

## Technical Documentation

### mapXP

Version 1.5

October 2022

© Daniel M. Ringel

### About

- mapXP is a java script based interactive map tool that can visualize relationship structures among objects in two-dimensional space and augment map objects with additional attributes using color, size, connecting edges/arcs as well as pop-ups.
- mapXP runs within common web browsers such as Safari, Chrome and Firefox.
- mapXP needs to be hosted on a webserver (or equivalent localhost)
- mapXP neither estimate and projects maps, nor does mapXP cluster objects. These steps need to be done before with whatever analysis and data you choose.

### Disclaimer

I do not warrant mapXP and any part of its functionality and code in any way. There is no guarantee that mapXP or its code is error-free. I do not offer technical support for mapXP. Use mapXP at your own risk and liability. By using mapXP, you agree to assume full liability for any damages that may result from its use. This technical documentation is intended to help you run mapXP on your own data. It is not a full documentation of the entire code, nor is it intended to serve as such.

### Hosting

Copy all files and folders onto your webspace. Make sure to initially keep file and folder names unchanged.

The following folders should be in your webspace folder that is connected to your domain:

- "images" (contains image files for objects that appear in pop-ups when linked in data file "network.json")
- "instructions" (contains these instructions)
- "meta" (contains "metadata.json" file that specifies which object attributes are shown and how)
- "network" (contains the "network.json" file in which object coordinates, attributes and relationships are stored)

The following files should be in your webspace folder that is connected to your domain:

- "index.html" (main page that loads SVG container in which maps are generated)
- "networkchart.css" (CSS file that specifies design and layout)
- "networkchart.js" (main code of mapXP that generates map and functionality)

## Data

mapXP loads a JSON format data file (`"/network/network.json"`) that includes information on nodes (i.e., objects) and links (i.e., relationships among nodes and their weights or strengths).

Nodes must include a unique `"node_index"`, a node `"name"`, as well as `"x"` and `"y"` coordinates on a two-dimensional plane. In addition, you can include any number of attributes. Make sure that each attribute is included in the information for each node.

Links must include a running `"link_index"`, a `"source"` node ID and `"target"` node ID that each correspond to a `"node_index"` from the Node list, as well as a `"weight"` that represents the relationship strength between the source and target node. You do not need to specify a link between every node. In particular, you should leave out nodes with a relationship weight of 0. You might also want to truncate all relationship strengths below a certain threshold to reduce the amount of data that needs to be loaded and visualized. Beware that many links quickly overload the capacity of a user's browser and system to display a map smoothly. Links can be asymmetric. That is, you can specify a different `"weight"` from node 1 to 2, than from node 2 to 1. If you have asymmetric relationships, make sure to set the property `"links"` in the `"metadata.json"` file to `"arcs"`. Else, it should be either `"edges"` (i.e., straight lines) or `"curvededges"` for symmetric bent lines (i.e., arcs without arrowheads).

The basic structure of `"network.json"` for the numerical example at [www.mapxp.app/mapxp](http://www.mapxp.app/mapxp) is as follows:

```
{
  "nodes":
  [
    {"node_index":0,"name":"1. SkewB","x":189,"y":150,"cluster":1,
    "share":8.22,"brand":"Meffert","image":"product_1.gif","price":8.99,"Memberships":"Single Submarket"},
    {"node_index":1,"name":"2. SkewB v2","x":-189,"y":-150,"cluster":1,
    "share":8.22,"brand":"Meffert","image":"product_2.gif","price":7.99,"Memberships":"Single Submarket"},
    .
    .
    .
    {"node_index":9,"name":"9. Speed Cube","x":-1034,"y":-530,"cluster":3,
    "share":12.33,"brand":"D-FantiX","image":"product_9.gif","price":5.5,"Memberships":"Single Submarket"}
  ],

  "links":
  [
    {"link_index":0,"source":0,"target":1,"weight":0.759},
    .
    .
    .
    {"link_index":9,"source":1,"target":9,"weight":0.204},
    {"link_index":45,"source":9,"target":8,"weight":0.311}
  ]
}
```

In addition to `"node_index"`, `"name"`, `"x"`, and `"y"`, each node has additional attributes in the numerical example. These are simply added to each node and can be numerical or text. They can be continuous variables, integers, characters, or strings. They can also refer to image files that mapXP is to load into its pop-up window when users mouse-over an object on the map. It is important that all nodes have the same additional attributes.

