SPL Project

Network of CRC Discussion Papers

Alona Zharova*

1 Introduction

Interdisciplinarity is an essential component of complex research projects. Therefore, the identification of research outcomes that deals with two and more disciplines contributes greatly to the decision-making process of the management team.

The current work, as a part of the research project on evaluation of research performance of Collaborative Research Center 649 "Economic Risk" (CRC), aims to show if the research results of the CRC are interdisciplinary. For this purpose, we analyze discussion papers (DP) that serve as a result of the CRC research activity. Each one has a code indicating the subject fields according to the well-known in the economic sciences classification of Journal of Economic Literature (JEL), see Table 2 in Appendix.

The report is structured as follows. The description of the implemented theory and design

*Humboldt-Universität zu Berlin, School of Business and Economics, Ladislaus von Bortkiewicz Chair of Statistics and Collaborative Research Center 649 “Economic Risk”, Spandauer Str. 1, 10178 Berlin, Germany, Tel: +49 (0)30 2093 5708, Email: zharovaa@hu-berlin.de

is presented in Section 2. Section 3 introduces the implementation. Section 4 shows experimental results from testing on JEL codes and DPs. Finally, Section 5 summarizes results of this work and draws conclusions.

2 Theory and Design

In order to show, if the interdisciplinarity took place in CRC, the network visualization is used. For the construction of the network map, that results from graph theory, the nodes and edges (links) have to be defined.

All identical links or from-to combinations must be aggregated into "from" and "to" and the sum of weights written in "type". There are several R packages that provide the possibility to model networks in R. The most used ones are "igraph"(5), "network"(2), "sna"(3) and "ndtv"(1).

The used here R package "igraph" as it provides the necessary for the purpose tools as well as allows to define such key elements of the network as size, shape, color and position.

3 Implementation

In order to produce the network map, the following steps are essential.

1. The prepared data are assigned to the nodes and edges variables:

```
nodesdp = read.csv("CRCDPtestnodes.csv", header = TRUE, as.is = T)
linksdp = read.csv("CRCDPtestlinks.csv", header = TRUE, as.is = T)
```

2. Converting of the data into an igraph network object:

```
net = graph.data.frame(linksdp, nodesdp, directed = TRUE)
```

The igraph object consists of vertex (vertices) and edges, can have direction (directed or undirected) and such node attributes as name, weight and type.

3. Plotting the network map

```
plot(net)
```

As it is shown later, in certain circumstances (too many nodes, great difference between available weights, etc.) the plot will not look nice. In this case, each attribute of the network object can be adjusted.

4 Empirical Study

Data Preparation

In practice the most of the time takes the data preparation. The structure of the data that were received from Research Data Center (RDC) of the CRC 649 is introduced in Table 1. There are totally 776 DP published in CRC 649 from 2005 to 2016, June. The initial data has missing values, some mistakes in JEL codes and IDs, concentration of all JEL codes in one cell (column) and multiple letters of the same JEL code for one DP

DP.Authors	DP.SFBNumber	DP.Abstract	DP.FileName	DP.YearPublished	DP.Quantlet	DP.DateOfIssue	DP.JournalPubl	DP.ProjectsName	BARCODE.300	JEL
Grzegorz R.D.	9	This paper [...]	SFB649DP2016-009.pdf	2016		29.02.2016	[...]	C7		E44, F41
...	...	...	...	...	...	...	...	...	...	...

Table 1: Initial information on CRC DP received from RDC

The transformation of the data into the format necessary for the further programming include the following steps (the code is provided in Figure 6 in Appendix):

1. creating an id column consisting of the year of DP and its issue number (line 20)
2. checking, if there are mistakes or missimg JEL codes (lines 23–30)
3. preparing a function for splitting of JEL codes into different columns (lines 44–47)
4. fixing mistakes in JEL column (lines 50–56)

As it follows from the Section 3, the two specific data frames with nodes and edges (links) are needed for the creation of the igraph object. In order to prepare the nodes data frame, the necessary variables are selected, the columns with JEL codes are splitted and the data is sorted according to ID, see the code in Figure 6, lines 59–80. For the creation of the edges data frame, the variables with DP and their JEL codes are selected as "from" and "to", where the weight is formed from the number of similar rows, see Figure 6, lines 83–106.

