SOMbrero : un package R pour les cartes auto-organisatrices

Nathalie Vialaneix

nathalie.vialaneix@inrae.fr
http://www.nathalievialaneix.eu

Toulouse R user group
January 27th, 2022

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

INRAE

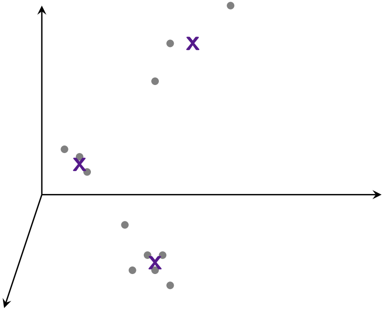
Outline

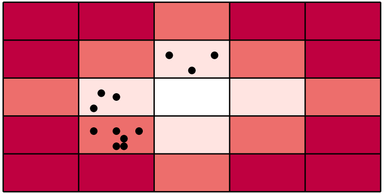
Self-organizing maps and SOMbrero

Other features with SOMbrero

Use case examples

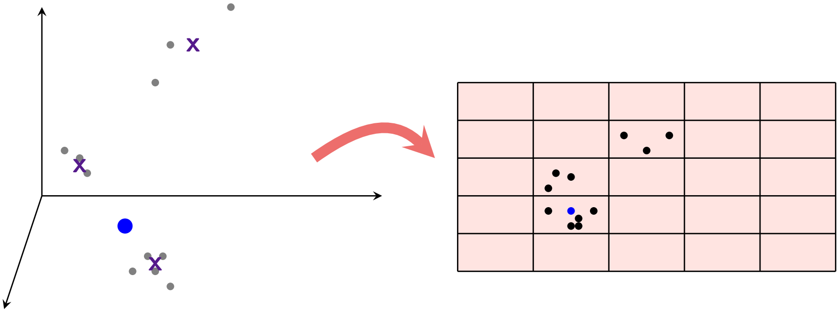
Basics on (standard) stochastic SOM





- $(x_i)_{i=1,\dots,n} \subset \mathbb{R}^d$ are affected to a unit $f(x_i) \in \{1, \dots, U\}$
- 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 (standard) stochastic SOM

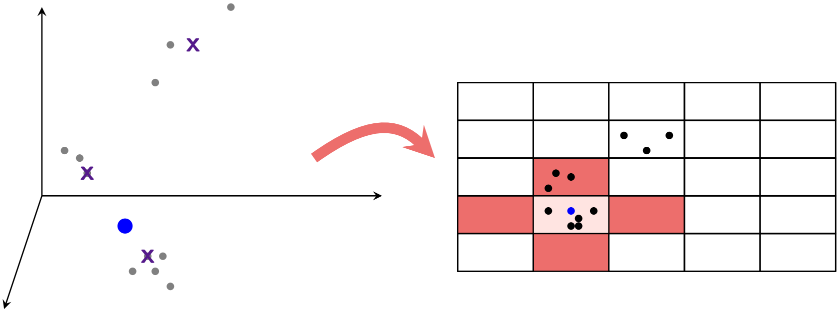


Iterative learning

(assignment step): x_i is picked at random within $(x_k)_k$ and affected to *best matching unit*:

$$f^t(x_i) = \arg\min_u \|x_i - p_u^t\|^2$$

Basics on (standard) stochastic SOM



Iterative learning

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

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

SOMbrero

- SOMbrero is an R package implementing stochastic variants of SOM (standard version and versions specific to non numeric data)

SOMbrero

- SOMbrero is an R package implementing stochastic variants of SOM (standard version and versions specific to non numeric data)
- specifically well adapted for non expert use and teachers:
  - many plots
  - shiny app

SOMbrero

- SOMbrero is an R package implementing stochastic variants of SOM (standard version and versions specific to non numeric data)
- specifically well adapted for non expert use and teachers:
  - many plots
  - shiny app
- reference website: http://sombrero.nathalievialaneix.eu
- Contributors: Élise Maigné, Jérôme Mariette, Madalina Olteanu, Fabrice Rossi and two interns (Julien Boelaert and Laura Bendhaïba)

Alternative tools

- Matlab: SOM Toolbox **[Kohonen, 2001]**

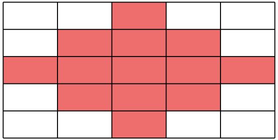
- R (CRAN):
  - class: batch training, crude implementation
  - som (2016): training (two-step batch), basic plots and quality criteria
  - popsom (2017): vectorized stochastic learning (fortran), stochastic learning (C++), batch (C), several plots and quality criteria
  - kohonen (2017): various variants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

