

Find a Gene Pt.2

```
library(heatmaply)
```

```
Loading required package: plotly
```

```
Loading required package: ggplot2
```

```
Attaching package: 'plotly'
```

```
The following object is masked from 'package:ggplot2':
```

```
  last_plot
```

```
The following object is masked from 'package:stats':
```

```
  filter
```

```
The following object is masked from 'package:graphics':
```

```
  layout
```

```
Loading required package: viridis
```

```
Loading required package: viridisLite
```

=====

Welcome to heatmaply version 1.4.2

Type citation('heatmaply') for how to cite the package.

Type ?heatmaply for the main documentation.

The github page is: <https://github.com/talgalili/heatmaply/>

Please submit your suggestions and bug-reports at: <https://github.com/talgalili/heatmaply/issues>

You may ask questions at stackoverflow, use the r and heatmaply tags:

<https://stackoverflow.com/questions/tagged/heatmaply>

=====

```
library(seqinr)
```

```
alignment <- read.alignment("/Users/rogeliocastro/Documents/Classes UCSD/BIMM 143/Gene_Pt2/
```

```
alignment_matrix <- as.matrix(alignment)
```

```
library(bio3d)
```

Attaching package: 'bio3d'

The following objects are masked from 'package:seqinr':

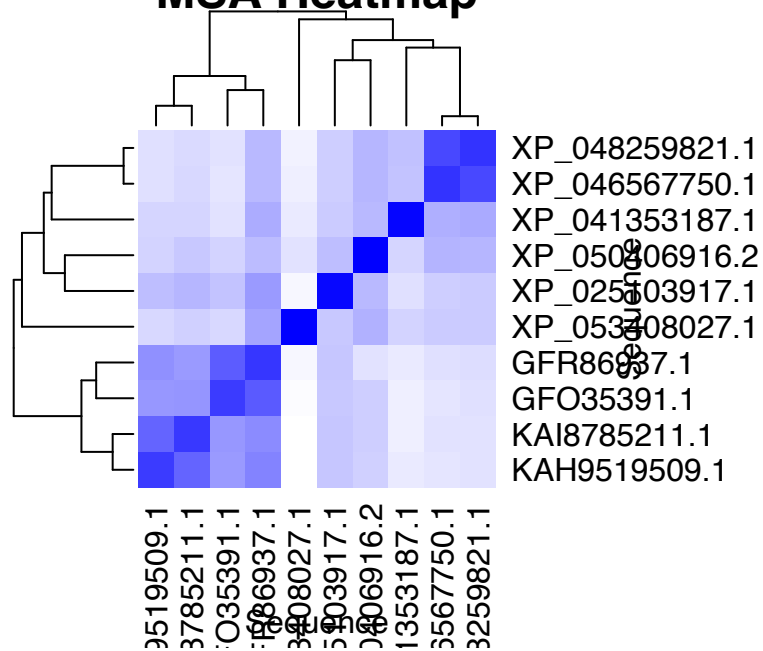
```
consensus, read.fasta, write.fasta
```

```
msa <- read.fasta("/Users/rogeliocastro/Documents/Classes UCSD/BIMM 143/Gene_Pt2/aln-fasta
```

```
identity_matrix <- seqidentity(msa)
```

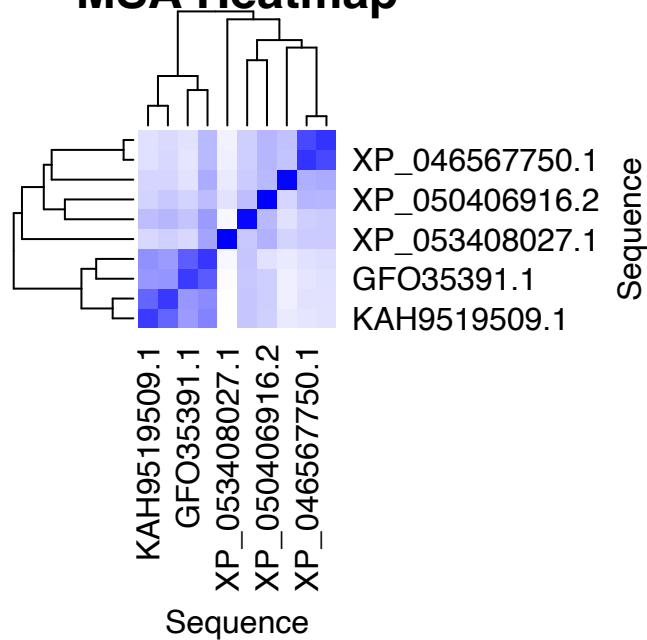
```
heatmap(identity_matrix, col = colorRampPalette(c("white", "blue"))(100), xlab = "Sequence
```

MSA Heatmap



```
heatmap(identity_matrix, col = colorRampPalette(c("white", "blue"))(100),
  xlab = "Sequence", ylab = "Sequence",
  main = "MSA Heatmap",
  cex.axis = 0.8, cex.lab = 0.8, mar = c(10, 10))
```

MSA Heatmap



```
library("rgl")
```

```
pdb_file <- "/Users/rogeliocastro/Downloads/5go9-pdb-bundle.tar.gz"
```

```

pdb_file <- "/Users/rogeliocastro/Downloads/5go9-pdb-bundle.tar.gz"
temp_dir <- tempdir() # Create a temporary directory
untar(pdb_file, exdir = temp_dir) # Extract the archive to the temporary directory

```

```
pdb_files <- list.files(temp_dir, pattern = "\\\\.pdb$", full.names = TRUE)
```

```
pdb_files <- list.files(temp_dir, pattern = "\\\\.pdb$", full.names = TRUE)
```

```

# Read the first PDB file from the list
pdb <- read.pdb(file = pdb_files[1])

```

```
pdb <- read.pdb(file = file.path(temp_dir, "5go9-pdb-bundle1.pdb"))
```

```
pdb2 <- read.pdb(file = file.path(temp_dir, "5go9-pdb-bundle2.pdb"))
```

```
library(bio3d)
```

```
pdb <- read.pdb(file = file.path(temp_dir, "5go9-pdb-bundle1.pdb"))
```

```
library(bio3d)
```

```
library(rgl)
```

```
# Read the PDBx/mmCIF file using the read.cif() function from bio3d
```

```
pdb <- read.cif("/Users/rogeliocastro/Downloads/5go9.cif")
```

Warning in read.cif("/Users/rogeliocastro/Downloads/5go9.cif"): beta version of
`read.cif`. please use with caution

Warning in read.cif("/Users/rogeliocastro/Downloads/5go9.cif"): helix/sheet
records could not be parsed

```
# Extract the atomic coordinates
```

```
xyz <- pdb$atom$xyz
```

```
# Open a 3D scene
```

```
open3d()
```

```
glX
```

```
1
```

```
# Create a mesh object using the atomic coordinates
```

```
mesh <- rgl::mesh3d(x = xyz[, 1], y = xyz[, 2], z = xyz[, 3])
```

```
#White Background
```

```
rgl::rgl.bg(color = "white")
```

Warning: 'rgl::rgl.bg' is deprecated.

Use 'bg3d' instead.

See help("Deprecated")

```
# Set the color of the mesh object
rgl::shade3d(mesh, material = material3d(color = "blue"))

# Adjust the view angle
rgl::view3d(theta = 120, phi = 20, zoom = 0.8)

# Save the 3D image as a snapshot
rgl::snapshot3d(filename = "protein_image.png")
```

Warning in rgl::snapshot3d(filename = "protein_image.png"): webshot = TRUE
requires the webshot2 package and Chrome browser; using rgl.snapshot() instead