visNetwork

Load library

```
library(igraph)
library(visNetwork)
library(networkD3)
library(dplyr)
```

Graph object

```
df = read.table("./data/data.tsv", header = T)
veccol = c(rep("pink",5), rep("light blue",6))
g = graph_from_data_frame(df)
net = g
```

Create nodes and edges data frames

visNetwork needs at least two pieces of information :

- a nodes data.frame, with id column
- an edges data.frame, with from and to columns, which make the link with id

```
nodes = data.frame(id = V(g)$name)
nodes$label = nodes$id
edges = data.frame(get.edgelist(g))
colnames(edges) = c('from','to')
head(nodes)
```

```
##     id label
## 1 POX1  POX1
## 2 FAA1  FAA1
## 3 TGL3  TGL3
## 4 TGL4  TGL4
## 5 FAA4  FAA4
## 6 FAS1  FAS1
```

```
head(edges)
```

```
##   from   to
## 1 POX1 FAA2
## 2 FAA1 POX1
## 3 TGL3 YJU3
## 4 TGL4 YJU3
## 5 TGL3 TGL4
## 6 FAA4 POX1
```

Visualize

```
visNetwork(nodes = nodes, edges = edges)
```

visNodes

Shape, title, borderWidth

```
visNetwork(nodes, edges) %>% visNodes(color = "red",shape = "square", title = "Gene Fusion Network", bo
```

Change color properties with multiple parameters

```
visNetwork(nodes, edges) %>%
  visNodes(color = list(
    background = "red",
    border = "blue",
    highlight = "yellow"))
```

Add hover property

```
visNetwork(nodes, edges) %>%
  visNodes(color = list(hover = "green")) %>%
  visInteraction(hover = TRUE)
```

Add shadow

```
visNetwork(nodes, edges) %>% visNodes(shadow = TRUE)
```

```
visNetwork(nodes, edges) %>% visNodes(shadow = list(enabled = TRUE, size = 50))
```

visEdges

Arrows

```
visNetwork(nodes, edges) %>% visEdges(arrows = 'from')
```

```
visNetwork(nodes, edges) %>% visEdges(arrows = 'to')
```

```
visNetwork(nodes, edges) %>% visEdges(arrows = 'middle')
```

```
visNetwork(nodes, edges) %>% visEdges(arrows = 'to, from')
```

```
visNetwork(nodes, edges) %>% visEdges(arrows = 'to, from, middle')
```

Smooth

Boolean | named list. Default to named list. When true, the edge is drawn as a dynamic quadratic bezier curve. The drawing of these curves takes longer than that of straight curves but it looks better.

- "enabled" : Boolean. Default to true. Toggle smooth curves on and off. This is an optional option. If any of the other properties in this object are set, this option will be set to true.
- "type" : String. Default to 'dynamic'. Possible options: 'dynamic', 'continuous', 'discrete', 'diagonalCross', 'straightCross', 'horizontal', 'vertical', 'curvedCW', 'curvedCCW', 'cubicBezier'.
- "roundness" : Number. Default to 0.5. Accepted range: 0 .. 1.0. This parameter tweaks the roundness of the smooth curves for all types EXCEPT dynamic.
- "forceDirection" : String or Boolean. Default to false. Accepted options: ['horizontal', 'vertical', 'none']. This options is only used with the cubicBezier curves. When true, horizontal is chosen, when false, the direction that is larger (x distance between nodes vs y distance between nodes) is used. If the x distance is larger, horizontal. This is ment to be used with hierarchical layouts.

```
visNetwork(nodes, edges) %>% visEdges(smooth = FALSE)
```

```
visNetwork(nodes, edges) %>%
  visEdges(smooth =
             list(
               enabled = TRUE,
               type = "diagonalCross"))
```

width

```
visNetwork(nodes, edges) %>% visEdges(width = 10)
```

color edges

```
visNetwork(nodes, edges) %>% visEdges(color = "red")
visNetwork(nodes, edges) %>% visEdges(color = list(color = "red", highlight = "yellow"))
```

```
visNetwork(nodes, edges) %>% visEdges(color = list(hover = "green")) %>%
 visInteraction(hover = TRUE)
```

shadow

```
visNetwork(nodes, edges) %>% visEdges(shadow = TRUE)
```

```
visNetwork(nodes, edges) %>% visEdges(shadow = list(enabled = TRUE, size = 5))
```

dashes

globally

```
visNetwork(nodes, edges) %>% visEdges(dashes = TRUE)
```

Group nodes

```
nodes$group = c(rep("a",6),rep("b",5))
head(nodes)
```

```
##     id label group
## 1 POX1  POX1     a
## 2 FAA1  FAA1     a
## 3 TGL3  TGL3     a
## 4 TGL4  TGL4     a
## 5 FAA4  FAA4     a
## 6 FAS1  FAS1     a
```

Color nodes by groups

```
visNetwork(nodes = nodes, edges = edges) %>% visOptions(selectedBy = "group")
```

Collapse / Uncollapse Nodes

```
visNetwork(nodes = nodes, edges = edges) %>% visOptions(collapse = TRUE)
```

Highlight nearest

```
visNetwork(nodes = nodes, edges = edges) %>% visOptions(highlightNearest = TRUE)
```

```
visNetwork(nodes = nodes, edges = edges) %>% visOptions(highlightNearest = list(enabled = T, degree = 2
```

Legends

```
# default, on group
visNetwork(nodes, edges, width = "100%") %>%
  visGroups(groupname = "a", color = "red") %>%
  visGroups(groupname = "b", color = "gold") %>%
  visLegend()
```

Placement and titles

```
visNetwork(nodes, edges, width = "100%") %>%
  visGroups(groupname = "a", color = "red") %>%
  visGroups(groupname = "b", color = "gold") %>%
visLegend(width = 0.1, position = "right", main = "Group")
```

Another example

```
nodes1 <- jsonlite::fromJSON("https://raw.githubusercontent.com/datastorm-open/datastorm-open.github.io

edges1 <- jsonlite::fromJSON("https://raw.githubusercontent.com/datastorm-open/datastorm-open.github.io


visNetwork(nodes1, edges1, height = "700px", width = "100%") %>%
  visOptions(selectedBy = "group",
             highlightNearest = TRUE,
             nodesIdSelection = TRUE) %>%
  visPhysics(stabilization = FALSE)
```