

Reorder.js: A JavaScript Library to Reorder Tables and Networks

Jean-Danie Fekete, INRIA



github.com/jdfekete/reorder.js

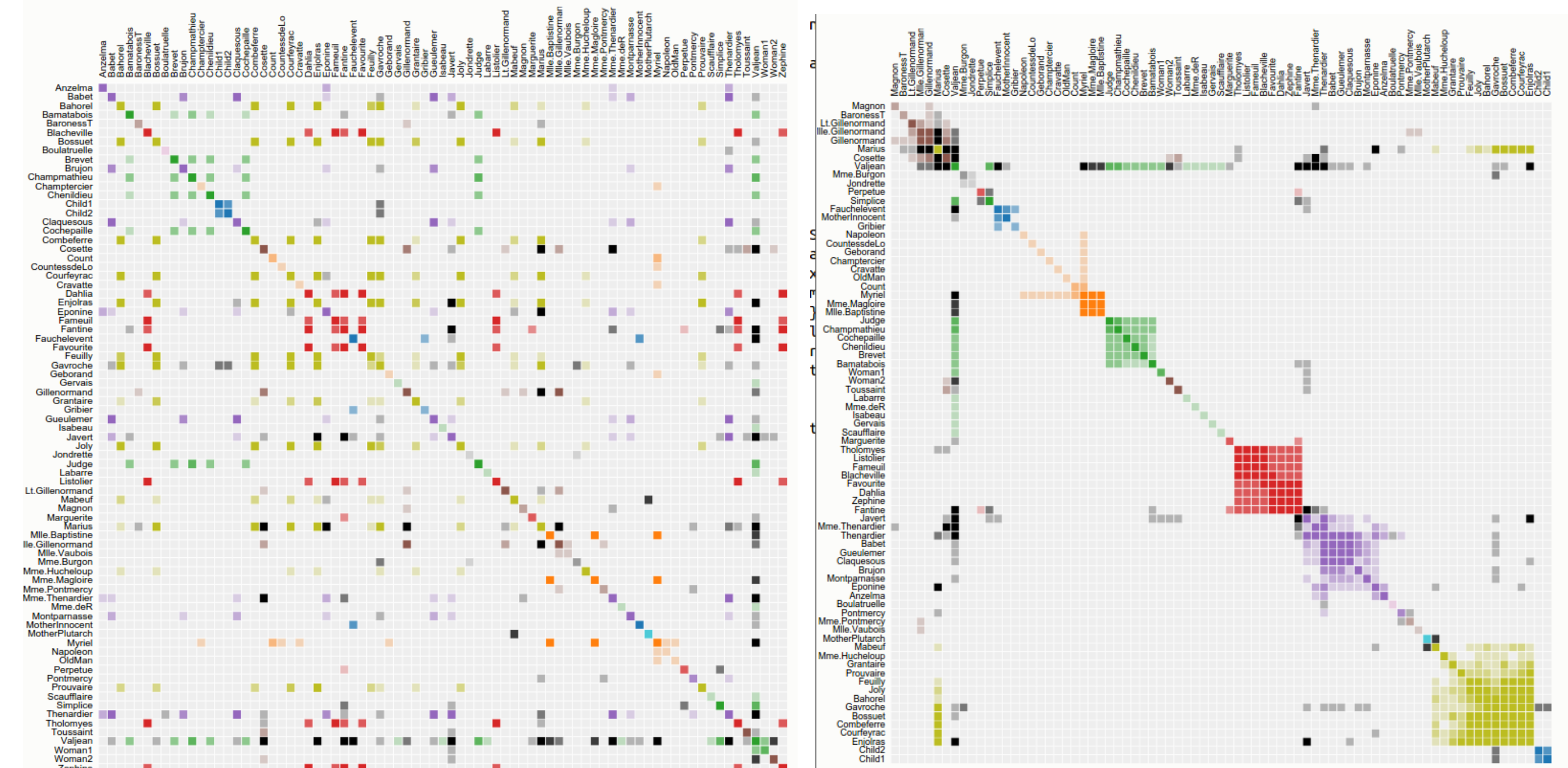
Reordering, also known as *seriation* and *linear ordering*, consists in assigning an order to rows/columns of a table or to vertices of a network to reveal structures.

Many algorithms exist to order table row/columns or graph vertices; `Reorder.js` provide the most effective ones for visualization in JavaScript/HTML5.

In addition to *algorithms*, `Reorder.js` provide convenient methods to *convert tables into graphs* and vice versa. It also provides *quality measures* for orderings.

