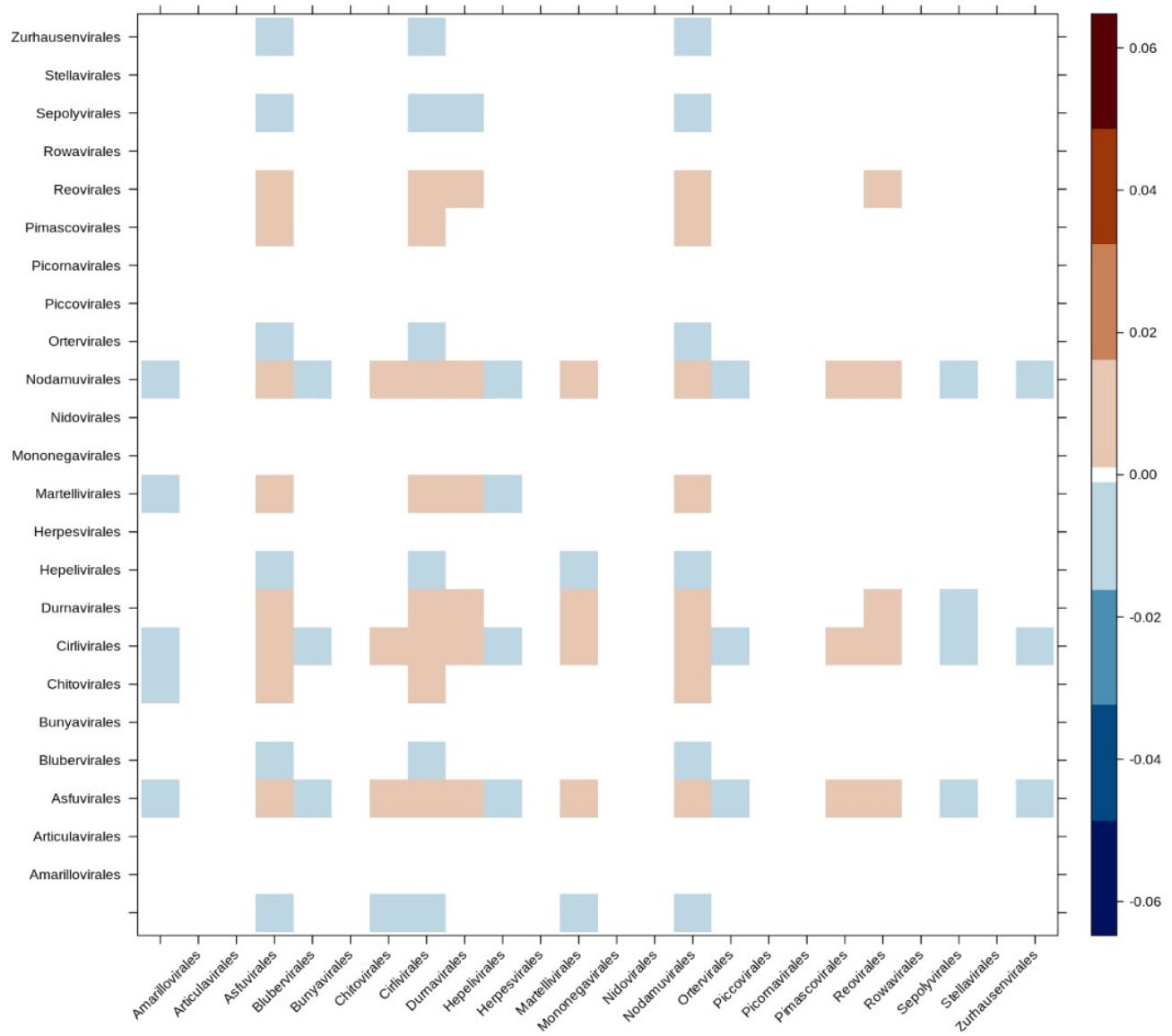
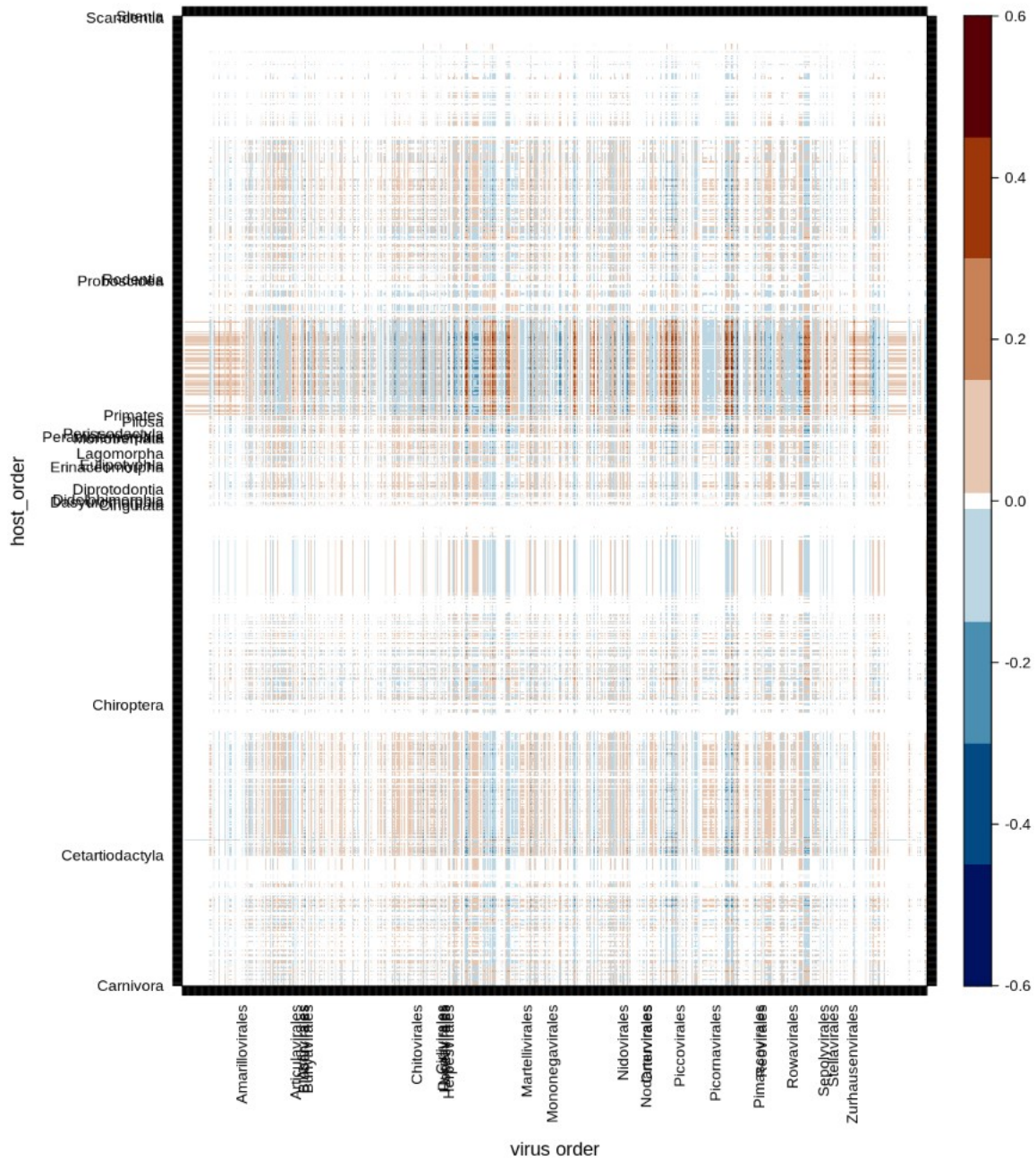Figure 5: Figure 5 : Virus-Virus recap (with median of cluster value for each order)
