Web-based data-dependant visualization with NDExEdit

Florian J. Auer [1]

[1]IT-Infrastructure for Translational Medical Research, University of Augsburg

03/30/2022

Contents

1 NDExEdit

NDExEdit allows you to modify a biological network visually in your browser. You don't need to create an account or install further software. All you need is a browser and a network, that conforms to the CX data model. NDExEdit is available at

https://frankkramer-lab.github.io/NDExEdit

NDExEdit is tightly connected to NDEx, a distribution platform specifically designed for biological networks.

One of the most powerful visualization tools in our toolbox are so-called mappings. They were introduced by Cytoscape as a manner of applying a network's properties onto its visualization.

A simple workflow with NDExEdit consists of importing your network file, inspect the data foundation and existing visualization, modifying some visual properties and export the network. Each of these steps is described in detail below.

2 Visualize gene occurances

2.1 Import combined subnetworks

To load a network into NDExEdit there are three options: - UUID: NDEx assigns unique IDs to all of its networks. If you know that, you can just paste it into the import field and get started. - Browse NDEx: The built-in browser allows you to look for a network on NDEx from within NDExEdit. To browse your account specific networks, you need to login. - Local file: If the network you wish to edit is not yet on NDEx, you can upload a local .cx file to work with.

Please note: A network can only be imported, if it conforms to the CX data model and if its size is acceptable.

The network is also available on the NDEx platform as "Combined patient-specific breast cancer subnetworks":

https://www.ndexbio.org/viewer/networks/079f4c66-3b77-11ec-b3be-0ac135e8bacf

We could search the network using the option to brows NDEx or simply load the network by its UUID:

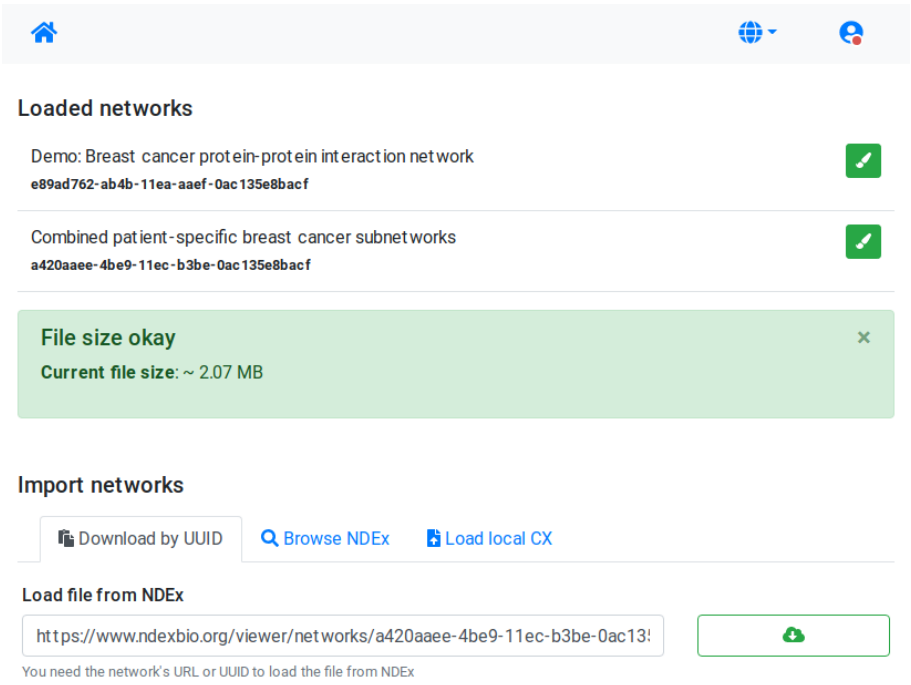


Figure 1: NDExEdit: upload networks

Once the network is available, open the network to edit its visual styles. In the beginning all nodes are cluttered together, because the network does neither contain a visualization nor a layout applied. To achieve the same layout as at the NDEx platform, we can simply apply a circular layout by selecting it from the "Layout graph" option above the graph.

