Stochastic Self-Organizing Map variants with the R package **SOMbrero**

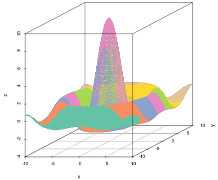
Nathalie Villa-Vialaneix

nathalie.villa@toulouse.inra.fr
`http://www.nathalievilla.org`

WSOM 2017
Nancy, June 29th 2017

Outline

1. SOMbrero: an R package for stochastic SOM
2. KORRESP
3. Dissimilarity data
4. Applications



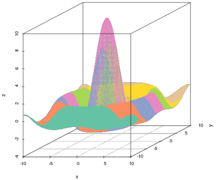
Outline

1 SOMbrero: an R package for stochastic SOM

2 KORRESP

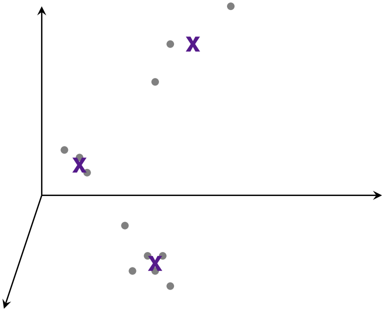
3 Dissimilarity data

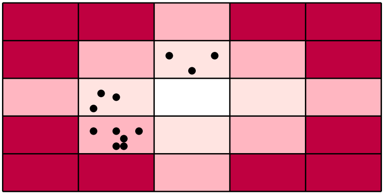
4 Applications



Basics on stochastic SOM

[Kohonen, 2001]

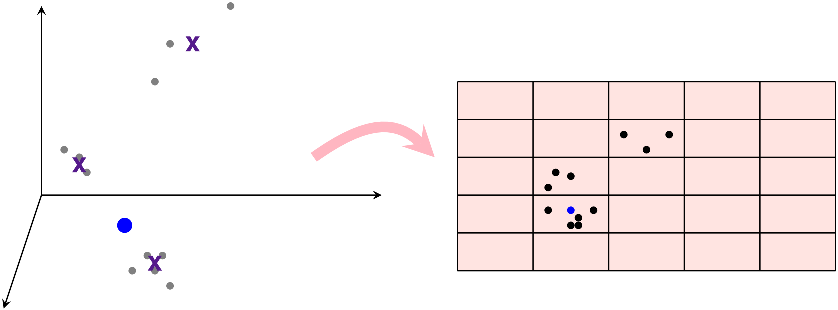




- $(\mathbf{x}_i)_{i=1,\ldots,n} \subset \mathbb{R}^d$ are affected to a unit $f(x_i) \in \{1, \ldots, U\}$
- the grid is equipped with a "distance" between units: $d(u, u')$ and observations affected to close units are close in $\mathbb{R}^d$
- every unit u corresponds to a prototype, p_u ($\mathbf{x}$) in $\mathbb{R}^d$

Basics on stochastic SOM

[Kohonen, 2001]

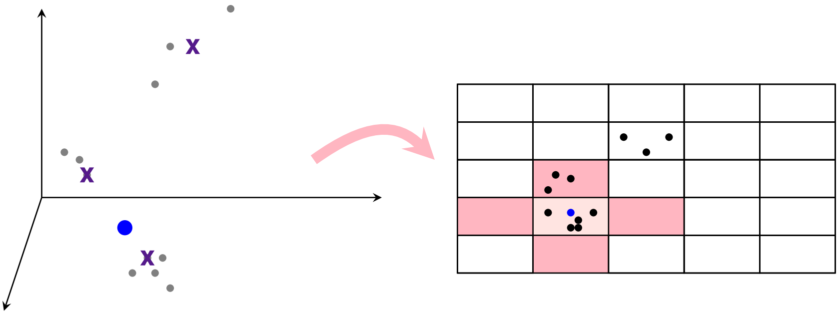


Iterative learning (assignment step): $\mathbf{x}_i$ is picked at random within $(\mathbf{x}_k)_k$ and affected to *best matching unit*:

$$f^t(x_i) = \arg\min_u \|\mathbf{x}_i - p_u^t\|^2$$

Basics on stochastic SOM

[Kohonen, 2001]



Iterative learning (representation step): all prototypes in neighboring units are updated with a gradient descent like step:

$$p_u^{t+1} \longleftarrow p_u^t + \mu(t) H^t(d(f(\mathbf{x}_i), u))(\mathbf{x}_i - p_u^t)$$

Implementations of SOM algorithm

- Matlab: SOM Toolbox [Kohonen, 2014]

Implementations of SOM algorithm

- Matlab: SOM Toolbox [Kohonen, 2014]

- R (CRAN only)
  - **class**: batch training, crude implementation
  - **som** (2016): training (two-step batch), basic plots and quality criteria
  - **popsom** (2017): vectorized stochastic learning (fortran), stochastic learning (C++), batch (C), several plots and quality criteria
  - **kohonen** (2017): various variants of SOM (including supervised) and super-clustering

SOMbrero

[Boelaert et al., 2014]

- **SOMbrero** is an R package implementing stochastic variants of SOM for non vectorial data

Specifically well adapted to...

non expert use and teaching

- first release: March 2013; latest release: Sept. 2016 (version 1.2)
- depends on R (version $\geq$ 3.1.0) http://www.r-project.org and on several packages available on CRAN:

  wordcloud, **igraph**, **RColorBrewer**, **scatterplot3d**, **knitr**, **shiny**
- available at https://cran.r-project.org/package=SOMbrero (licence GPL) and can be installed from inside R using

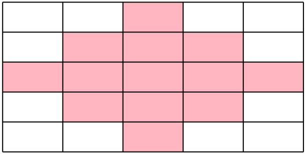
```
install.packages("SOMbrero")
```

Training

```
mysom <- trainSOM(iris[ ,1:4], ...)
```

Options to train the SOM:

- grid: square grid, with arbitrary width and length
  - *distance between units*: standard distances as in `dist` or `"letremy"` (Euclidean then `"maximum"`)
  - *neighborhood relationship*: Gaussian or `"letremy"`



- prototypes: initialized randomly, with a PCA, with random observations from the training sample
- preprocessing: centering, scaling to unit variance or nothing
- training: number of iterations, standard or Heskes's assignment step

$$f^t(\mathbf{x}_i) \leftarrow \arg\min_{u=1,\ldots,U} \sum_{u'=1}^{U} H^t(d(u,u'))\|\mathbf{x}_i - p_{u'}^{t-1}\|^2$$

Diagnostic tools

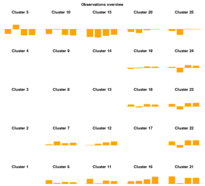
```
quality(mysom)
```

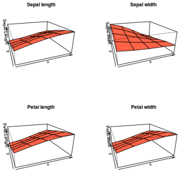
- *topographic error*: average frequency (over the samples) for which the prototypes that comes closest is in the direct neighborhood on the grid of the BMU
- *quantization error*

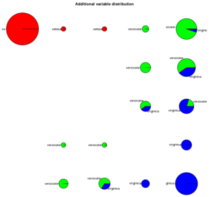
$$\mathcal{Q} = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}_i - p_{f(\mathbf{x}_i)}\|^2$$

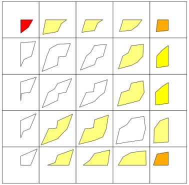
Plots...

```r
plot(mysom,
     what = c("observations", "prototypes", "add"),
     type = ..., ...)
```



