Drilling into Erasmus learning mobility flows between countries 2014-2024

Vladimir Batagelj

IMFM Ljubljana, UP IAM Koper, UL FMF Ljubljana

Sredin seminar

Ljubljana April 16, 2025 at 05 : 03

Outline

Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si

Current version of slides (April 16, 2025 at 05 : 03): slides PDF

https://github.com/bavla/wNets/

Erasmus

Erasmus+ is a European Union (EU) program designed to support education, training, youth, and sport across Europe. Established in 1987, it aims to provide opportunities for individuals to study, train, gain work experience, and volunteer abroad, while also fostering cooperation and innovation in these fields.

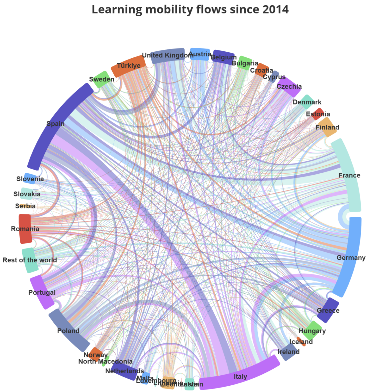
Key features of Erasmus+ are (1) mobility opportunities, (2) cooperation projects, (3) policy development, and (4) sport initiatives.

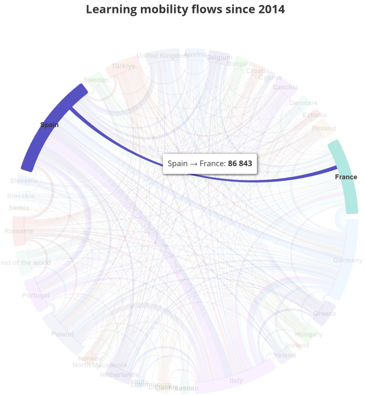
Erasmus+ is funded by the EU, with a budget of over €26 billion for the 2021-2027 period, making it one of the largest programs of its kind. It is open to EU member states, as well as non-EU countries associated with the program. Millions of individuals and thousands of organizations participate annually.

At the bottom of the Erasmus+ page Data visualization on learning mobility projects, the "Learning mobility flows since 2014" chart can be found.

Erasmus

interactive chart

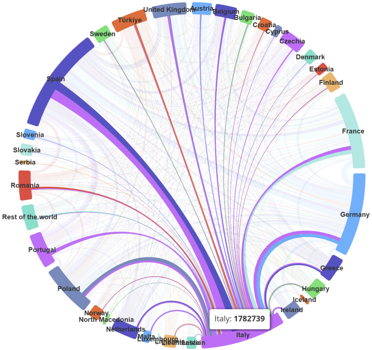


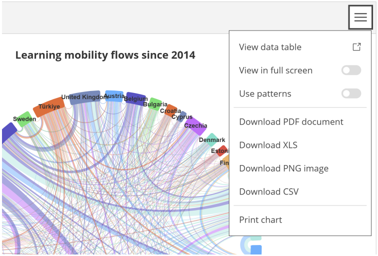


The interactive chart shows mobility flows between countries since 2014. The colors are related to the sending country.

Erasmus

. . . interactive chart





For example, moving the mouse over Italy will highlight all its in/outbound flows and the total count of participants. The same can be done at the flow level.

The interactive chart provides an option to download the network data.

I saved the network data on the file `Learning-mobility-flows-since-2014.csv`. The dataset contains the following countries:

Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Liechtenstein, Lithuania, Luxembourg, Malta, Netherlands, North Macedonia, Norway, Poland, Portugal, Rest of the world, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Türkiye, United Kingdom.

I used the Deepseek to obtain the corresponding ISO 3166-1 alpha-2 country codes and the total population estimate for each country.

I converted the collected data into Pajek files `ErasmusFlows.net`, `ErasmusFlowsISO.nam`, and `PopTotal.vec`. The created Pajek files are available at GitHub/Vlado.

Erasmus

. . . creating network

```
> wdir <- "C:/Users/vlado/DL/data/erasmus/flows"; setwd(wdir)
> source("https://raw.githubusercontent.com/bavla/Rnet/master/R/Pajek.R"
> F <- read.csv("Learning-mobility-flows-since-2014.csv")
> str(F)
'data.frame':   1223 obs. of  4 variables:
 $ Category  : chr  "highcharts-hv9y537-3" "highcharts-hv9y537-4" "highc
 $ X..from.  : chr  "Austria" "Austria" "Austria" "Austria" ...
 $ X..to.    : chr  "Austria" "Belgium" "Bulgaria" "Croatia" ...
 $ X..weight.: int  8394 6825 603 2389 780 3808 3962 1104 7014 15250 ...
> from <- factor(F$X..from.); to <- factor(F$X..to.)
> uvFac2net(from,to,w=F$X..weight.,Net="ErasmusFlows.net")
> C <- read.csv("ISO2.csv",strip.white=TRUE)
> str(C)
'data.frame':   35 obs. of  2 variables:
 $ country: chr  "Austria" "Belgium" "Bulgaria" "Croatia" ...
 $ ISO2   : chr  "AT" "BE" "BG" "HR" ...
> V <- 1; levels(V) <- C$ISO2
> uvFac2net(V,V,Net="ErasmusFlowsISO.nam")
> Pop <- read.csv("pop.csv",strip.white=TRUE)
> str(Pop)
'data.frame':   35 obs. of  3 variables:
 $ n      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ country: chr  "Austria" "Belgium" "Bulgaria" "Croatia" ...
 $ pop    : num  9000000 11500000 6800000 4000000 1200000 10700000 59000
> vector2vec(Pop$pop,Vec="PopTotal.vec")
```