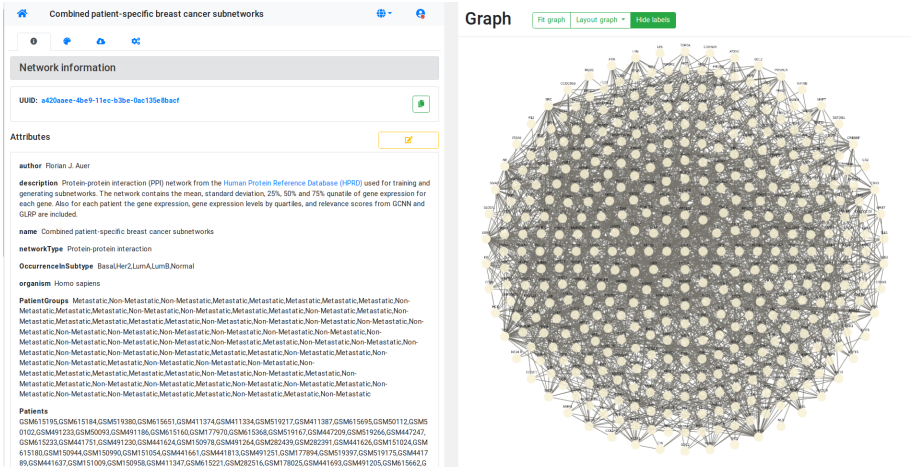


Figure 2: Blank network with circular layout applied

2.2 Create mappings

Firstly create adjust the node labels to be displayed in the center of the nodes instead on to of its. Again, because the network initially does not contain any visual styles, on loading a default style analogous with the visualization on the NDEx platform is automatically loaded. Therefore, some properties are already present, which influence the default visual representation of the nodes and edges. Those default values also apply if attributes used for a mapping are not set for a node or edge (fallback).

To change the default node label position, we switch to the edit tab and click on the yellow edit button right below the node properties to start editing.

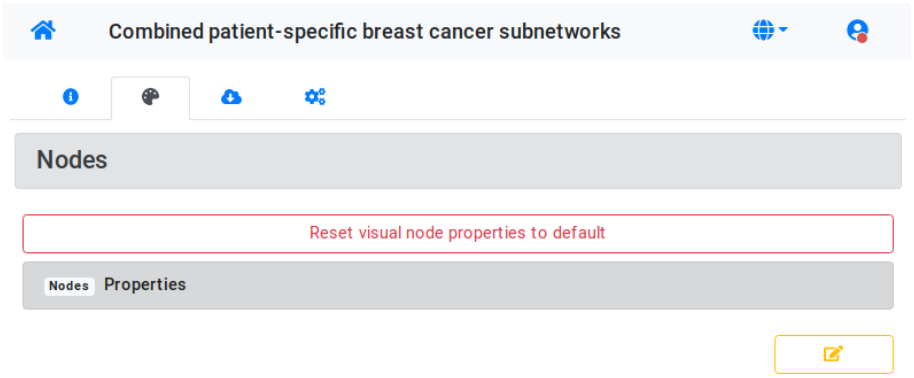


Figure 3: Edit tab with shown node properties subchapter

There is also a button directly below the nodes chapter to reset all set visual properties to the default values.

The entry responsible for the label position is the `NODE_LABEL_POSITION` property. Changing the value to `C,C,c,0.00,0.00` sets the label position with its center (first `C`) to the center of the node (second `C`).