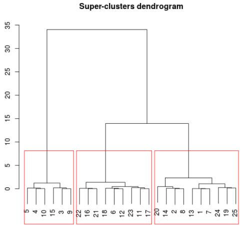


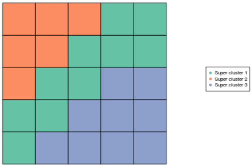


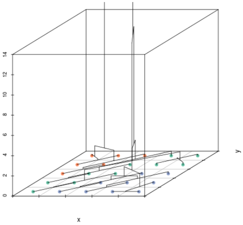


Super-clustering

```r
mysom.sc <- superClass(mysom)
```





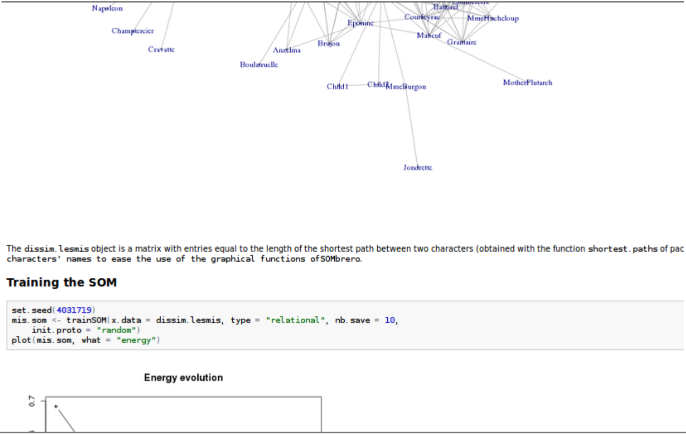


Start with SOMbrero

- 3 datasets corresponding to the three types of data that **SOMbrero** can handle (`iris`, `presidentielles2002` and `lesmis`, a graph from “Les Misérables”)

Start with SOMbrero

- 3 datasets corresponding to the three types of data that **SOMbrero** can handle (`iris`, `presidentielles2002` and `lesmis`, a graph from "Les Misérables")
- comprehensive (HTML) vignettes included in the package and available on the website



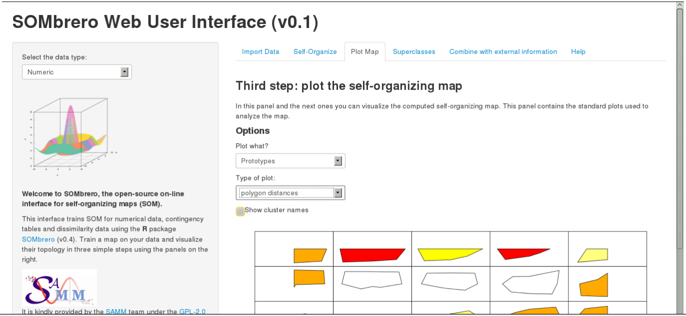
The `dissim.lesmis` object is a matrix with entries equal to the length of the shortest path between two characters (obtained with the function `shortest.paths` of pac characters' names to ease the use of the graphical functions of SOMbrero.

Training the SOM

```
set.seed(4031719)
mis.som <- trainSOM(x.data = dissim.lesmis, type = "relational", nb.save = 10,
    init.proto = "random")
plot(mis.som, what = "energy")
```

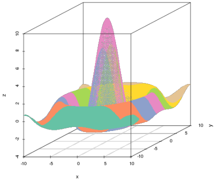
Start with SOMbrero

- 3 datasets corresponding to the three types of data that **SOMbrero** can handle (`iris`, `presidentielles2002` and `lesmis`, a graph from "Les Misérables")
- comprehensive (HTML) vignettes included in the package and available on the website
- Web User Interface (made with **shiny**) for using the package even if you do not know R programming language (included in the package with `sombreroGUI()` Tested and approved on an historian!



Outline

1. SOMbrero: an R package for stochastic SOM
2. KORRESP
3. Dissimilarity data
4. Applications



Extensions to non vectorial data 1

KORRESP [Cottrell and Letrémy, 2005]

Data: contingency table $\mathbf{T} = (n_{ij})_{ij}$ with p rows and q columns transformed into a numeric dataset $\mathbf{X}$:

$$\mathbf{X} = \begin{array}{cc|l} \text{columns} & \text{rows} & \\ \hline \textit{row profile} & & \text{rows} \\ & \textit{column profile} & \text{columns} \end{array}$$

with

- $\forall\, i = 1, \ldots, p$ and $\forall\, j = 1, \ldots, q$, $\mathbf{x}_{ij} = \frac{n_{ij}}{n_{i.}} \times \sqrt{\frac{n}{n_{.j}}}$

Extensions to non vectorial data 1

KORRESP [Cottrell and Letrémy, 2005]

Data: contingency table $\mathbf{T} = (n_{ij})_{ij}$ with p rows and q columns transformed into a numeric dataset $\mathbf{X}$:

$$\mathbf{X} = \begin{array}{cc} \text{columns} & \text{rows} & \\ \left[\begin{array}{cc} \quad & \textit{augmented row profile} \\ \textit{augmented column profile} & \quad \end{array}\right] & & \begin{array}{l} \text{rows} \\ \text{columns} \end{array} \end{array}$$

with

- $\forall\, i = 1, \ldots, p$ and $\forall\, j = q+1, \ldots, q+p$, $\mathbf{x}_{ij} = \mathbf{x}_{k(i)+p,j}$ with $k(i) = \arg\max_{k=1,\ldots,q} \mathbf{x}_{ik}$

Extensions to non vectorial data 1

KORRESP [Cottrell and Letrémy, 2005]

Data: contingency table $\mathbf{T} = (n_{ij})_{ij}$ with p rows and q columns transformed into a numeric dataset $\mathbf{X}$:

$$\mathbf{X} = \begin{array}{cc|l} \text{columns} & \text{rows} & \\ \hline \textit{row profile} & \textit{augmented row profile} & \text{rows} \\ \textit{augmented column profile} & \textit{column profile} & \text{columns} \end{array}$$

- assignment uses reduced profile
- representation uses augmented profile
- alternatively process row profiles and column profiles

Also available in **SOMbrero**

```r
mysom <- trainSOM(presidentielles2002, type = "korresp")
plot(mysom, what = "obs", type = "names")
```

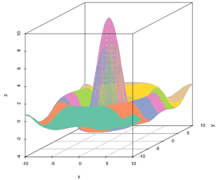
Outline

1 SOMbrero: an R package for stochastic SOM

2 KORRESP

3 Dissimilarity data

4 Applications



What are dissimilarity data?

Sometimes, $(\mathbf{x}_i)_i \subset \mathcal{X}$, $\mathcal{X} \neq \mathbb{R}^d$...

What are dissimilarity data?

Sometimes, $(\mathbf{x}_i)_i \subset \mathcal{X}$, $\mathcal{X} \neq \mathbb{R}^d$...

In this case, data can frequently be described by a pairwise relationship:

a kernel (similarity measure) $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$...

- *symmetric*: $K(z_i, z_j) = K(z_j, z_i)$
- *positive*: $\sum_{i,i'=1}^{N} \alpha_i \alpha_{i'} K(z_i, z_{i'}) \geq 0$

What are dissimilarity data?

Sometimes, $(\mathbf{x}_i)_i \subset \mathcal{X}$, $\mathcal{X} \neq \mathbb{R}^d$...

In this case, data can frequently be described by a pairwise relationship:

a kernel (similarity measure) $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$...

- *symmetric*: $K(z_i, z_j) = K(z_j, z_i)$
- *positive*: $\sum_{i,i'=1}^N \alpha_i \alpha_{i'} K(z_i, z_{i'}) \geq 0$

$\Rightarrow \exists$ Hilbert space $(\mathcal{H}, \langle ., . \rangle)$ and a feature map $\phi : \mathcal{X} \rightarrow \mathcal{H}$:

$$K(z, z') = \langle \phi(z), \phi(z') \rangle$$

What are dissimilarity data?

Sometimes, $(\mathbf{x}_i)_i \subset \mathcal{X}$, $\mathcal{X} \neq \mathbb{R}^d$...