Matrix representation

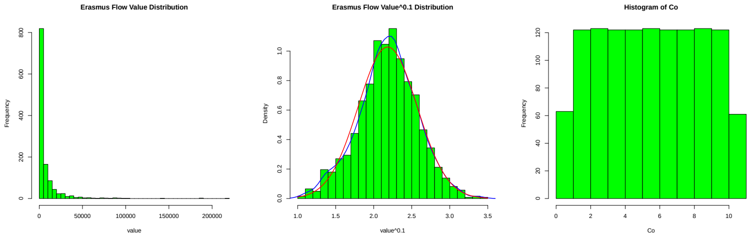
comments

- Larger, $n > 20$, dense graphs can't be presented readably with a graphical layout. For the Erasmus network, the number of nodes is $n = 35$, and the density $\gamma = 0.9984$. For dense graphs of moderate size (up to some hundreds of nodes) a better option is the matrix representation.
- What about weights? They can be represented by link thickness or level of grey of matrix cells. The problem is a very large range and the distribution of weights – most weights give almost white cells. For Erasmus $w_{\text{min}} = 1$ and $w_{\text{max}} = 217003$. Monotonic transformations such as $w' = a \cdot w$, $a > 0$ or $w' = \sqrt{w}$ or $w' = \log(w)$, etc. In our case, we used $w' = w^{0.1}$.
- A better ordering of rows/cols in the matrix representation can be obtained by network clustering [Batagelj etal.(2014)]. Additional reordering of subtrees can be made manually using R or Pajek by reordering nodes in the hierarchy.

```
> wdir <- "C:/Users/vlado/DL/data/erasmus/flows"; setwd(wdir)
> source("https://raw.githubusercontent.com/bavla/Rnet/master/R/Pajek.R"
> Z <- P <- net2matrix("ErasmusFlows.net")
> n <- nrow(P); V <- sort(P[P>0])
> hist(V,col="green",border="black",breaks=50,
+    xlab="value",main="Erasmus Flow Value Distribution")
> hist(V**0.1,col="green",border="black",breaks=20,prob=TRUE,
+    xlab="value^0.1",main="Erasmus Flow Value^0.1 Distribution")
> lines(density(V**0.1,n=64),lwd=2,col="blue")
> m <- mean(V**0.1); s <- sd(V**0.1)
> m
[1] 2.183272
> s
[1] 0.3881439
> curve(dnorm(x,m,s),from=1,to=3.5,lwd=2,col="red",xaxt="n",yaxt="n",
+    add=TRUE)
> b <- rep(0,11); b[11] <- max(V); Co <- P
> for(i in 1:10) b[i] <- V[round((-1+2*i)*length(V)/20)]
> for(i in 1:n) for(j in 1:n)
+  {k <- 1; while(P[i,j]>b[k]) k <- k+1; Co[i,j] <- k}
> b
 [1]   53   350   786   1327   2112   3178   4621   7323   13074   30142   217003
> hist(Co,col="green",breaks=0:11)
```

Erasmus network

. . . monotonic transformations and weight distributions



Monotonic (increasing) transformation $f : \mathbb{R} \to \mathbb{R}$

$$x < y \Rightarrow f(x) \leq f(y)$$

They preserve the ordering of weights.
Let $w' = f \circ w$ then $w(x) < w(y) \Rightarrow w'(x) \leq w'(y)$.

Hubs and authorities

To each node v of a network $\mathcal{N} = (\mathcal{V}, \mathcal{L}, w)$ we assign two values: quality of its content (*authority*) x_v and quality of its references (*hub*) y_v.
A good authority is selected by good hubs; and good hub points to good authorities (see Kleinberg).

$$x_v = \sum_{u:(u,v)\in\mathcal{L}} w(u,v) y_u \qquad \text{and} \qquad y_v = \sum_{u:(v,u)\in\mathcal{L}} w(v,u) x_u$$

Let $\mathbf{W}$ be a matrix of network $\mathcal{N}$ and $\mathbf{x}$ and $\mathbf{y}$ authority and hub vectors. Then we can write these two relations as $\mathbf{x} = \mathbf{W}^T\mathbf{y}$ and $\mathbf{y} = \mathbf{W}\mathbf{x}$.
We start with $\mathbf{y} = [1, 1, \ldots, 1]$ and then compute new vectors $\mathbf{x}$ and $\mathbf{y}$. After each step, we normalize both vectors. We repeat this until they stabilize. We can show that this procedure converges.
The limit vector $\mathbf{x}^*$ is the principal eigenvector of the matrix $\mathbf{W}^T\mathbf{W}$; and $\mathbf{y}^*$ of matrix $\mathbf{W}\mathbf{W}^T$.