Testing

When running the code described in Section 3 and plotting the igraph network object, introduced in Figure 1, it is hard to identify anything. Thus, the further adjustment of network attributes is needed.

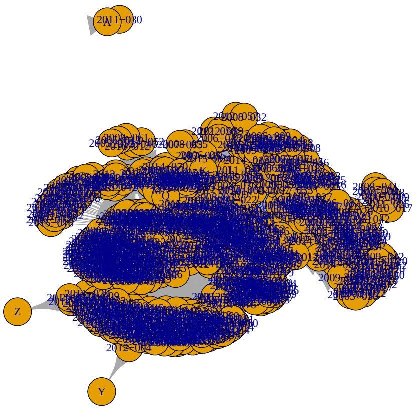

Figure 1: The result of the initial run of the code

After the removing of all labels and assigning of the total number of JEL "citations" by each DP to the nodes that introduce the JEL codes, the Figure 2 is received.

```
V(net)$label = ""
deg = igraph::degree(net, mode = "all")
V(net)$size <- (deg)
```

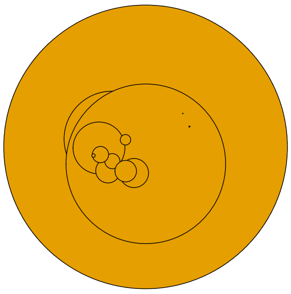

Figure 2: Testing without labels and with node degrees for JEL codes

As it could be seen from the Figure 2, some nodes are too large and overlap with each other. The Figure 3 introduce the result of the node size standardization.

```
deg = igraph::degree(net, mode = "all")
V(net)$size <- (deg)^0.5
```

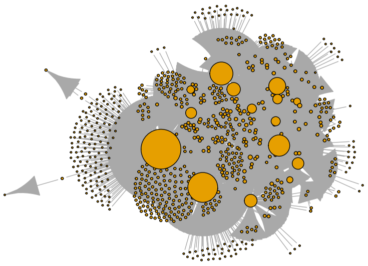

Figure 3: Testing with node degrees and standardized node size

Next, the arrow size is reduced, the labels are added to each JEL code and the text size is changed depending on the node size, as it is shown in Figure 4.

```
V(net)$label = ifelse(!is.na(V(net)$outlet.type), V(net)$outlet.name, NA)
```

```
E(net)$arrow.size = 0.2
```

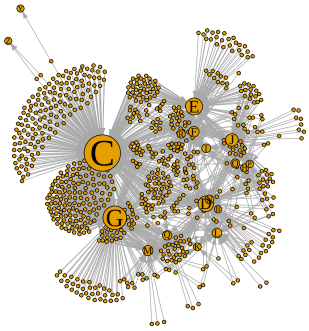

Figure 4: Testing with adjusted text size depending on node size and arrow size

The final result, received after the generating of colors based on node type, is introduced in Figure 2.

```
colrs = c("light blue", "gold")
V(net)$color = colrs[V(net)$art]
V(net)$label.color = "black"
```