Training

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

Options to train the SOM:

- grid: square/hexagonal 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

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

Diagnostic tools

```
quality(mysom)
```

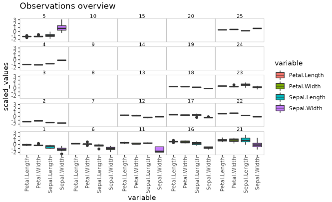
- *topographic error*: average frequency (over the samples) for which the prototypes that comes closest is in the direct neighborhood on the grid of the BMU

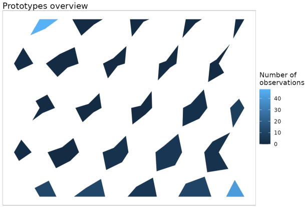
- *quantization error*

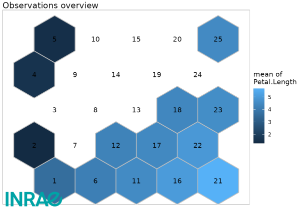
$$\mathcal{Q} = \frac{1}{n} \sum_{i=1}^{n} \|x_i - p_{f(x_i)}\|^2$$

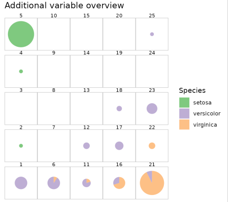
Plots...

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



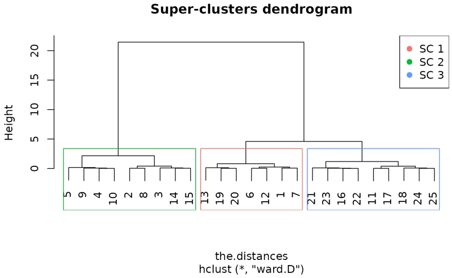


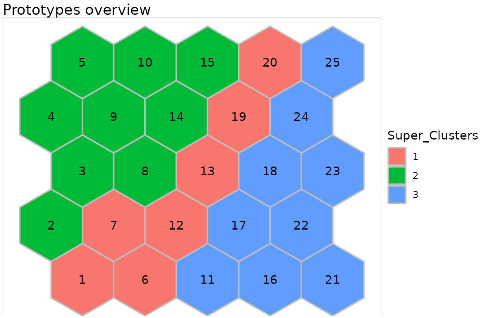




Super-clustering

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



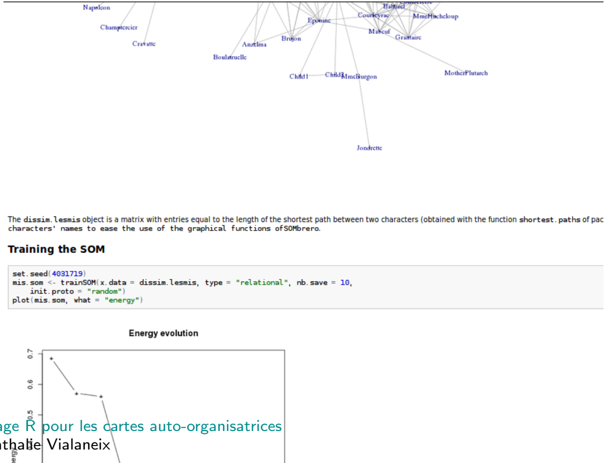


Start with SOMbrero

- 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.