Erasmus network

hubs and authorities

Rank	hub (source) Value	Id	authority (destination) Value	Id
1	0.4667	France	0.6120	Spain
2	0.4475	Italy	0.3985	Italy
3	0.4120	Germany	0.3230	Germany
4	0.3829	Spain	0.2894	France
5	0.2720	Poland	0.2363	United Kingdom
6	0.1793	Türkiye	0.1956	Portugal
7	0.1472	Romania	0.1886	Poland
8	0.1392	Netherlands	0.1526	Ireland
9	0.1361	United Kingdom	0.1193	Belgium
10	0.1339	Portugal	0.1142	Netherlands
11	0.1123	Rest world	0.1108	Greece
12	0.1017	Greece	0.0976	Finland
13	0.0910	Czechia	0.0962	Czechia
14	0.0905	Belgium	0.0938	Sweden
15	0.0797	Austria	0.0912	Romania
16	0.0777	Hungary	0.0904	Türkiye
17	0.0700	Finland	0.0882	Austria
18	0.0641	Sweden	0.0800	Malta
19	0.0625	Bulgaria	0.0764	Rest world
20	0.0610	Lithuania	0.0687	Hungary
21	0.0494	Slovakia	0.0619	Norway
22	0.0484	Denmark	0.0568	Denmark
23	0.0417	Croatia	0.0436	Lithuania
24	0.0400	Ireland	0.0400	Croatia
25	0.0383	Norway	0.0389	Bulgaria
26	0.0358	Latvia	0.0348	Slovenia
27	0.0347	Slovenia	0.0334	Slovakia
28	0.0278	Estonia	0.0296	Latvia
29	0.0147	N Macedonia	0.0261	Estonia
30	0.0128	Cyprus	0.0254	Cyprus
31	0.0120	Serbia	0.0153	Iceland
32	0.0087	Malta	0.0134	Luxembourg
33	0.0068	Iceland	0.0098	N Macedonia
34	0.0052	Luxembourg	0.0077	Serbia
35	0.0008	Liechtenstein	0.0007	Liechtenstein

In stochastic (Markov) normalization **S** of the network weight matrix **W**

$$S[u,v] = \frac{w(u,v)}{\sum_v w(u,v)}$$

we consider its entry $S[u,v]$ as the probability of transition from node u to node v.

```
> HubAut <- function(P,eps=0.0000005,rep=30){
+   m <- ncol(P); y <- rep(1,m)
+   for(i in 1:rep){yo <- y
+     x <- t(P) %*% y; x <- x/sqrt(sum(x**2))
+     y <- P %*% x; y <- y/sqrt(sum(y**2))
+     err <- sum(abs(y-yo))
+     if(err<eps) break
+     # cat(i,err,"\n")
+   }
+   return(list(hub=y,aut=x,rep=i,err=err))
+ }
>
> S <- t(apply(P, 1, function(x) x/sum(x)))
> R <- HubAut(S)
> data.frame(hub=S$hub,aut=S$aut)
```

Erasmus network

Stochastic (Markov) hubs and authorities

```
> data.frame(hub=R$hub,aut=R$aut)
                        hub        aut
Austria           0.1817131 0.10890332
Belgium           0.1782539 0.11070990
Bulgaria          0.1604739 0.06425047
Croatia           0.1577055 0.07142584
Cyprus            0.1409318 0.05535802
Czechia           0.1584400 0.12199506
Denmark           0.1720647 0.07922070
Estonia           0.1444778 0.04819643
Finland           0.1804099 0.10652933
France            0.1912070 0.29249107
Germany           0.1735351 0.40000708
Greece            0.1818980 0.13927542
Hungary           0.1644013 0.08599074
Iceland           0.1371526 0.03579342
Ireland           0.1973568 0.10170555
Italy             0.2067337 0.37901789
Latvia            0.1465599 0.04559441
Liechtenstein     0.1468488 0.01355996
Lithuania         0.1525463 0.06114302
Luxembourg        0.1738805 0.03368790
Malta             0.1482544 0.07501223
Netherlands       0.1894975 0.13333959
North Macedonia   0.1267503 0.02356214
Norway            0.1762025 0.06933105
Poland            0.1899846 0.19716810
Portugal          0.1904467 0.19972449
Rest of the world 0.1636026 0.10606983
Romania           0.1636682 0.11339274
Serbia            0.1359451 0.02300272
Slovakia          0.1399368 0.04995594
Slovenia          0.1541224 0.06998306
Spain             0.1779972 0.54245429
Sweden            0.1883508 0.08880944
Türkiye           0.1628406 0.11681930
United Kingdom    0.2152683 0.19656926
```

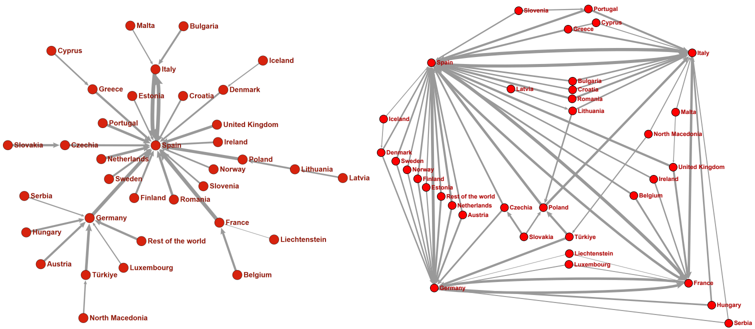
Skeletons

To get insight into the structure of a large (or/and) dense network we can reduce it to its skeleton by removing less important links and/or nodes [Batagelj (2011)].

- Most often the spanning tree, link cut, or node cut are used.
- In the closest k-neighbor skeleton for each node only the largest k incident links are preserved. Invariant for monotonic transformations.
- The Pathfinder algorithm was proposed in the 1980s by Schvaneveldt [Schvaneveldt etal.(1988), Schvaneveldt(1990), Batagelj etal.(2014)]. It removes from the network with a dissimilarity weight all links that do not satisfy the triangle inequality – if a shorter path exists that connects the link's end nodes then the link is removed.
- Cores are a very efficient tool to determine the most cohesive (active) subnetworks [1]. The subset of nodes $\mathbf{C} \subseteq \mathcal{V}$ induces a P_s core at level t if for all $v \in \mathbf{C}$ it holds $\mathrm{wdeg}_{\mathbf{C}}(v) \geq t$, and $\mathbf{C}$ is the maximum such subset.

