

# Generating oncoplots using maftools

---

Oncoplots (also called waterfall plots) are useful plots that help us visualize frequently affected genes in a cohort. This type of plots can also display information about the number and type of mutations, and can include annotations about the characteristics of the cohort, such as clinical data. The main structure of this plot is to show in each column a sample and in each row a gene. Showing mutations for the displayed genes for each sample with a color code.

There are several tools that can be used to generate this type of plots. Today we will be using the R package maftools (Mayakonda et al., 2018).

We will now be using the TCGA-BLCA data in maf format to generate our own oncoplots.

## Reading MAF files into maftools

First we need to set the directory where we will be working.

```
setwd("/home/manager/data/Driver_detection-20220823T163817Z-001/Driver_detection/")
```

Now, we'll call the maftools package.

```
library(maftools)
```

Today's data is already merged in one MAF file. To read any MAF or MAF like file using maftools use the following command.

```
BLCA = read.maf(maf = "TCGA-BLCA.maf")
```

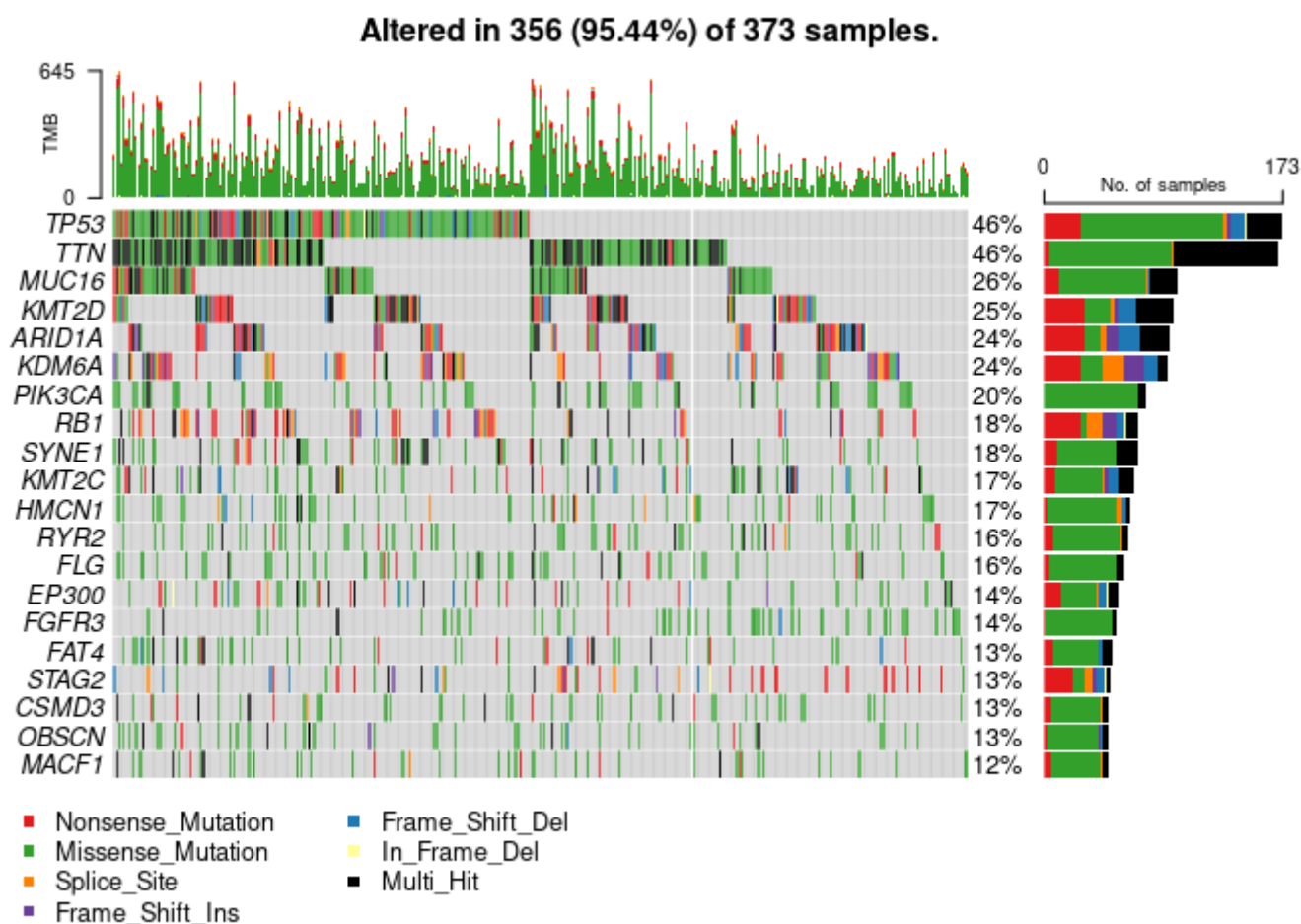
With this we have read the MAF file and stored the data in the object BLCA.

While reading the MAF file you can notice that in the console some information about the data is being displayed. This includes information about the number of silent variants in the data, genes with possible flags and, in our case, it states that we have not included clinical data.

## Creating an oncoplot

Now that we have all the data from the cohort stored in one object, we will plot a basic oncoplot.

```
oncoplot(maf= BLCA)
```



As you can see, with this simple command you can get an oncoplot displaying 20 genes.

With the default parameters the oncoplot will also show the percentage of samples in the cohort that have alterations in the shown genes, as well as the type of mutations affecting the gene through a code of colors.