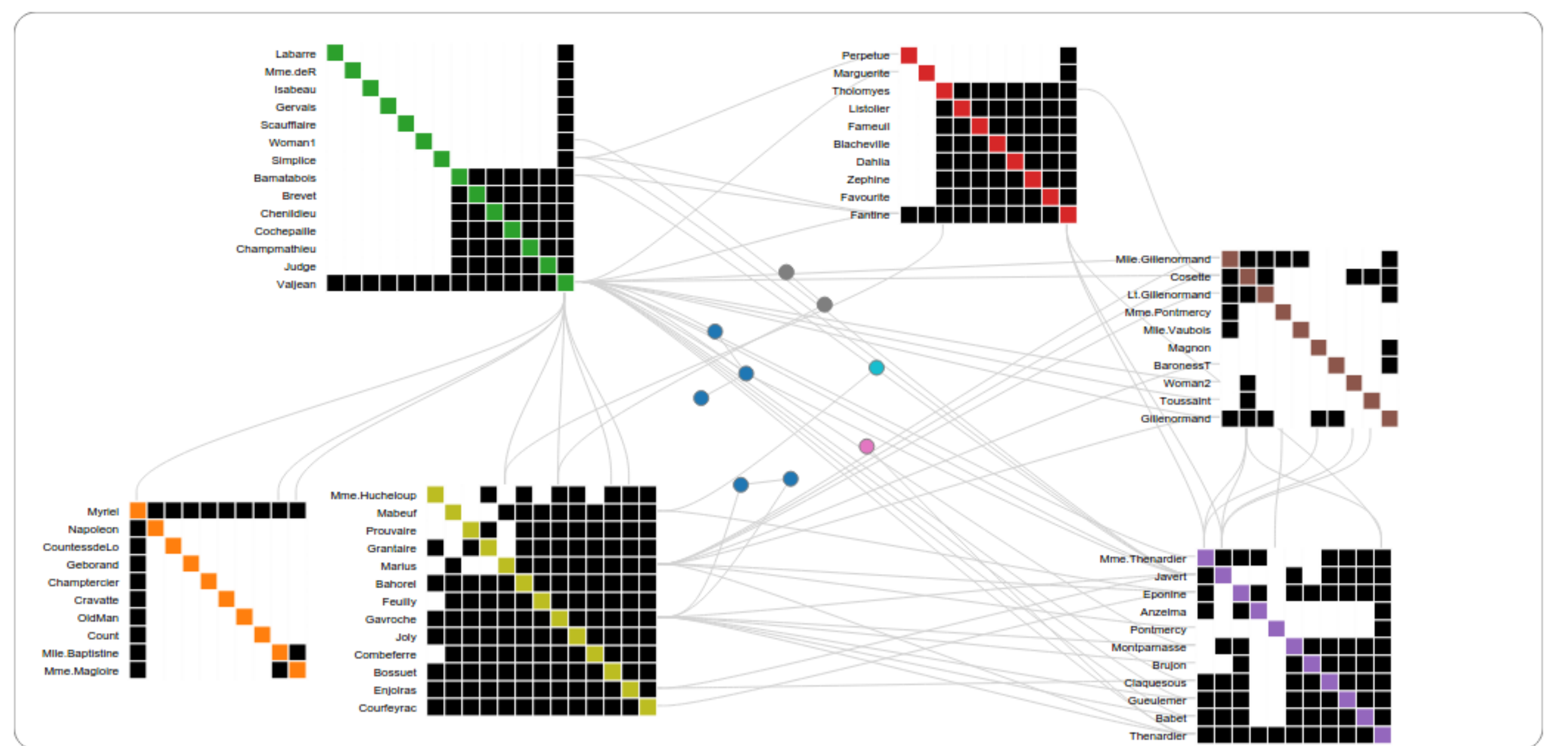
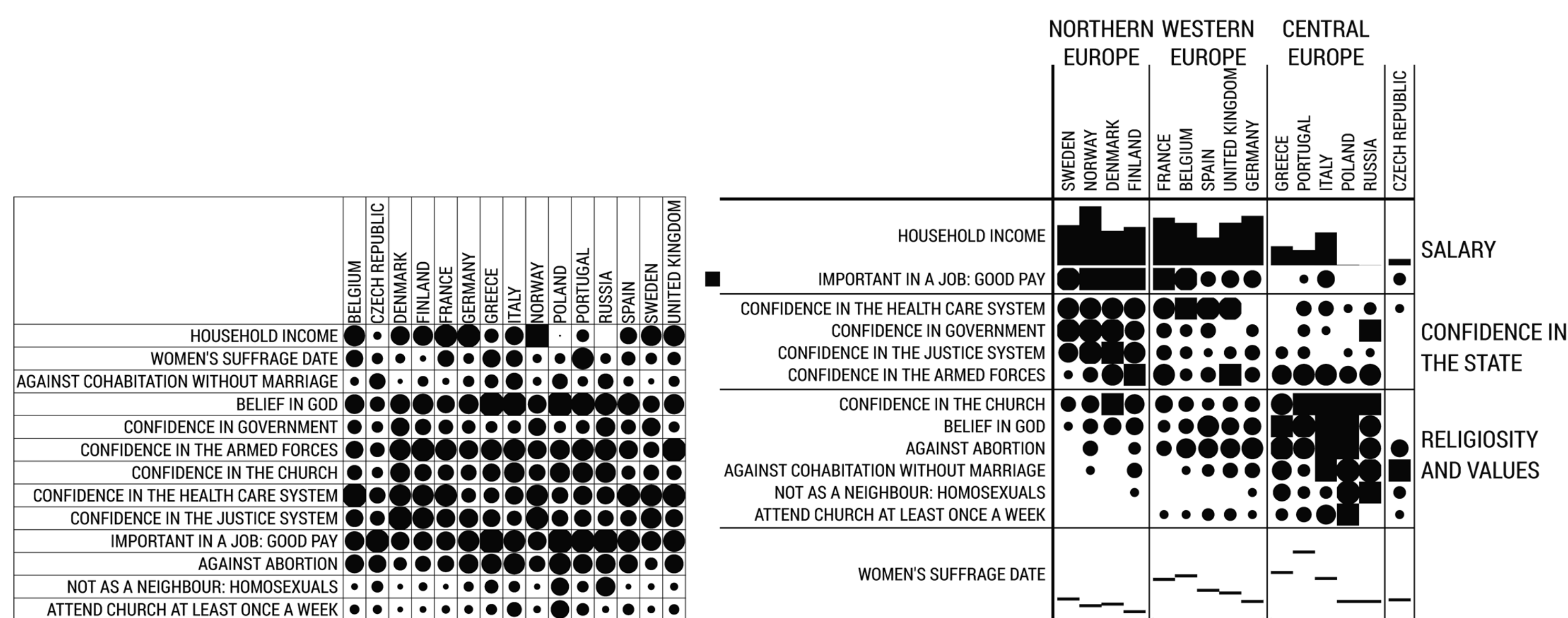
`Reorder.js` can be used in conjunction with `D3.js` or in standalone applications. It is already used in several systems.