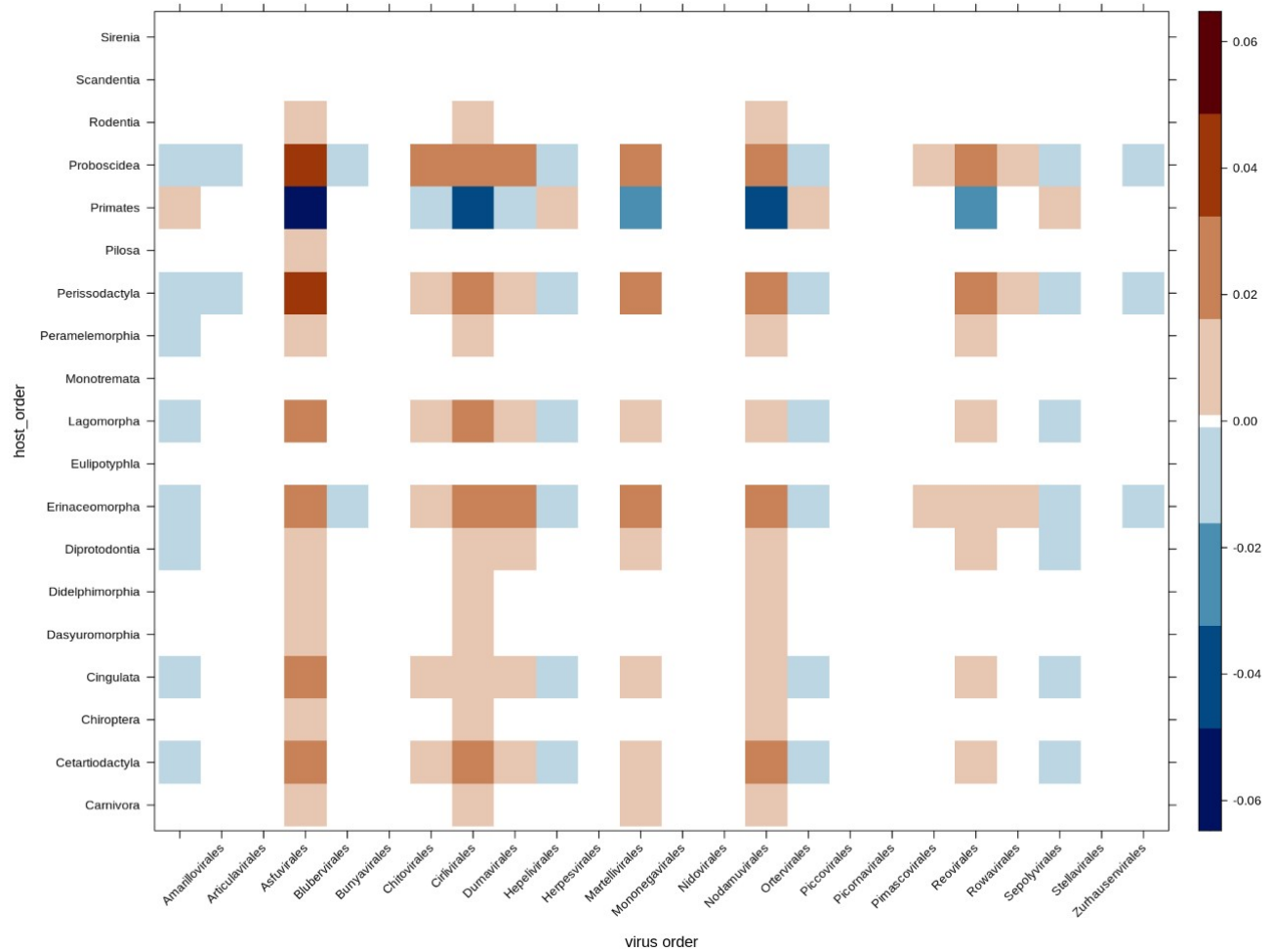
Figure 6: Figure 6 : Host-Virus cluster matrix

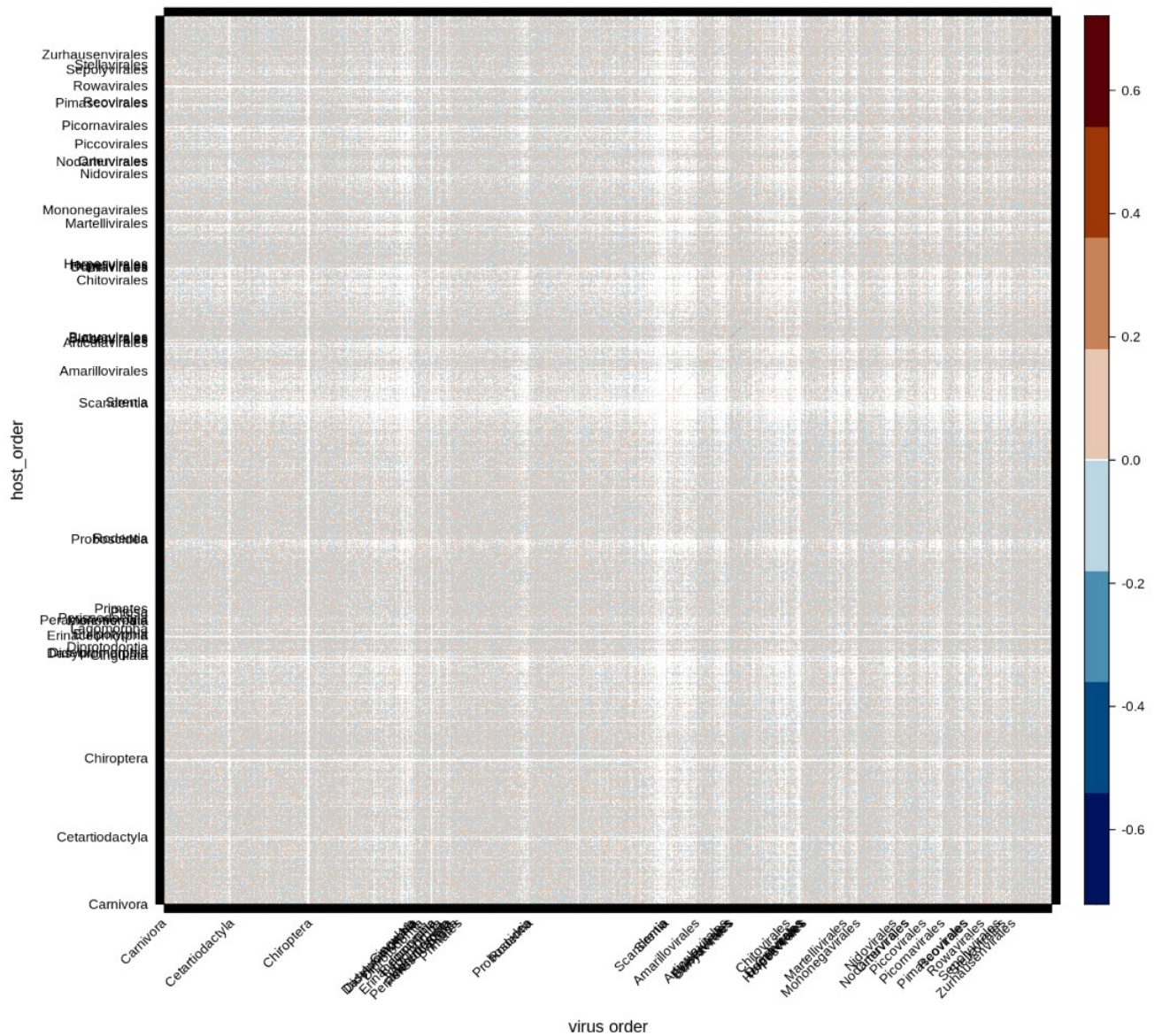Figure 7: Figure 7 : Host-Virus