1-neighbors and 2-neighbors

. . . first and second choice

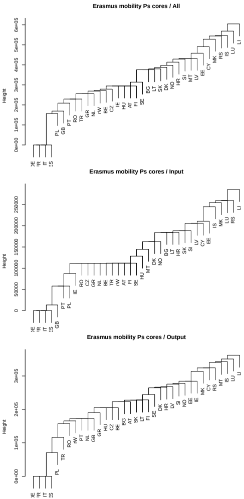


Observations

1. The 1-neighbors skeleton highlights Spain as the main attractor in the network.
2. The 2-neighbors skeleton shows the dominant role of Spain, Germany, France, and Italy.
3. These observations are confirmed by the Ps cores approach.

P_s-cores

Rank	All Id	All Value	Input Id	Input Value	Output Id	Output Value
1	DE	609063	DE	287693	DE	364594
2	FR	609063	FR	287693	FR	364594
3	IT	609063	IT	287693	IT	364594
4	ES	609063	ES	287693	ES	364594
5	PL	452314	GB	274340	PL	294156
6	GB	439822	PT	229822	TR	248328
7	PT	400014	PL	229822	RO	207249
8	RO	379701	IE	200266	rW	198970
9	TR	379701	RO	176038	PT	191225
10	GR	353090	CZ	176038	NL	191225
11	NL	353090	GR	176038	GB	191225
12	rW	339887	NL	176038	GR	174407
13	BE	336319	BE	176038	HU	159516
14	CZ	330134	TR	176038	CZ	159516
15	IE	314423	rW	175804	BE	159516
16	HU	314423	AT	175804	BG	141731
17	AT	314423	FI	175804	AT	141526
18	FI	314423	SE	175804	SK	136878
19	SE	295197	HU	159244	LT	136878
20	BG	233448	MT	143246	FI	136050
21	LT	233448	DK	125031	SE	120105
22	SK	229052	NO	125031	DK	100006
23	DK	221538	BG	103421	HR	98028
24	NO	211331	LT	103421	LV	96748
25	HR	195283	HR	103421	SI	88877
26	SI	179996	SK	99455	NO	86535
27	MT	176232	SI	99187	EE	80157
28	LV	176232	LV	81938	IE	80157
29	EE	150575	CY	81600	MK	50478
30	CY	118367	EE	76830	CY	40446
31	MK	80685	IS	42888	RS	40232
32	RS	64736	MK	33208	MT	24158
33	IS	62144	LU	28600	IS	21770
34	LU	40258	RS	27942	LU	13761
35	LI	4358	LI	2216	LI	2412



P_s-cores code

```
> Rnet <- "https://raw.githubusercontent.com/bavla/Rnet/master/R/''
> source(paste0(Rnet,"ClusNet.R")
> Ps <- read.csv("PsCores.csv",strip.white=TRUE)
> ca <- coreDendro(Ps$aId,Ps$aValue)
> plot(ca,hang=-1,main="Erasmus mobility Ps cores / All")
> ci <- coreDendro(Ps$iId,Ps$iValue)
> plot(ci,hang=-1,main="Erasmus mobility Ps cores / Input")
> co <- coreDendro(Ps$oId,Ps$oValue)
> plot(co,hang=-1,main="Erasmus mobility Ps cores / Output")
> plot(ci,main="Erasmus mobility Ps cores / Input")
> hi <- ci$height; ci$height <- max(Ps$iValue)-ci$value
> plot(ci,main="Erasmus mobility Ps cores / Input")
> ha <- ca$height; ca$height <- max(Ps$aValue)-ca$value
> plot(ca,main="Erasmus mobility Ps cores / All")
> ho <- co$height; co$height <- max(Ps$oValue)-co$value
> plot(co,main="Erasmus mobility Ps cores / Output")
```

Pathfinder

The Pathfinder procedure was proposed in the 1980s by Schvaneveldt [Schvaneveldt etal.(1988), Schvaneveldt(1990), Vavpetič etal.(2009)] for simplifying weighted networks, where the weight measures a dissimilarity between nodes.

It is based on Minkowski operation $a \boxed{r} b = \sqrt[r]{a^r + b^r}$. For $r = 1$, $r = 2$, and $r = \infty$ we get $a \boxed{1} b = a + b$, $a \boxed{2} b = \sqrt{a^2 + b^2}$, and $a \boxed{\infty} b = \max(a, b)$.

For a path $\pi = (v_1, v_2, \ldots, v_k)$ of length k we define its weight $w(\pi) = w(v_1, v_2) \boxed{r} w(v_2, v_3) \boxed{r} \ldots \boxed{r} w(v_{k-1}, v_k)$.

The Pathfinder procedure removes from a given network $\mathcal{N}$ every link (u, v) with its weight larger than the minimum weight of all u-v paths of length at most q. The resulting simplified network is denoted $\mathrm{PFnet}(\mathcal{N}, r, q)$.