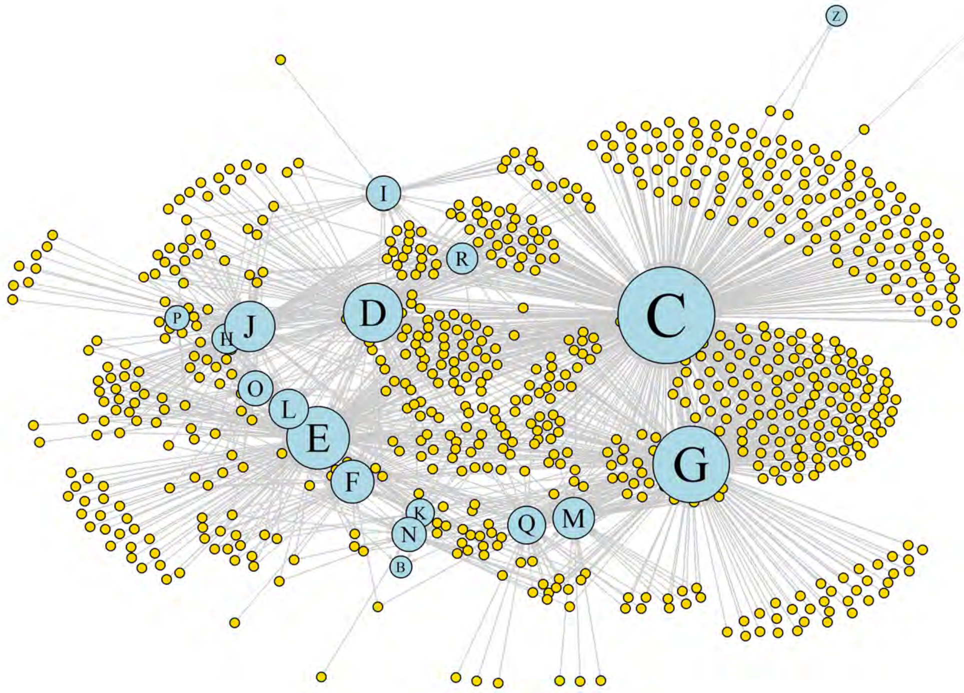

Figure 5: The adjusted network map

CRCnetjel

The "igraph" package provides the possibility to check which other layouts are available for the data. As it can be concluded from the Figure 7 in Appendix, this option could be used when the dataset is not too large.

```
layouts = grep("^layout_", ls("package:igraph"), value = TRUE)[-1]
layouts = layouts[!grepl("bipartite|merge|norm|sugiyama|tree", layouts)]
for (layout in layouts) {
    print(layout)
    l = do.call(layout, list(net))
    plot(net, edge.arrow.mode = 0, layout = l, main = layout)
}
```

Moreover, if the nodes should be shifted manually, the animation mode can be used. The screen shot in shown in Figure 8 in Appendix.

```
tkid = tkplot(net)
l = tkplot.getcoords(tkid)
plot(net, layout = l)
```

5 Conclusions

Summarizing this work, the network of collaborating disciplines can be clearly seen in Figure 5. The small gold circles introduce the discussion papers, whereas the nodes lead to the bigger blue circles indicating the JEL code of the corresponding research area. The size of each circle denotes the number of referrings from discussion papers. Summing up the results, it can be concluded that the interdisciplinarity of research results of the CRC could be shown through the network mapping of its DPs.

References

[1] Bender-deMoll, S. (2016) ndtv: Network Dynamic Temporal Visualizations. *https://CRAN.R-project.org/package=ndtv (accessed 03 Jul 2016).*

[2] Butts, C.T. (2008) network: a Package for Managing Relational Data in R. *Journal of Statistical Software* **24**(2) *http://www.jstatsoft.org/v24/i02/paper (accessed 03 Jul 2016).*

[3] Butts, C.T. (2014) sna: Tools for Social Network Analysis. *https://CRAN.R-project.org/package=sna (accessed 03 Jul 2016).*

[4] CRC Project on GitHub, `http://www.github.com/QuantLet/CRC` accessed 02 Jul 2016.

[5] Csardi, G. and Nepusz, T. (2006) The igraph software package for complex network research. *InterJournal, Complex Systems http://igraph.org (accessed 03 Jul 2016).*

[6] Gohil, A. R Data Visualization Cookbook. *Packt Publshing*, 2015.

[7] JEL (Journal of Economic Literature) Classification System, `https://www.aeaweb.org/econlit/jelCodes.php?view=jel` accessed 02 Jul 2016.

[8] Ognyanova, K. Network Analysis and Visualization with R and igraph. *NetSciX 2016 School of Code Workshop. http://kateto.net/networks-r-igraph (accessed 03 Jul 2016).*

[9] QuantNet, `http://quantlet.de/d3/ia/` accessed 02 Jul 2016.