recap (with median of cluster value for each order)

Figure 8: Figure 8 : Global matrix of clustering communicability, with random edges connection. Number of edges for each nodes (Host and virus) is concerved. X and Y are ordered the same way as fig. 1
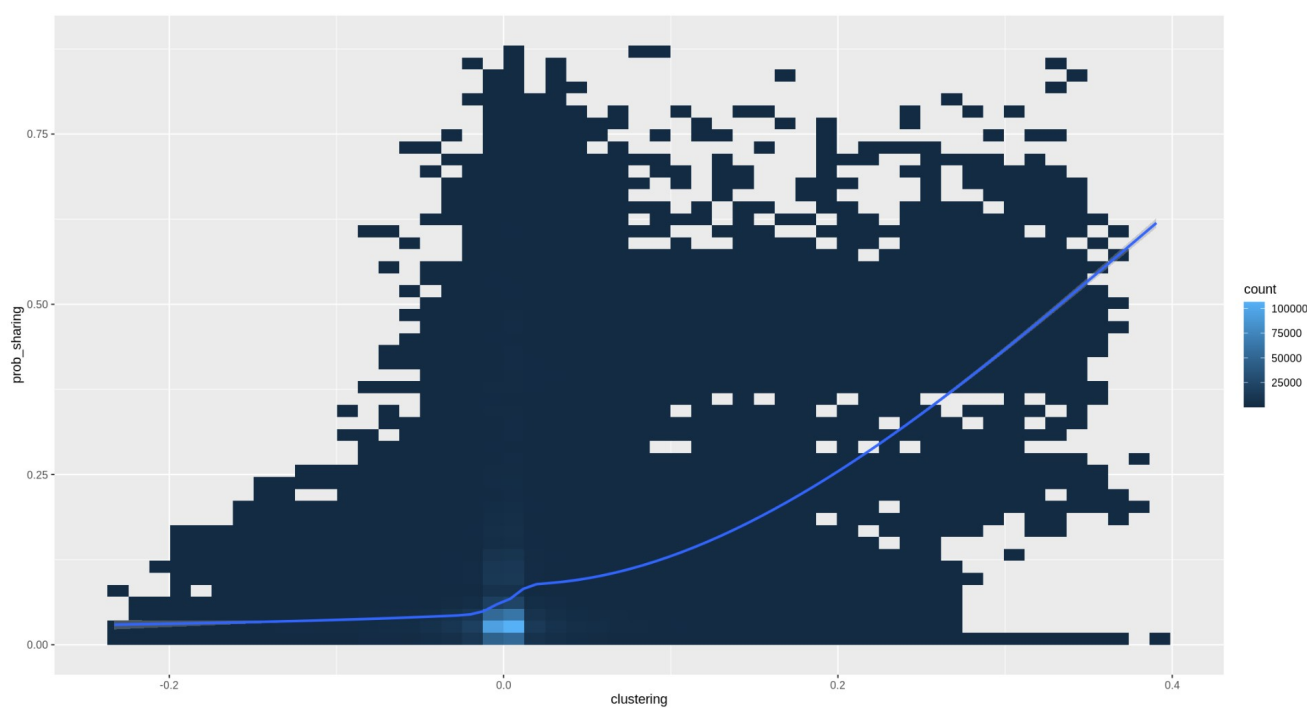
