Chemviz vignette

Julia Gustavsen

August 16, 2016

Contents

1 Introduction

Scientists can have information about chemical compounds and then want a simple way to visualize the chemical structure and also the to understand the chemical similarity of compounds (Tanimoto similarity). This can be done in Cytoscape using the Chemviz plugin

In this vignette we will use the R package rcellminer to generate some data that we will use to look a some drug compounds. Then, in Cytoscape, we will then see how these drugs are chemically related, look at the 2-dimensional structures of these compounds and then colour the compounds by their mechanism of action to see if it is related to their similarity.

2 Required materials:

- Cytoscape (www.cytoscape.org)
- Chemviz - plugin for Cytoscape (http://www.cgl.ucsf.edu/cytoscape/chemViz2/index.shtml)
- Rpackages: RCy3,rcellminer, RColorBrewer

3 Load the packages

```
library(RCy3)
```

```
## Warning: package 'RCy3' was built under R version 3.5.2
```

```
library(rcellminer)
```

```
## Warning: package 'rJava' was built under R version 3.5.2
```

```
library(RColorBrewer)
source("./functions_to_add_to_RCy3/working_with_namespaces.R")
```

```
## Warning: package 'httr' was built under R version 3.5.2
```

```
## Warning: package 'RJSONIO' was built under R version 3.5.2
```

4 Get data from rcellminer

```
df <- getFeatureAnnot(rcellminerData::drugData)[["drug"]]

moaToCompounds <- getMoaToCompounds()
Moa_names <- names(moaToCompounds)

df_knownMoaDrugs <- subset(df, MOA %in% Moa_names)

## use only those with greater than 10 experiments
df_with_knownMoaDrugs <- subset(df_knownMoaDrugs, TOTAL_EXPS > 10)
```

For display purposes we will chop off long names after 12 characters.

```
long_names <- df_with_knownMoaDrugs$NAME[nchar(df_with_knownMoaDrugs$NAME) > 12]
chopped_long_names <- gsub("^(.{12})(.*)$", "\\1", long_names)
df_with_knownMoaDrugs$NAME[nchar(df_with_knownMoaDrugs$NAME) > 12 ] <- chopped_long_names
```

5 Set up connection between R and Cytoscape

```
# first, delete existing windows to save memory:
commandsHelp("help chemviz")
```

```
## [1] "Available commands for 'chemviz':"
```

```
##  [1] "calculate mcss"           "create attributes"
##  [3] "create similarity"        "hide results"
##  [5] "paint structures"         "remove structures"
##  [7] "search"                   "settings"
##  [9] "show compound structures" "show compound table"
## [11] "show results"
```

6 Format dataframe from RCellMiner to format that can be sent to Cytoscape

```
g <- new("graphNEL",
         nodes = df_with_knownMoaDrugs$NAME,
```

```r
        edgemode = "undirected")
createNetworkFromGraph(g)
```

```
## Loading data...
## Applying default style...
## Applying preferred layout...
```

```
## networkSUID
##          52
```

```r
nodeData<- subset(df_with_knownMoaDrugs, select=c("NAME", "SMILES", "NSC", "MOA"))
colnames(nodeData)[1] <- "id"

loadTableData(nodeData, 'id')
```

```
## [1] "Success: Data loaded in defaultnode table"
```

Network successfully sent to Cytoscape. The node attributes have also been sent.

```
##                                                                                                     file
## "/Users/juliagustavsen/Documents/random_projects/Rcy3/gsoc_Rcy3_vignettes/nodes_for_chemviz.png"
```

Mitoxantrone	Aphidicoline	Amonafide	Buthionine s	He
Morpholinodo	Anthrapyrazo	Cyanomorphol	Zalcitabine	Hali
6-Mercaptopu	Colchicine	Actinomycin	Thiotepa	Tes
Vincristine	Beta-Thiogua	A-TGdR	Fluorodopan	Dau
Yoshi 864	Azacitidine	5-HP	Teniposide	Dox
Maytansine	Acivicin	Rubidazone	Asaley	Dex
M-AMSA	CHIP	Dihydro-5-az	Deoxydoxorub	"N,N

7 Create similarity network

Next based on the chemical properties of each node (which are associated using the SMILES strings (Simplified Molecular Input Line Entry System which are line representations of chemical structures) we will use chemviz to calculate the similarity of each drug based on its chemical properties (using Tanimoto similarity which is a distance-based measure of chemical similarity).