6 Appendix

Code	Research field
A	General Economics and Teaching
B	History of Economic Thought, Methodology, and Heterodox Approaches
C	Mathematical and Quantitative Methods
D	Microeconomics
E	Macroeconomics and Monetary Economics
F	International Economics
G	Financial Economics
H	Public Economics
I	Health, Education, and Welfare
J	Labor and Demographic Economics
K	Law and Economics
L	Industrial Organization
M	Business Administration and Business Economics / Marketing / Accounting / Personnel Economics
N	Economic History
O	Economic Development, Innovation, Technological Change, and Growth
P	Economic Systems
Q	Agricultural and Natural Resource Economics / Environmental and Ecological Economics
R	Urban, Rural, Regional, Real Estate, and Transportation Economics
Y	Miscellaneous Categories
Z	Other Special Topics

Table 2: JEL Classification System

```
# Close windows and clear variables
graphics.off()
rm(list = ls(all = TRUE))

# Install packages / Load library
libraries = c("igraph")
lapply(libraries, function(x) if (!(x %in% installed.packages())) {install.packages(x)})
lapply(libraries, library, quietly = TRUE, character.only = TRUE)

##===============================================##
# Data preparation

test  = TRUE
corrections = TRUE

# Load data
crcdp = read.csv("CRCDP.csv", header = TRUE, as.is = TRUE, sep = ";")

# Create id column
crcdp$id = gsub(pattern = "SFB649DP|.pdf", replacement = "", x = crcdp$DP_FileName)

# Check if there are mistakes or missing JEL
if (test){
```

```
# wrong ids, too long ids
crcdp[nchar(crcdp$id) != 8, "id"]
# Papers without any JEL:
sum(crcdp$JEL == ""|crcdp$JEL == "\n ")
crcdp[crcdp$JEL == ""|crcdp$JEL == "\n ",]
crcdp = crcdp[!(crcdp$JEL == ""|crcdp$JEL == "\n "),]
}

# Create column with SP project area
crcdp$SP = gsub(pattern = "[[:digit:]]|[[:punct:]]|[[:space:]]",
    replacement = "",
x = crcdp$DP_ProjectsName)
crcdp$SP = strtrim(crcdp$SP, 1) # leave only one letter of first SP

# Create other columns
crcdp$outlet.name = crcdp$DP_Titel
crcdp$outlet.type = NA
crcdp$art = 2
crcdp$outlet = crcdp$DP_FileName

# Prepare function for splitting of JEL codes into different columns
jel_split = function(string){
string = gsub(pattern = "[[:digit:]]|[[:space:]]",replacement = "", string)
return(strsplit(x = string, split = "[[:punct:]]")[[1]])
}

# Fixe mistakes in JEL column
if(corrections){
crcdp$JEL = gsub(pattern = "and", replacement = ",", x = crcdp$JEL)
crcdp$JEL = gsub(pattern = "\nE44\nF34", replacement = "\nE44,\nF34",
x = crcdp$JEL)
crcdp$JEL = gsub(pattern = "\003", replacement = "", x = crcdp$JEL)
crcdp$JEL = gsub(pattern = "GO", replacement = "G0", x = crcdp$JEL)
}

# Prepare Nodes Data Frame
crcdp1 = crcdp[, c("id", "outlet", "outlet.type", "outlet.name", "art", "SP
    ")]
crcdp2 = data.frame(id = sort(unique(unlist(lapply(X = crcdp$JEL, FUN =
    jel_split)))),
outlet = sort(unique(unlist(lapply(X = crcdp$JEL,FUN = jel_split)))),
outlet.type = 1,
outlet.name = sort(unique(unlist(lapply(X = crcdp$JEL,FUN = jel_split)))),
art = 1,
SP = NA,
stringsAsFactors = FALSE
)

if (test){
tmp = vector()
for (i in 1:length(crcdp2$id)){
tmp[i] = any(crcdp2$id[i] == toupper(letters))
}
# wrong JEL codes
crcdp2$id[!tmp]
}
```

```

# Write Nodes Data Frame
crcdp_output = rbind(crcdp1, crcdp2)
write.csv(x = crcdp_output, file = "CRCnetjel_nodes.csv", row.names = FALSE
    )