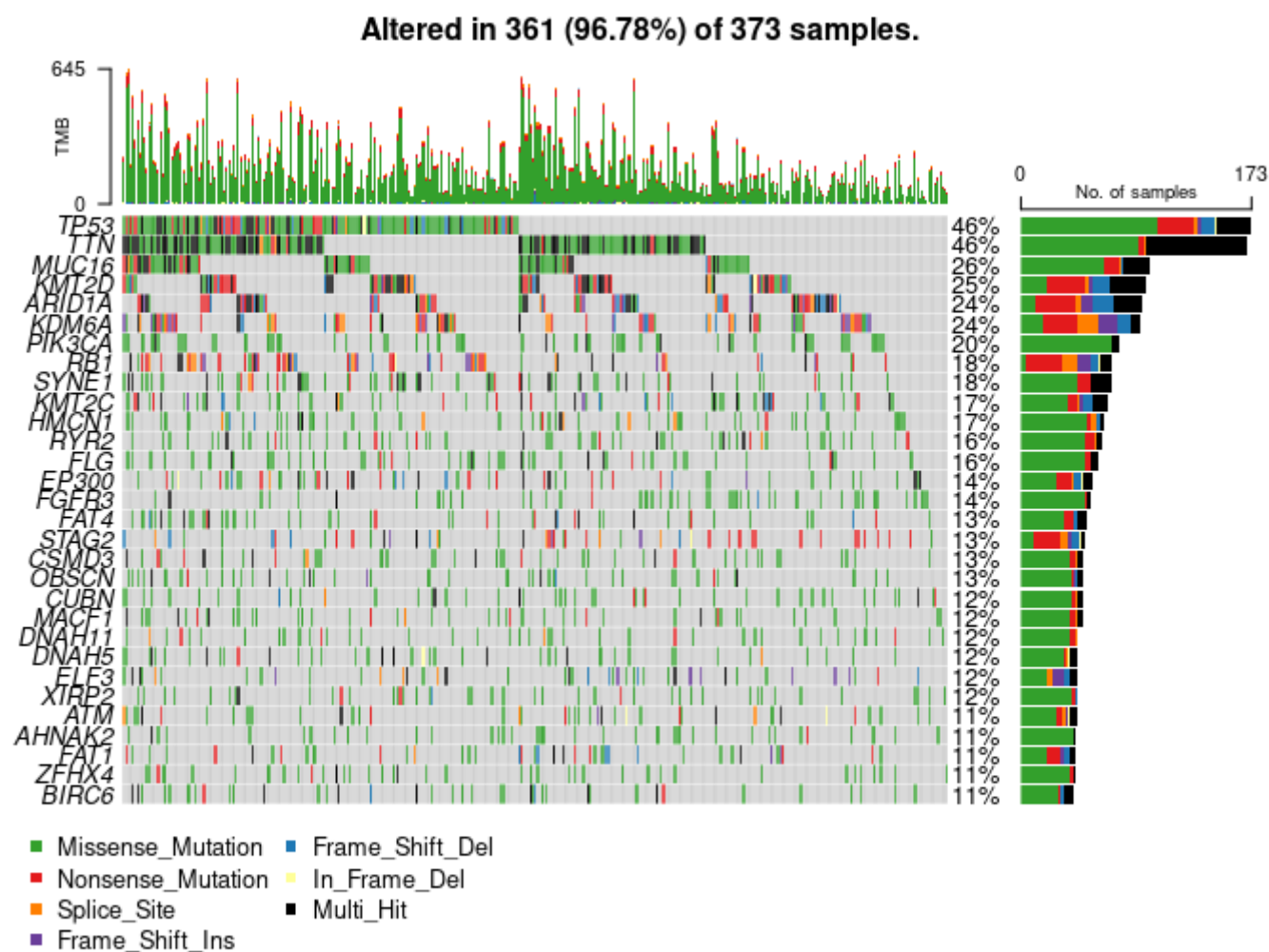
From this oncoplot, how do you think maftools selects the genes to show in the plot?

Compare these genes with the data you obtained from dndscv.

We can change the number of genes displayed with the option "top".

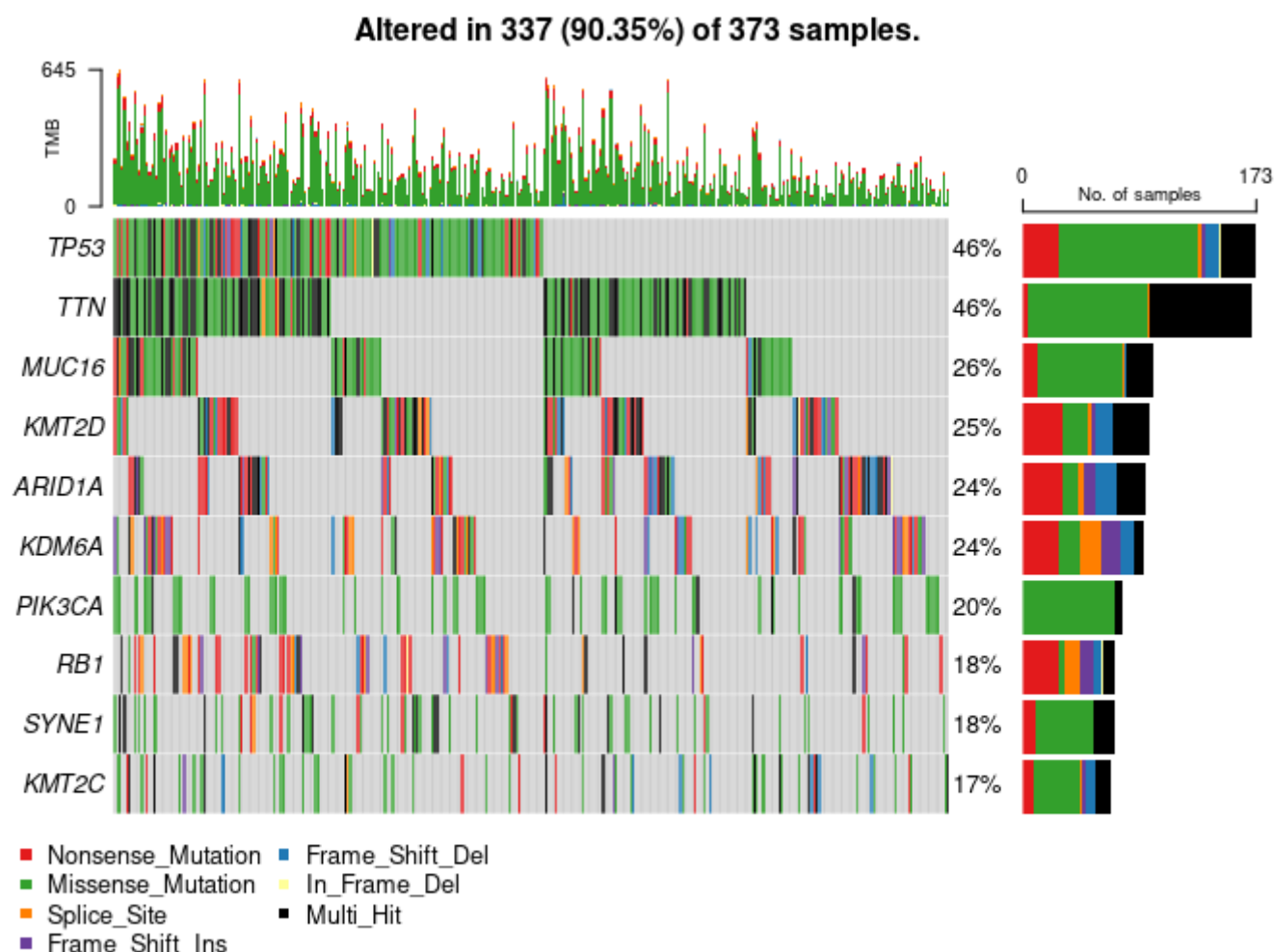
For example, if we want to show 30 genes we can type the following.

```
oncoplot(maf=BLCA, top=30)
```



