1 Aims

So the point of this is multi-faceted:

- To provide a tutorial/introduction and access to the use of some of the crappy R code I wrote in the process of analysing peaklist MALDI imaging data during my thesis.
- To provide some example code on how (not) to use R, `knitr`, and LaTeX, including referencing using `bibtex`.

2 Disclaimer

This code is a steaming pile of crap. Use it with extreme skeptisism – as you should use any software, and do any analysis, skeptisism is the lifeblood of a scientist, embrace it. When I get some free time I plan on re-writing all the plotting functions from scratch to make them nicer. Which should make this steaming pile of crap slightly more bearable.

Also, I really ought to have wrapped my code up in a package, but I am shit and haven't, so for the time being all the relevant functions can be loaded into the workspace by calling

```
source('localFunctions.R')
```

the fact that this is here may prompt me to getting around to cleaning it up and wrapping it up in package form at some point in the future, but probably not in the immediate future.

3 Setup and Reading Peaklists

I set the current `dataset_name` to a variable,

```
dataset_name <- "A1"
```

This reads in peaklist files from

`<parent_folder_name>/<dataset_name>/<peaklist_folder_name>`

and returns a combined peaklist `data.frame` object after writing the three files:

`<dataset_name>_comprehensive_peaklist.txt`
`<dataset_name>_fExists.txt`
`<dataset_name>_LXY.txt`

to

`./<data_folder>`

`dataset_name` is required, but the other arguments are optional and default to:

- `peaklist_folder_name = "peaklists"`
- `parent_folder_name = "."`
- `data_folder = "./data"`

Note that the function `readPeaklists` reads peaklists in batches of 1000 at a time, and at the end of each batch prints the name of the last peaklist file to inform the user of progress, finally it prints the total number of empty peaklists read in.

Once created the files written to `data_folder` can be read in easily by calling

```
pl_all  <- load_peaklist(dataset_name)
LXY     <- load_LXY(dataset_name)
fExists <- load_fExists(dataset_name)
```

respectively. These `load_*` functions also accept an optional `data_folder` argument if an alternative location is used to store these files.

4 DIPPS

Now say you have produced some peakgroups one way or another, and now you have two regions you want to compare using DIPPS. For example, here I have annotation of the center of cancer tumours stored in an xml 'ROI' file. So I'll read the annotations into R using the `XML` package and merge them onto the `LXY` variable as a 'ROI' column with value 'None' for spectra not in any annotated region.

```
library(XML)
fname <- 'A1_annotation.xml'
doc   = xmlInternalTreeParse(fname)
rois  = xpathSApply(doc,
                    "/ClinProtSpectraImport/Class",
                    xmlGetAttr,"Name")
nSpec = xpathSApply(doc,
                     "/ClinProtSpectraImport/Class",
                     xmlSize)
spec  = xpathSApply(doc,
                     "/ClinProtSpectraImport/Class/Element",
                     xmlGetAttr,"Path")
temp  = data.frame(Peaklist = Peaklist_ID(spec),
                    ROI = rep(rois,nSpec),
```

```
                    stringsAsFactors = FALSE)
LXY <- merge(LXY,temp,
             all.x = TRUE)
LXY[is.na(LXY$ROI),'ROI'] = "None"
```

We can take a quick look at these annotations by plotting them. Figure 1 demonstrates this, as well as providing a simple (less confusing?) example of a straightforward way to make spatial plots without using my gargantuan `spatialPlot` function (although you could equally make this plot using `spatialPlot` if you really wanted to.

```
p = (ggplot(LXY,aes(x=X,y=Y,
                    fill=ROI,
                    alpha= 1-(ROI=='None')),
            colour=NA)
     + geom_tile()
     + coord_fixed()
     + guides(alpha = FALSE)
     + scale_x_reverse(breaks=seq(75,175, 50))
     + scale_y_continuous(breaks=seq(50, 200, 50))
     + ylab("")
     + xlab("")
)
print(p)
```

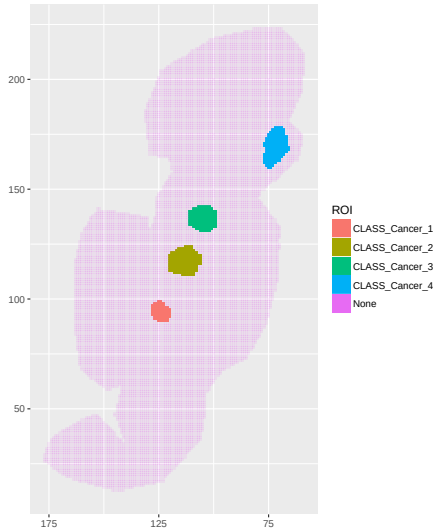

Figure 1: Annotation Regions