To begin let's look at some of the commands available to use in chemviz:

```
commandsHelp("help chemviz")
```

```
## [1] "Available commands for 'chemviz':"
```

```
##  [1] "calculate mcss"           "create attributes"
##  [3] "create similarity"        "hide results"
##  [5] "paint structures"         "remove structures"
##  [7] "search"                   "settings"
##  [9] "show compound structures" "show compound table"
## [11] "show results"
```

Now let's look at the arguments we can use for creating a similarity network:

```
commandsHelp("chemviz/create%20similarity")
```

```
## [1] "Available arguments for 'chemviz create similarity':"
```

```
## [1] "createNewNetwork" "network"          "nodeList"
```

Set the properties that we would like to use

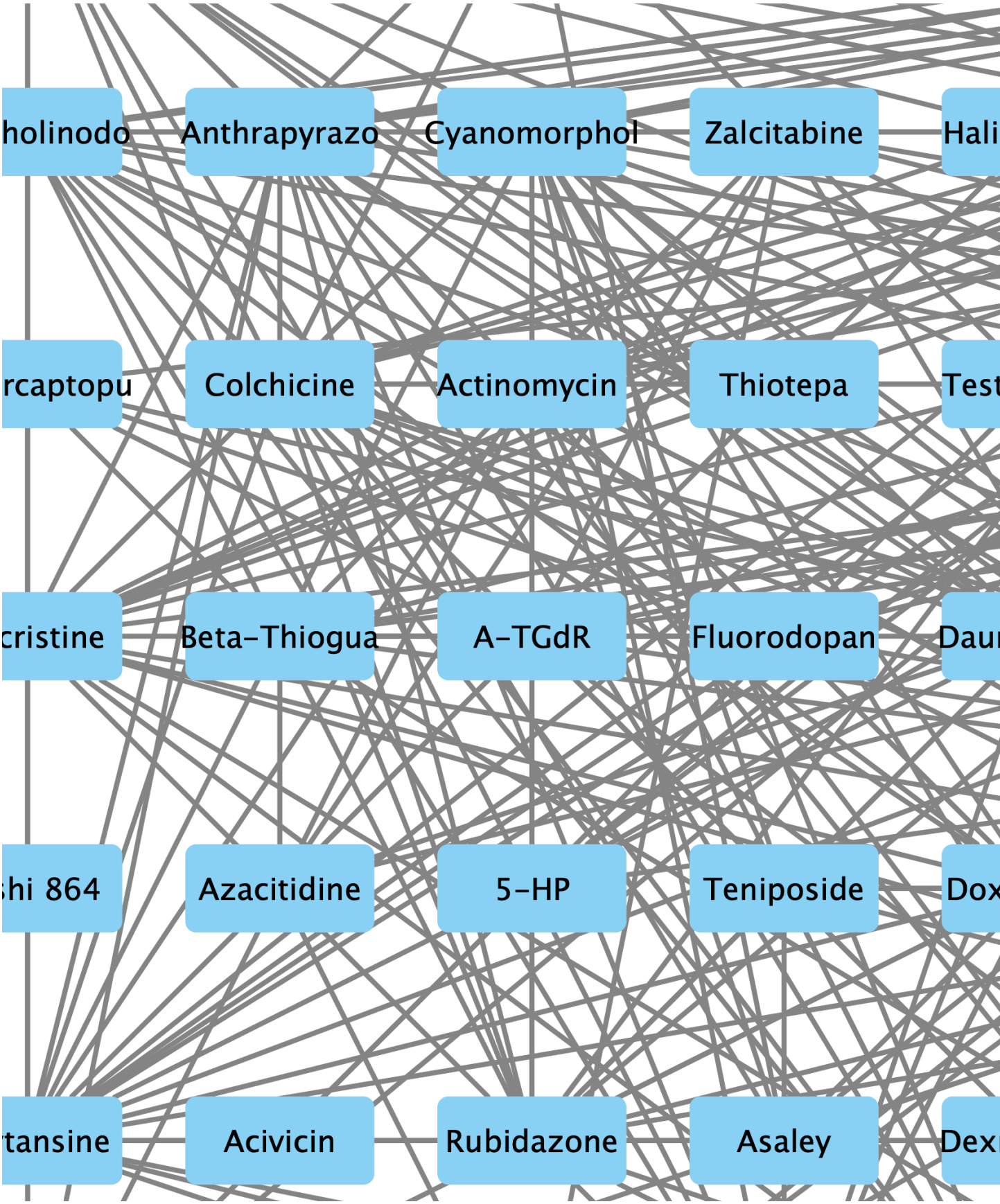
```
properties.list <- list(createNewNetwork = TRUE,
                        network = "current",
                        nodeList = "all")

command.name <- "chemviz/create%20similarity"

# chemviz_cw <- setCommandProperties(cw,
#                        command.name,
#                        properties.list,
#                        copy.graph.to.R = FALSE)

string.cmd = paste0(command.name, " createNewNetwork=TRUE"," network=",getNetworkSuid(), " nodeList=all
commandsGET(string.cmd)
```