Or if we want to only display 10 genes we just type:

```
oncoplot(maf=BLCA, top=10)
```

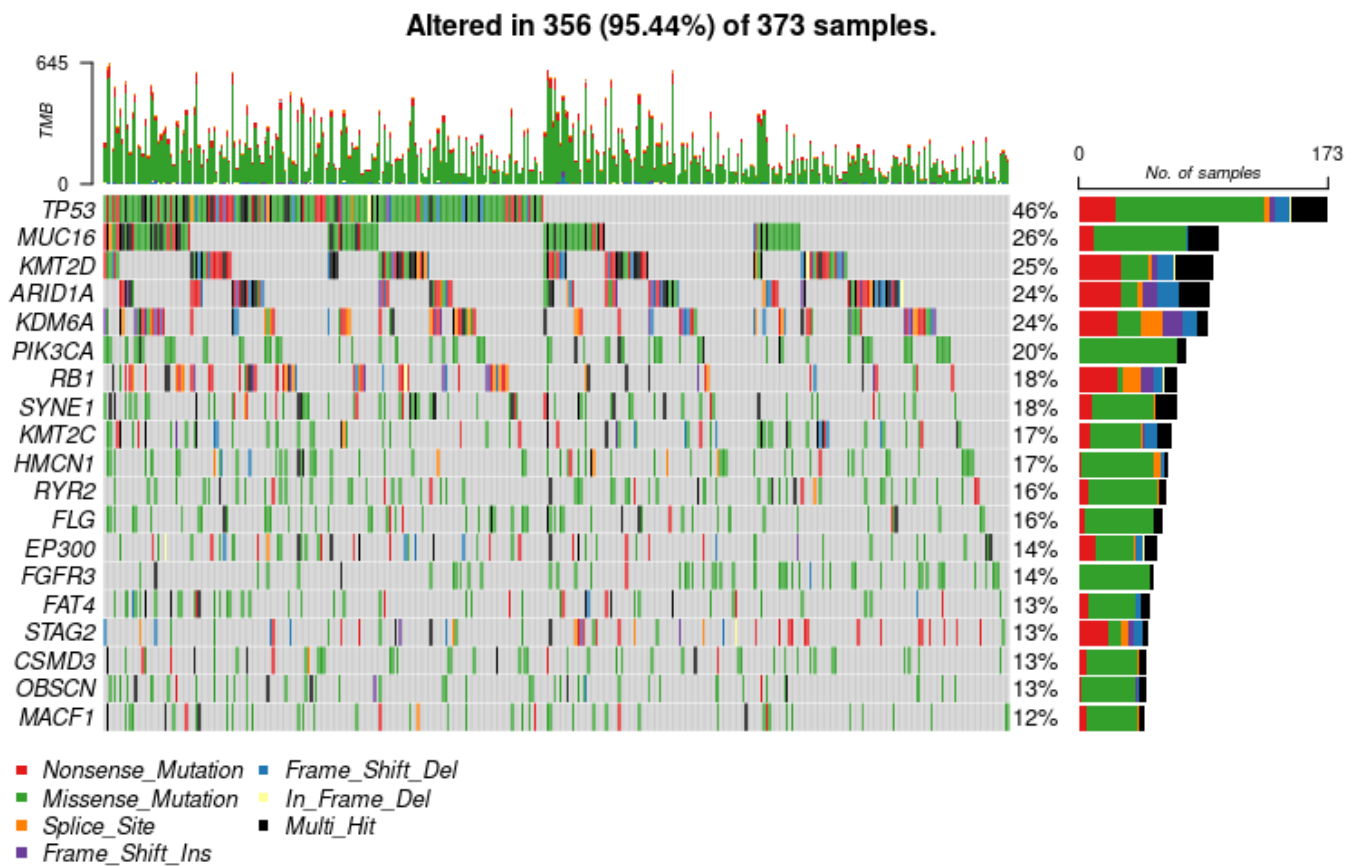


As you may have noticed, the genes shown are selected only considering the number of mutations affecting them, and the size or composition of these genes is not taken into account.

For example the TTN gene (polymorphic gene) is a very large gene. Because of its size it is expected to have a higher amount of mutations affecting it. Therefore it may not be relevant to this type of cancer. If you return to the information displayed when reading the maf file you can also see that the TTN gene was one of the genes with possible flags.

You can customize your oncoplot taking this type of considerations. For example let's try to exclude the TTN gene from our oncoplot.