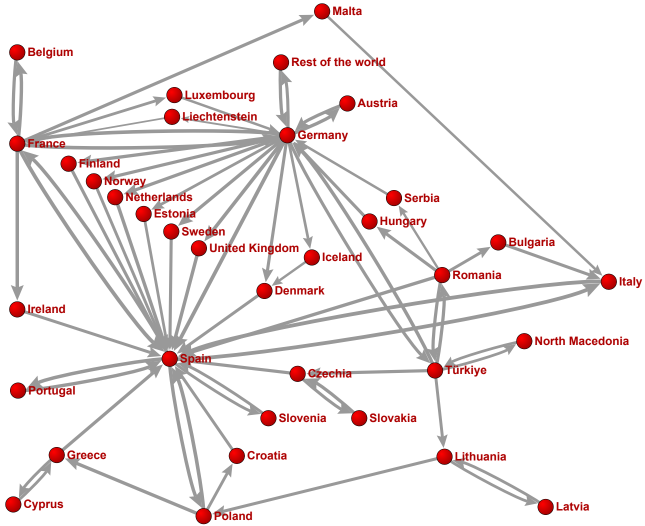
Pathfinder

The Erasmus network weight w (number of visits) is a similarity measure. The Pathfinder procedure requires a dissimilarity measure d. A similarity w can be converted into a dissimilarity d in different ways. For example, $d_1 = w_{\max} - w$ or $d_2 = w_{\max}/w$. We will use the second option.

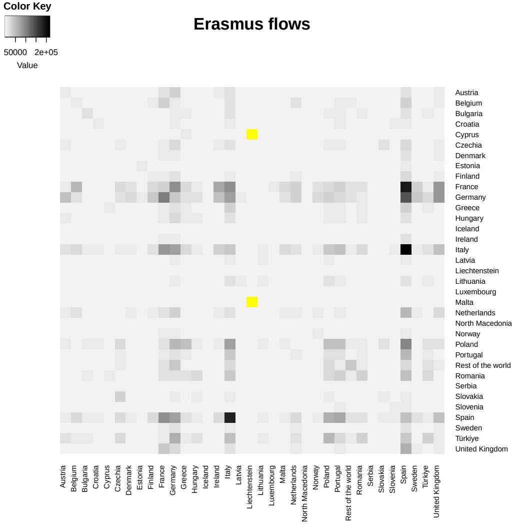
```
read ErasmusFlow.net in Pajek
Network/Create new network/Transform/Line values/Ln
select ErasmusFlow.net as the First network
Network/Create new network/Transform/Line values/Power [-1]
Network/Create new network/Transform/Line values/Multiply by [217003]
Network/Create new network/Transform/Reduction/Pathfinder* [0]
select Ln as the Second network
Networks/Cross intersection/Second
draw network
```

Erasmus network Pathfinder skeleton shows Spain and Germany as central countries followed by France, Italy, Romania, Turkiye, and Poland.

Erasmus network Pathfinder skeleton



Matrix representation of Erasmus network



Matrix based (dis)similarities

For clustering units (nodes) we need a dissimilarity matrix D. In a square weight matrix, its weights can be sometimes considered (or transformed into) a dissimilarity.

$$D[u,v] = f(w[u,v], w[v,u]), \quad f(x,y) = f(y,x)$$

Often we use rows (and columns) as node descriptions and apply a selected dissimilarity on them

$$D[u,v] = d(w[u,.], w[v,.])$$

Typical dissimilarities are the *Euclidean distance*

$$d_e(\mathbf{x},\mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^2}$$

and the *Salton* or *cosine index*

$$S(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x} \bullet \mathbf{y}}{\sqrt{\mathbf{x}^2 \cdot \mathbf{y}^2}},\ d_s(\mathbf{x},\mathbf{y}) = 1 - S(\mathbf{x},\mathbf{y}) \text{ or } d_a(\mathbf{x},\mathbf{y}) = \frac{\arccos S(\mathbf{x},\mathbf{y})}{\pi}$$

where $\mathbf{x} \bullet \mathbf{y} = \sum_i x_i \cdot y_i$ and $\mathbf{x}^2 = \mathbf{x} \bullet \mathbf{x}$.

Corrected (dis)similarities

$$w[u, \cdot] = [w[u,1], \ldots, w[u,i], \ldots, w[u,u], \ldots, w[u,v], \ldots, w[u,k]]$$

$$w[v, \cdot] = [w[v,1], \ldots, w[v,i], \ldots, w[v,u], \ldots, w[v,v], \ldots, w[v,k]]$$

In traditional (dis)similarities, comparing $w[u,i]$ and $w[v,i]$ we are comparing how u relates to i with how v relates to i. The problem arises for $i = u$ and $i = v$. We would need to compare $w[u,u]$ with $w[v,v]$ and $w[u,v]$ with $w[v,u]$. This leads to *corrected* (dis)similarities.

Corrected Euclidean distance and Salton index

Corrected Euclidean distance

$$d'_e(u,v) = \sqrt{(w[u,v]-w[v,u])^2 + (w[u,u]-w[v,v])^2 + \sum_{t \notin \{u,v\}} (w[u,t]-w[v,t])^2}$$

Corrected Salton index of the link $(u,v) \in \mathcal{L}$

$$S'(u,v) = \frac{w[u,.] \bullet w[v,.] + (w[u,u]-w[u,v]) \cdot (w[v,v]-w[v,u])}{\sqrt{w[u,.]^2 \cdot w[v,.]^2}}$$

It has the following properties

1. $S'(u,v) \in [-1,1]$
2. $S'(u,v) = S'(v,u)$
3. $S'(u,u) = 1$
4. $w : L \to \mathbb{R}_0^+ \Rightarrow S'(u,v) \in [0,1]$
5. $S'(\alpha u, \beta v) = S'(u,v), \quad \alpha,\beta > 0$
6. $S'(\alpha u, u) = 1, \quad \alpha > 0$

Normalizations

activity or Balassa index

In networks with weights with a large range usually a few strong nodes prevail. To diminish or neutralize the influence of size on results different normalizations were proposed and used [Batagelj and Mrvar(2003), Matveeva etal.(2023)].