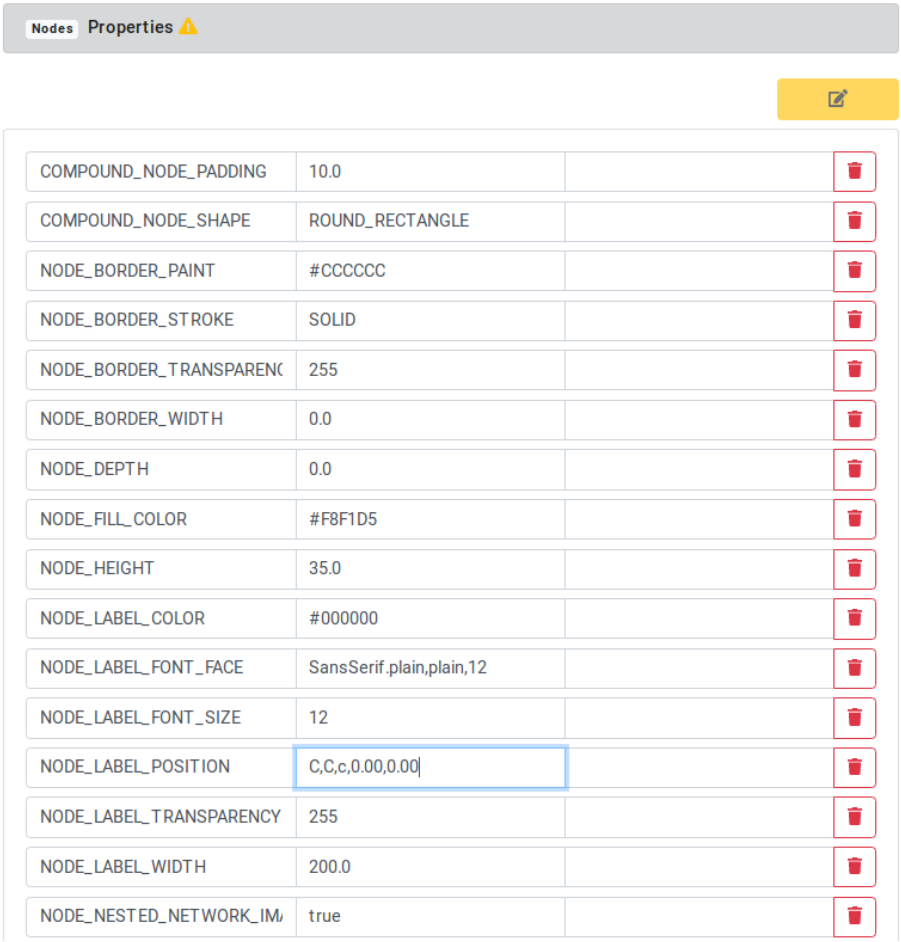


Figure 4: Editing node properties

The yellow symbol next to the node properties subchapter header indicates the properties are in the editing mode, and therefore nothing else can be changed in the meanwhile. The edit mode only can be left with accepting or canceling the edits. This applies to any edits of properties or mappings on the page!

Figure 5: Accepting or cancling the changes

Next we will create a continuous mapping based on the number of occurrences of the genes across the different patients (`Occurrence` node attribute). At the continuous subchapter we start typing the attribute name and select the occurrence, type for example “width” in the style property input box and select the prompted “NODE_WIDTH”, and click on the green plus to start defining the mapping.

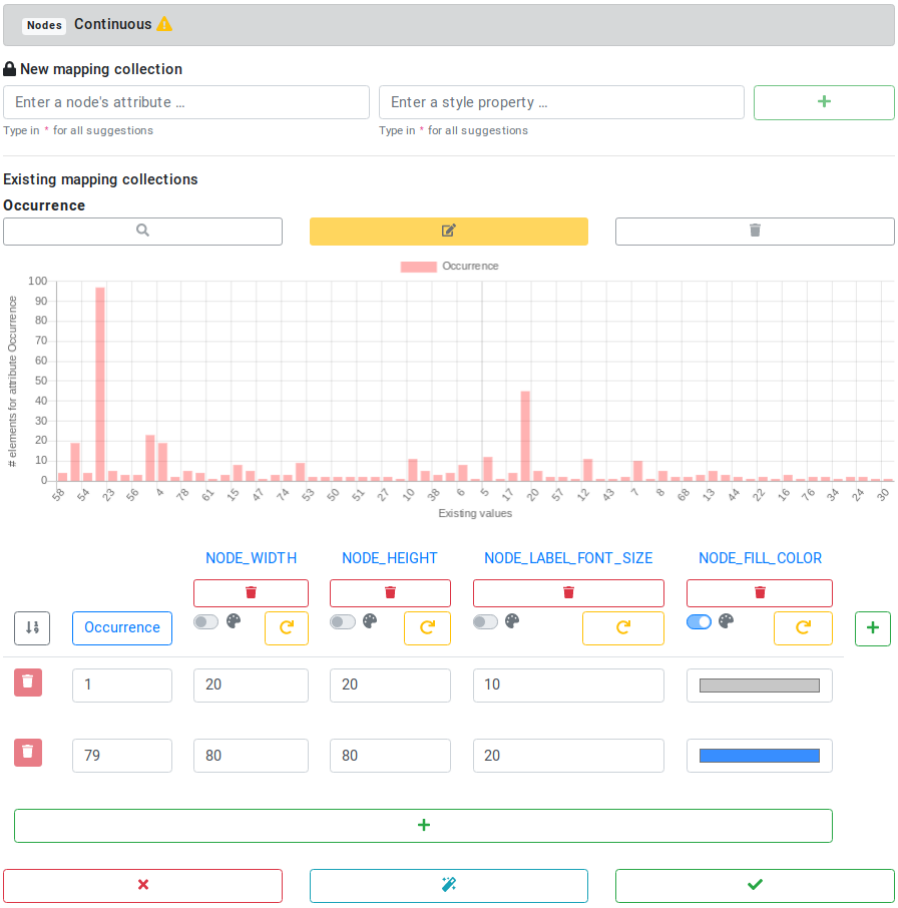


Figure 6: Create an continuous mapping for node color and size, node label size base on gene occurances

