# Open Sankey v1.1 – Documentation – 05/2015

\*\*\*\*\*

Changes compared to v1:

* Adding of recycling flows (when target node is located before source node).
* Javascript code of the form “[x,y] = [1,2];” is not recognized by Google Chrome, therefore when Chrome is detected on page “sankey.php”, the script “manual\_sankey\_chrome.js” is loaded instead of “manual\_sankey.js” (above code is replaced by “x=1; y=2;”).

\*\*\*\*\*

## 0) Based on D3.

Open Sankey is based on Mike Bostock’s D3 javascript library (http://d3js.org). A local “d3.v3.js” file is provided in “open-sankey/sources/js”.

## 1) Organization of SVG elements in the web page.

Starting from a blank page:

::Library:Application Support:SnapNDrag:screenshot_01.jpg

We created a first node named « pommes » :



- “gg\_node0” is the group containing the graphical representation of the node (a rectangle) and the associated label.

- Position of node X is defined in “gg\_nodeX” by attribute “transform”. Attributes “x” and “y” have no meaning for the svg element “g” but are used to store data more easily. Here, 430 and 230 are the default position values given by the “add\_new\_node()” function.

- “title” contains the id of the node, which is displayed on mouseover (after a few seconds).

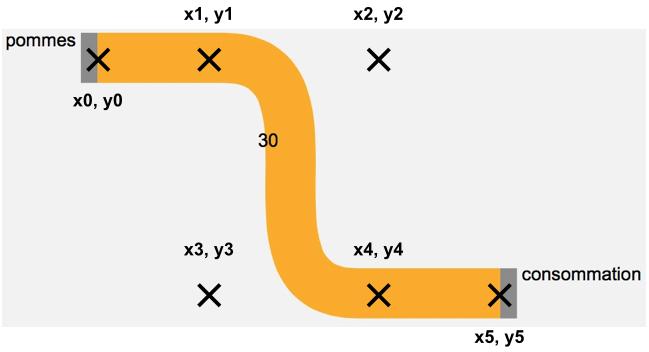
We create a second node named “consommation” and a link of value 30 connecting node “pommes” (0) with node “consommation” (1):



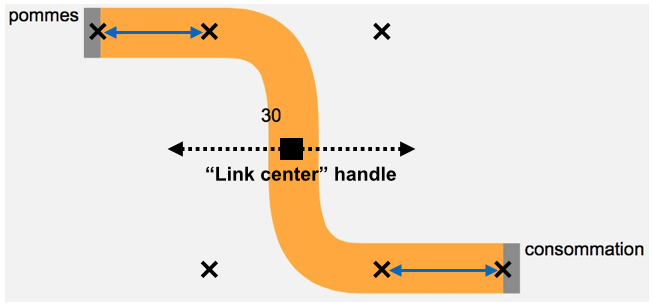
- In the “node0” and “node1”, we can see that attribute “height” is now 30. For each node, the attribute corresponds to the maximum between the sum of input links and the sum of output links.

- A “gg\_link0” has been created. It contains the graphical representation of the link (a path of id “link0”, the link label (“link\_value0”), the label of the link uncertainty (“link\_sd\_value0”), and a square (“link\_center0”) only visible on mouseover and indicating, in the case of a link between two vertical nodes, where (on the x-axis) the link should go up or down. By default, this square is located at the midle of source and target nodes.

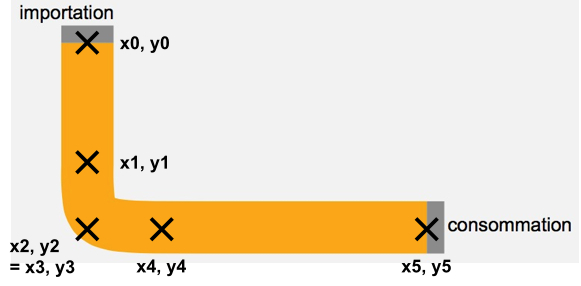
- The “stroke-width” attribute of the de path object “link0” corresponds to the value of the link (concretely, the link is a line with a thickness).