Let $T = \sum_{e \in \mathcal{L}} w(e)$ and for $(u, v) \in \mathcal{L}$ (Balassa index)

$$A(u, v) = \frac{w[u, v] \cdot T}{\mathrm{woutdeg}(u) \cdot \mathrm{windeg}(v)}$$

then the *activity normalization* w'

$$w'(u, v) = \log_2 A(u, v)$$

ClusNet.R functions

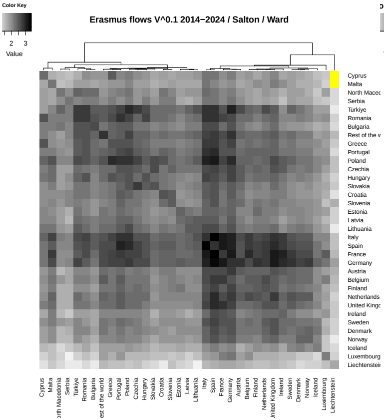
Balassa, Salton, Euclid

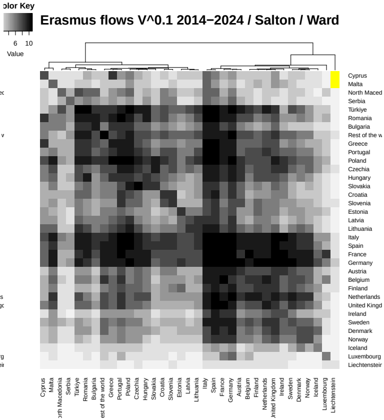
```
Balassa <- function(P){
   R <- rowSums(P); C <- rowSums(t(P)); T <- sum(R); Z <- P
   for(u in 1:nrow(P)) for(v in 1:ncol(P)) Z[u,v] <- P[u,v]*T/R[u]/C[v]
   Z <- log2(Z); Z[Z == -Inf] <- 0; return(Z)
}

CorSalton <- function(W){
   S <- W; diag(S) <- 1; n = nrow(S)
   for(u in 1:(n-1)) for(v in (u+1):n) S[v,u] <- S[u,v] <-
      (as.vector(W[u,]%*%W[v,])+(W[u,u]-W[v,u])*(W[v,v]-W[u,v]))/
      sqrt(as.vector(W[u,]%*%W[u,])*as.vector(W[v,]%*%W[v,]))
   return(S)
}

CorEuclid <- function(W){
   D <- W; diag(D) <- 0; n = nrow(D)
   for(u in 1:(n-1)) for(v in (u+1):n) D[v,u] <- D[u,v] <-
      sqrt(sum((W[u,]-W[v,])**2) + 2*(W[u,u]-W[u,v])*(W[v,u]-W[v,v]))
   return(D)
}
```

ClusNet.R

Erasmus mobilty flow matrix

Salton clustering





Observations

1. The "cross" formed by $C_1 =$ (Italy, Spain, France, Germany) – strong activity with almost all countries in both directions.
2. Intense diagonal "squares" – clusters: $C_2 =$ (Türkiye, Romania, Bulgaria, Rest of the world, Greece, Portugal, Poland, Czechia, Hungary), $C_3 =$ (Poland, Czechia, Hungary, Slovakia), $C_4 =$ (Croatia, Slovenia), $C_5 =$ (Estonia, Latvia, Lithuania), $C_6 =$ (Austria, Belgium, Finland, Netherlands, United Kingdom, Ireland, Sweden, Denmark, Norway), $C_1 \cup C_6$, $C_7 =$ (Sweden, Denmark, Norway, Iceland)
3. Out-diagonal "rectangles": Luxembourg $\times$ (France, Germany), Greece $\times$ Cyprus, (Croatia, Slovenia) $\times$ (North Macedonia, Serbia), etc.
4. In the cross, $C_4 \cup C_5 \cup$ Slovakia less often select France, etc.

Salton clustering

```
> library(gplots); source("ClusNet.R")
> n <- nrow(P); Z <- Co <- P; Co[P == 0] <- NA
> for(u in 1:n) for(v in 1:n) Z[u,v] <- P[u,v]**0.1
> t <- hclust(1-as.dist((CorSalton(Z)+CorSalton(t(Z)))/2),method="ward.D
> t$merge <- flip(18,flip(7,flip(20,flip(30,flip(33,t$merge)))))
> t$merge <- flip(24,flip(28,flip(19,flip(4,t$merge))))
> b <- rep(0,11); b[11] <- max(V)
> for(i in 1:10) b[i] <- V[round((-1+2*i)*length(V)/20)]
> for(i in 1:n) for(j in 1:n){k <- 1;
+    while(P[i,j]>b[k]) k <- k+1; Co[i,j] <- k}
> heatmap.2(Co,Rowv=as.dendrogram(t),Colv="Rowv",
+    dendrogram="column",scale="none",revC=TRUE,
+    margins=c(8,8),cexRow=0.8,cexCol=0.8,
+    col=colorpanel(30,low="grey95",high="black"),na.color="yellow",
+    trace="none", density.info="none", keysize=0.8, symkey=FALSE,
+    main=paste("Erasmus flows V^0.1 2014-2024 / Salton / Ward",sep=""))
> Z <- Balassa(P)
> t <- hclust(as.dist((CorEu(Z)+CorEu(t(Z)))/2),method="ward.D")
> pdf(file="EF14bala.pdf",width=30,height=30)
> par(cex.main=3); Z[Z == 0] <- NA
> heatmap.2(Z,Rowv=as.dendrogram(t),Colv="Rowv",dendrogram="column",
+    scale="none",revC=TRUE,col=bluered(100),na.color="yellow",
+    trace="none", density.info="none", keysize=0.8, symkey=FALSE,
+    margins=c(15,15),cexRow=2,cexCol=2,
+    main=paste("Erasmus flows 2014-2024 / Balassa / Ward",sep=""))
> dev.off()
```