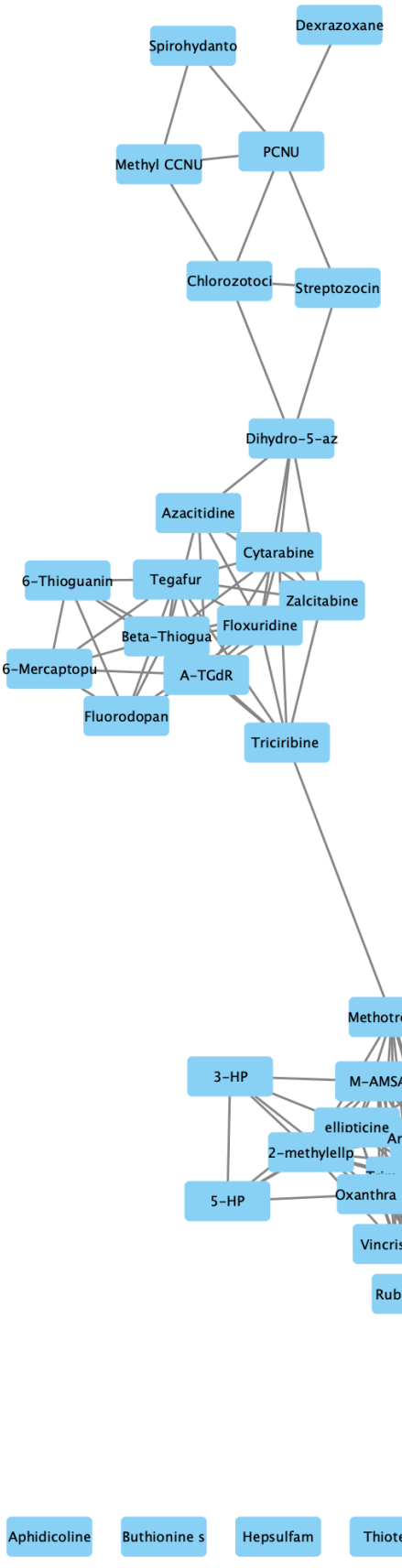
```
##                                                                                                  file
## "/Users/juliagustavsen/Documents/random_projects/Rcy3/gsoc_Rcy3_vignettes/chemviz_similarity_net.png
```


holinodo
Anthrapyrazo
Cyanomorphol
Zalcitabine
Hali
rcaptopu
Colchicine
Actinomycin
Thiotepa
Test
cristine
Beta-Thiogua
A-TGdR
Fluorodopan
Daur
hi 864
Azacitidine
5-HP
Teniposide
Dox
tansine
Acivicin
Rubidazone
Asaley
Dex