The distribution of the occurrence values along the node ids is shown in the histogram above to simplify selection of appropriate values. Since we have 79 patients included, and the node ids start at 1, we define the mapping with these values as limits. After saving the mapping, we can explore the defined mapping, e.g. the created color gradient.

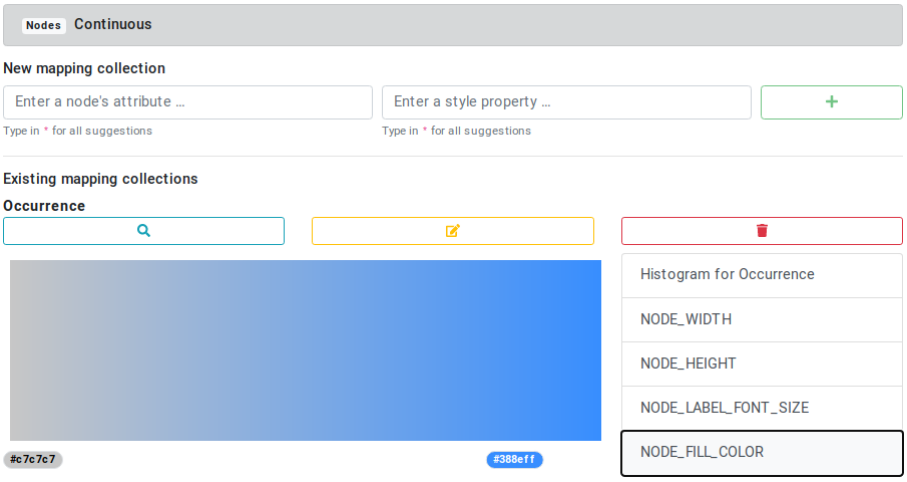


Figure 7: Created continuous mapping for gene occurrences

Also the graph is updated immediately after:

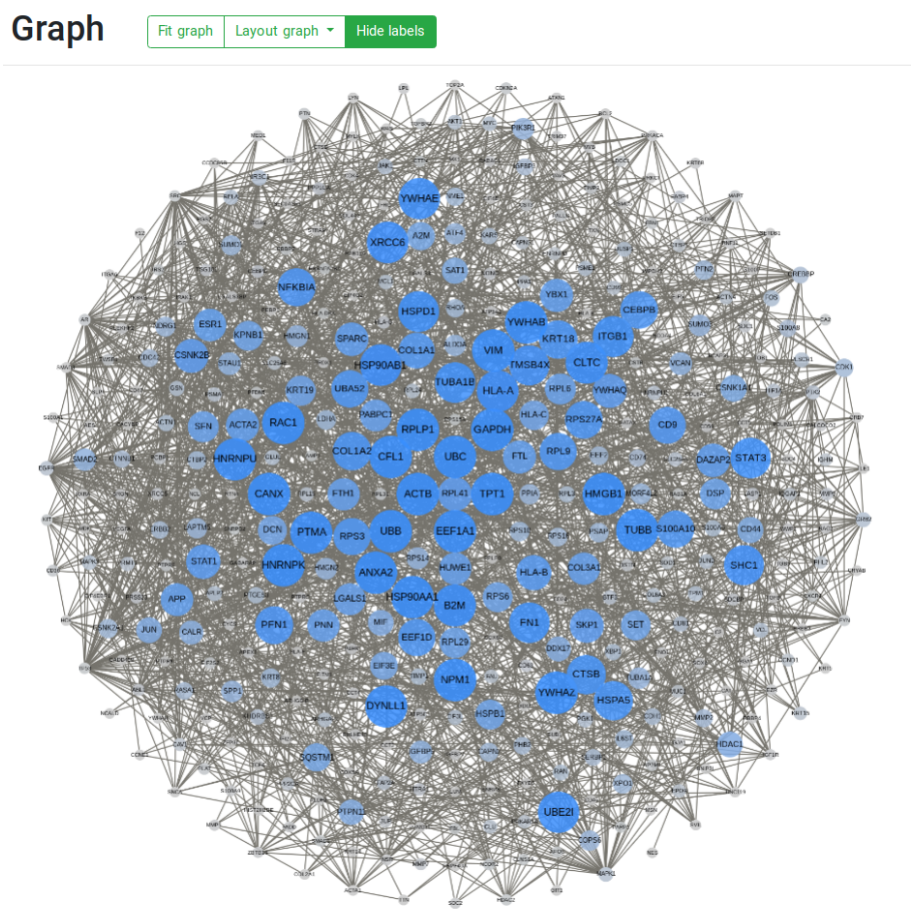


Figure 8: Network visualization based on gene occurences across patients

The network now can be exported in different formats, namely PNG, JPEG, or downloaded as CX file.

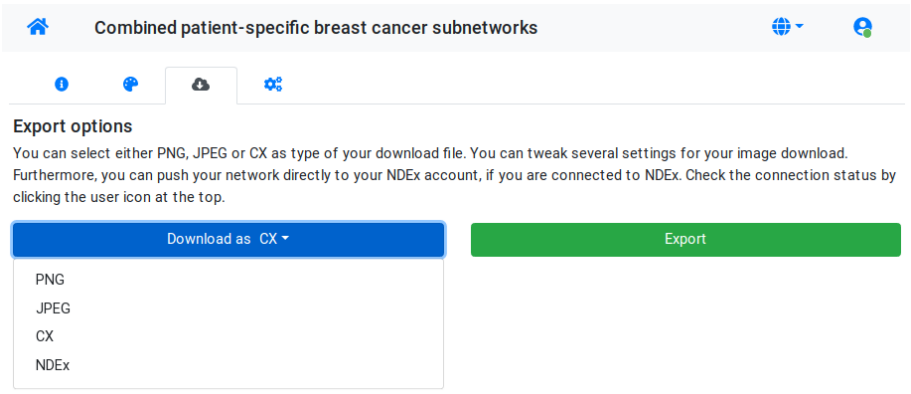


Figure 9: Export options for the network with included visualization

Additionally, the network can directly uploaded the the NDEx platform when logged in. The symbol in the upper right indicates the connection status.

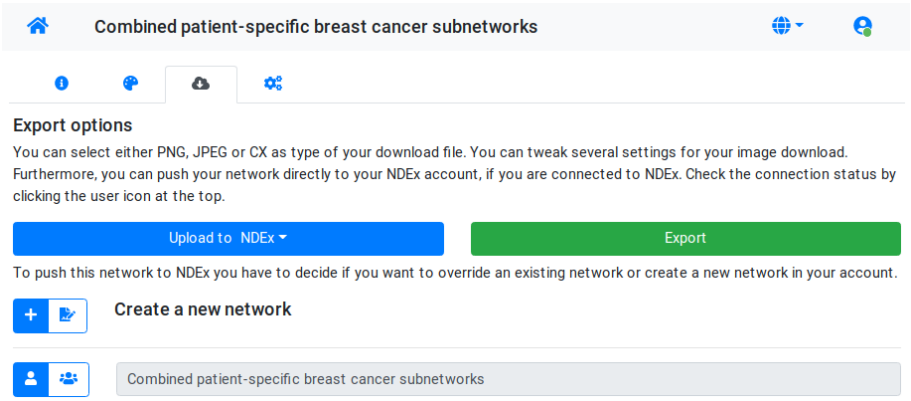


Figure 10: Export the network directly to the NDEx platform

3 Visualize one patient

Here we create a visualization for one patient equivalent to the visualization generated by MetaRelSubNetVis. As sample we choose the metastatic patient GSM615368. We create a visualization based on the relevance score for node and node label size, and gene expression level for node coloring.

First adjust the default node color (property `NODE_FILL_COLOR`) to gray (#888888), and the node label the same as before to the center. Analogously, we set the default edge color (property `EDGE_UNSELECTED_PAINT`) to a light gray (#cccccc).

Now we create a discrete mapping for the gene expression level. The different levels `LOW`, `NORMAL`, and `HIGH` should be mapped to blue, yellow and red.

Also we mark the results of the molecular tumor board analysis for this patient by a green border.

