

Overview

Spectre provides functions that enable quick and easy plotting of cellular data. These are facilitated within two key functions:

1. **make.colour.plot** will save a dot plot with an 'X' and 'Y' axis. If no colour axis (col.axis) is specified, the data points will be coloured by density. If you specify another column as col.axis, then each point will be coloured using a continuous colour scheme (e.g. for marker expression) or a factor colour scheme (e.g. for sample name, group name, cluster etc).
2. **make.multi.plot** will enable the generation of multiple plots with the same X and Y parameters in a single image. This can be used to plot multiple marker expression patterns, divide up the data by some factor, or both.

Each of these will display and save a .png of the specified dataset.

Basic plot types

```
## Load packages
library(Spectre)
package.check()
package.load()

## This is the demo data we will use
Spectre::demo.clustered

## Set a working directory where the plots will be saved
setwd("/Users/thomasa/Desktop")
```

Review column names

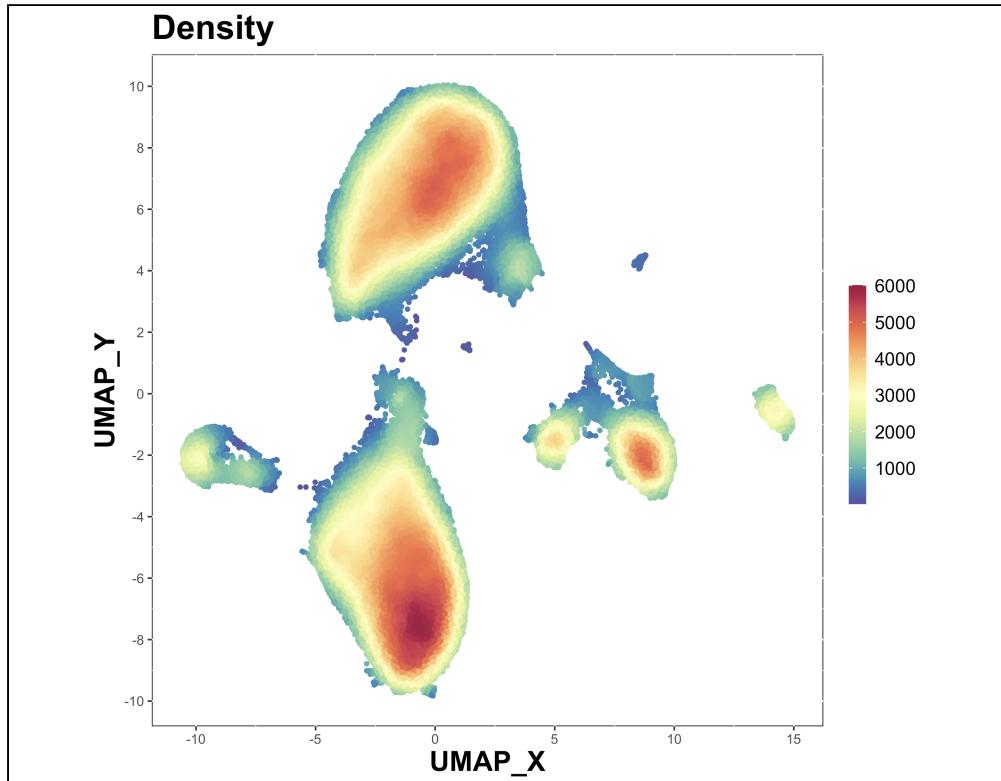
We can create an X-Y colour plot using any columns in the dataset. First, let's see the column names for our data.

```
as.matrix(names(demo.clustered))
```

Density plot

A basic X-Y plot where each dot point is coloured by the relative plot density. *Note: the plots that colour cells by density take longer to create than other types of colour plots.*

