|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| D3.js | Coala.js | Sigmajs | Arbors.js | Getspringy |
| * Many tutorials * Relatively good performance with large data sets * Extensive GitHub explanation * Gigantic library of examples * Works with big data sets | * Drop-in replacement of D3 force layout * Higher quality layout then d3.js * More stable when interacted with (“no jitter”) * Automatically generates constraints to avoid overlapping of nodes by other nodes and edges * Layout constrains can be set (like grouping, flow layout, alignments with axes etc. * Many tutorials * Very “snappy” | * Very good documentation on GitHub * Interactivity oriented * Is actively developed * Probably high performance * Can use Graphs in Json and GEXF | * Very simple (easy to learn) * Good documentation by creator | * Very VERY simple * good documentation on the creator’s page |
| * “jittery” | * Works best with smaller graphs with fewer than 100 nodes | * Optimisations on larger datasets is unknown | * Very simple (view commands) * Bad performance with large data sets * From 2011 (not very up-to-date) | * Super limited in its possibilities * Optimisations on larger datasets is unknown |
|  | * <https://ialab.it.monash.edu/webcola/examples/browsemovies.html> (that’s literally what we want to make but with films and actors) |  |  |  |