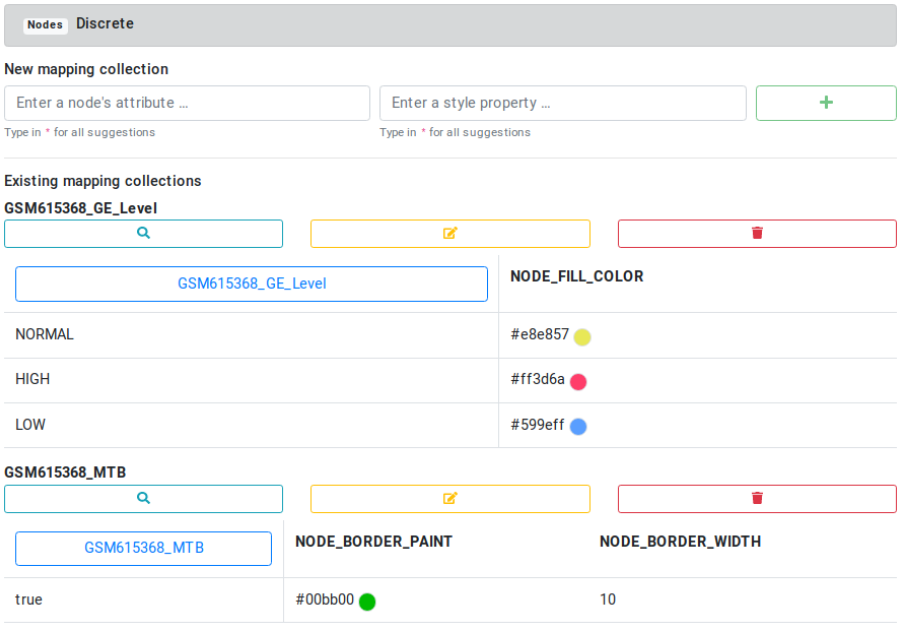


Figure 11: Discrete mapping for the gene expression levels and MTB results

Next create the continuous mapping for the node size. The mapping is limited by the min. value of 0.000298, and the max. value 0.000922 for relevance scores. The values of the relevance scores for this patient are stored in the node attribute `GSM615368_Score`.

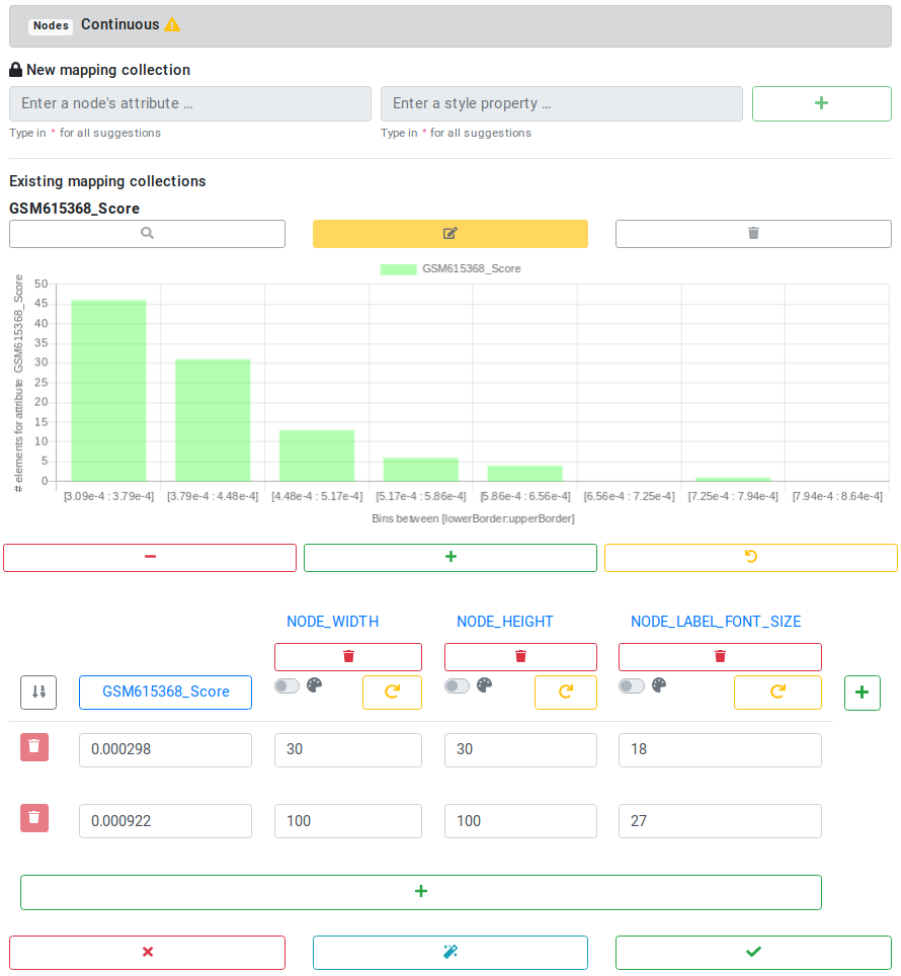


Figure 12: Continous mapping based on the relevance score of patient GSM615368

The finished network looks as follows:

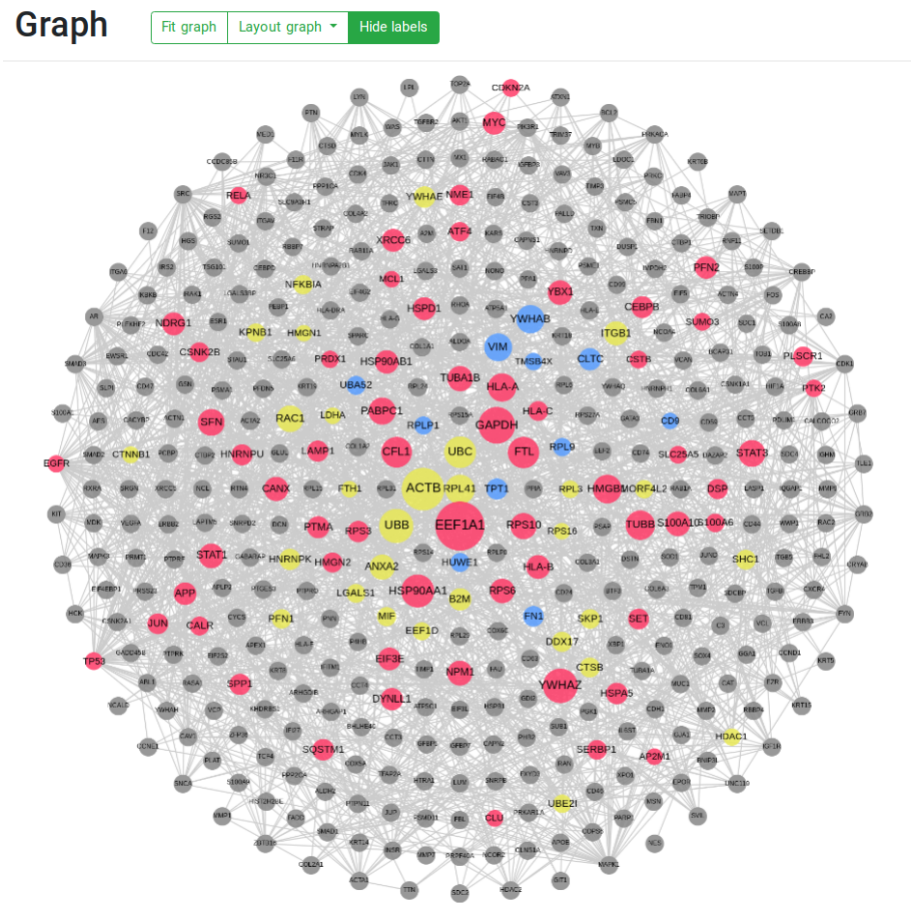
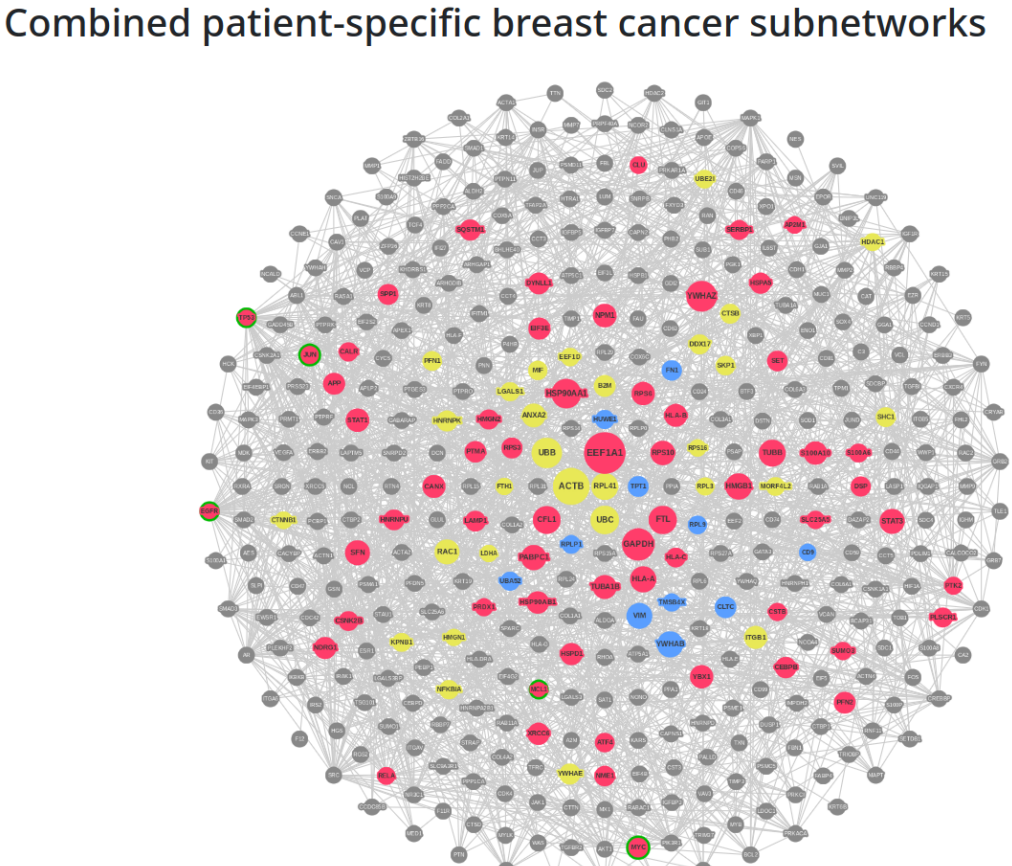