In this case, data can frequently be described by a pairwise relationship:

a dissimilarity $\delta : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+$...

- *symmetric*: $\delta(z_i, z_j) = \delta(z_j, z_i)$
- *positive*: $\delta(z_i, z_j) \geq 0$ and $\delta(z_i, z_i) = 0$

What are dissimilarity data?

Sometimes, $(\mathbf{x}_i)_i \subset \mathcal{X}$, $\mathcal{X} \neq \mathbb{R}^d$...

In this case, data can frequently be described by a pairwise relationship:

a dissimilarity $\delta : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+$...

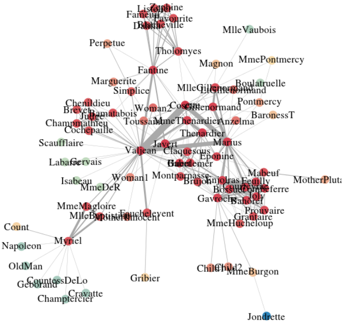
- *symmetric*: $\delta(z_i, z_j) = \delta(z_j, z_i)$
- *positive*: $\delta(z_i, z_j) \geq 0$ and $\delta(z_i, z_i) = 0$

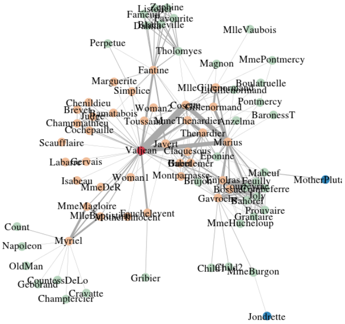
$\Rightarrow \exists$ Euclidean spaces $(\mathcal{E}_1, \langle ., . \rangle_1)$ and $(\mathcal{E}_2, \langle ., . \rangle_2)$ and feature maps $\phi_1 : \mathcal{X} \rightarrow \mathcal{E}_1$, $\phi_2 : \mathcal{X} \rightarrow \mathcal{E}_2$:

$$\delta(z, z') = \langle \phi_1(z), \phi_1(z') \rangle_1 - \langle \phi_2(z), \phi_2(z') \rangle_2$$

Practical examples

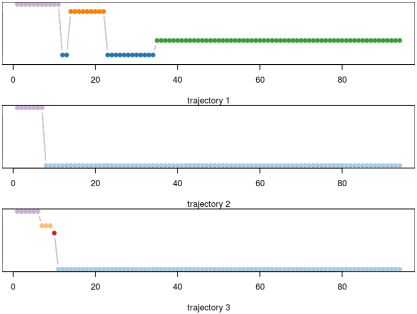
- kernel/dissimilarity for nodes of a graph:
  - based on the Laplacian (heat kernel / commute time kernel...) [Kondor and Lafferty, 2002, Fouss et al., 2007]
  - shortest path length





Practical examples

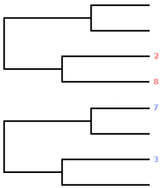
- kernel/dissimilarity for nodes of a graph:
  - based on the Laplacian (heat kernel / commute time kernel...) **[Kondor and Lafferty, 2002, Fouss et al., 2007]**
  - shortest path length
- kernel/dissimilarity for strings
  - χ^2 dissimilarity emphasizes the contemporary identical situations, whether these identical moments are contiguous or not
  - Optimal-matching dissimilarities [Needleman and Wunsch, 1970] (or “edit distance”, “Levenshtein distance”)



trajectory 1 is closer to trajectory 2 than to trajectory 3 according to OM distance and the opposite according to χ^2 distance

Practical examples

- kernel/dissimilarity for nodes of a graph:
  - based on the Laplacian (heat kernel / commute time kernel...) **[Kondor and Lafferty, 2002, Fouss et al., 2007]**
  - shortest path length
- kernel/dissimilarity for strings
  - χ^2 dissimilarity emphasizes the contemporary identical situations, whether these identical moments are contiguous or not
  - Optimal-matching dissimilarities [Needleman and Wunsch, 1970] (or "edit distance", "Levenshtein distance")
- in metagenomics (with phylogeny information)
  - unifrac, weighted unifrac... **[Lozupone et al., 2007]**



Extension of SOM to data described by a kernel or a dissimilarity

[Olteanu and Villa-Vialaneix, 2015a]

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathbb{R}^d$

Initialization:

randomly set p_1^0, ..., p_U^0 in $\mathbb{R}^d$

for $t = 1 \rightarrow T$ **do**

pick at random $i \in \{1, \ldots, n\}$

Assignment

$$f^t(\mathbf{x}_i) = \arg\min_{u=1,\ldots,U} \|\mathbf{x}_i - p_u^t\|^2$$

for all $u = 1 \rightarrow U$ **do** Representation

$$p_u^{t+1} = p_u^t + \mu(t) H^t(d(f^t(\mathbf{x}_i), u)) \left(\mathbf{x}_i - p_u^t\right)$$

end for

end for

Extension of SOM to data described by a kernel or a dissimilarity

[Olteanu and Villa-Vialaneix, 2015a]

Data: $(\mathbf{x}_i)_{i=1,\dots,n} \in \mathcal{X}$

1: Initialization:
    randomly set p_1^0, ..., p_U^0 in $\mathbb{R}^d$
2: **for** $t = 1 \to T$ **do**
3: pick at random $i \in \{1, \dots, n\}$
4: Assignment

$$f^t(\mathbf{x}_i) = \arg\min_{u=1,\dots,U} \|\mathbf{x}_i - p_u^t\|^2$$

5: **for all** $u = 1 \to U$ **do** Representation
6:

$$p_u^{t+1} = p_u^t + \mu(t) H^t(d(f^t(\mathbf{x}_i), u)) \left(\mathbf{x}_i - p_u^t\right)$$

7: **end for**
8: **end for**

Extension of SOM to data described by a kernel or a dissimilarity

[Olteanu and Villa-Vialaneix, 2015a]

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathcal{X}$

Initialization:

$$p_u^0 = \sum_{i=1}^n \beta_{ui}^0 \phi(\mathbf{x}_i) \text{ (convex combination)}$$

for $t = 1 \rightarrow T$ **do**

  pick at random $i \in \{1, \ldots, n\}$

  Assignment

$$f^t(\mathbf{x}_i) = \arg\min_{u=1,\ldots,U} \|\mathbf{x}_i - p_u^t\|^2$$

  for all $u = 1 \rightarrow U$ **do** Representation

$$p_u^{t+1} = p_u^t + \mu(t) H^t(d(f^t(\mathbf{x}_i), u)) \left(\mathbf{x}_i - p_u^t\right)$$

  end for

end for

Extension of SOM to data described by a kernel or a dissimilarity

[Olteanu and Villa-Vialaneix, 2015a]

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathcal{X}$

1: Initialization:
$p_u^0 = \sum_{i=1}^n \beta_{ui}^0 \phi(\mathbf{x}_i)$ (convex combination)
2: **for** $t = 1 \rightarrow T$ **do**
3: pick at random $i \in \{1, \ldots, n\}$
4: Assignment

$$f^t(\mathbf{x}_i) = \arg\min_{u=1,\ldots,U} \|\phi(\mathbf{x}_i) - p_u^t\|^2_{\mathcal{H}}$$

5: **for all** $u = 1 \rightarrow U$ **do** Representation
6:

$$p_u^{t+1} = p_u^t + \mu(t) H^t(d(f^t(\mathbf{x}_i), u)) \left(\mathbf{x}_i - p_u^t\right)$$

7: **end for**
8: **end for**

Extension of SOM to data described by a kernel or a dissimilarity

[Olteanu and Villa-Vialaneix, 2015a]

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathcal{X}$

1: Initialization:
$p_u^0 = \sum_{i=1}^n \beta_{ui}^0 \phi(\mathbf{x}_i)$ (convex combination)