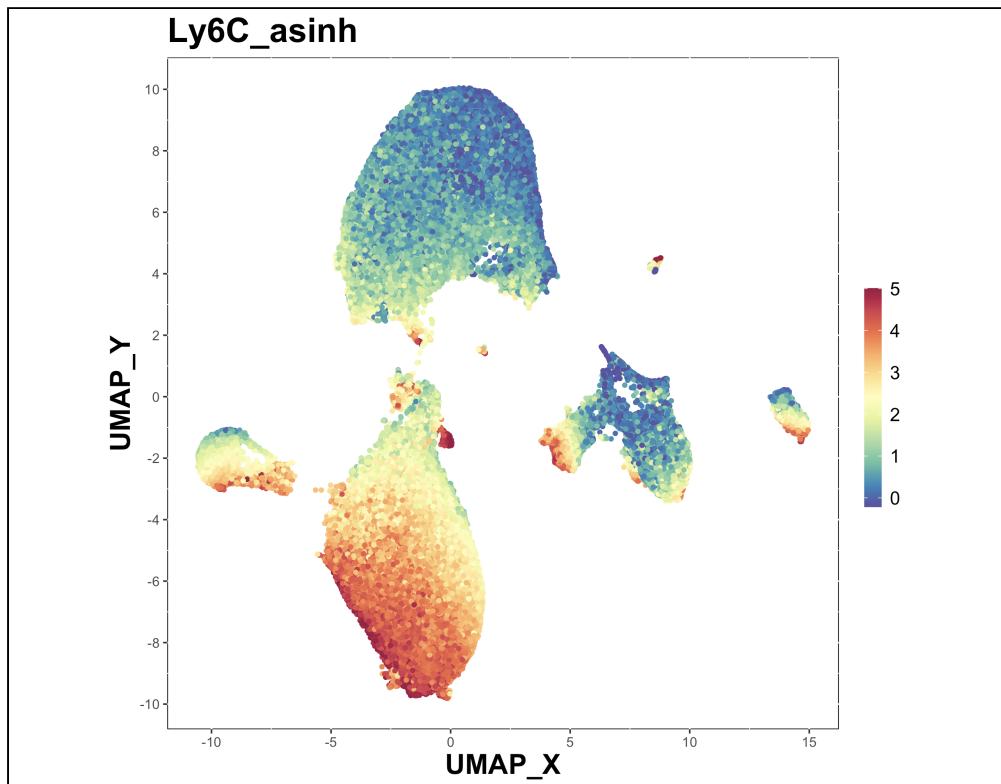
```
make.colour.plot(dat = demo.clustered,
                  x.axis = "UMAP_X",
                  y.axis = "UMAP_Y")
```



Colour plot

A basic X-Y plot where each dot point is coloured by some column. In this case, a continuous variable (such as a cellular marker).

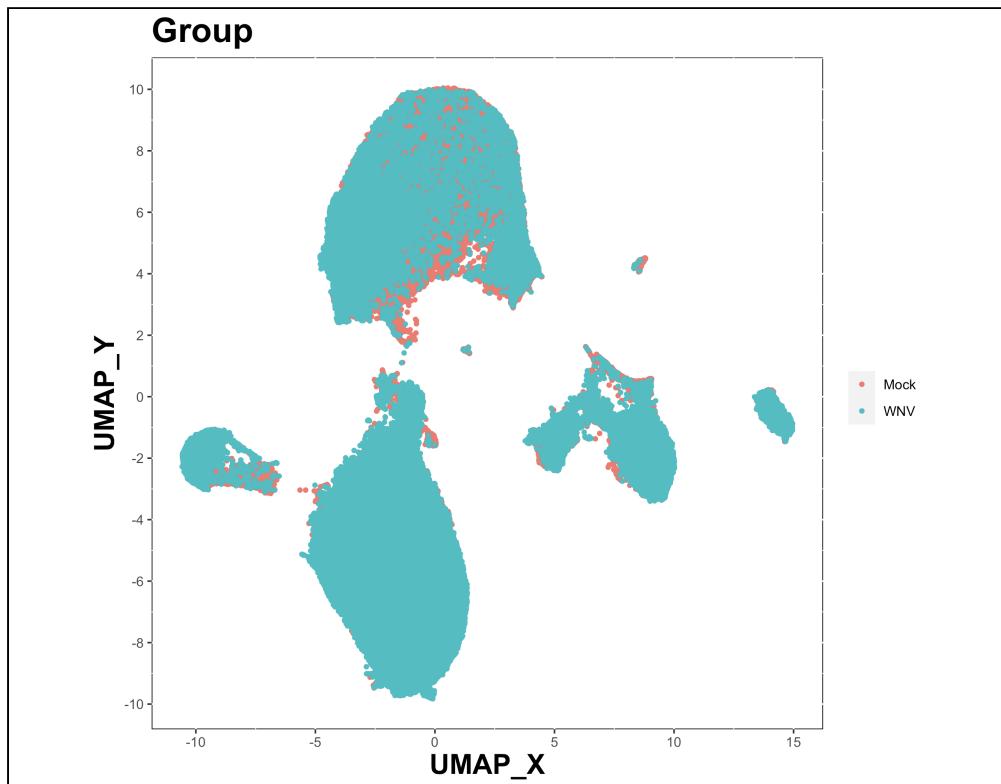
```
make.colour.plot(dat = demo.clustered,
                  x.axis = "UMAP_X",
                  y.axis = "UMAP_Y",
                  col.axis = "Ly6C_asinh")
```



Factor plot

Similar to the colour plot above, but coloured by some factor (e.g. Group).