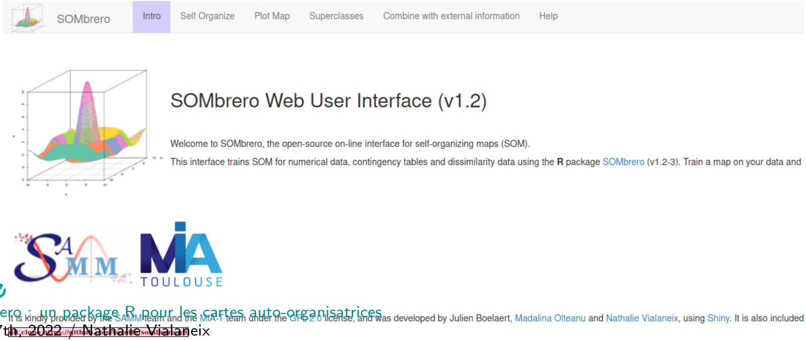
Training the SOM

```
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

Self-organizing maps and SOMbrero

Other features with SOMbrero

Use case examples

Specific features in SOMbrero

- it can cope with missing data (version 1.4-1)

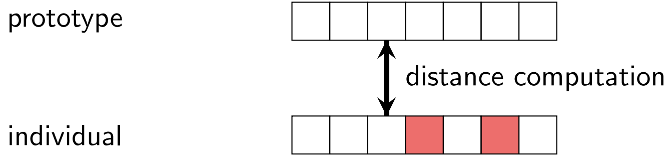
Specific features in SOMbrero

- it can cope with missing data (version 1.4-1)
- it can handle non numeric datasets ("relational" datasets and contingency tables)

Missing data

How does that work?

- during the training phase, missing values are not used

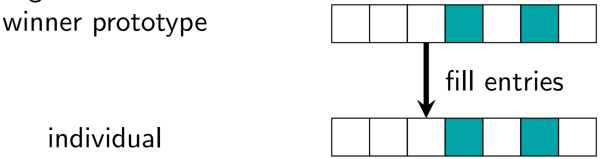


Missing data

How does that work?

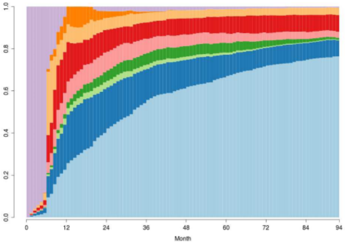
- during the training phase, missing values are not used

- when the training is finished, prototypes can be used to impute (fill in) the missing entries



Relational data 1: Career paths [Olteanu and Villa-Vialaneix, 2015]

Survey "Génération 98": labor market status (9 categories) on more than 16,000 people having graduated in 1998 during 94 months.[1]

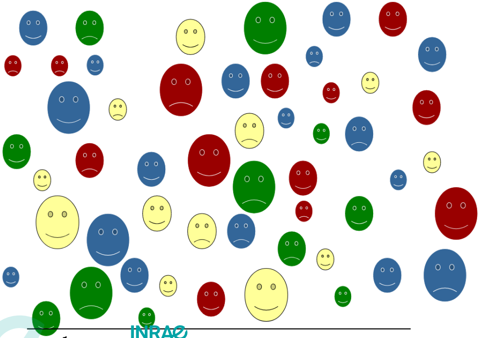


[1] Available thanks to Génération 1998 à 7 ans - 2005, [producer] CEREQ, [diffusion] Centre Maurice Halbwachs (CMH).

Relational data 1: Career paths [Olteanu and Villa-Vialaneix, 2015]

Survey "Génération 98": labor market status (9 categories) on more than 16,000 people having graduated in 1998 during 94 months.[1]

How to cluster career paths into homogeneous groups?

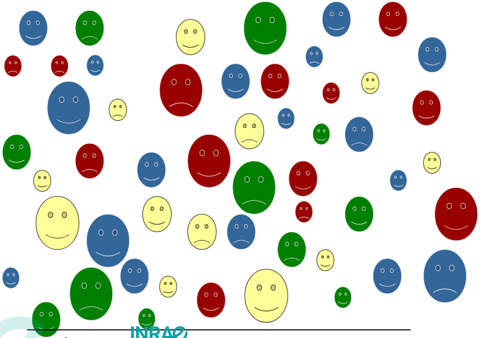


[1] Available thanks to Génération 1998 à 7 ans - 2005, [producer] CEREQ, [diffusion] Centre Maurice Halbwachs (CMH).

Relational data 1: Career paths [Olteanu and Villa-Vialaneix, 2015]

Survey "Génération 98": labor market status (9 categories) on more than 16,000 people having graduated in 1998 during 94 months.[1]

How to cluster career paths into homogeneous groups?



It is all about distance...

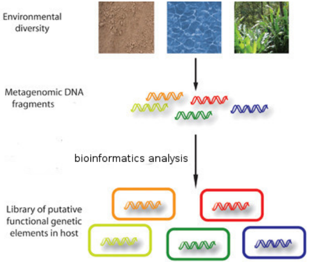
- χ^2 dissimilarity emphasizes the contemporary identical situations
- Optimal-matching dissimilarities is more focused on the sequences similarities [Needleman and Wunsch, 1970] (or "edit distance", "Levenshtein distance")

[1] Available thanks to: Génération 1998 à 7 ans - 2005, [producer] CEREQ, [diffusion] Centre Maurice Halbwachs (CMH).

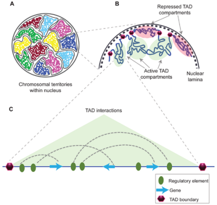
Relational data 2: a collection of NGS data...

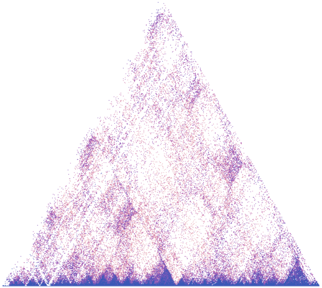
DNA barcoding *Astraptes fulgerator*

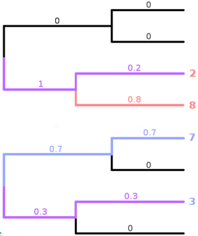
optimal matching (edit) distances to differentiate species



Hi-C data







Metagenomics