We currently just have the network in the grid format, but now with edges connecting the nodes we can do a layout that can help us visualize the connections.

```
layoutNetwork("force-directed")
```

```
##
## "/Users/juliagustavsen/Documents/random_projects/Rcy3/gsoc_Rcy3_vignettes/chemviz_similarity_net_lay
```


Dexrazoxane
Spirohydanto
PCNU
Methyl CCNU
Chlorozotoci
Streptozocin
Dihydro-5-az
Azacitidine
Cytarabine
6-Thioguanin
Tegafur
Zalcitabine
Floxuridine
Beta-Thiogua
6-Mercaptopu
A-TGdR
Fluorodopan
Triciribine
Methotr
3-HP
M-AMSA
ellipticine
2-methylellp
5-HP
Oxanthra
Vincris
Rub
Aphidicoline
Buthionine s
Hepsulfam
Thiote

8 Add 2D chemical structures to nodes

Another thing we can do with chemviz is to add the two dimensional chemical structure on to the nodes of our networks.

Let's look again at the commands available in Chemviz:

```
commandsHelp("help chemviz")
```

```
## [1] "Available commands for 'chemviz':"
```

```
##  [1] "calculate mcss"           "create attributes"
##  [3] "create similarity"        "hide results"
##  [5] "paint structures"         "remove structures"
##  [7] "search"                   "settings"
##  [9] "show compound structures" "show compound table"
## [11] "show results"
```

And the arguments needed for the command we want to use: "paint structures"

```
commandsHelp("help chemviz/paint%20structures")
```

```
## [1] "Available arguments for 'chemviz paint structures':"
```

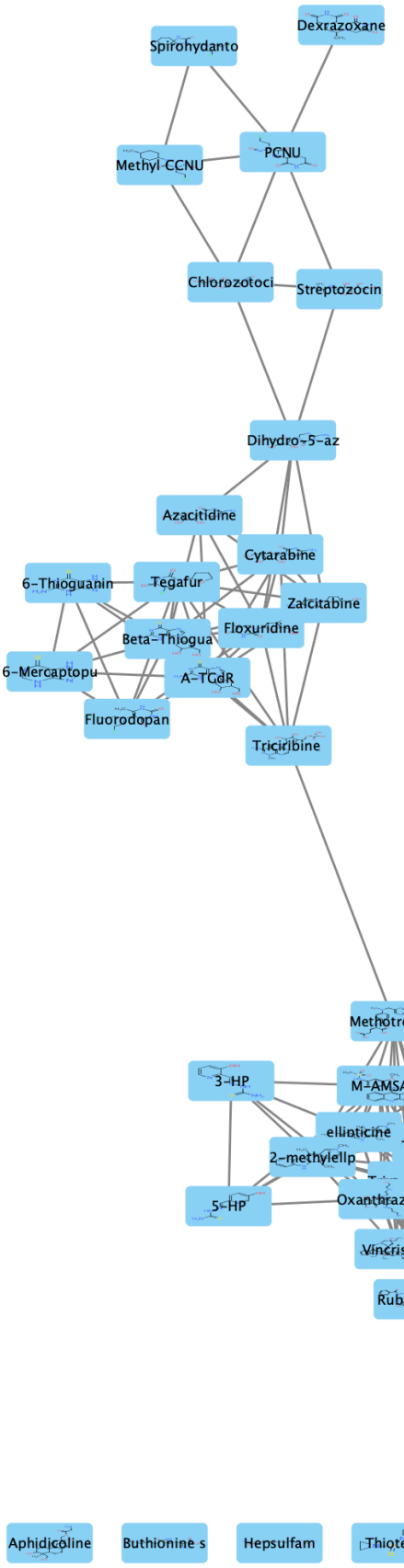
```
## [1] "nodeList"
```

```
command.name <- "chemviz/paint%20structures"

string.cmd = paste0(command.name, " nodeList=all")
commandsGET(string.cmd)
```

Now we have all of the chemical structures displayed on the nodes of our network.

```
##
## "/Users/juliagustavsen/Documents/random_projects/Rcy3/gsoc_Rcy3_vignettes/chemviz_similarity_net_node
```