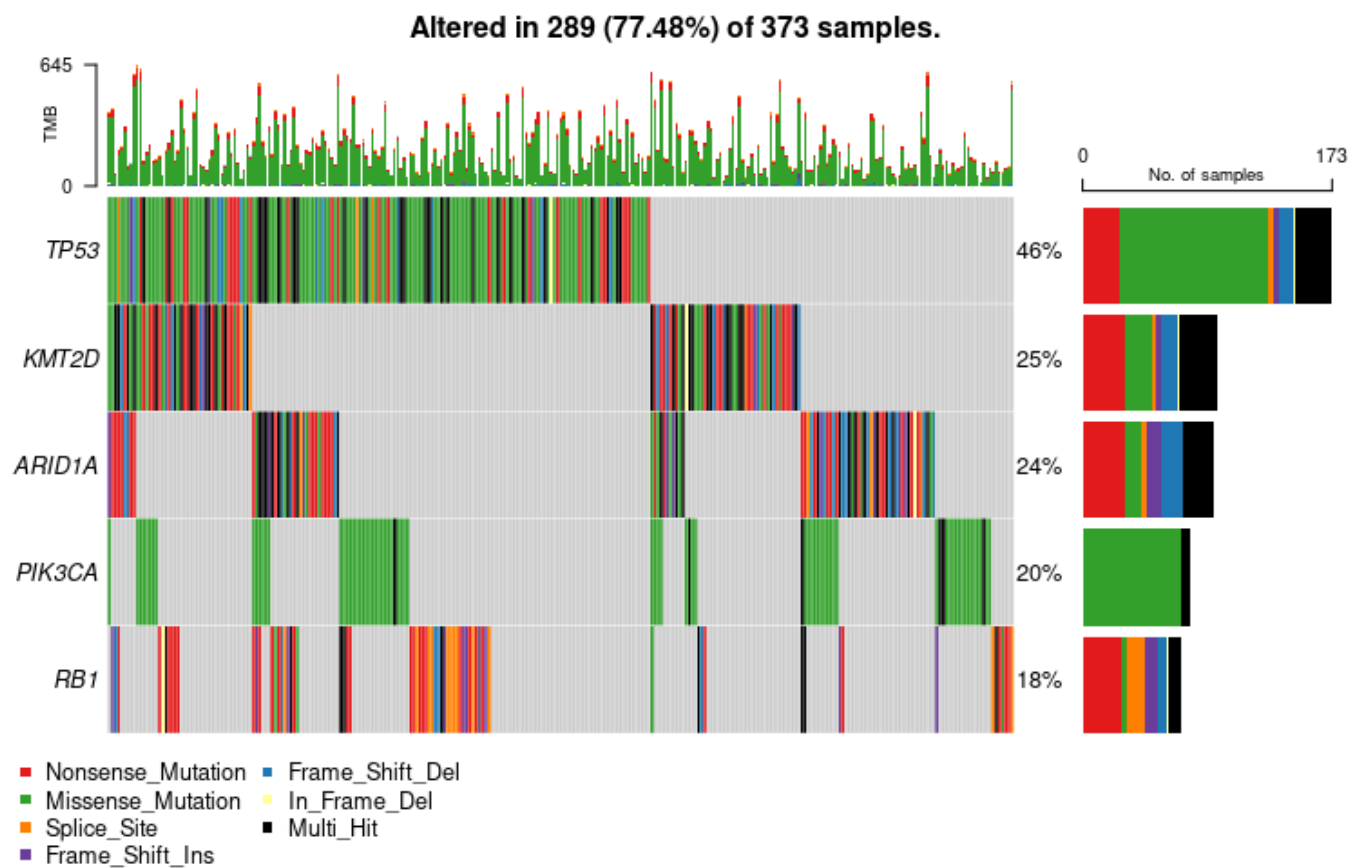
```
oncoplot(BLCA, genesToIgnore = c("TTN"))
```



Now that we have used `dnscv` to identify driver genes, we can also select those genes to generate a specific oncoplot with them.

Let's try to generate an oncoplot with 5 significant genes we identified with `dnscv`.

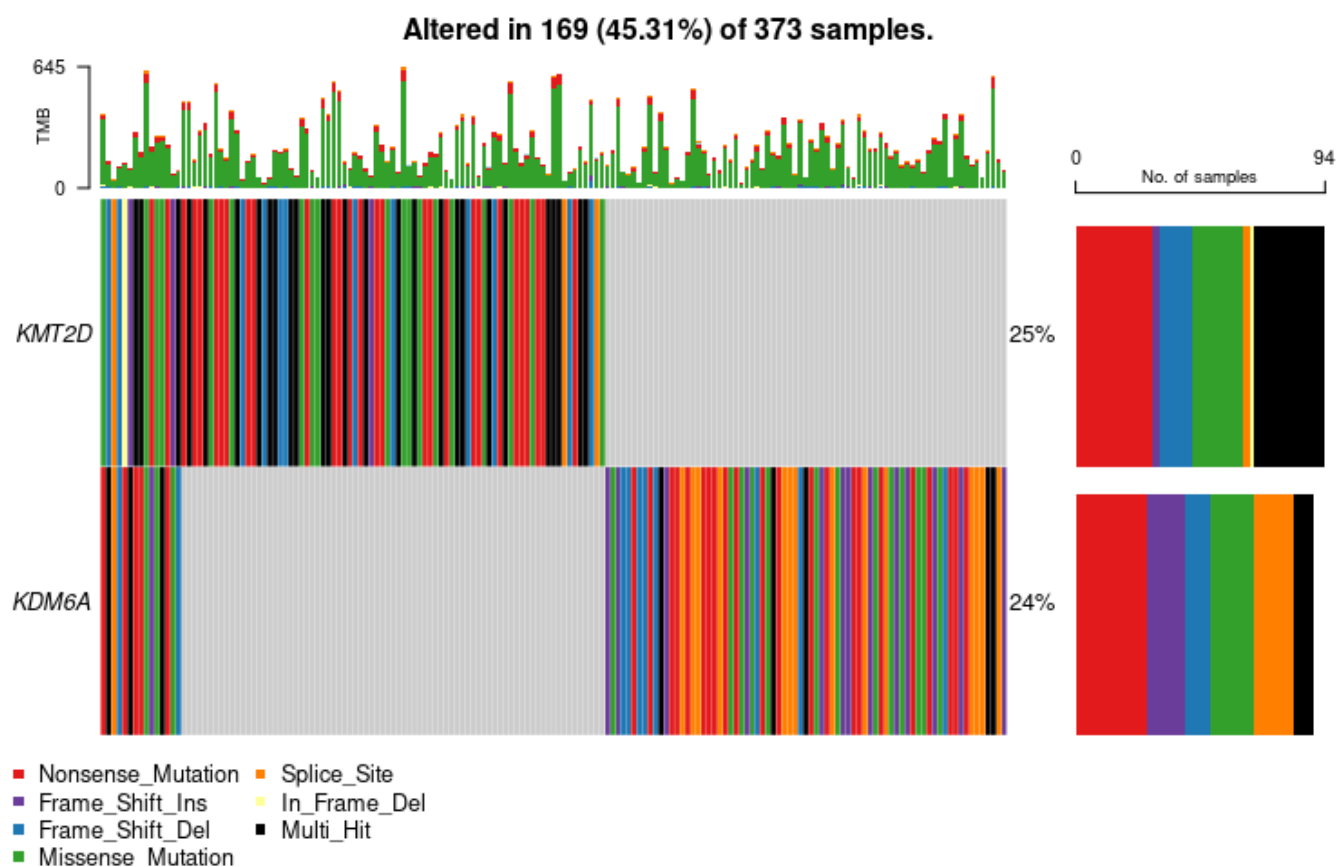
```
oncoplot(BLCA, genes = c("TP53", "PIK3CA", "ARID1A", "KMT2D", "RB1"))
```



As mentioned at the beginning, the plots are also known as waterfall plots due to their appearance. What do you think causes the plot to look like this? (Compare samples and genes) Do you see any patterns?

To make this comparison easier let's create an oncoplot only for the genes *KMT2D* and *KDM6A*.

```
oncoplot(BLCA, genes = c("KMT2D", "KDM6A"))
```



What can you see here? (Remember each "column" is one sample)

The KDM6A and KMT2D genes have shown frequently to be mutually exclusive in bladder cancer (Sangster et al., 2022). In our cohort you can see that most samples that have a mutation in one of these genes do not have a mutation in the other. Just a few samples have mutations in both genes. This type of relationship between these genes can indicate that both genes play similar roles in this type of cancer.