dissemblance between samples is better captured when phylogeny between species is taken into account (unifrac distances)

Principles for learning from relational data

Euclidean case (kernel K)
rewrite all quantities using:

- K to compute distances and dot products
- linear or convex combinations of $(\phi(x_i))_i$ to describe all unobserved elements (centers of gravity and so on...)

Principles for learning from relational data

Euclidean case (kernel K)
rewrite all quantities using:

- K to compute distances and dot products
- linear or convex combinations of $(\phi(x_i))_i$ to describe all unobserved elements (centers of gravity and so on...)

Works for: PCA, k-means, linear regression, ...

Principles for learning from relational data

Euclidean case (kernel K)
rewrite all quantities using:

- K to compute distances and dot products
- linear or convex combinations of $(\phi(x_i))_i$ to describe all unobserved elements (centers of gravity and so on...)

Works for: PCA, k-means, linear regression, ...

non Euclidean case (non Euclidean dissimilarity D): do almost the same using a pseudo-Euclidean framework

[Goldfarb, 1984]

$\exists$ two Euclidean spaces $\mathcal{E}_+$ and $\mathcal{E}_-$ and two mappings ϕ_+ and ϕ_- st:

$$D(x, x') = \|\phi_+(x) - \phi_+(x')\|^2_{\mathcal{E}_+} - \|\phi_-(x) - \phi_-(x')\|^2_{\mathcal{E}_-}$$

Note on drawbacks of RSOM

Two main drawbacks:

- For $T \sim \gamma n$ iterations, complexity of RSOM is $\mathcal{O}(\gamma n^3 U)$ (compared to $\mathcal{O}(\gamma U d n)$ for numeric) **[Rossi, 2014]**

Note on drawbacks of RSOM

Two main drawbacks:

- For $T \sim \gamma n$ iterations, complexity of RSOM is $\mathcal{O}(\gamma n^3 U)$ (compared to $\mathcal{O}(\gamma U d n)$ for numeric) **[Rossi, 2014]**

  Exact solution proposed in **[Mariette et al., 2017]** to reduce the complexity to $\mathcal{O}(\gamma n^2 U)$ with additional storage memory of $\mathcal{O}(Un)$

Note on drawbacks of RSOM

Two main drawbacks:

- For $T \sim \gamma n$ iterations, complexity of RSOM is $\mathcal{O}(\gamma n^3 U)$ (compared to $\mathcal{O}(\gamma U d n)$ for numeric) **[Rossi, 2014]**

  Exact solution proposed in **[Mariette et al., 2017]** to reduce the complexity to $\mathcal{O}(\gamma n^2 U)$ with additional storage memory of $\mathcal{O}(Un)$

- For the non Euclidean case, the learning algorithm can be very unstable (saddle points)

Note on drawbacks of RSOM

Two main drawbacks:

- For $T \sim \gamma n$ iterations, complexity of RSOM is $\mathcal{O}(\gamma n^3 U)$ (compared to $\mathcal{O}(\gamma U d n)$ for numeric) **[Rossi, 2014]**