ClusNet.R functions

reorder clustering

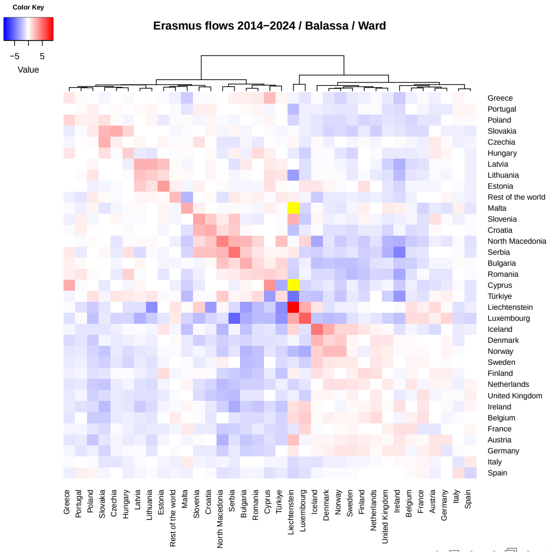
```
toFather <- function(tm){
  n <- nrow(tm); T <- rep(0,2*n+1)
  for(i in 1:n){
    for(j in 1:2){
      p <- tm[i,j]
      if(p<0) T[-p] <- i+n+1 else T[n+1+p] <- i+n+1
    }
  }
  return(T)
}

minCl <- function(u,v,T){
  if(min(u,v)==0) return(T[max(u,v)])
  # cat(u," ",v,":",T[u]," ",T[v],"\n")
  if(u==v) return(u)
  return( if(T[u]<T[v]) minCl(T[u],v,T) else minCl(u,T[v],T) )
}

flip <- function(k,T) {t <- T[k,1]; T[k,1] <- T[k,2];
  T[k,2] <- t; return(T)}

clOrder <- function(M,k) if(k<0) return(-k) else
  return(c(clOrder(M,M[k,1]),clOrder(M,M[k,2])))
```

Erasmus mobility flow clustering. . . reordering


```
> hm <- function(){
+  heatmap.2(Z,Rowv=as.dendrogram(t),Colv="Rowv",
+   dendrogram="column",scale="none",revC=TRUE,
+   margins=c(8,8),cexRow=0.8,cexCol=0.8,
+   col=colorpanel(30,low="grey95",high="black"),na.color="yellow",
+   trace="none", density.info="none", keysize=0.8, symkey=FALSE,
+   main=paste("Erasmus flows V^0.1 2014-2024 / Salton / Ward",sep=""))
+ }
> for(u in 1:n) for(v in 1:n) Z[u,v] <- P[u,v]**0.1
> t <- hclust(1-as.dist((CorSalton(Z)+CorSalton(t(Z)))/2),method="ward.D
> Z[P == 0] <- NA
> s <- t; hm()
> F <- toFather(t$merge); N <- rownames(Z); n <- length(N)
> minCl(which(N=="Italy"),which(N=="Liechtenstein"),F) - n
[1] 33
> t$merge <- flip(33,t$merge); hm()
> minCl(which(N=="Italy"),which(N=="Luxembourg"),F) - n
[1] 30
> t$merge <- flip(30,t$merge); hm()
> minCl(which(N=="Iceland"),which(N=="Sweden"),F) - n
[1] 20
> t$merge <- flip(20,t$merge); hm()
  . . . . .
> t <- s
> t$merge <- flip(18,flip(7,flip(20,flip(30,flip(33,t$merge)))))
> t$merge <- flip(24,flip(28,flip(19,flip(4,t$merge)))); hm()
```

Erasmus mobility flow matrix

Balassa clustering



Observations

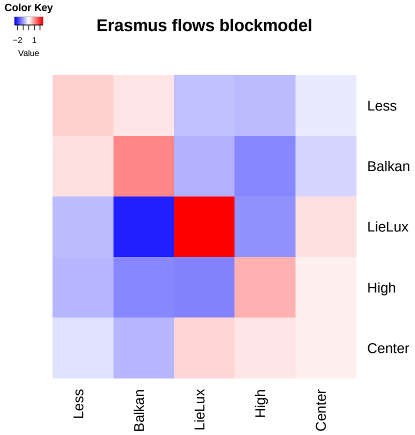
1. Three main clusters $B_l =$ (Greece : Türkiye) – less developed, $B_h =$ (Iceland : Spain) – high developed. and $B_L =$(Liechtenstein, Luxembourg). Most cells inside squares are red and out-diagonal rectangles are mostly blue – exchange between countries from the same cluster is above expected, and below expected between different clusters.
2. Red diagonal "squares" – clusters: $B_1 =$ (Slovakia, Czechia, Hungary), $B_2 =$(Latvia, Lithuania, Estonia), $B_3 =$ (Slovenia, Croatia, North Macedonia, Serbia), $B_4 =$ (North Macedonia, Serbia, Bolgaria, Romania), B_L. The exchange between Cyprus and Türkiye is bellow expected. In the main cluster B_h we can identify a subcluster $B_5 =$ (Iceland, Denmark, Norway, Sweden, Finland, Netherlands, United Kingdom). Within the clusters B_1, B_2, B_3, B_4, B_L, and (Iceland, Denmark, Norway) visits are much above expected.

