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

\*\*\*\*\*

Changements par rapport à la v1:

* Ajout de flux de recyclage (noeud cible situé avant le noeud source).
* Les codes javascript de la forme “[x,y] = [1,2];” ne sont pas reconnus par Google Chrome. Si Chrome est détecté lors du chargement de “sankey.php”, le script “manual\_sankey\_chrome.js” est chargé plutôt que “manual\_sankey.js” (le code ci-dessus est remplacé par “x=1; y=2;”).

\*\*\*\*\*

## 0) Basé sur D3.

Open Sankey est basé sur la librairie javascript D3 de Mike Bostock (http://d3js.org). Un fichier local “d3.v3.js” est fourni dans “open-sankey/sources/js”.

## 1) Organisation des SVG dans la page internet

Quand on part d’une feuille blanche :

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

On crée un premier nœud appelé « pommes » :



- « gg\_node0 » est le groupe qui contient la représentation graphique du nœud (un rectangle) et le label associé.

- Le placement du nœud X est défini dans « gg\_nodeX » par l’attribut « transform », les attributs « x » et « y » n’ont pas de signification pour le groupe mais sont utilisés pour stocker les données. En l’occurrence, 430 et 230 sont les valeurs par défaut données dans la fonction add\_new\_node().

- « title » contient l’id du nœud qui apparaît quand on laisse la souris dessus quelques secondes.

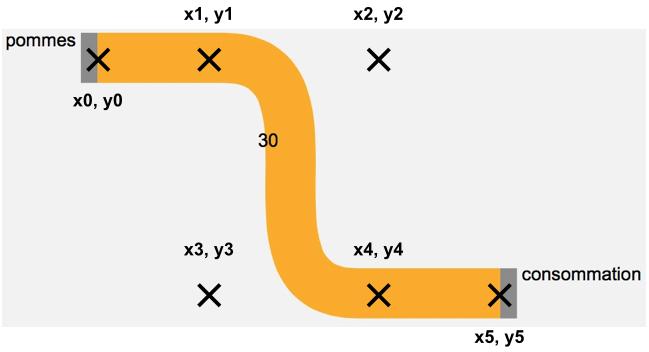
On crée un deuxième nœud appelé « consommation » et un lien d’une valeur de 30 qui relie le nœud « pommes » (0) au nœud « consommation » (1) :



- Dans les objets « node0 » et « node1 », on peut voir que l’attribut « height » est passé à 30. Pour chaque nœud, cet attribut correspond au maximum entre la somme des liens entrant et la somme des liens sortant.

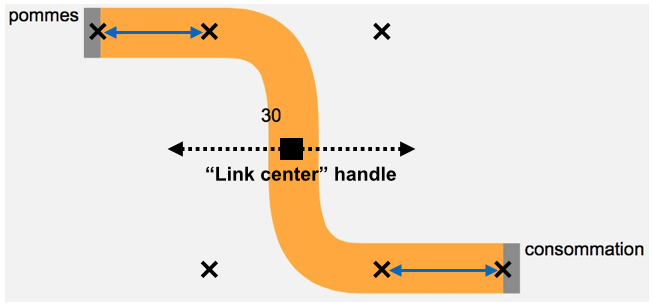
- Un objet « gg\_link0 » a été créé qui contient la représentation graphique du lien (un path d’identifiant « link0 »), le label du lien (« link\_value0 »), le label de l’incertitude sur le lien (« link\_sd\_value0 »), et un carré « link\_center0 » visible uniquement lorsque l’on passe la souris dessus qui indique, dans le cas d’un lien entre deux nœuds verticaux, à quel endroit le lien doit monter ou descendre. Par défaut ce carré se trouve au milieu des nœuds source et cible.

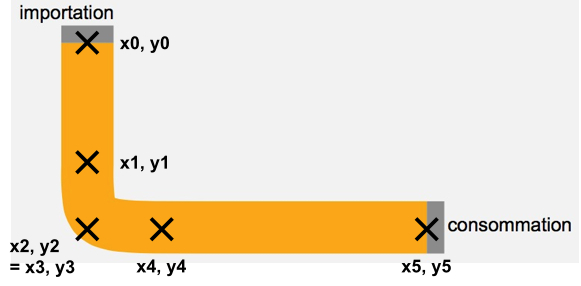
- L’attribut « stroke-width » de l’objet path « link0 » correspond à la valeur du lien (concrètement le lien est un trait auquel on donne une épaisseur).