Oncoplots can help us detect certain relationships between affected genes, as shown above.

## Adding annotations

We can also add other annotations to our oncoplots, such as transitions and transversions. We can add this information like this:

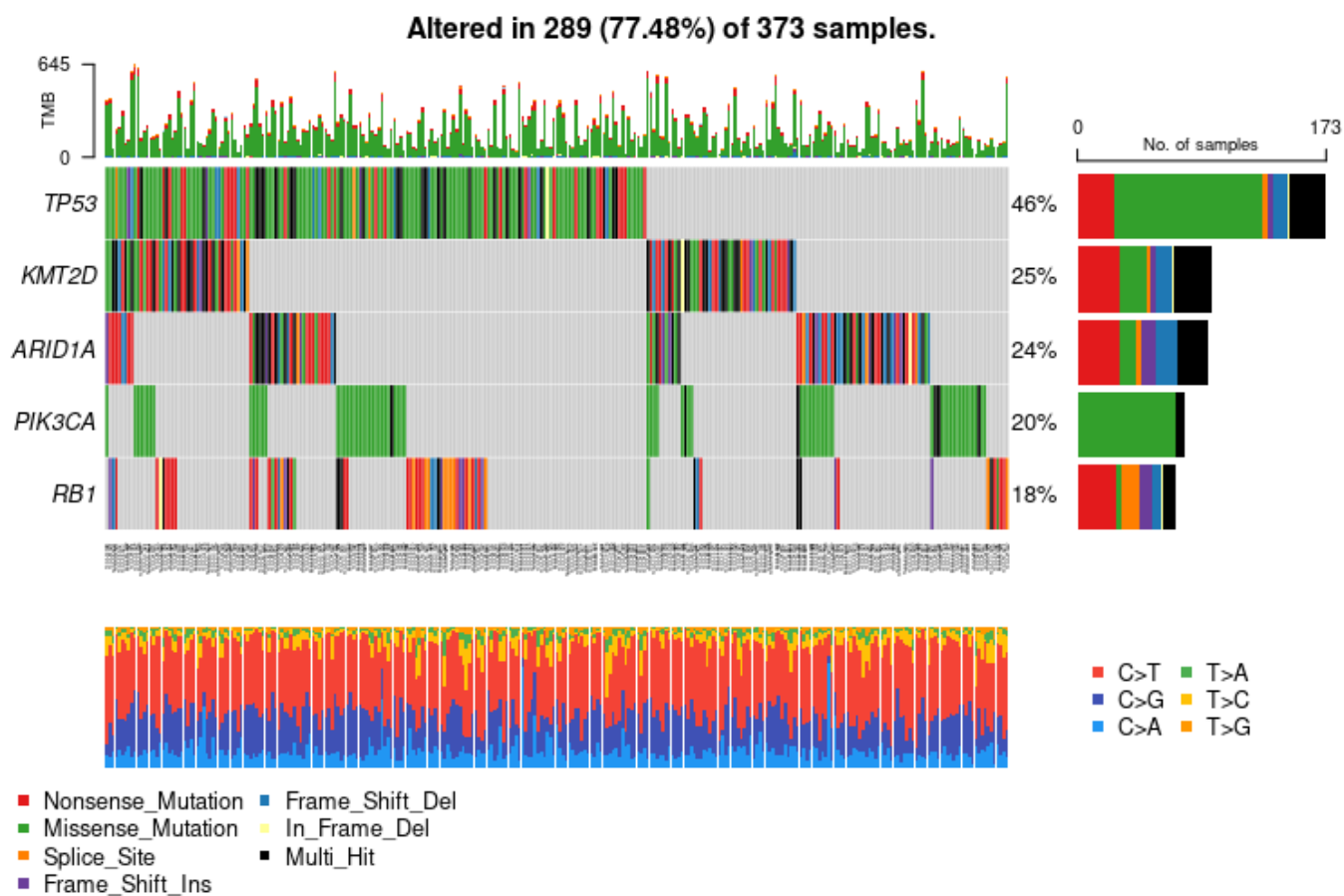
```
oncoplot(BLCA, genes = c("TP53", "PIK3CA", "ARID1A", "KMT2D", "RB1"), draw_titv
= TRUE)
```

Being able to visualize transitions and transversions can help us identify certain patterns that could be related to a specific source of DNA damage.

We can also use the annotations of the variant classification that is displayed by default to identify oncogenes and tumor suppressors. What type of mutations do you expect on oncogenes? And what type do you expect on tumor suppressors?

Depending on your cohort, it can also be useful to have the IDs of the samples on the plot. You can add them with the option "showTumorSampleBarcodes".

```
oncoplot(BLCA, genes = c("TP53", "PIK3CA", "ARID1A", "KMT2D", "RB1"), draw_titv
= TRUE, showTumorSampleBarcodes=TRUE)
```