Observations

3. Countries from the cluster B_I are selecting Malte bellow expected. The exchange between B_5 and B_L is bellow expected.
4. Exchange between Cyprus and Greece is above expected.
5. Exchange of Italy, Spain and Estonia with other countries is mostly close to as expected.

Erasmus mobilty flow blockmodel

Balassa clustering



1. **Less:** Greece, Portugal, Poland, Slovakia, Czechia, Hungary, Latvia, Lithuania, Estonia, Rest of the world, Malta,
2. **Balkan:** Slovenia, Croatia, North Macedonia, Serbia, Bulgaria, Romania, Cyprus, Türkiye,
3. **LieLux:** Liechtenstein, Luxembourg,
4. **High:** Iceland, Denmark, Norway, Sweden, Finland, Netherlands, United Kingdom,
5. **Center:** Ireland, Belgium, France, Austria, Germany, Italy, Spain.

Blockmodeling

```
> BlockModel <- function(P,q,p,lab){
+   m <- length(table(p))
+   B <- matrix(0,nrow=m,ncol=m); rownames(B) <- colnames(B) <- lab
+   for(i in 1:m){I <- q[p==i]
+     for(j in 1:m){J <- q[p==j]; B[i,j] <- sum(P[I,J])} }
+   return(B)
+ }
> q <- clOrder(t$merge,34)
> p <- c(rep(1,11),rep(2,8),rep(3,2),rep(4,7),rep(5,7))
> lab <- c("Less","Balkan","LieLux","High","Center")
> B <- BlockModel(P,q,p,lab)
> BB <- Balassa(B)
> BC <- apply(BB,1:2,function(x) min(2.5,max(-2.5,x)))
> heatmap.2(BC,Rowv=FALSE,Colv="Rowv",
+    dendrogram="none",scale="none",revC=FALSE,
+    margins=c(8,8),cexRow=1.5,cexCol=1.5,
+    col=bluered(100),na.color="yellow",
+    trace="none", density.info="none", keysize=0.8, symkey=FALSE,
+    main=paste("Erasmus flows blockmodel",sep=""))
> t$labels[q[p==1]]
[1] "Greece"         "Portugal"          "Poland"          "Slovakia"
[5] "Czechia"        "Hungary"           "Latvia"          "Lithuania"
[9] "Estonia"        "Rest of the world" "Malta"
> t$labels[q[p==2]]
[1] "Slovenia"         "Croatia"          "North Macedonia" "Serbia"
[6] "Romania"          "Cyprus"           "Türkiye"
> t$labels[q[p==3]]
```

Conclusions

1. Why is Spain the most attractive country?
2. How can the blue between less and high developed countries be reduced?
3. This is exploratory network analysis. Collect and use additional data (neighbors relation, population size, GDP, etc.).
4. Temporal version of the network.

Acknowledgments

The computational work reported in this presentation was performed using R and Pajek. The code and data are available at GitHub/Vlado.

This work is supported in part by the Slovenian Research Agency (research program P1-0294 and research projects J5-2557, J1-2481, and J5-4596), and prepared within the framework of the COST action CA21163 (HiTEc).

References I

- Batagelj, V, Doreian, P, Ferligoj, A, Kejžar, N (2014). Understanding Large Temporal Networks and Spatial Networks: Exploration, Pattern Searching, Visualization and Network Evolution. Wiley. ISBN: 978-0-470-71452-2
- Batagelj, V (2011). Large-Scale Network Analysis. in John Scott, Peter J. Carrington eds. The SAGE Handbook of Social Network Analysis SAGE Publications.
- Batagelj, V, & Mrvar, A (2003). Density based approaches to network analysis; Analysis of Reuters terror news network. Workshop on link analysis for detecting complex behavior (LinkKDD2003). Retrieved August 27, 2003, from http://www.cs.cmu.edu/dunja/LinkKDD2003/papers/Batagelj.pdf
- Batagelj, V, Zaveršnik, M (2011) Fast algorithms for determining (generalized) core groups in social networks. Advances in Data Analysis and Classification, Volume 5, Number 2, 129–145.
- Matveeva, N, Batagelj, V, Ferligoj, A (2023). Scientific collaboration of post-Soviet countries: the effects of different network normalizations Scientometrics, Volume 128, issue 8, Pages: 4219 – 4242

References II

- Schvaneveldt, RW (Editor) (1990). Pathfinder associative networks: Studies in knowledge organization. Norwood, NJ: Ablex. book
- Schvaneveldt, RW, Dearholt, DW, Durso, FT (1988). Graph theoretic foundations of Pathfinder networks. Computers and Mathematics with Applications, 15, 337-345.
- Vavpetič, A, Batagelj, V, Podpečan, V (2009). "An implementation of the Pathfinder algorithm for sparse networks and its application on text network." In 12th International Multiconference Information Society, vol. A, pp. 236-239.