# Prepare Edges Data Frame
crced = data.frame()
crced_entry = function(crced, from, to, weight){
tmp.num = dim(crced)[1] + 1
crced[tmp.num, "from"] = from
crced[tmp.num, "to"] = to
crced[tmp.num, "weight"] = weight
return(crced)
}

for (tid in crcdp$id){
num = grep(pattern = tid, crcdp$id)
# crcdp[num, "JEL"]
jel = table(jel_split(crcdp[num, "JEL"]))
# if any(nchar(names(jel))>1)
for (let in names(jel)){
crced = crced_entry(crced, from = tid, to = let, weight = jel[let])
}
}

if (test)
crced[nchar(crced$to) != 1, ]

# Write Edges Data Frame
write.csv(x = crced, file = "CRCnetjel_edges.csv", row.names = FALSE)

##===============================================##
# Ploting

# Read data
nodesdp = read.csv("CRCnetjel_nodes.csv", header = TRUE, as.is = TRUE)
linksdp = read.csv("CRCnetjel_edges.csv", header = TRUE, as.is = TRUE)

# Examine the data:
head(nodesdp)
head(linksdp)
nrow(nodesdp); length(unique(nodesdp$id))
nrow(linksdp); nrow(unique(linksdp[, c("from", "to")]))

linksdp = linksdp[order(linksdp$from, linksdp$to), ]

# Convert the data to an igraph object:
net = graph.data.frame(linksdp, nodesdp, directed = TRUE)

# Examine the resulting object:
class(net)
net

# Remove loops from the graph:
net = simplify(net, remove.multiple = FALSE, remove.loops = TRUE)
```

```r

# Access nodes, edges, and their attributes:
E(net)
V(net)
V(net)$outlet
V(net)$outlet.type
V(net)$outlet.name
V(net)$art
head(E(net))

# Generate colors based on type_node:
colrs = c("light blue", "gold")
V(net)$color = colrs[V(net)$art]

# Change arrow size and edge color:
E(net)$arrow.size = 0.2

# Compute node degrees (#links) and use that to set node size:
deg = igraph::degree(net, mode = "all")
V(net)$size = (deg)^0.5
sizes = V(net)$size[V(net)$art == 1]
a = 4
b = 20
V(net)$size = (V(net)$size-min(sizes))/(max(sizes)-min(sizes))*(b-a)+a
# V(net)$size[V(net)$art==1] = max(V(net)$size[V(net)$art==2])
V(net)$size[V(net)$art == 2] = 2

# JEL codes will have name labels, DPs will not:
V(net)$label = ""
V(net)$label[V(net)$type == FALSE] = nodesdp$outlet[V(net)$type == FALSE]
V(net)$label.cex = V(net)$size/6
V(net)$label.font = 1 # 2: bold labels
V(net)$label.family = "serif" # serif, sans, mono, symbol

# Plot
png(file = "CRCnetjel1.png", width = 10, height = 10, units="in", res = 600)
par(mar = c(0, 0, 0, 0))
plot(net,
vertex.label = ifelse(!is.na(V(net)$outlet.type), V(net)$outlet.name, NA),
vertex.label.color = "black",
edge.color = "gray80"
#    ,vertex.size = ifelse(!is.na(V(net)$outlet.type),6,2)
)
dev.off()

##================================================##
# Highlighting aspects of the network
png(file = "CRCnetjelhis.png")
hist(linksdp$weight)
mean(linksdp$weight)
sd(linksdp$weight)
dev.off()

##================================================##
# Check layouts available for our network
```

```
png(file = "CRCnetjellayouts.png")
layouts = grep("^layout_", ls("package:igraph"), value=TRUE)[-1]
layouts = layouts[!grepl("bipartite|merge|norm|sugiyama|tree", layouts)]
par(mfrow=c(3,5), mar=c(1,1,1,1))
for (layout in layouts) {
print(layout)
l = do.call(layout, list(net))
plot(net, edge.arrow.mode=0, layout=l, main=layout) }
dev.off()


##===============================================##
# Animation

tkid = tkplot(net) # tkid is the id of the tkplot that will open
l = tkplot.getcoords(tkid) # grab the coordinates from tkplot
plot(net, layout = l)
```