Character co-occurrences in Victor Hugo "Les Misérables" . Each colored cell represents two characters that appeared in the same chapter; darker cells indicate characters that co-occurred more frequently.



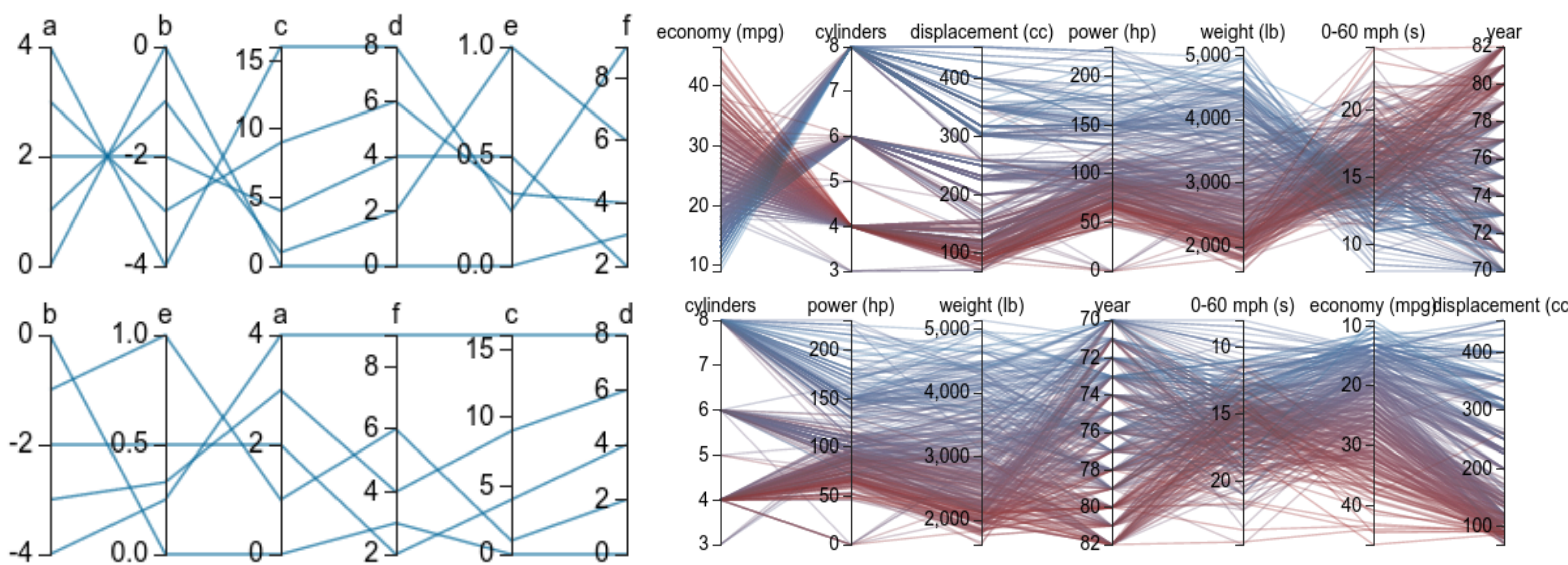
Rows/columns sorted alphabetically: no high-level structure is visible