Figure 13: Discrete mapping based on the gene expression level of patient GSM615368 on NDExEdit



The only look almost the same due to some difference in the layout algorithm, which mirrors the nodes at the x axis.

4 Session info

```
sessionInfo()
## R version 4.0.3 (2020-10-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.3 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C               LC_TIME=de_DE.UTF-8        LC_COLLATE=en_US.UTF
##  [6] LC_MESSAGES=en_US.UTF-8    LC_PAPER=de_DE.UTF-8       LC_NAME=C                  LC_ADDRESS=C
## [11] LC_MEASUREMENT=de_DE.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] XML_3.99-0.8        httr_1.4.2          RJSONIO_1.3-1.6     pacman_0.5.1        devtools_2.4.2      u
##  [8] timeDate_3043.102   pander_0.6.4        xtable_1.8-4        stringr_1.4.0       BiocStyle_2.18.1    k
## [15] igraph_1.2.7        gplots_3.1.1        dplyr_1.0.7         RColorBrewer_1.1-2  survival_3.2-7
##
## loaded via a namespace (and not attached):
##  [1] pkgload_1.2.3       jsonlite_1.7.2      splines_4.0.3       gtools_3.9.2        assertthat_0.2.1    b
##  [8] stats4_4.0.3        remotes_2.4.1       yaml_2.2.1          sessioninfo_1.1.1   pillar_1.6.4        t
## [15] digest_0.6.28       htmltools_0.5.2     Matrix_1.2-18       plyr_1.8.6          pkgconfig_2.0.3     b
## [22] webshot_0.5.2       processx_3.5.2      tibble_3.1.5        generics_0.1.1      ellipsis_0.3.2      b
## [29] cachem_1.0.6        BiocGenerics_0.36.1 cli_3.0.1           mime_0.12           magrittr_2.0.1      c
## [36] memoise_2.0.0       evaluate_0.14       fs_1.5.0            fansi_0.5.0         pkgbuild_1.2.0      g
## [43] tools_4.0.3         formatR_1.11        lifecycle_1.0.1     callr_3.7.0         compiler_4.0.3      j
## [50] tinytex_0.34        rlang_0.4.12        grid_4.0.3          htmlwidgets_1.5.4   crosstalk_1.1.1     k
## [57] testthat_3.1.0      DBI_1.1.1           curl_4.3.2          markdown_1.1        R6_2.5.1            k
## [64] utf8_1.2.2          rprojroot_2.0.2     desc_1.4.0          KernSmooth_2.23-17  stringi_1.7.5       p
## [71] vctrs_0.3.8         tidyselect_1.1.1    xfun_0.27
```