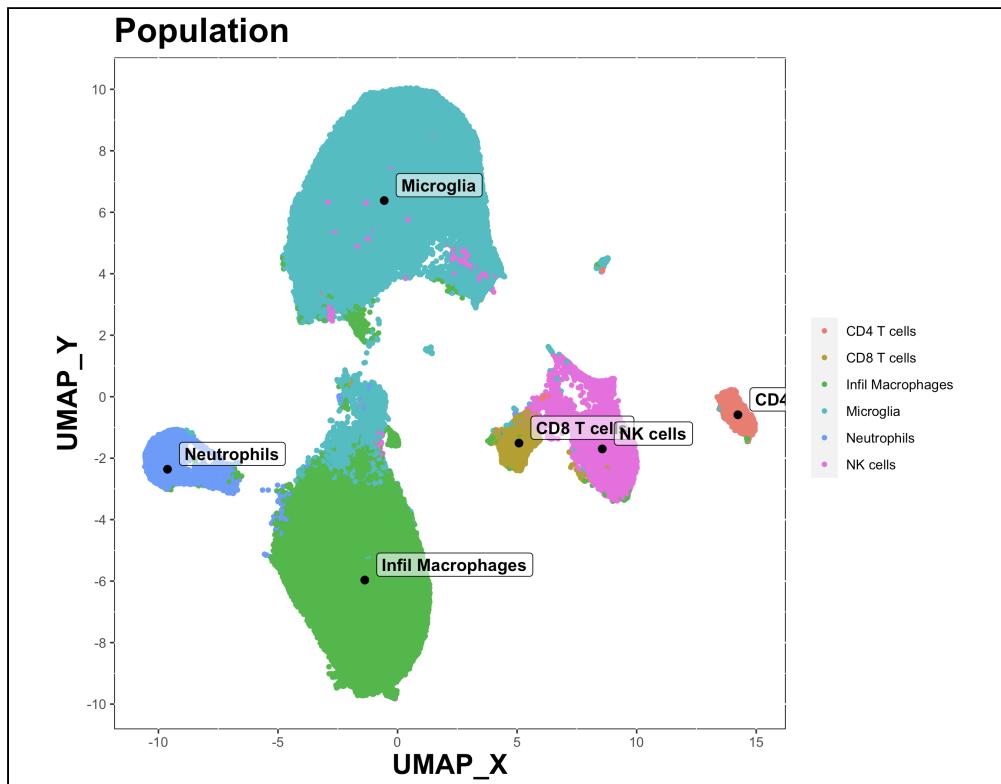
```
make.colour.plot(dat = demo.clustered,
                  x.axis = "UMAP_X",
                  y.axis = "UMAP_Y",
                  col.axis = "Group",
                  col.type = "factor")
```



Factor plot with labels

Similar to the factor plot above, but coloured by some factor (e.g. Cluster) where the of cells belonging to that factor are labelled.

```
make.colour.plot(dat = demo.clustered,
                  x.axis = "UMAP_X",
                  y.axis = "UMAP_Y",
                  col.axis = "Population",
                  col.type = 'factor',
                  add.label = TRUE)
```