- L’attribut « d » de l’objet path « link0 » définit la courbe à tracer. « d » est de la forme « M x0,y0 L x1,y1 C x2,y2 x3,y3 x4,y4 L x5,y5" :

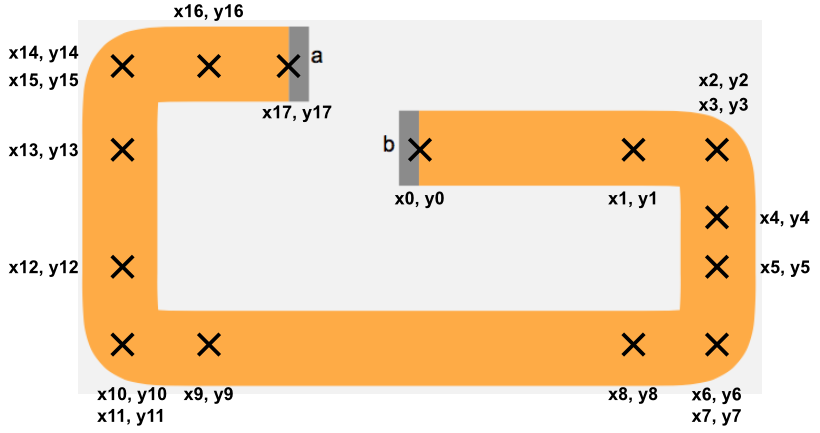
x2,y2 et x3,y3 sont respectivement le point de contrôle initial et le point de contrôle final d’une courbe de Bézier cubique qui relie x1,y1 à x4,y4.

La distance entre certains points peut être modifiée avec la poignée « link\_center » :

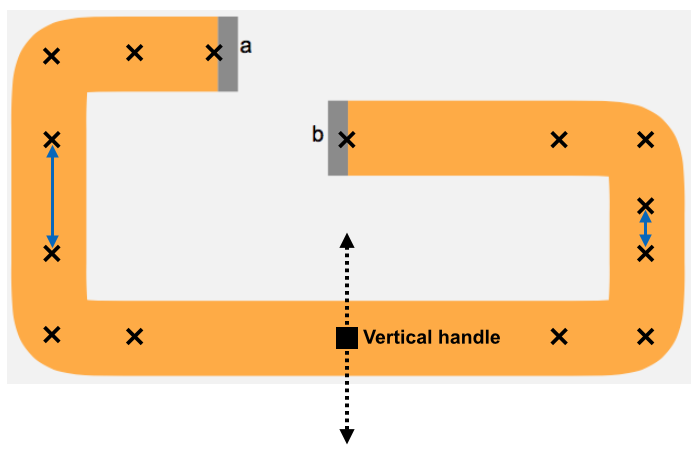


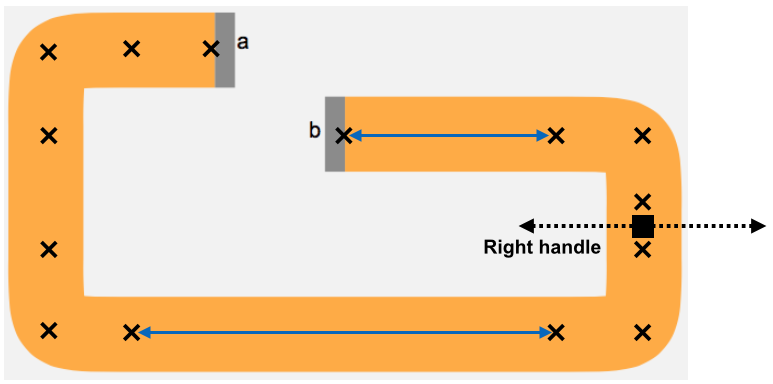
- Dans le cas d’un lien reliant un nœud horizontal à un nœud vertical le schéma est le suivant :

- Dans le cas d’un flux de recyclage (nœud cible situé avant le nœud source) :



Certaines distances peuvent être modifiées à l’aide de 3 poignées identifiés « left\_horiz\_shift », « right\_horiz\_shift » et « vert\_shift » dans le code javascript.