Rows/columns sorted using the "optimal leaf ordering" algorithm shows structures.



Bertifier uses it to order of table rows/columns with additional constraints. For more information, see: bertifier.com/ and www.aviz.fr/Bertifier

Nodetrix uses it to order the small adjacency matrices at creation time. For more information, see: github.com/IRT-SystemX/nodetrix and www.aviz.fr/Research/Nodetrix



Parallel Coordinates can be improved by ordering the axes by correlation order, and flipping axes when next to a negatively correlated axis, avoiding the *knots*. See syntagmatic.github.io/parallel-coordinates/

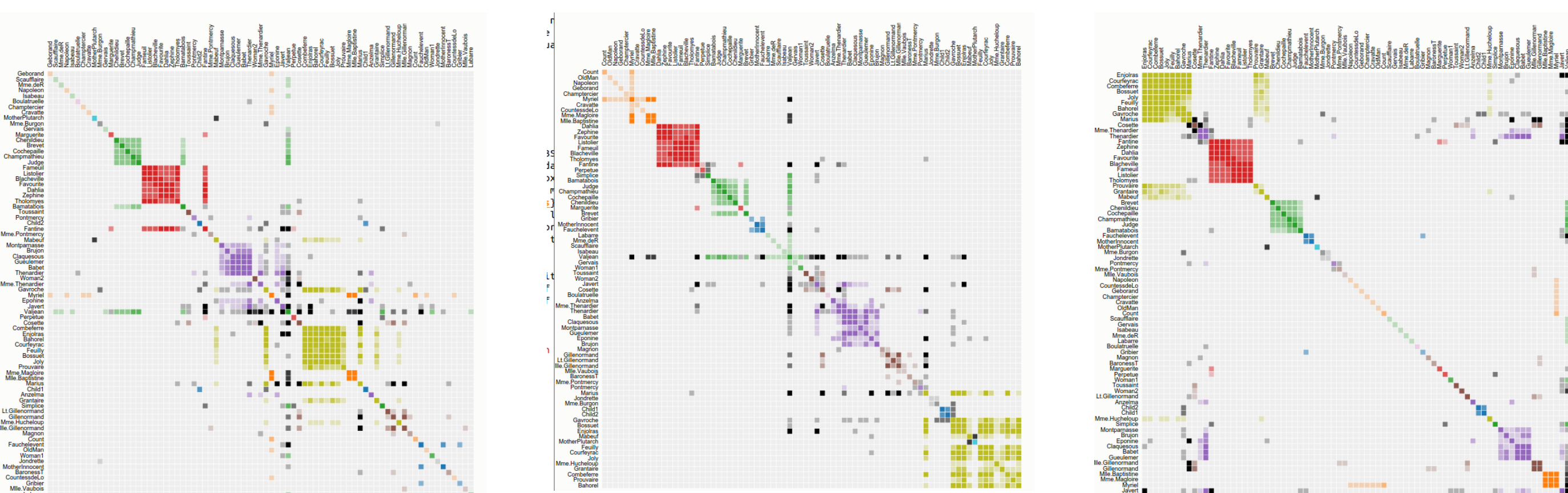
Rich API with

- Unifying concepts: *table*, *graph*, *order* or *permutation*, *distance-matrix*
- 8 High-level reordering algorithms: *Barycenter*, *Bfs*, *Correspondence*, *Cuthill-McKee*, *Optimal Leaf Ordering*, *PCA*, *Spectral*
- Quality measures: *bandwidth*, *crossing counts*, *edge sum*, etc.
- Lower-level algorithms: *iterative eigenvectors*, *distance functions*, *random permutations*.
- Conversions from graph to tables and vice-versa

Simple coding:

```
require("science");
require("reorder.v1");

var mat = [
  [0, 1, 0, 1, 0],
  [1, 0, 1, 0, 1],
  [0, 1, 0, 1, 1],
  [1, 1, 1, 0, 0]
];
var perm = reorder.leafOrder()(mat);
var permuted_mat = reorder.stablepermute(mat, perm);
```



Visit us at www.aviz.fr

Visit us at www.inria.fr