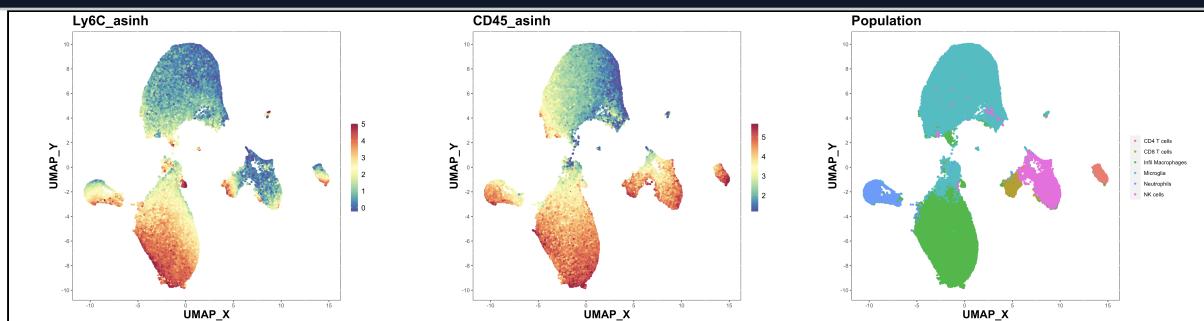


Multi plots

Multi plot

Provide the function with a vector of column names to colour the plots by, and it will create one plot entry, and save it as a single image.

```
make.multi.plot(dat = demo.clustered,
                 x.axis = "UMAP_X",
                 y.axis = "UMAP_Y",
                 plot.by = c("Ly6C_asinh", "CD45_asinh", "Population"))
```

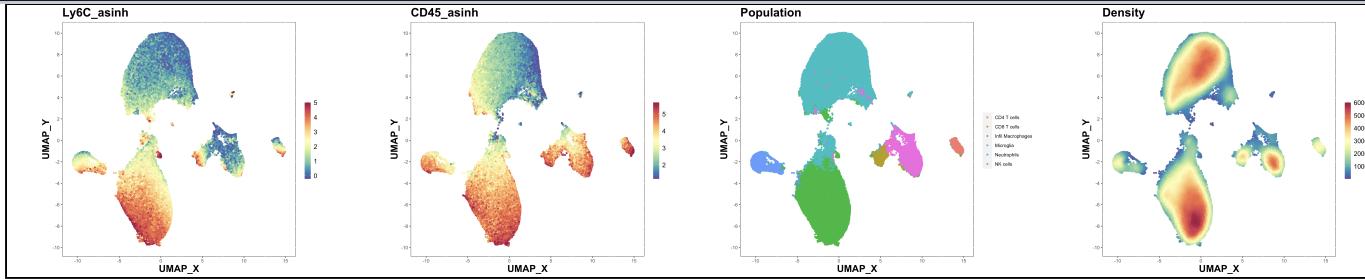


Multi plot + density

Same as the multi plot above, but adding a density plot at the end. *Note: density calculations will mean the function runtime will be longer, compared to the same not including a density plot.*

```
make.multi.plot(dat = demo.clustered,
                 x.axis = "UMAP_X",
```

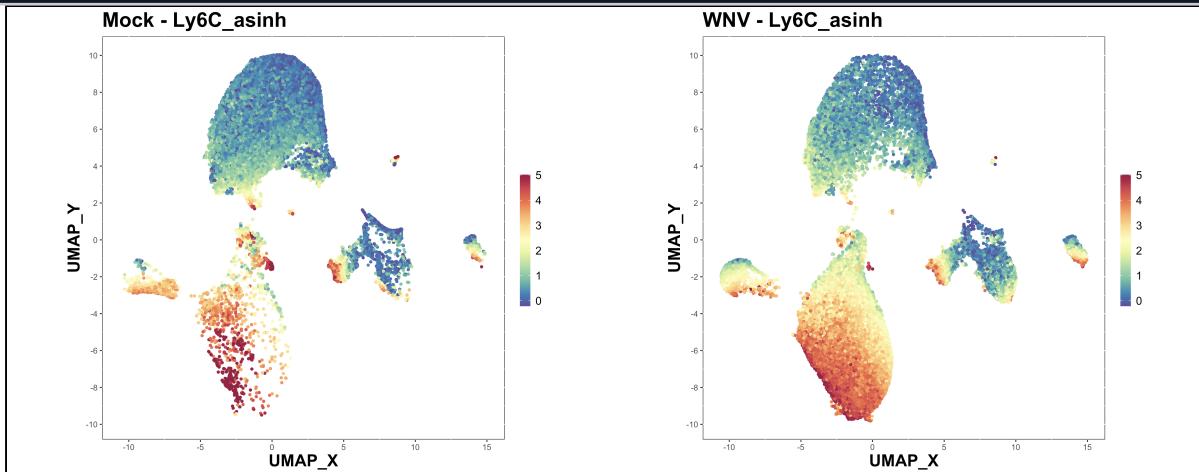
```
y.axis = "UMAP_Y",
plot.by = c("Ly6C_asinh", "CD45_asinh", "Population"),
add.density = TRUE)
```