## :::::::Desktop:left_horiz_handle.png2) Stockage des variables javascript

**NB : la liste des attributs des noeuds et des liens doit être mise à jour dans le fichier “souces/php\_sankey/attributes.php”.**

Les nœuds sont stockés dans la variable (liste) « nodes » et ont pour attributs :

* id,
* color,
* name,
* orientation // vertical ou horizontal
* x, // position du nœud
* y,
* x\_label, // position du label
* y\_label,
* input\_links, // liste ordonnée des liens entrant (graphiquement, pour un nœud horizontal le premier lien entrant se situe le plus à gauche, pour un nœud vertical, le premier lien entrant se situe le plus en haut)
* output\_links,
* input\_offsets, // liste ordonnée des décalages entre les liens. Si v\_X est la valeur du lien X, pour un noeud vertical, on aura : y\_X = y\_Noeud + input\_offsets[X] + v\_X / 2
* output\_offsets,
* total\_input\_offset, // somme des valeurs de tous les liens entrant
* total\_output\_offset,
* merged\_name // (facultatif) indique le nœud qui doit être rejoint en cas de clic sur « agréger les flux », par exemple « pommes » et « poires » peuvent être agrégés dans « fruits ».

Les liens sont stockés dans la variable « links » et ont pour attributs :

* id,
* color,
* source, // id du nœud source
* target,
* source\_name, // nom du nœud source
* target\_name,
* value,
* sd\_value, // valeur de l’écart-type associé au flux
* x\_label, // position du label
* y\_label,
* x\_center, // position du carré « link\_center », valeur comprise entre 0 et 1 (en pratique, on fait le choix que ces valeurs extrêmes ne puissent pas être atteintes) : 0 signifierait que le lien descend/monte immédiatement à la sortie du nœud source, 1 signifierait que le lien descend/monte uniquement à l’arrivée au niveau du nœud cible, 0.5 signifie que le lien descend/monte à l’abscisse située au milieu des nœuds source et cible.
* x\_sd\_label, // position du label intervalle de confiance
* y\_sd\_label

Les nœuds et liens filtrés sont stockés respectivement dans les variables « filtered\_nodes » et « filtered\_links ».

## 3) Divers

--- DOSSIERS SOURCES/USER\_SANKEY ---

- auto\_diagrams : contient le fichier javascript généré automatiquement à partir des csv uploadés,

- diagram\_uploads : contient les diagrammes (txt) chargés par les utilisateurs et les fichiers javascript du même nom générés à partir de ceux-ci,

- downloaded\_diagrams : contient les diagrammes sauvegardés (téléchargés par les utilisateurs),

- layout\_uploads : contient les diagrammes (txt) chargés par les utilisateurs pour servir de layout à de nouvelles données,

- supply\_uploads : contient les fichiers supply (csv) chargés par les utilisateurs,

- use\_uploads : contient les fichiers use (csv) chargés par les utilisateurs.

--- FONCTIONNEMENT DE LA SAUVEGARDE DES DIAGRAMMES ---

- la pages web sankey.php contient un modal "sauvegarder" avec un bouton d'id "save\_layout\_button",

- le script manual\_sankey.js observe ce bouton et déclenche onclick la fonction save\_layout, puis save\_links, puis save\_filtered, puis save\_env qui postent des infos au serveur (ajax) et exécutent les scripts php du même nom.

--- FONCTIONNEMENT DU CHARGEMENT D’UN DIAGRAMME ---

- l’utilisateur charge un diagramme txt qui est copié dans le dossier ‘diagram\_uploads’ avec l’extension ‘.txt’ et un nom aléatoire,

- le path de ce fichier est contenu dans la variable de session $\_SESSION[“file\_path”],

- le fichier txt est traduit en fichier js avec le fichier ‘diagram\_txt\_to\_js.php’ qui fait appel aux fonctions de ‘functions\_txt\_to\_js.php’,

- $\_SESSION[“path”] passe d’une extension ‘.txt’ à une extension ‘.js’,

- s’il n’y a pas d’erreur, le navigateur est redirigé vers ‘sankey.php’ avec la variable $\_SESSION[“sankey\_type”] = “saved\_diagram” et il va ajouter le lien vers le fichier $\_SESSION[“file\_path”].