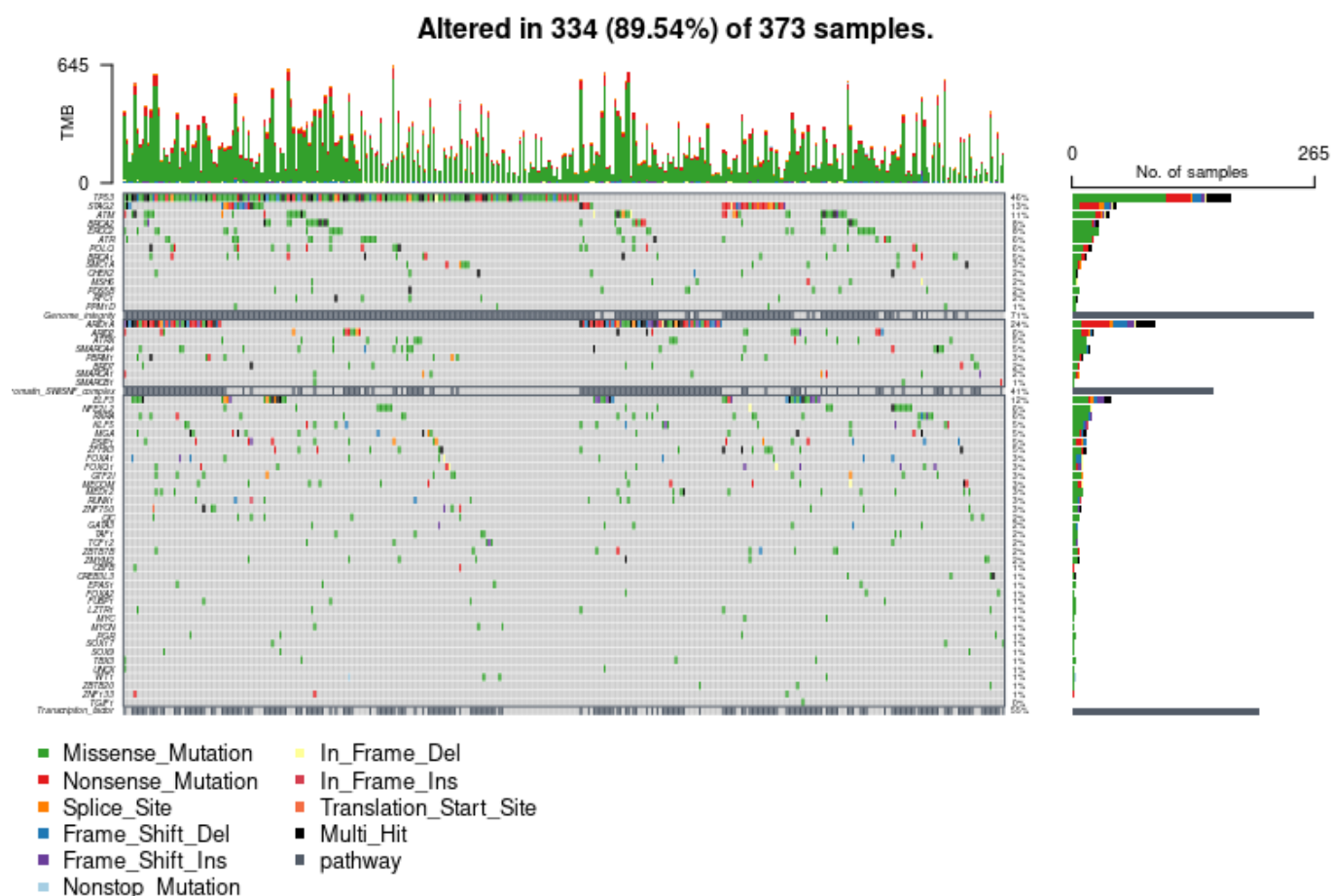


## Adding pathways

Maftools has the option to plot affected pathways and the genes corresponding to those pathways. By using the option `pathways = 'auto'`, maftools will display the three most affected pathways.

```
oncoplot(maf=BLCA, pathways= 'auto', gene_mar =5, fontSize =0.3)
```





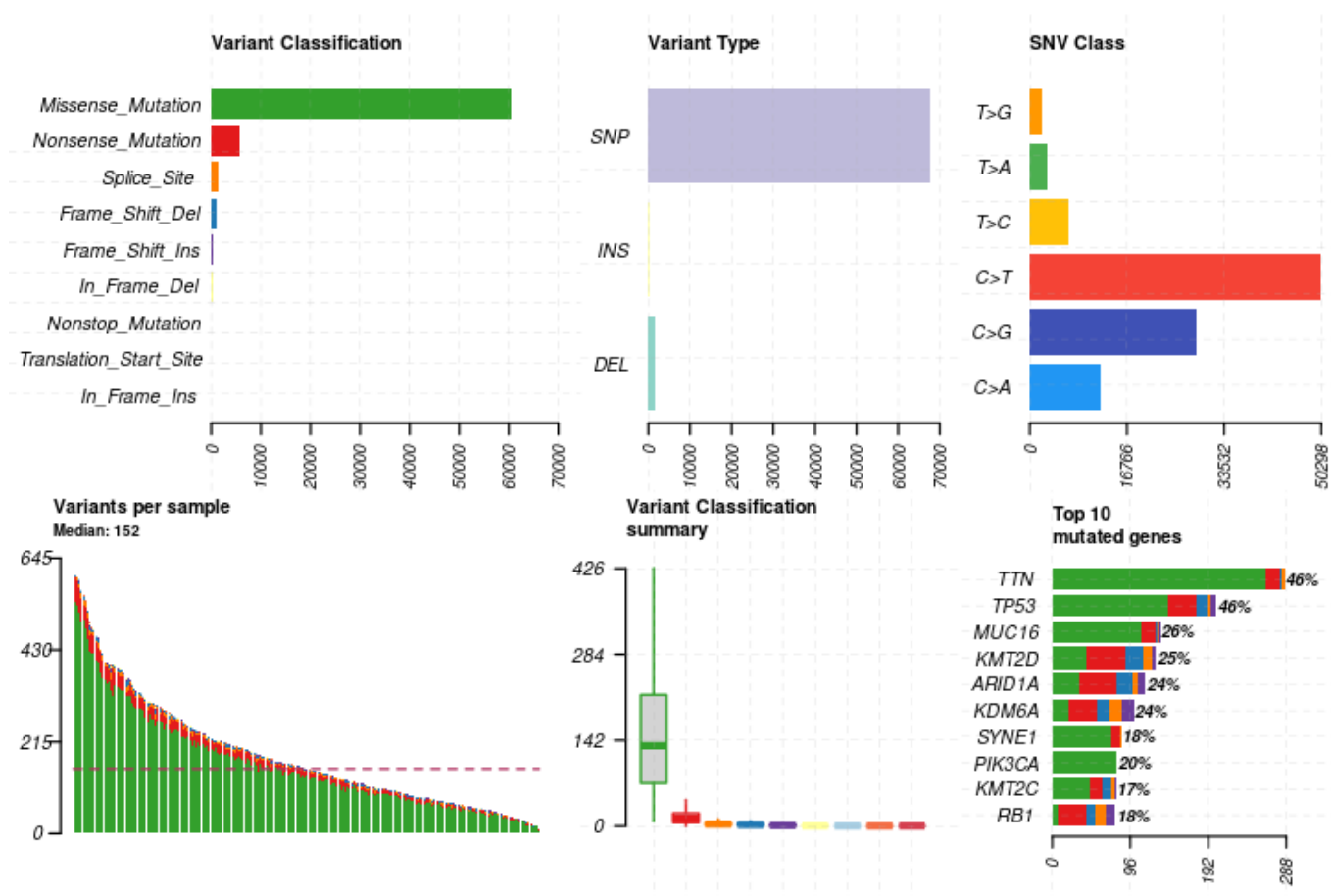
The option "gene\_mar" in the previous command changes the margin width in gene names, and "fontSize" changes the size of the gene names to improve readability in the plot.

## Other maftools plots

Maftools can generate other useful plots. For example it can plot a summary of all variants, showing variants per sample, variant type and classification.