  Exact solution proposed in **[Mariette et al., 2017]** to reduce the complexity to $\mathcal{O}(\gamma n^2 U)$ with additional storage memory of $\mathcal{O}(Un)$

- For the non Euclidean case, the learning algorithm can be very unstable (saddle points)

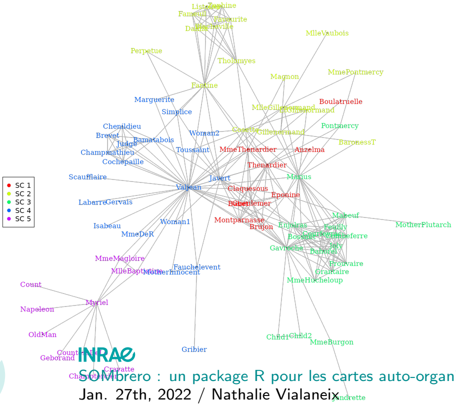
  clip or flip? **[Chen et al., 2009]** $\Rightarrow$ not provided in SOMbrero

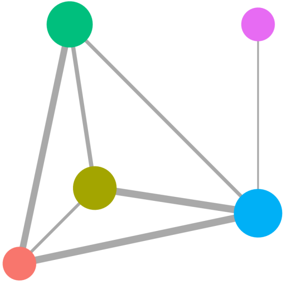
SOMbrero also includes

- the KORRESP algorithm (extension of Correspondence Analysis to SOM) [**Cottrell and Letrémy, 2005**]

SOMbrero also includes

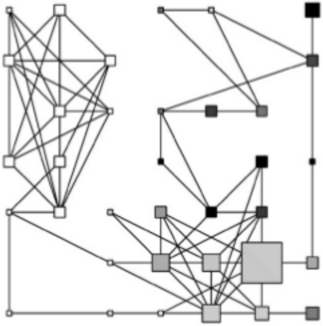
- the KORRESP algorithm (extension of Correspondence Analysis to SOM) **[Cottrell and Letrémy, 2005]**
- specific functions to manipulate graphs with the relational algorithm and obtain simplified representation





Outline

RSOM for mining a medieval social network



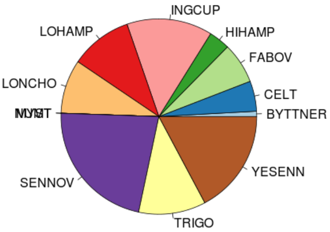
Graph induced by clusters:

- has nice relations with space and time
- emphasizes leading people
- has helped to identify problems in the database (namesakes)

[Boulet et al., 2008]

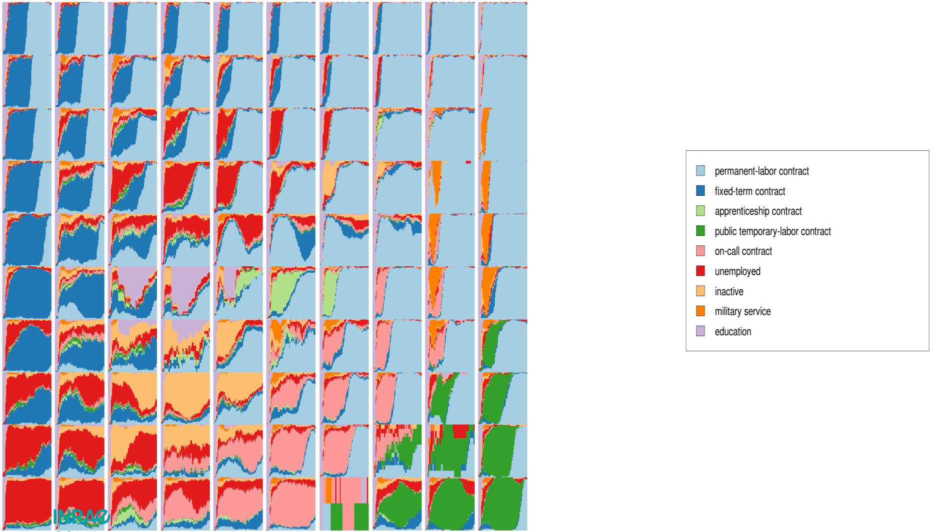
But: biggest communities are still very

RSOM for typology of *Astraptes fulgerator* from DNA barcoding



Almost perfect clustering (identifying a possible label error on one sample) with (in addition) information on relations between species.

RSOM for typology of school-to-time transitions



Madalina Olteanu, Fabrice Rossi,
Marie Cottrell, Laura Bendhaïba
and Julien Boelaert






Jérôme Mariette


Élise Maigné

References

Boulet, R., Jouve, B., Rossi, F., and Villa, N. (2008).
Batch kernel SOM and related Laplacian methods for social network analysis.
Neurocomputing, 71(7-9):1257–1273.

Chen, Y., Garcia, E., Gupta, M., Rahimi, A., and Cazzanti, L. (2009).
Similarity-based classification: concepts and algorithm.
Journal of Machine Learning Research, 10:747–776.

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

Goldfarb, L. (1984).
A unified approach to pattern recognition.
Pattern Recognition, 17(5):575–582.

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

Mariette, J., Rossi, F., Olteanu, M., and Villa-Vialaneix, N. (2017).
Accelerating stochastic kernel som.
In Verleysen, M., editor, *XXVth European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2017)*, pages 269–274, Bruges, Belgium. i6doc.

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. (2015).
On-line relational and multiple relational SOM.
Neurocomputing, 147:15–30.

Rossi, F. (2014).
How many dissimilarity/kernel self organizing map variants do we need?
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 3–23, Mittweida, Germany. Springer Verlag, Berlin, Heidelberg.