2: **for** $t = 1 \to T$ **do**

3: pick at random $i \in \{1, \ldots, n\}$

4: Assignment

$$f^t(\mathbf{x}_i) = \arg \min_{u=1,\ldots,U} \|\phi(\mathbf{x}_i) - p_u^t\|_{\mathcal{H}}^2$$

5: **for all** $u = 1 \to U$ **do** Representation

6:

$$p_u^{t+1} = p_u^t + \mu(t) H^t(d(f^t(\mathbf{x}_i), u)) \left(\phi(\mathbf{x}_i) - p_u^t\right)$$

7: **end for**

8: **end for**

Extension of SOM to data described by a kernel or a dissimilarity

[Olteanu and Villa-Vialaneix, 2015a]

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathcal{X}$

1: Initialization:

$p_u^0 = \sum_{i=1}^n \beta_{ui}^0 \phi(\mathbf{x}_i)$ (convex combination)

2: **for** $t = 1 \rightarrow T$ **do**

3: pick at random $i \in \{1, \ldots, n\}$

4: Assignment

$$f^t(\mathbf{x}_i) = \arg\min_{u=1,\ldots,U} \sum_{j,j'=1}^n \beta_{uj}^t \beta_{uj'}^t K(\mathbf{x}_j, \mathbf{x}_{j'}) - 2 \sum_{j=1}^n \beta_{uj}^t K(\mathbf{x}_j, \mathbf{x}_i)$$

5: **for all** $u = 1 \rightarrow U$ **do** Representation

6:

$$\beta_u^{t+1} = \beta_u^t + \mu(t) H^t(d(f^t(\mathbf{x}_i), u)) \left(\mathbf{1}_i - \beta_u^t\right)$$

7: **end for**

8: **end for**

Extension of SOM to data described by a kernel or a dissimilarity

[Olteanu and Villa-Vialaneix, 2015a]

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathcal{X}$

Initialization:

$$p_u^0 \sim \sum_{i=1}^n \beta_{ui}^0 \mathbf{x}_i \text{ (convex combination)}$$

for $t = 1 \to T$ **do**

  pick at random $i \in \{1, \ldots, n\}$

  Assignment

$$f^t(\mathbf{x}_i) = \arg \min_{u=1,\ldots,U} \delta(p_u^t, \mathbf{x}_i)$$

  for all $u = 1 \to U$ **do** Representation

$$p_u^{t+1} = p_u^t + \mu(t) H^t(d(f^t(\mathbf{x}_i), u)) \left(\sim \mathbf{x}_i - p_u^t\right)$$

  end for

end for

Extension of SOM to data described by a kernel or a dissimilarity

[Olteanu and Villa-Vialaneix, 2015a]

Data: $(\mathbf{x}_i)_{i=1,\dots,n} \in \mathcal{X}$

1: Initialization:
$p_u^0 \sim \sum_{i=1}^n \beta_{ui}^0 \mathbf{x}_i$ (convex combination)

2: **for** $t = 1 \rightarrow T$ **do**

3: pick at random $i \in \{1, \dots, n\}$

4: Assignment

$$f^t(\mathbf{x}_i) = \arg\min_{u=1,\dots,U} \sum_{j=1}^n \beta_{uj}^t \delta(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{2} \sum_{j,j'=1}^n \beta_{uj}^t \beta_{uj'}^t \delta(\mathbf{x}_j, \mathbf{x}_{j'})$$

5: **for all** $u = 1 \rightarrow U$ **do** Representation

6:

$$\beta_u^{t+1} = \beta_u^t + \mu(t) H^t(d(f^t(\mathbf{x}_i), u)) \left(\mathbf{1}_i - \beta_u^t\right)$$

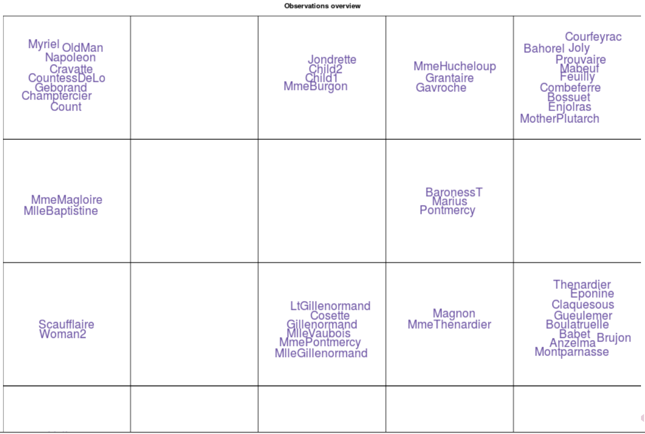
7: **end for**

8: **end for**

Relational SOM is available in **SOMbrero**

Kernel SOM equivalent to RSOM for δ square distance induced by the kernel

```
mysom <- trainSOM(dissim.lesmis, type = "relational")
plot(mysom, what="obs", type="names")
```



Problems with RSOM and KSOM

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathcal{X}$

1: Initialization: $p_u^0 = \sum_{i=1}^n \beta_{ui}^0 \phi(\mathbf{x}_i)$ (convex combination)
2: **for** **do**
3: pick randomly $i \in \{1, \ldots, n\}$
4: Assignment
5: **for all** $u = 1 \to U$ **do** Representation
6:
7: **end for**
8: **end for**

Problems with RSOM and KSOM

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathcal{X}$

Initialization: $p_u^0 = \sum_{i=1}^n \beta_{ui}^0 \phi(\mathbf{x}_i)$ (convex combination)
for $t = 1 \rightarrow T$ **do**
    pick randomly $i \in \{1, \ldots, n\}$
    Assignment

$$f^t(\mathbf{x}_i) = \arg \min_{u=1,\ldots,U} \|\phi(\mathbf{x}_i) - p_u^t\|^2_{\mathcal{H}}$$

    for all $u = 1 \rightarrow U$ **do** Representation

$$p_u^{t+1} = p_u^t + \mu(t)\left(\phi(\mathbf{x}_i) - p_u^t\right)$$

    end for
end for

Problems with RSOM and KSOM

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathcal{X}$

Initialization: $p_u^0 = \sum_{i=1}^n \beta_{ui}^0 \phi(\mathbf{x}_i)$ (convex combination)

for $t = 1 \to T$ **do**

  pick randomly $i \in \{1, \ldots, n\}$

  Assignment

$$f^t(\mathbf{x}_i) = \arg\min_{u=1,\ldots,U} \sum_{j,j'=1}^n \beta_{uj}^t \beta_{uj'}^t K(\mathbf{x}_j, \mathbf{x}_{j'}) - 2 \sum_{j=1}^n \beta_{uj}^t K(\mathbf{x}_j, \mathbf{x}_i)$$

  for all $u = 1 \to U$ **do** Representation

$$\beta_u^{t+1} = \beta_u^t + \mu(t) H^t(d(f^t(\mathbf{x}_i), u))(\mathbf{1}_i - \beta_u^t)$$

  end for

end for

Problems with RSOM and KSOM

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathcal{X}$

1: Initialization: $p_u^0 = \sum_{i=1}^n \beta_{ui}^0 \phi(\mathbf{x}_i)$ (convex combination)

2: **for** $t = 1 \to T$ **do**

3: pick randomly $i \in \{1, \ldots, n\}$

4: Assignment

$$f^t(\mathbf{x}_i) = \arg\min_{u=1,\ldots,U} \sum_{j,j'=1}^n \beta_{uj}^t \beta_{uj'}^t K(\mathbf{x}_j, \mathbf{x}_{j'}) - 2\sum_{j=1}^n \beta_{uj}^t K(\mathbf{x}_j, \mathbf{x}_i) \quad \to \mathcal{O}(n^2U)$$