It is straight forward to automatically generate such a JSON data file from the results of your analysis (e.g., projecting a dimensionality reduced vectors onto a two-dimensional plane using t-SNE or MDS) on most platforms such as MATLAB, Python or R. mapXP is only a front-end to your analysis results that enables you and others to easily visually explore the generated maps that can be augmented with additional attributes. Any map projection (e.g., with t-SNE) or clustering (e.g., with community detection or label propagation) needs to be done within your analysis (code) on your computer.

### Metadata

The file `"metadata.json"` inside the `"meta"` folder specifies how mapXP visualizes objects and their attributes. The filename and path to the `"metadata.json"` file is specified inside the `"networkchart.js"` file around line 167, where you can change it, if need be.

You must list and configure all attributes that appear with your objects (nodes) in the `"nodes"` section of the `"network.json"` file in the `"metadata.json"` file.

Each attribute must have the following properties:

- `"variable"`: the original variable name exactly(!) as it appears in the `"network.json"` data file
- `"variableName"`: what you would like the variable to appear as in mapXP
- `"type"`: the type of variable. Options are `"categorical"`, `"quantitative"`.
- `"colorArray"`: required for categorical variables only. Accepts 9 different built-in color arrays named `"auto1"` to `"auto9"`. Or you set your own array as done in the numerical example to make the map colors mode accessible to the colorblind. Note that quantitative variables will automatically be visualized as a heatmap whose colors are specified in the `"networkchart.js"` file as `"divergingcolors"` around line 44
- `"colorOrder"`: required for each quantitative variable. Defines which way the heatmap is generated (high values either dark red or dark blue). Takes values `"ascending"` or `"descending"`.
- `"inColorDropdown"`: required for each quantitative variable. Controls which quantitative variable appears in the color-dropdown in mapXP such that it can be visualized as heatmap. Accepts `"true"` or `"false"`.

The following attribute properties are optional:

- `"currency"` [optional]: specify a currency symbol or letters such as \$ or USD before a number (for a currency attribute such as price). Possible for quantitative and categorical variables (for maximum flexibility). If you want to add a space in the currency, put the

space in the string itself. For instance, `{"currency": "$ "}`, with a space after the \$, will create a space between the \$ and the number.

- `"unit"` [optional]: Behaves exactly like `"currency"` except that it appears after the number. Note that you can have currency and unit simultaneously (makes little sense to me, but who knows!).
- `"tooltip"` [optional]: controls whether the attribute and its value appear in the tooltip pop-up on the map when the user mouses-over an object. Set to `"yes"` or `"no"`.
- `"tooltipOrder"` [conditionally optional]: Must include if you set `"tooltip"` to `"yes"`. Defines in which order attributes appear in the tooltips on the map. Number starts at 1 (first attribute to appear at the top of the tooltip).

In addition to the attribute properties, you can also control how mapXP initially presents your data when it is loaded. The `"features"` section of the `"metadata.json"` files includes the following properties that control mapXP behavior:

- `"sameSize"`: defines how big bubbles that represent objects are when the user selects `"Same Size"` from the size drop-down.
- `"links"`: set to either `"edges"` (straight lines) or `"curvededges"` (arcs for symmetric relationships) or `"arcs"` (arcs with arrowheads for asymmetric relationships).
- `"fileName"`: specifies the path and name of your data file `"network.json"`.
- `"selectedColor"`: Defines which attribute is pre-set for bubble color. Must use the exact same spelling as specified in the `"network.json"` data file. You can specify `"samecolor"` if you want all bubbles to initially appear in grey.
- `"selectedSize"`: Defines which attribute is pre-set for bubble size. Must use the exact same spelling as specified in the `"network.json"` data file. You can specify `"samesize"` if you want all bubbles to initially appear in equal sizes (where the size is specified above with `"sameSize"`).
- `"sameColorPosition"`: Defines where the option `"Same Color"` appears in the `"Color"` drop-down. Set to 0 for the top position.
- `"sameSizePosition"`: Defines where the option `"Same Size"` appears in the `"Size"` drop-down. Set to 0 for the top position.
- `"showLabels"`: set to `"true"` or `"false"`. Controls `"Show Names"` checkbox on initial load of mapXP.
- `"sizeLabels"`: set to `"true"` or `"false"`. Controls `"Adaptive Name Size"` checkbox on initial load of mapXP.
- `"showTooltip"`: set to `"true"` or `"false"`. Controls `"Attribute Pop-up"` checkbox on initial load of mapXP.
- `"zoomLevel"`: set from 0 to 1.0. Controls zoom level on initial load of mapXP.
- `"pathsPercentage"`: set from 0 to 1.0. Controls percent strongest relationships shown as edges/curvededges/arcs on initial load of mapXP.
- `"rotateAngle"`: set to an angle from 0 to 360 degrees. Controls rotation of map on initial load of mapXP.
- `"verticalFlip"`: set to `"true"` or `"false"`. Controls whether map is flipped vertically or not on initial load of mapXP.

For the numerical example of mapXP, the "metadata.json" file looks as follows:

```
[
  {
    "variable": "cluster",
    "variableName": "Submarket",
    "type": "categorical",
    "colorArray": [
      "#882255",
      "#117733",
      "#88CCEE",
      "#DDCC77",
      "#AA4499",
      "#44AA99",
      "#332288",
      "#999933",
      "#CC6677",
      "#DDDDDD",
      "#000000"
    ],
    "tooltip": "no"
  }, {
    "variable": "brand",
    "variableName": "Brand",
    "type": "categorical",
    "colorArray": [
      "#882255",
      "#117733",
      "#88CCEE",
      "#DDCC77",
      "#AA4499",
      "#44AA99",
      "#332288",
      "#999933",
      "#CC6677",
      "#DDDDDD",
      "#000000"
    ],
    "tooltip": "yes",
    "tooltipOrder": 1
  }, {
    "variable": "price",
    "variableName": "Price",
    "type": "quantitative",
    "inColorDropdown": true,
    "colorOrder": "descending",
    "currency": "$",
    "tooltip": "yes",
    "tooltipOrder": 2
  }, {
    "variable": "share",
    "variableName": "Marketshare",
    "type": "quantitative",
    "inColorDropdown": true,
    "colorOrder": "ascending",
    "unit": "%",
  }
]
```

```

        "tooltip": "yes",
        "tooltipOrder": 3
    }, {
        "variable": "Memberships",
        "variableName": "Memberships",
        "type": "categorical",
        "colorArray": [
            "black",
            "rgb(160, 160, 160)"
        ],
        "tooltip": "yes",
        "tooltipOrder": 4
    }, {
        "features": {
            "sameSize": 10,
            "links": "curvededges",
            "fileName": "network/network.json",
            "selectedColor": "cluster",
            "selectedSize": "share",
            "sameColorPosition": 0,
            "sameSizePosition": 0,
            "showLabels": true,
            "sizeLabels": true,
            "showTooltip": true,
            "zoomLevel": 0.8,
            "pathsPercentage": 100,
            "rotateAngle": 143,
            "verticalFlip": false
        }
    }
]

```

## Imported Libraries

mapXP is built upon the D3.js library. A link is provided to this library in the `index.html` file

```
<script src="https://d3js.org/d3.v4.min.js"></script>
```

It also requires the D3 color schemes as specified in `index.html`:

```
<script src="https://d3js.org/d3-scale-chromatic.v1.min.js"></script>
```

Further, mapXP requires jQuery and jQuery UI as specified in the `index.html` file:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.min.js"></script>
```

You might consider downloading and saving these files on your webserver (don't forget to change the paths in the `index.html`). You should have the following files in a new folder

"scripts" that you refer to:

- d3-scale-chromatic.v1.min.js
- d3.v4.min.js
- jquery-ui.css
- jquery-ui.min.css
- jquery-ui.min.js
- jquery.min.js