Figure 6: The R code

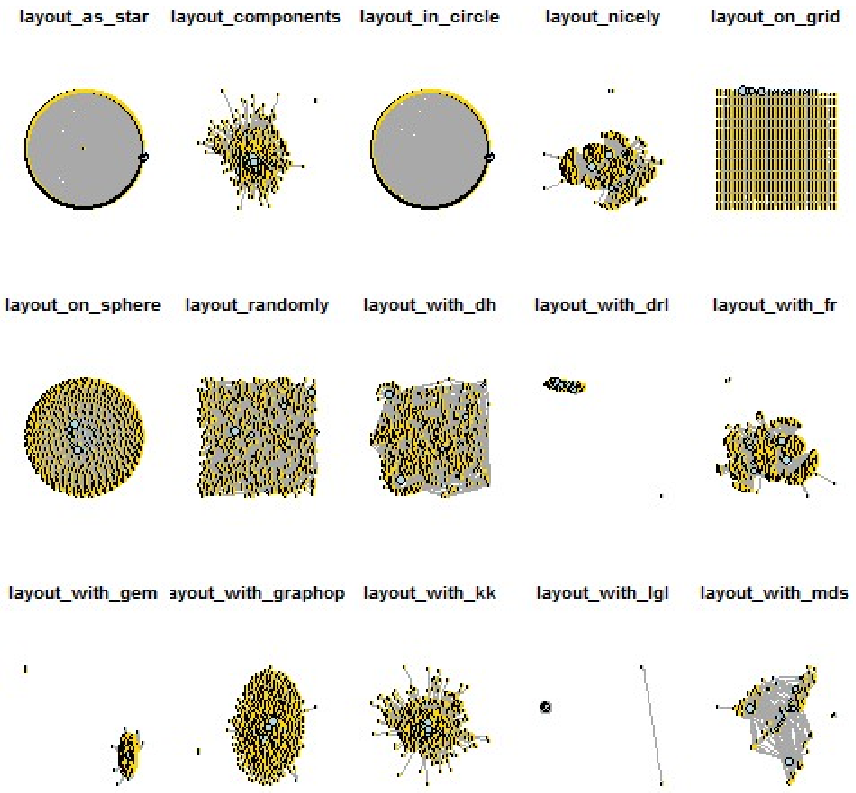

Figure 7: Available layouts for produced network map

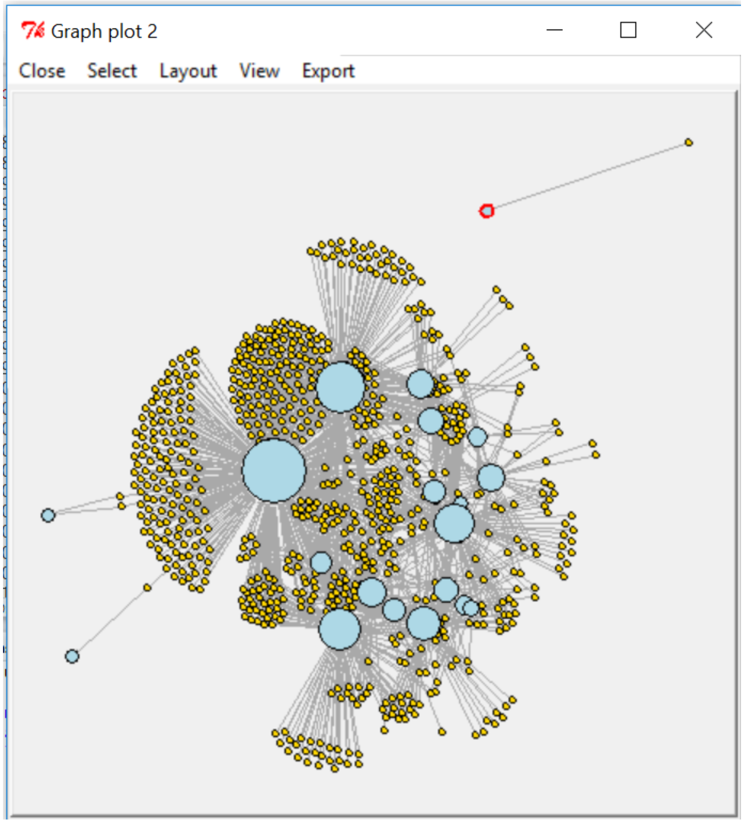

Figure 8: The animation with tkplot