5: **for all** $u = 1 \to U$ **do** Representation

6:

$$\beta_u^{t+1} = \beta_u^t + \mu(t) H^t(d(f^t(\mathbf{x}_i), u))(\mathbf{1}_i - \beta_u^t) \quad \to \mathcal{O}(nU)$$

7: **end for**

8: **end for**

Problems with RSOM and KSOM

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathcal{X}$

1: Initialization: $p_u^0 = \sum_{i=1}^n \beta_{ui}^0 \phi(\mathbf{x}_i)$ (convex combination)

2: **for** $t = 1 \rightarrow \gamma n$ **do**

3: pick randomly $i \in \{1, \ldots, n\}$

4: Assignment

$$f^t(\mathbf{x}_i) = \arg \min_{u=1,\ldots,U} \sum_{j,j'=1}^n \beta_{uj}^t \beta_{uj'}^t K(\mathbf{x}_j, \mathbf{x}_{j'}) - 2 \sum_{j=1}^n \beta_{uj}^t K(\mathbf{x}_j, \mathbf{x}_i) \quad \rightarrow \mathcal{O}(n^2 U)$$

5: **for all** $u = 1 \rightarrow U$ **do** Representation

6:

$$\beta_u^{t+1} = \beta_u^t + \mu(t) H^t(d(f^t(\mathbf{x}_i), u))(\mathbf{1}_i - \beta_u^t) \quad \rightarrow \mathcal{O}(nU)$$

7: **end for**

8: **end for**

$\rightarrow$ algorithm complexity: $\mathcal{O}(\gamma n^3 U)$ (compared to $\mathcal{O}(\gamma U d n)$ for numeric)

Reducing the stochastic K-SOM complexity

[Mariette et al., 2017]

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathcal{X}$

1: Initialization: $p_u^0 = \sum_{i=1}^n \beta_{ui}\phi(\mathbf{x}_i)$ (convex combination)
2: **for** $t = 1 \rightarrow \gamma n$ **do**
3: pick at random $i \in \{1, \ldots, n\}$
4: Assignment $f^t(\mathbf{x}_i) = \arg\min_{u=1,\ldots,U} \sum_{j,j'=1}^n \beta_{uj}^t \beta_{uj'}^t K(\mathbf{x}_j, \mathbf{x}_{j'}) - 2\sum_{j=1}^n \beta_{uj}^t K(\mathbf{x}_j, \mathbf{x}_i)$
5: **for all** $u = 1 \rightarrow U$ **do** Representation
6: $\beta_u^{t+1} = \beta_u^t + \mu(t) H^t(d(f^t(\mathbf{x}_i), u))(\mathbf{1}_i - \beta_u^t)$
7: **end for**
8: **end for**

Reducing the stochastic K-SOM complexity

[Mariette et al., 2017]

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathcal{X}$

Initialization: $p_u^0 = \sum_{i=1}^n \beta_{ui}\phi(\mathbf{x}_i)$ (convex combination)

for $t = 1 \to \gamma n$ **do**

  pick at random $i \in \{1, \ldots, n\}$

  Assignment $$f^t(\mathbf{x}_i) = \arg\min_{u=1,\ldots,U} \underbrace{\sum_{j,j'=1}^n \beta_{uj}^t \beta_{uj'}^t K(\mathbf{x}_j, \mathbf{x}_{j'})}_{A_u^t} \underbrace{-2\sum_{j=1}^n \beta_{uj}^t K(\mathbf{x}_j, \mathbf{x}_i}_{B_{ui}^t}$$

  for all $u = 1 \to U$ **do** Representation

    $$\beta_u^{t+1} = \beta_u^t + \mu(t) H^t(d(f^t(\mathbf{x}_i), u))(\mathbf{1}_i - \beta_u^t)$$

  end for

end for

Reducing the stochastic K-SOM complexity

[Mariette et al., 2017]

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathcal{X}$

1: Initialization: $p_u^0 = \sum_{i=1}^n \beta_{ui}\phi(\mathbf{x}_i)$ (convex combination)

2: **for** $t = 1 \to \gamma n$ **do**

3: pick at random $i \in \{1, \ldots, n\}$

4: Assignment $f^t(\mathbf{x}_i) = \arg\min_{u=1,\ldots,U} A_u^t - 2B_{ui}^t$

5: **for all** $u = 1 \to U$ **do** Representation

6: $\beta_u^{t+1} = \beta_u^t + \mu(t)H^t(d(f^t(\mathbf{x}_i), u))(\mathbf{1}_i - \beta_u^t)$

7: **end for**

8: **end for**

Reducing the stochastic K-SOM complexity

[Mariette et al., 2017]

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathcal{X}$

1: Initialization: $p_u^0 = \sum_{i=1}^n \beta_{ui}\phi(\mathbf{x}_i)$ (convex combination)

2: **for** $t = 1 \rightarrow \gamma n$ **do**

3: pick at random $i \in \{1,\ldots,n\}$

4: Assignment $f^t(\mathbf{x}_i) = \arg\min_{u=1,\ldots,U} A_u^t - 2B_{ui}^t$

5: **for all** $u = 1 \rightarrow U$ **do** Representation

6: $\beta_u^{t+1} = \beta_u^t + \underbrace{\mu(t)H^t(d(f^t(\mathbf{x}_i),u))}_{\lambda_u(t)}(\mathbf{1}_i - \beta_u^t)$

7: **end for**

8: **end for**

Reducing the stochastic K-SOM complexity

[Mariette et al., 2017]

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathcal{X}$

Initialization: $p_u^0 = \sum_{i=1}^n \beta_{ui}\phi(\mathbf{x}_i)$ (convex combination)
for $t = 1 \to \gamma n$ **do**
    pick at random $i \in \{1, \ldots, n\}$
    Assignment $f^t(\mathbf{x}_i) = \arg\min_{u=1,\ldots,U} A_u^t - 2B_{ui}^t$
    for all $u = 1 \to U$ **do** Representation
        $\beta_u^{t+1} = (1 - \lambda_u(t))\beta_u^t + \lambda_u(t)\mathbf{1}_i$
    end for
end for

Reducing the stochastic K-SOM complexity

[Mariette et al., 2017]

Data: $(\mathbf{x}_i)_{i=1,\dots,n} \in \mathcal{X}$

Initialization: $p_u^0 = \sum_{i=1}^n \beta_{ui}\phi(\mathbf{x}_i)$ (convex combination)

$A_u^0 = \sum_{j,j'=1}^n \beta_{uj}^0 \beta_{uj'}^0 K(\mathbf{x}_j, \mathbf{x}_{j'})$

$B_{ui}^0 = \sum_{j=1}^n \beta_{uj}^0 K(\mathbf{x}_j, \mathbf{x}_i)$

for $t = 1 \rightarrow \gamma n$ **do**

  pick at random $i \in \{1, \dots, n\}$

  Assignment $f^t(\mathbf{x}_i) = \arg\min_{u=1,\dots,U} A_u^t - 2B_{ui}^t$

  for all $u = 1 \rightarrow U$ **do** Representation

    $\beta_u^{t+1} = (1 - \lambda_u(t))\beta_u^t + \lambda_u(t)\mathbf{1}_i$

  end for

end for

Reducing the stochastic K-SOM complexity

[Mariette et al., 2017]

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathcal{X}$

Initialization: $p_u^0 = \sum_{i=1}^n \beta_{ui}\phi(\mathbf{x}_i)$ (convex combination)

$A_u^0 = \sum_{j,j'=1}^n \beta_{uj}^0 \beta_{uj'}^0 K(\mathbf{x}_j, \mathbf{x}_{j'})$