- The attribute “d” of path object « link0 » defines the curve to be drawn. “d” is of form “M x0,y0 L x1,y1 C x2,y2 x3,y3 x4,y4 L x5,y5" :

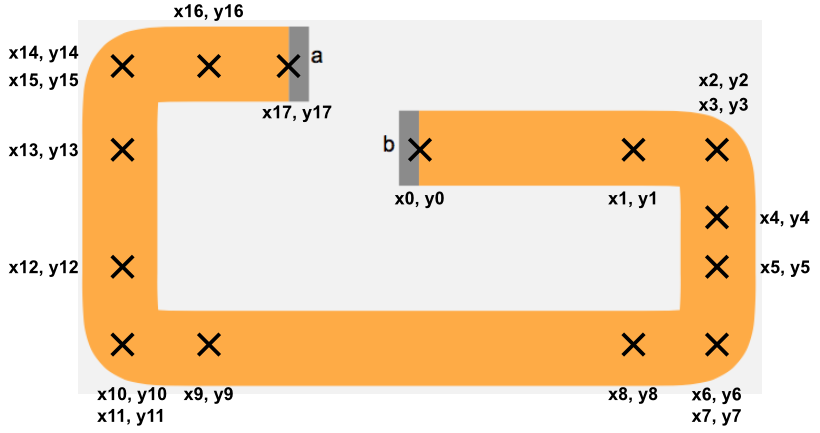
x2,y2 and x3,y3 are respectively the initial and final control points of a cubic Bézier curve connecting x1,y1 to x4,y4.

Distances between some of the points can be changed using the “link\_center” handle:

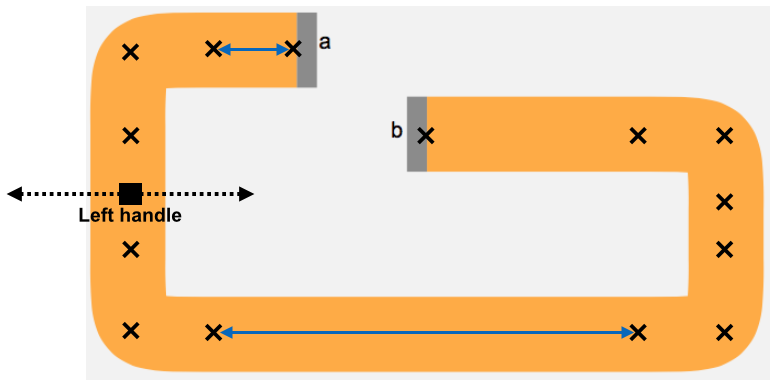
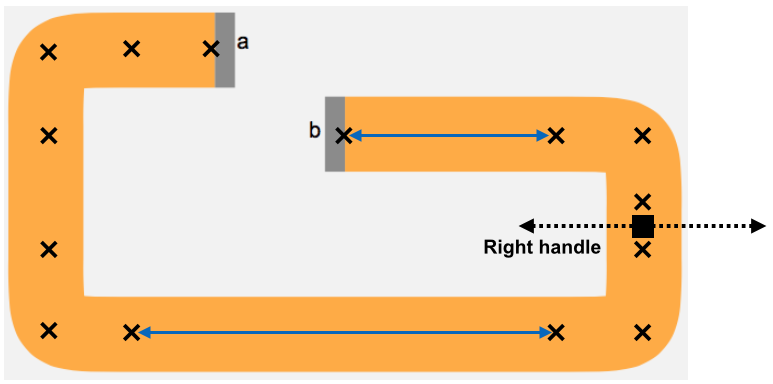
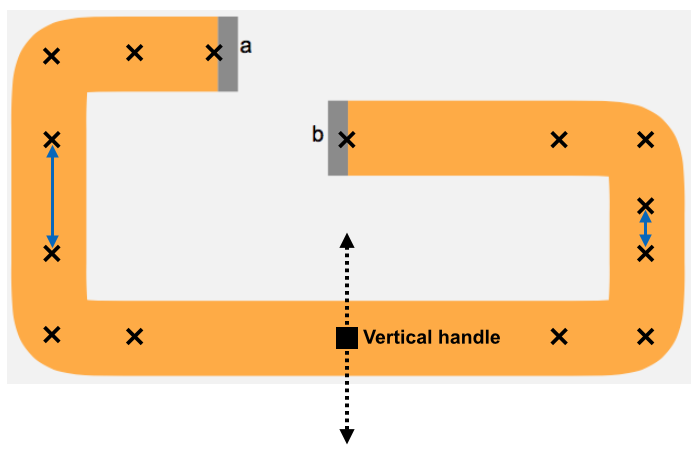
- In the case of a link connecting a horizontal node to a vertical node, the situation is the following:



- In the case of a recycling flow (target node before source node):



Distances between some of the points can be changed using 3 different handles: they are respectively identified as “left\_horiz\_shift”, “right\_horiz\_shift” and “vert\_shift” in the javascript code.



## 2) Javascript variables

**NB : The list of nodes and links attributes must be updated in the file “souces/php\_sankey/attributes.php”.**

Nodes are stored in the variable “nodes” and have the following attributes:

* id,
* color,
* name,
* orientation // vertical or horizontal
* x, // position of the node
* y,
* x\_label, // position of the label
* y\_label,
* input\_links, // ordered list of input links (graphically, for a horizontal node, the first link is on the left, and for a vertical node, the first link is on the top)
* output\_links,
* input\_offsets, // ordered list of offsets between links. If v\_X is the value of link X, for a vertical node, we will have: y\_X = y\_Node + input\_offsets[X] + v\_X / 2
* output\_offsets,
* total\_input\_offset, // sum of all input links
* total\_output\_offset,
* merged\_name // (not mandatory) indicates which node must be joined in the case of a click on “aggregate flows”. For instance, “apple” and “pears” can be aggregated into “fruits”.

Links are stored in the variable « links » and have the following attributes:

* id,
* color,
* source, // id of the source node
* target,
* source\_name, // name of the source node
* target\_name,
* value,
* sd\_value, // value of the standard deviation associated to the flow
* x\_label, // position of the label
* y\_label,
* x\_center, // position of the square « link\_center », value between 0 and 1 (in practice these extreme values cannot be reached) : 0 would mean that the link goes up/down immediately after exiting the source node; 1 would mean that the link goes up/down just before the target node; 0.5 means the link goes up/down at the middle of source and target nodes.
* x\_sd\_label, // position of the uncertainty label (in fact a confidence interval corresponding to 2\*standard deviation)
* y\_sd\_label

Filtered nodes and links are stored in variables “filtered\_nodes” and “filtered\_links”.

## 3) Other stuff

--- FOLDERS SOURCES/USER\_SANKEY ---

- auto\_diagrams : stores javascript files automatically generated based on supply/use csv uploads,

- diagram\_uploads : stores diagrams (txt) loaded by users and the javascript files generated on this basis,

- downloaded\_diagrams : stores saved diagrams (downloaded by users),

- layout\_uploads : stores diagrams (txt) loaded by users to serve as layout for updated data (with same nodes and links),

- supply\_uploads : stores supply (csv) files loaded by users,

- use\_uploads : stores use (csv) files loaded by users.

--- SAVE DIAGRAM FUNCTIONS ---

- The web page “sankey.php” contains a modal with a button of id "save\_layout\_button",

- When clicked, the script “sources/js\_sankey/manual\_sankey.js” triggers the fonction save\_layout, then save\_links, then save\_filtered, then save\_env which post information to the server (ajax) and execute the php scripts of the same names.

--- LOAD DIAGRAM FUNCTIONS ---

- The user loads a diagram (txt) which is copied in the folder “sources/user\_sankey/diagram\_uploads” with the ‘.txt’ extension and a random name,

- The path of this file is stored in the session variable $\_SESSION[“file\_path”],

- The txt file is translated into a javascrit file using the script “diagram\_txt\_to\_js.php” which itself calls “functions\_txt\_to\_js.php”,

- $\_SESSION[“path”] goes from a ‘.txt’ to a ‘.js’ extension,

- If there is no error, the browser is redirected to ‘sankey.php’ with the session variable $\_SESSION[“sankey\_type”] = “saved\_diagram” and will add the link to the file $\_SESSION[“file\_path”].