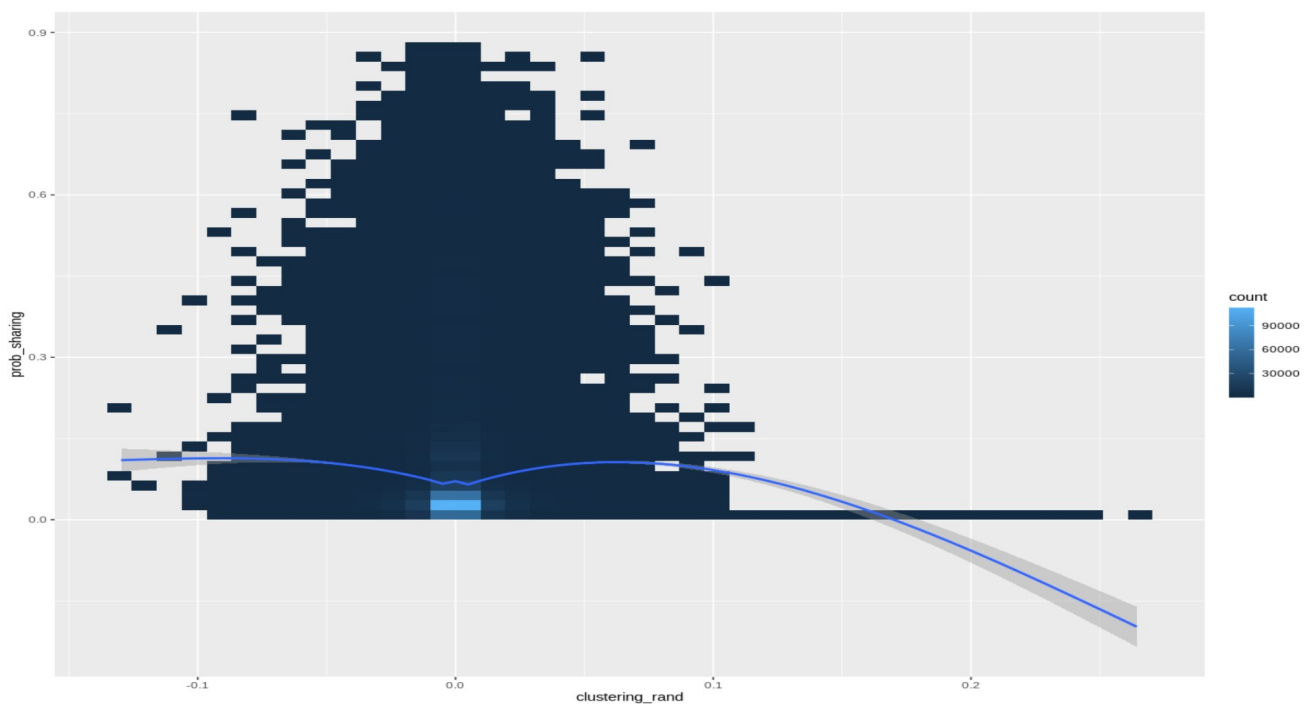Figure 9: Figure 9 : comparison with Albery sharing metric.

Figure 10: Figure 10 : comparison with Albery sharing metric for a random network