$B_{ui}^0 = \sum_{j=1}^n \beta_{uj}^0 K(\mathbf{x}_j, \mathbf{x}_i)$

for $t = 1 \to \gamma n$ **do**

  pick at random $i \in \{1, \ldots, n\}$

  Assignment $\quad f^t(\mathbf{x}_i) = \arg\min_{u=1,\ldots,U} A_u^t - 2B_{ui}^t$

  for all $u = 1 \to U$ **do** Representation

    $\beta_u^{t+1} = (1 - \lambda_u(t))\beta_u^t + \lambda_u(t)\mathbf{1}_i$

$B_{ui'}^{t+1} = \sum_{j=1}^n \beta_{uj}^{t+1} K(\mathbf{x}_{i'}, \mathbf{x}_j) = (1 - \lambda_u(t))B_{ui'}^t + \lambda_u(t)K(\mathbf{x}_i, \mathbf{x}_{i'})$

  end for

end for

Reducing the stochastic K-SOM complexity

[Mariette et al., 2017]

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathcal{X}$

Initialization: $p_u^0 = \sum_{i=1}^n \beta_{ui}\phi(\mathbf{x}_i)$ (convex combination)
$A_u^0 = \sum_{j,j'=1}^n \beta_{uj}^0 \beta_{uj'}^0 K(\mathbf{x}_j, \mathbf{x}_{j'})$
$B_{ui}^0 = \sum_{j=1}^n \beta_{uj}^0 K(\mathbf{x}_j, \mathbf{x}_i)$
for $t = 1 \to \gamma n$ **do**
    pick at random $i \in \{1, \ldots, n\}$
    Assignment $\quad f^t(\mathbf{x}_i) = \arg\min_{u=1,\ldots,U} A_u^t - 2B_{ui}^t$
    for all $u = 1 \to U$ **do** Representation
        $\beta_u^{t+1} = (1 - \lambda_u(t))\beta_u^t + \lambda_u(t)\mathbf{1}_i$

$$B_{ui'}^{t+1} = \sum_{j=1}^n \beta_{uj}^{t+1} K(\mathbf{x}_{i'}, \mathbf{x}_j) = (1 - \lambda_u(t))B_{ui'}^t + \lambda_u(t)K(\mathbf{x}_i, \mathbf{x}_{i'})$$

$$A_u^{t+1} = \sum_{j,j'=1}^n \beta_{uj}^{t+1}\beta_{uj'}^{t+1} K(\mathbf{x}_j, \mathbf{x}_{j'}) =$$
$$(1 - \lambda_u(t))^2 A_u^t + \lambda_u(t)^2 K(\mathbf{x}_i, \mathbf{x}_i) + 2\lambda_u(t)(1 - \lambda_u(t))B_{ui}^t$$

    end for
end for

Reducing the stochastic K-SOM complexity

[Mariette et al., 2017]

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathcal{X}$

Initialization: $p_u^0 = \sum_{i=1}^n \beta_{ui}\phi(\mathbf{x}_i)$ (convex combination)

$A_u^0 = \sum_{j,j'=1}^n \beta_{uj}^0 \beta_{uj'}^0 K(\mathbf{x}_j, \mathbf{x}_{j'}) \quad \to \mathcal{O}(n^2 U)$

$B_{ui}^0 = \sum_{j=1}^n \beta_{uj}^0 K(\mathbf{x}_j, \mathbf{x}_i) \quad \to \mathcal{O}(nU)$

for $t = 1 \to \gamma n$ **do**

  pick at random $i \in \{1, \ldots, n\}$

  Assignment $\quad f^t(\mathbf{x}_i) = \arg\min_{u=1,\ldots,U} A_u^t - 2B_{ui}^t$

  for all $u = 1 \to U$ **do** Representation

    $\beta_u^{t+1} = (1 - \lambda_u(t))\beta_u^t + \lambda_u(t)\mathbf{1}_i$

$B_{ui'}^{t+1} = \sum_{j=1}^n \beta_{uj}^{t+1} K(\mathbf{x}_{i'}, \mathbf{x}_j) = (1-\lambda_u(t))B_{ui'}^t + \lambda_u(t) K(\mathbf{x}_i, \mathbf{x}_{i'})$

$A_u^{t+1} = \sum_{j,j'=1}^n \beta_{uj}^{t+1}\beta_{uj'}^{t+1} K(\mathbf{x}_j, \mathbf{x}_{j'}) =$

$(1-\lambda_u(t))^2 A_u^t + \lambda_u(t)^2 K(\mathbf{x}_i, \mathbf{x}_i) + 2\lambda_u(t)(1-\lambda_u(t))B_{ui}^t$

  end for

end for

Reducing the stochastic K-SOM complexity

[Mariette et al., 2017]

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathcal{X}$

Initialization: $p_u^0 = \sum_{i=1}^n \beta_{ui}\phi(\mathbf{x}_i)$ (convex combination)

$A_u^0 = \sum_{j,j'=1}^n \beta_{uj}^0 \beta_{uj'}^0 K(\mathbf{x}_j, \mathbf{x}_{j'}) \quad \rightarrow \mathcal{O}(n^2 U)$

$B_{ui}^0 = \sum_{j=1}^n \beta_{uj}^0 K(\mathbf{x}_j, \mathbf{x}_i) \quad \rightarrow \mathcal{O}(nU)$

for $t = 1 \rightarrow \gamma n$ **do**

    pick at random $i \in \{1, \ldots, n\}$

    Assignment $\quad f^t(\mathbf{x}_i) = \arg\min_{u=1,\ldots,U} A_u^t - 2B_{ui}^t \quad \rightarrow$ does not depend on n

    for all $u = 1 \rightarrow U$ **do** Representation

        $\beta_u^{t+1} = (1 - \lambda_u(t))\beta_u^t + \lambda_u(t)\mathbf{1}_i$

$B_{ui'}^{t+1} = \sum_{j=1}^n \beta_{uj}^{t+1} K(\mathbf{x}_{i'}, \mathbf{x}_j) = (1 - \lambda_u(t))B_{ui'}^t + \lambda_u(t) K(\mathbf{x}_i, \mathbf{x}_{i'})$

$A_u^{t+1} = \sum_{j,j'=1}^n \beta_{uj}^{t+1} \beta_{uj'}^{t+1} K(\mathbf{x}_j, \mathbf{x}_{j'}) =$

$(1 - \lambda_u(t))^2 A_u^t + \lambda_u(t)^2 K(\mathbf{x}_i, \mathbf{x}_i) + 2\lambda_u(t)(1 - \lambda_u(t))B_{ui}^t$

    end for

end for

Reducing the stochastic K-SOM complexity

[Mariette et al., 2017]

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathcal{X}$

Initialization: $p_u^0 = \sum_{i=1}^n \beta_{ui}\phi(\mathbf{x}_i)$ (convex combination)

$A_u^0 = \sum_{j,j'=1}^n \beta_{uj}^0 \beta_{uj'}^0 K(\mathbf{x}_j, \mathbf{x}_{j'})$ $\rightarrow \mathcal{O}(n^2 U)$

$B_{ui}^0 = \sum_{j=1}^n \beta_{uj}^0 K(\mathbf{x}_j, \mathbf{x}_i)$ $\rightarrow \mathcal{O}(nU)$

for $t = 1 \rightarrow \gamma n$ **do**

  pick at random $i \in \{1, \ldots, n\}$

  Assignment $f^t(\mathbf{x}_i) = \arg\min_{u=1,\ldots,U} A_u^t - 2B_{ui}^t$ $\rightarrow$ does not depend on n

  for all $u = 1 \rightarrow U$ **do** Representation

    $\beta_u^{t+1} = (1 - \lambda_u(t))\beta_u^t + \lambda_u(t)\mathbf{1}_i$

$B_{ui'}^{t+1} = \sum_{j=1}^n \beta_{uj}^{t+1} K(\mathbf{x}_{i'}, \mathbf{x}_j) = (1 - \lambda_u(t))B_{ui'}^t + \lambda_u(t) K(\mathbf{x}_i, \mathbf{x}_{i'})$ $\rightarrow \mathcal{O}(nU)$