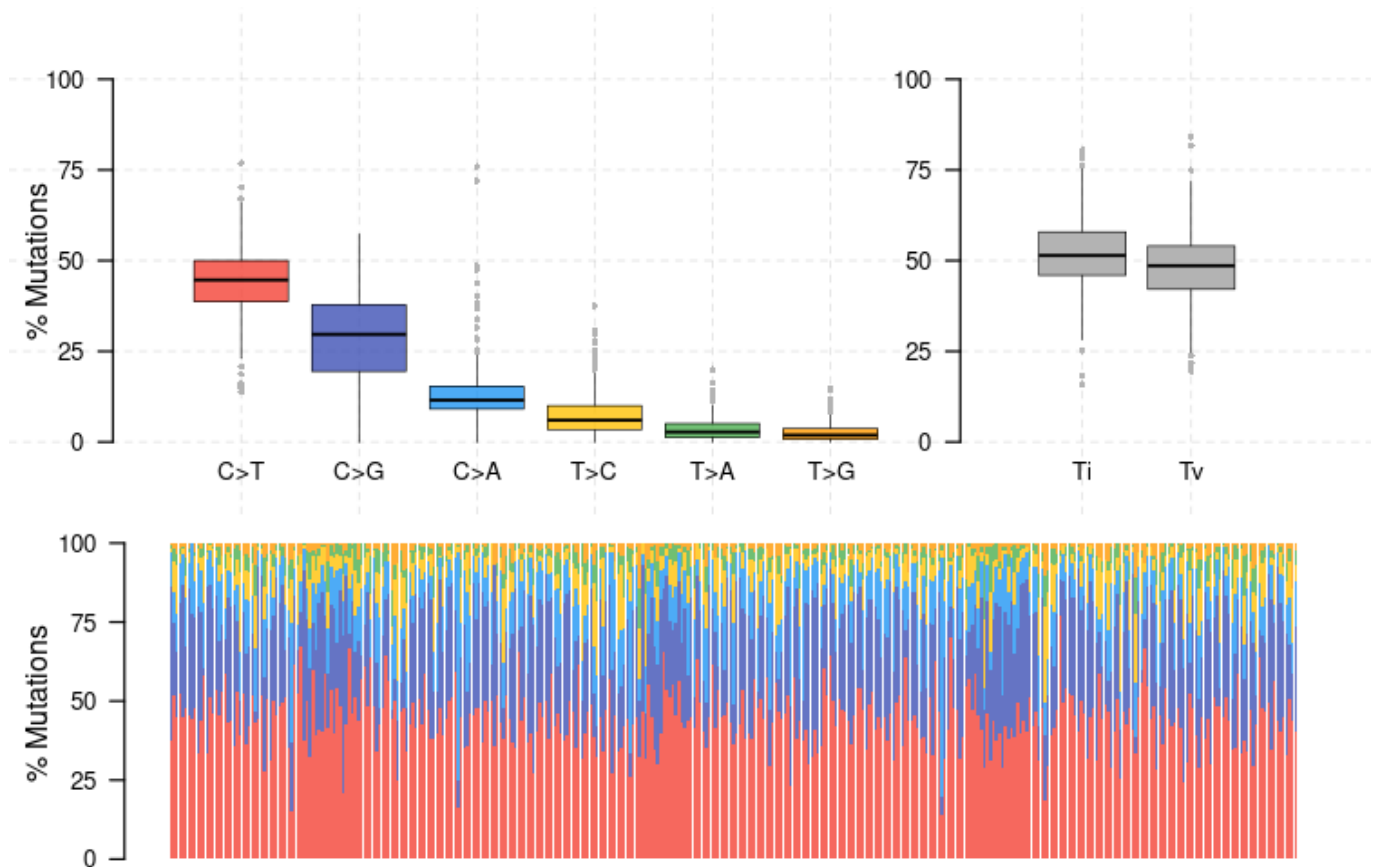
Use the next command to get the summary plot.

```
plotmafSummary(maf=BLCA, addStat = 'median', dashboard = TRUE)
```



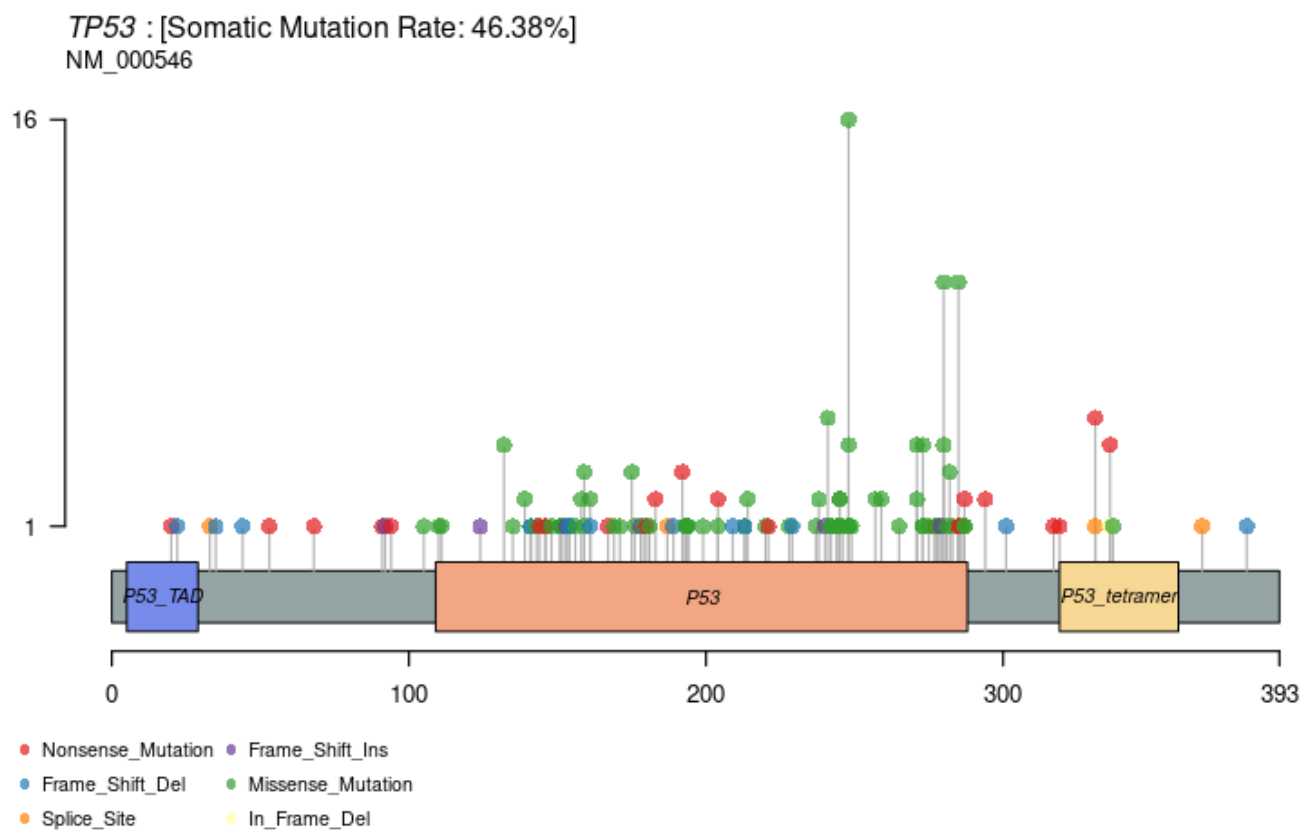
We can also generate a specific plot for transitions and transversions. To do so, we first get the transition and transversion data with "titv" and then plot it with "plotTiTv".

```
BLCA_titv = titv(maf = BLCA, plot=FALSE)
plotTiTv(res=BLCA_titv)
```