Multi plot - divided by the values in some column

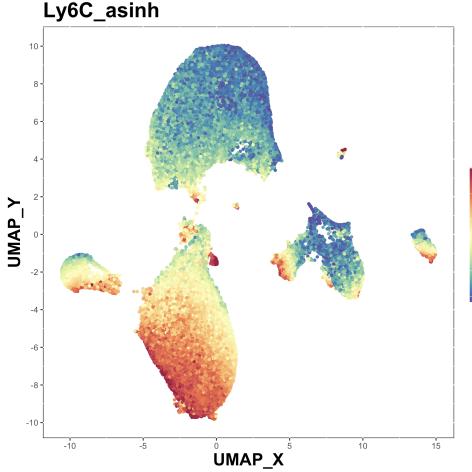
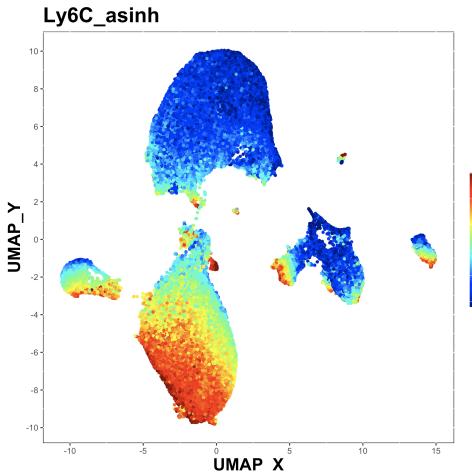
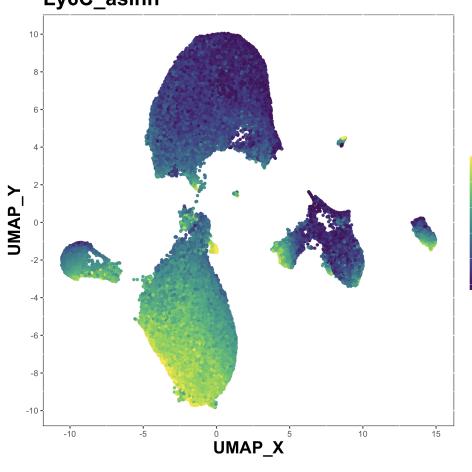
Similar to the plots above, but done in repeating fashion, dividing the data by the values in a specified column (e.g. in this case, by group)

```
make.multi.plot(dat = demo.clustered,
                 x.axis = "UMAP_X",
                 y.axis = "UMAP_Y",
                 plot.by = c("Ly6C_asinh"),
                 divide.by = "Group")
```



Additional arguments

Colour schemes

Function	Plot
Spectral <pre data-bbox="96 361 854 534">make.colour.plot(dat = demo.clustered, x.axis = "UMAP_X", y.axis = "UMAP_Y", col.axis = "Ly6C_asinh", colours = "spectral")</pre>	 <p>Ly6C_asinh</p> <p>UMAP_Y</p> <p>UMAP_X</p> <p>0 1 2 3 4 5</p>
Jet <pre data-bbox="96 878 854 1051">make.colour.plot(dat = demo.clustered, x.axis = "UMAP_X", y.axis = "UMAP_Y", col.axis = "Ly6C_asinh", colours = "jet")</pre>	 <p>Ly6C_asinh</p> <p>UMAP_Y</p> <p>UMAP_X</p> <p>0 1 2 3 4 5</p>
Viridis <pre data-bbox="96 1394 854 1567">make.colour.plot(dat = demo.clustered, x.axis = "UMAP_X", y.axis = "UMAP_Y", col.axis = "Ly6C_asinh", colours = "viridis")</pre>	 <p>Ly6C_asinh</p> <p>UMAP_Y</p> <p>UMAP_X</p> <p>0 1 2 3 4 5</p>

Function	Plot
Inferno	<p>Ly6C_asinh</p> <p>UMAP_Y</p> <p>UMAP_X</p>
Magma	<p>Ly6C_asinh</p> <p>UMAP_Y</p> <p>UMAP_X</p>

Coloured hex plot

To help avoid overplotting, cells can be 'binned' into hexagons on the colour plot, where each hexagon can be coloured by the average expression of the cells captured. This addition was implemented by Givanna Putri, and was inspired by the [Schex](#) package from [Saskia Freytag](#).

```
make.colour.plot(dat = demo.clustered,
                  x.axis = "UMAP_X",
                  y.axis = "UMAP_Y",
                  col.axis = "Ly6C_asinh",
                  hex = TRUE)
```