$A_u^{t+1} = \sum_{j,j'=1}^n \beta_{uj}^{t+1} \beta_{uj'}^{t+1} K(\mathbf{x}_j, \mathbf{x}_{j'}) =$
$(1 - \lambda_u(t))^2 A_u^t + \lambda_u(t)^2 K(\mathbf{x}_i, \mathbf{x}_i) + 2\lambda_u(t)(1 - \lambda_u(t))B_{ui}^t$ $\rightarrow \mathcal{O}(U)$

  end for

end for

Reducing the stochastic K-SOM complexity

[Mariette et al., 2017]

Data: $(\mathbf{x}_i)_{i=1,\ldots,n} \in \mathcal{X}$

1: Initialization: $p_u^0 = \sum_{i=1}^n \beta_{ui}\phi(\mathbf{x}_i)$ (convex combination)

2: $A_u^0 = \sum_{j,j'=1}^n \beta_{uj}^0 \beta_{uj'}^0 K(\mathbf{x}_j, \mathbf{x}_{j'})$ $\quad \rightarrow \mathcal{O}(n^2 U)$

3: $B_{ui}^0 = \sum_{j=1}^n \beta_{uj}^0 K(\mathbf{x}_j, \mathbf{x}_i)$ $\quad \rightarrow \mathcal{O}(nU)$

4: **for** $t = 1 \rightarrow \gamma n$ **do**

5: pick at random $i \in \{1, \ldots, n\}$

6: Assignment $\quad f^t(\mathbf{x}_i) = \arg\min_{u=1,\ldots,U} A_u^t - 2B_{ui}^t$ $\quad \rightarrow$ does not depend on n

7: **for all** $u = 1 \rightarrow U$ **do** Representation

8: $\beta_u^{t+1} = (1 - \lambda_u(t))\beta_u^t + \lambda_u(t)\mathbf{1}_i$

$$B_{ui'}^{t+1} = \sum_{j=1}^n \beta_{uj}^{t+1} K(\mathbf{x}_{i'}, \mathbf{x}_j) = (1 - \lambda_u(t))B_{ui'}^t + \lambda_u(t)K(\mathbf{x}_i, \mathbf{x}_{i'}) \quad \rightarrow \mathcal{O}(nU)$$

$$A_u^{t+1} = \sum_{j,j'=1}^n \beta_{uj}^{t+1}\beta_{uj'}^{t+1} K(\mathbf{x}_j, \mathbf{x}_{j'}) =$$
$$(1 - \lambda_u(t))^2 A_u^t + \lambda_u(t)^2 K(\mathbf{x}_i, \mathbf{x}_i) + 2\lambda_u(t)(1 - \lambda_u(t))B_{ui}^t \quad \rightarrow \mathcal{O}(U)$$

9: **end for**

10: **end for**

Final complexity: $\mathcal{O}(\gamma n^2 U)$ with additional storage memory of $\mathcal{O}(U)$ and $\mathcal{O}(Un)$.

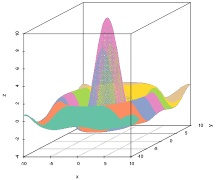
Outline

1 SOMbrero: an R package for stochastic SOM

2 KORRESP

3 Dissimilarity data

4 Applications

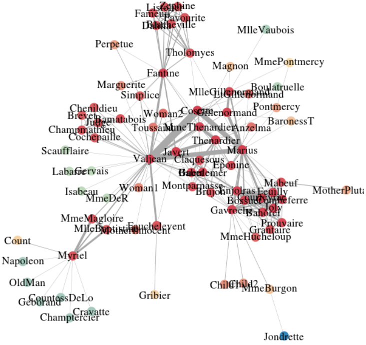


SOMbrero is well designed to handle graphs...

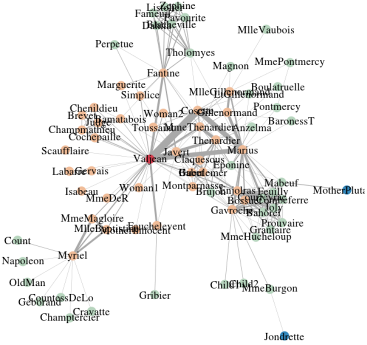
[Olteanu and Villa-Vialaneix, 2015b]

Graphs are kernel/dissimilarity data

Commute time kernel



Shortest path length



SOMbrero is well designed to handle graphs...

[Olteanu and Villa-Vialaneix, 2015b]

Graphs are kernel/dissimilarity data

Analysis of SOM results

SOM produces a map *st*:

- each node of the graph is associated to a unit of the map
- “close” nodes (according to the similarity/dissimilarity measure)

SOMbrero is well designed to handle graphs...

[Olteanu and Villa-Vialaneix, 2015b]

Graphs are kernel/dissimilarity data

Analysis of SOM results

SOM produces a map *st*:

- each node of the graph is associated to a unit of the map
- “close” nodes (according to the similarity/dissimilarity measure)

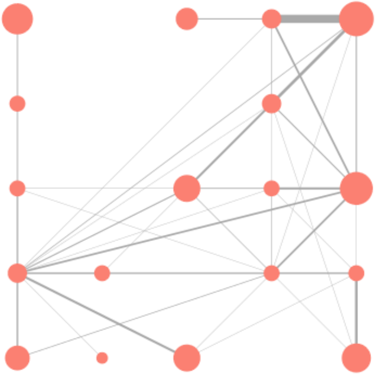
Using the result to produce a simplified representation of the graph (function `projectIGraph`), which is the representation of a meta-graph:

- meta-nodes are positionned on (non empty) clusters of the map;
- size of a meta-node is proportionnal to the number of nodes classified in this unit;
- edges between meta-nodes have width proportional to the number of edges between nodes of the two corresponding units.

SOMbrero is well designed to handle graphs...

[Olteanu and Villa-Vialaneix, 2015b]

Graphs are kernel/dissimilarity data

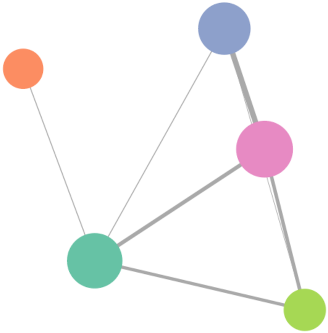


SOMbrero is well designed to handle graphs...

[Olteanu and Villa-Vialaneix, 2015b]

Graphs are kernel/dissimilarity data

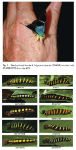
Combined with super-clustering, positionning meta-nodes at the centers of gravity of the clusters on the map (`superClass` + `projectIGraph`)

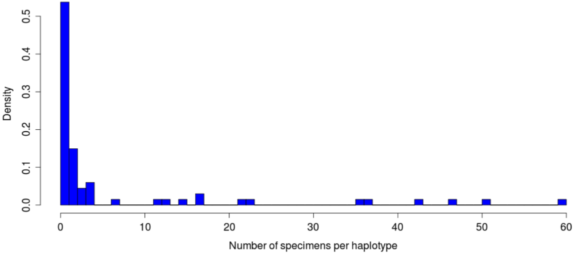


SC1: main characters (including Valjean), SC2: Bishop Myriel and related characters, SC3: Gavroche and related characters, SC4: rest of the