Another plot that we can create is a lollipop plot. This plot shows the positions and amino acid changes that are caused in the protein by the mutations present. You can create this type of plot for your protein of interest with the following command:

```
#We will use TP53 as an example  
lollipopPlot(maf=BLCA, gene="TP53", showMutationRate=TRUE)
```



You can label the changes by using the option "labelPos" and stating the position of the change you would like to annotate.

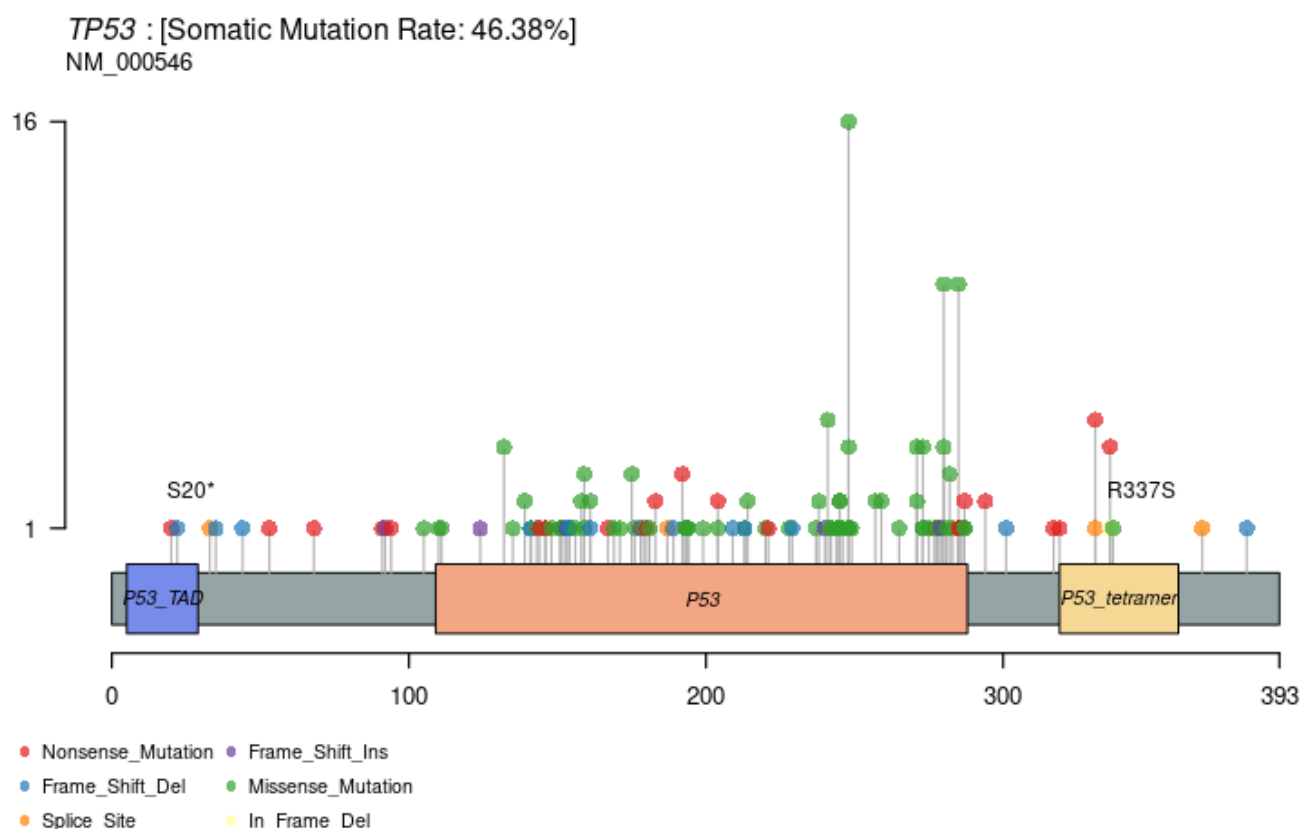
Here's an example.

```
#Here, we will label teh change in the position 20.
lollipopPlot(maf=BLCA, gene="TP53", showMutationRate=TRUE, labelPos=20)
```



You can label as many positions as you want.

```
lollipopPlot(maf=BLCA, gene="TP53", showMutationRate=TRUE,  
labelPos=c(20, 337))
```



An important thing to consider for these plots is that to be able to create them, you need the information of the change of amino acid to be included in your maf file. As well, the name of the column for this information can vary between maf files. By default maftools will look for the name "AAChange". If the column has another name, you should use the option "AACol" to state the name used in your maf file.

## When you have per sample mafs

This is not the case for this practical, but it is common to have per sample mafs. In these cases you can merge them together using maftools and create your plots as above. If you need to merge several maf files you can use the following commands.

```
mafs = list.files(path = "/directory/maf_files/", pattern = "*.\\..maf$",
full.names = TRUE)
all_samples = merge_mafs(mafs = mafs)
```

In this command you must indicate the path to the directory where you have your files. The names will be stored in "mafs" and you can then use this with "merge\_mafs" option of maftools. In this example the final merged maf is stored in "all\_samples".

### References:

Mayakonda A, Lin D, Assenov Y, Plass C, Koeffler PH (2018). Maftools: efficient and comprehensive analysis of somatic variants in cancer. Genome Research. doi: 10.1101/gr.239244.118.

Sangster, A. G., Gooding, R. J., Garven, A., Ghaedi, H., Berman, D. M., & Davey, S. K. (2022). Mutually exclusive mutation profiles define functionally related genes in muscle invasive bladder cancer. *PloS one*, 17(1), e0259992. <https://doi.org/10.1371/journal.pone.0259992>