Dexrazoxane
Spirohydanto
PCNU
Methyl-CCNU
Chlorozotoci
Streptozocin
Dihydro-5-az
Azacitidine
Cytarabine
6-Thioguanin
Tegafur
Zalcitabine
Floxuridine
Beta-Thiogua
6-Mercaptopu
A-TGdR
Fluorodopan
Triciribine
3-HP
ellipticine
2-methylellp
5-HP
Aphidicoline
Buthionine s
Hepsulfam

9 Colour nodes by mechanism of action (MOA)

The nodes that we are examining have mechanisms of action (MOA) associated with them. We can then colour these nodes by their MOA.

```
MOA_classes <- unique(df_with_knownMoaDrugs$MOA)
number_of_unique_MOA <- length(MOA_classes)
colours_for_MOA_classes <- colorRampPalette(brewer.pal(12, "Set3"))(number_of_unique_MOA)
```

We have 14 different MOA classes and have used RColorBrewer to generate different colours for the different classes.

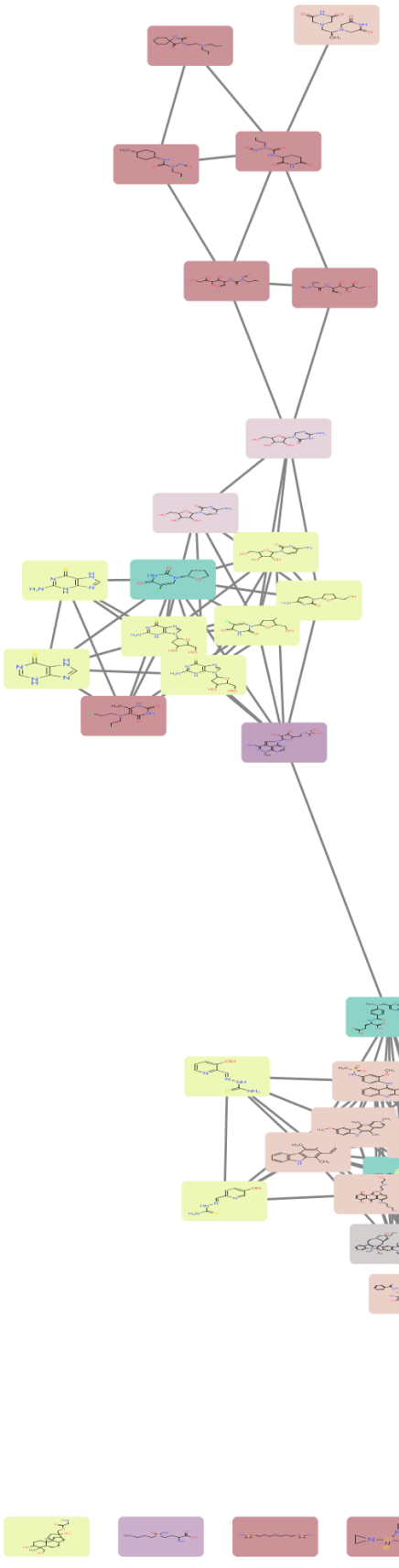
```
setNodeColorMapping("MOA",
                    MOA_classes,
                    colours_for_MOA_classes,
                    mapping.type = "d",
                    default.color = "#000000")



## node font looks ugly, let's turn it off for now
setNodeFontSizeDefault(0)
```

We have coloured the nodes nicely, but we do not know which ones are associated with which classes. To know more about this we will print out the legend from Cytoscape. At the moment there is no automated way to do this so we need to go into Cytoscape, click on the "Style" tab and then click on the little arrow (that has a mouseover text of "Options"). Once the menu opens there you will find a dialogue that lets you export a legend (in gif, svg, or pdf formats). Once exported we will look at the legend beside our new coloured network.

```
##
## "/Users/juliagustavsen/Documents/random_projects/Rcy3/gsoc_Rcy3_vignettes/chemviz_similarity_net_col
```



Node Fill Color	MOA
	A2
	A7
	AM
	DNMT
	Db
	Df
	Ds
	HDAC
	Ho
	Rs
	STK
	T1
	T2
	Tu