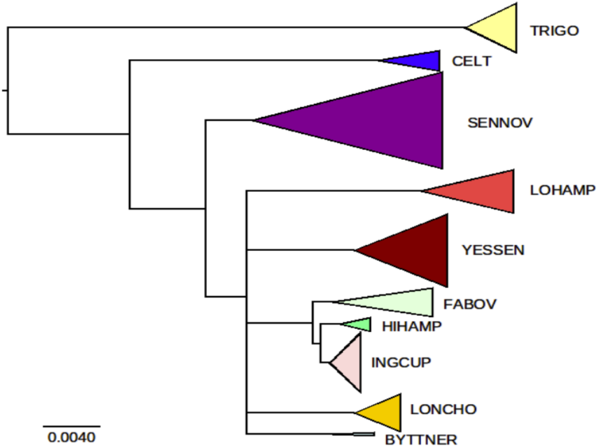
Astraptes Fulgerator data

- 465 samples (67 haplotypes)
- 663 sites CO1
- Low separation level for the species (1 to 19 haplotypes within each species)
- 10 described species [Hebert et al., 2004]

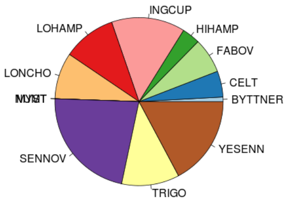




Phylogenetic tree for the *Astraptes* data



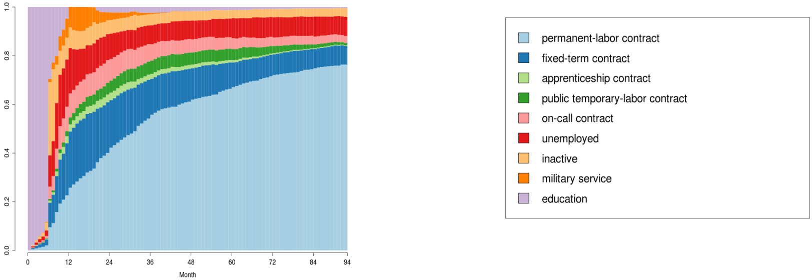
Relative distribution of the ten species and MDS projection of the samples



Topographic error: 0.0022

Relational SOM applied to career paths

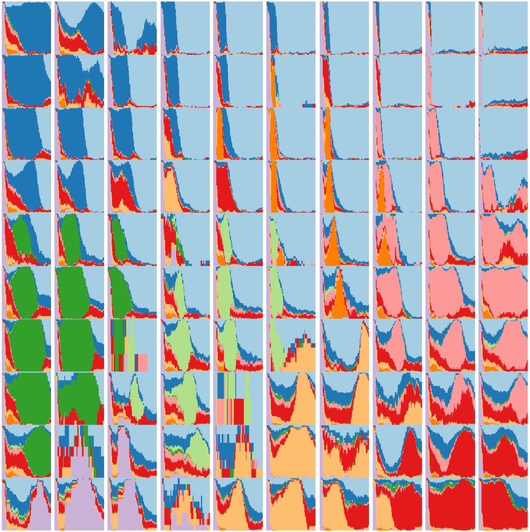
[Massoni et al., 2013]

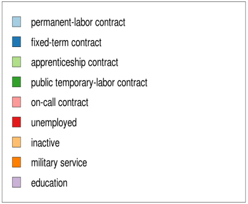


Two 10x10 maps trained on the 16040 sequences with :

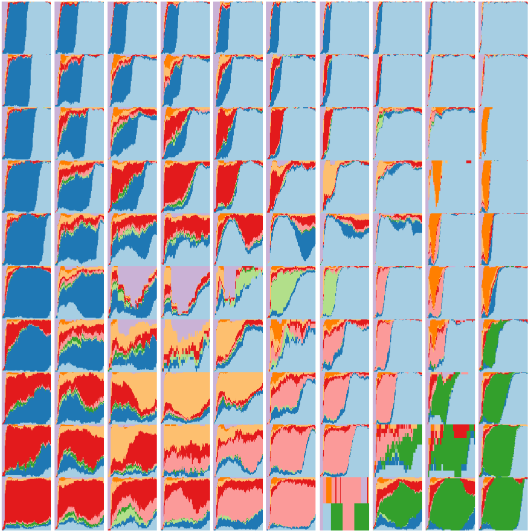
- The χ^2 distance - emphasizes the contemporaneity of the situations
- The LCS distance - emphasizes the transitions, the order of the situations and the common subsequences

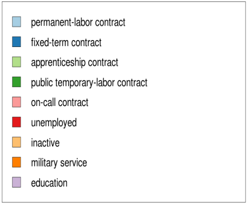
Results with the χ^2 distance





Results with the LCS distance

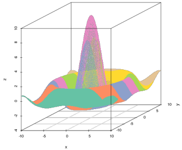




Perspectives

SOMbrero would benefit from:

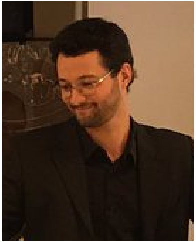
- faster implementation using C/C++
- nicer graphics with *e.g.*, **ggplot2**
- more flexible way to handle topology of the map
- (currently under implementation) kernel computation
- better coverage with **testhat**
- public **shiny** interface
- ...



Do not hesitate to use it and send us your feedback!

Special thanks to



Come in Toulouse in 2019 to learn more about R...

http://user2019.fr

Boelaert, J., Bendhaïba, L., Olteanu, M., and Villa-Vialaneix, N. (2014).
SOMbrero: an r package for numeric and non-numeric self-organizing maps.
In Villmann, T., Schleif, F., Kaden, M., and Lange, M., editors, *Advances in Self-Organizing Maps and Learning Vector Quantization (Proceedings of WSOM 2014)*, volume 295 of *Advances in Intelligent Systems and Computing*, pages 219–228, Mittweida, Germany. Springer Verlag, Berlin, Heidelberg.

Cottrell, M. and Letrémy, P. (2005).
How to use the Kohonen algorithm to simultaneously analyse individuals in a survey.
Neurocomputing, 63:193–207.

Fouss, F., Pirotte, A., Renders, J., and Saerens, M. (2007).
Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation.
IEEE Transactions on Knowledge and Data Engineering, 19(3):355–369.

Hebert, P., Penton, E., Burns, J., Janzen, D., and Hallwachs, W. (2004).
Ten species in one: DNA barcoding reveals cryptic species in the neotropical skipper butterfly astraptes fulgerator.
Genetic Analysis, 101(41):14812–14817.

Kohonen, T. (2001).
Self-Organizing Maps, 3rd Edition, volume 30.
Springer, Berlin, Heidelberg, New York.

Kohonen, T. (2014).
MATLAB Implementations and Applications of the Self-Organizing Map.
Unigrafia Oy, Helsinki, Finland.

Kondor, R. and Lafferty, J. (2002).
Diffusion kernels on graphs and other discrete structures.
In Sammut, C. and Hoffmann, A., editors, *Proceedings of the 19th International Conference on Machine Learning*, pages 315–322, Sydney, Australia. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

Lozupone, C., Hamady, M., Kelley, S., and Knight, R. (2007).
Quantitative and qualitative β eiversity measures lead to different insights into factors that structure microbial communities.
Applied and Environmental Microbiology, pages 1576–1585.

Mariette, J., Olteanu, M., and Villa-Vialaneix, N. (2017).
Efficient interpretable variants of online SOM for large dissimilarity data.
Neurocomputing, 225:31–48.

Massoni, S., Olteanu, M., and Villa-Vialaneix, N. (2013).
Which distance use when extracting typologies in sequence analysis? An application to school to work transitions.
In *International Work Conference on Artificial Neural Networks (IWANN 2013)*, Puerto de la Cruz, Tenerife.

Needleman, S. and Wunsch, C. (1970).
A general method applicable to the search for similarities in the amino acid sequence of two proteins.
Journal of Molecular Biology, 48(3):443–453.

Olteanu, M. and Villa-Vialaneix, N. (2015a).
On-line relational and multiple relational SOM.
Neurocomputing, 147:15–30.

Olteanu, M. and Villa-Vialaneix, N. (2015b).
Using SOMbrero for clustering and visualizing graphs.
Journal de la Société Française de Statistique, 156(3):95–119.