**NgchmEmbed Javascript Library Reference**

**Overview:**

The ngchmEmbed Javascript API is a small minimized (ngchmEmbed-min.js) javascript library for embedding NG-CHM (.ngchm) heat map files into an html page. It enables two types of heat map embedding: embedding a static ngchm and embedding an expandable ngchm.

A static ngchm is a fully expanded heat map embedded in an html page at the location of its container DIV. An expandable ngchm is a heat map that is represented by a thumbnail image that can be clicked on to expand to a full-sized heat map.

The library contains the following function for both types of ngchm embedding: **embedNGCHM**

Using the above function, the user can embed NG-CHM heat maps into their own HTML pages. The embedded heat maps will be contained within an iFrame that is configured inside of a DIV element that the user has placed on their page. Any JS logic required by the embedded map will be contained within that iFrame so as not to override any other JS libraries defined on the parent html page.

**HTML Page Setup:**

There are 3 requirements to using the ngchmEmbed library to place NG-CHMs in an HTML page:

1. Including the ngchmEmbed-min.js script into the page. (this is usually done at the top of the page and must precede items 2 and 3 on the page)
2. Placing a DIV element on the page for the display of the NG-CHM. (this must precede item 3 on the page).
3. Placing a JS script block with calls to ngchmEmbed functions. (this can be done at the end of the page or anywhere after the placement of the DIV from item 2 above).

**Example of simple html page for embedding NG-CHMs:**

This example shows how to display NG-CHMs for each of the source types available. The first two calls to embedNGCHM display the same NG-CHM heat map two different ways, one standard and one expandable (using the filename srcType). The second also shows the use of a query selector as the first parameter entry for the embedNGCHM call. The remaining show how to use the srcType values: url, blob, and base64. In the case of the latter two, a variable name would be provided for the blob or base64 variable being passed into the function.

<HTML>

<HEAD>

<script type=*"text/Javascript"* src=*"./ngchmEmbed-min.js"*></script>;

</HEAD>

<BODY>

<DIV id=*'ngChmDiv1'*></DIV>

<DIV id=*'ngChmDiv2'*></DIV>

<DIV id=*'ngChmDiv3'*></DIV>

<DIV id=*'ngChmDiv4'*></DIV>

<DIV id=*'ngChmDiv5'*></DIV>

<script type=*"module"*>

embedNGCHM ('ngChmDiv1', 'fileName', 'tcga\_rnaseq\_blca\_v1.0\_gene\_gene.ngchm'});

embedNGCHM ('#ngChmDiv2', 'fileName', 'tcga\_rnaseq\_blca\_v1.0\_gene\_gene.ngchm',{expandable: true});

embedNGCHM ('ngChmDiv3', 'url', 'http://www.anysite.com/test.ngchm');

embedNGCHM ('ngChmDiv4', 'blob', blobVar);

embedNGCHM ('ngChmDiv5', 'base64', bas64Var);

</script>

</BODY>

</HTML>

**Function: embedNGCHM**

**embedNGCHM (selector, srcType, srcSpec, params = {})**

**Required Parameters:**

**selector** – The selector can be one of three types of parameter values:

* A string parameter containing the id of the DIV in which to display the NG-CHM.
* A true query selector (e.g. #divName) for the DIV.
* The DOM Element for the DIV.

**srcType** - A string parameter containing source type for the provided SrcSpec

**Valid SrcTypes:**

* **fileName** – Specifies that the name (or path/name) of an .ngchm file will be provided as the SrcSpec.
* **base64** – Specifies that a base64 string will be provided as the SrcSpec.
* **blob** – Specifies that a blob object will be provided as the SrcSpec.
* **url** – Specifies that a URL path to an .ngchm will be provided as the SrcSpec.

**srcSpec** – A string parameter containing the file name, variable name or string for a base64 string, or a variable name for a blob object.

**Note on providing a fileName as SrcType:** .ngchm files provided to this function must be located on the web server in the web root directory or a directory below web root. If the .ngchm is located in web root, just the file name (e.g. *‘test\_map.ngchm’*) need be provided. If in a subdirectory, then a path and file name (e.g. *‘./subdir1/test\_map.ngchm’*) are to be provided.

**Optional Parameter:**

**params** – The ngchmEmbed library provides the default parameters necessary for embedding NG-CHM heat maps in your document. Using this optional parameter the user may override some or all of those defaults. The params parameter accepts a Javascript object in the format parameter: value (e.g. *{expandable: true, style: "height:100vh; width:100vw; border-style:none;"}*).

**Default Parameters specific to both types of embedded heat maps:**

Each of the below default parameters may be overridden by utilizing the *params* parameter in the *embedNGCHM* function request.

* **widgetPath: 'ngchmWidget-min.js'** **-** The path to ngchmWidget-min.js file. If in servers top level web content directory, just use file name to path to another sub-directory within web content: e.g. *'./subdir/ngchmWidget-min.js'* to path to a URL location e.g. 'http://widgetloc.com/ngchmWidget-1.2.0-min.js' (one we have tested with: 'http://cloudflare.com/..../ngchmWidget-1.2.0-min.js').
* **scrolling: ‘no’ -** Optional configuration for scrolling within the iFrame containing the embedded heat map.
* **style: - "height:100vh; width:100vw; border-style:none;" -** Style configuration for the iFrame containing the embedded heat map.
* **sandbox: - 'allow-scripts allow-same-origin allow-popups allow-forms allow-modals allow-downloads' –** Sandbox parameters to be applied to the iFrame containing the NG-CHM. Without these sandbox values applied, some functionality in the embedded NG-CHM will not be available.
* **docStyle: 'margin: 1em 0 0 5%; width: 90%;'** **-** Style of the top-level DIV within the embedded iFrame.
* **embedStyle: 'display: flex; flex-direction: column; background-color: white; height: 80vh; border: 2px solid gray; padding: 5px'** - Style of the second-level div that contains the embedded NG-CHM
* **customJS: ‘’ –** The path to an additional custom.js file. If in servers top level web content directory, just use file name to path to another sub-directory within web content: e.g. './subdir/anotherCustom.js' to path to a URL location e.g. *'http://widgetloc.com/ anotherCustom js'*.
* **expandable: false -** This parameter would be set to ***true*** for displaying expandable NG-CHM heat maps. Expandable embedded NG-CHMs differ from regular embedded NG-CHMs in that the viewer is first provided with a thumbnail image for the NG-CHM that they may click on to expand the NG-CHM to full size. A “collapse” button is provided in expanded mode to allow the user to collapse the NG-CHM back to the thumbnail size.

**Default Parameters specific only to expandable embedded heat maps:**

* **thumbStyle: ‘height: 150px; width: 150px; border-style:none;'** **–** The display style for the thumbnail on the html page.
* **thumbnail: A base64 image of a standard heat map thumbnail image –** The file name for the thumbnail image to be displayed. A file name (following the same location and pathing rules as the .ngchm file name) or a base64 image for a thumbnail image may be provided as an override. The user is encouraged to use a thumbnail that matches the map to be displayed. The NG-CHM Viewer application contains a feature for creating thumbnail .png files from an open NG-CHM.