Class 18: Mutational Signatures

Aparajita Pranjal

6/2/23

1. Exploring a cancer sequencing data portal

I chose to work on **lung squamous cell carcinomas.**

- **Q.** How many cancer samples are included in the dataset? **487 samples**
- **Q.** Which is the most mutated gene? **TP53**
- **Q.** Which is the most common treatment undergone by patients? **Cisplatin**

2. Downloading cancer sequencing data

Downloaded lusc data from cBioPortal and added to class18 folder.

3. Generating mutational matrices and visualizing mutational profiles

```
#BiocManager::install("maftools")
#BiocManager::install("BSgenome.Hsapiens.UCSC.hg19")
#BiocManager::install("MutationalPatterns")
```

```
# Read maf file
library(maftools)
coad = read.maf('data_mutations.txt')
```

```
-Reading
-Validating
--Removed 14038 duplicated variants
-Silent variants: 60539
-Summarizing
--Possible FLAGS among top ten genes:
  TTN
  MUC16
  USH2A
  SYNE1
-Processing clinical data
--Missing clinical data
-Finished in 18.3s elapsed (14.6s cpu)
```

```r
# Generate mutational matrix (SBS96 context)
mm_coad = trinucleotideMatrix(maf = coad, prefix = 'chr', add = TRUE,
                              ref_genome = "BSgenome.Hsapiens.UCSC.hg19")
```

```
Warning: replacing previous import 'S4Arrays::read_block' by
'DelayedArray::read_block' when loading 'SummarizedExperiment'


Attaching package: 'BiocGenerics'


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

    IQR, mad, sd, var, xtabs


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

    anyDuplicated, aperm, append, as.data.frame, basename, cbind,
    colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
    get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
    match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
    Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
    table, tapply, union, unique, unsplit, which.max, which.min


Attaching package: 'S4Vectors'
```

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

    findMatches


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

    expand.grid, I, unname



Attaching package: 'Biostrings'


The following object is masked from 'package:base':

    strsplit


-Extracting 5' and 3' adjacent bases
-Extracting +/- 20bp around mutated bases for background C>T estimation
-Estimating APOBEC enrichment scores
--Performing one-way Fisher's test for APOBEC enrichment
---APOBEC related mutations are enriched in  30.128 % of samples (APOBEC enrichment score >
-Creating mutation matrix
--matrix of dimension 469x96
```

```
mm_coad = t(mm_coad$nmf_matrix)
```

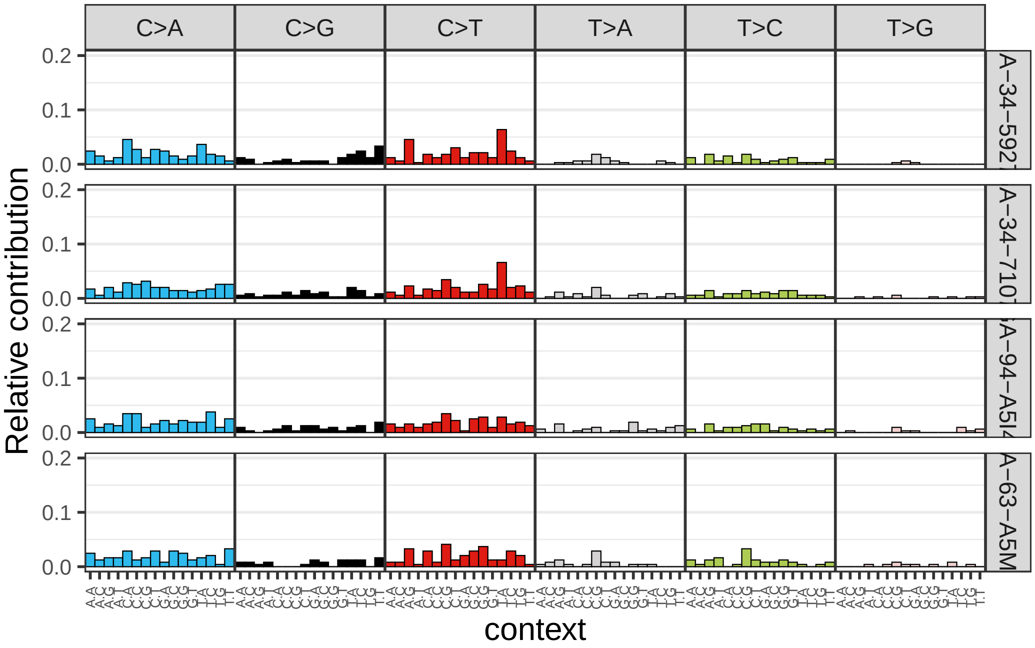
Generating mutational profiles from lusc data:

```
# Generate mutational profiles (4 random samples)
library(MutationalPatterns)
```

```
Loading required package: NMF


Loading required package: registry


Loading required package: rngtools


Loading required package: cluster


NMF - BioConductor layer [OK] | Shared memory capabilities [NO: bigmemory] | Cores 2/2
```

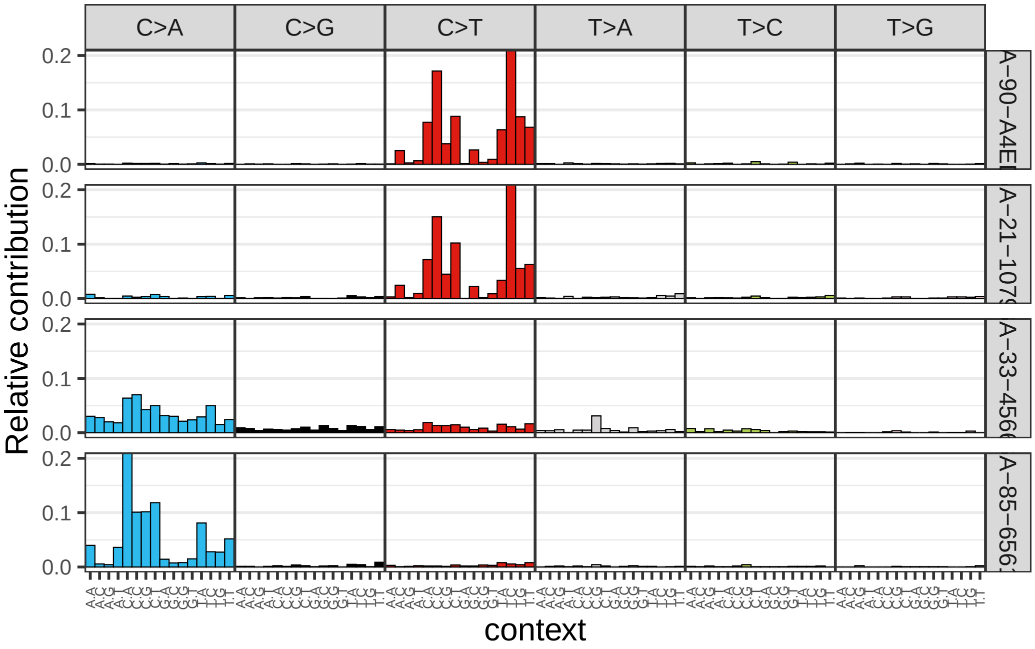
```
  To enable shared memory capabilities, try: install.extras('
NMF
')


Attaching package: 'NMF'


The following object is masked from 'package:S4Vectors':

    nrun
```

```r
set.seed(11111) # fixing the seed for random number generation

samples_to_plot = sample(1:ncol(mm_coad),4) # selecting 4 random samples
plot_96_profile(mm_coad[,samples_to_plot], condensed = T)
```



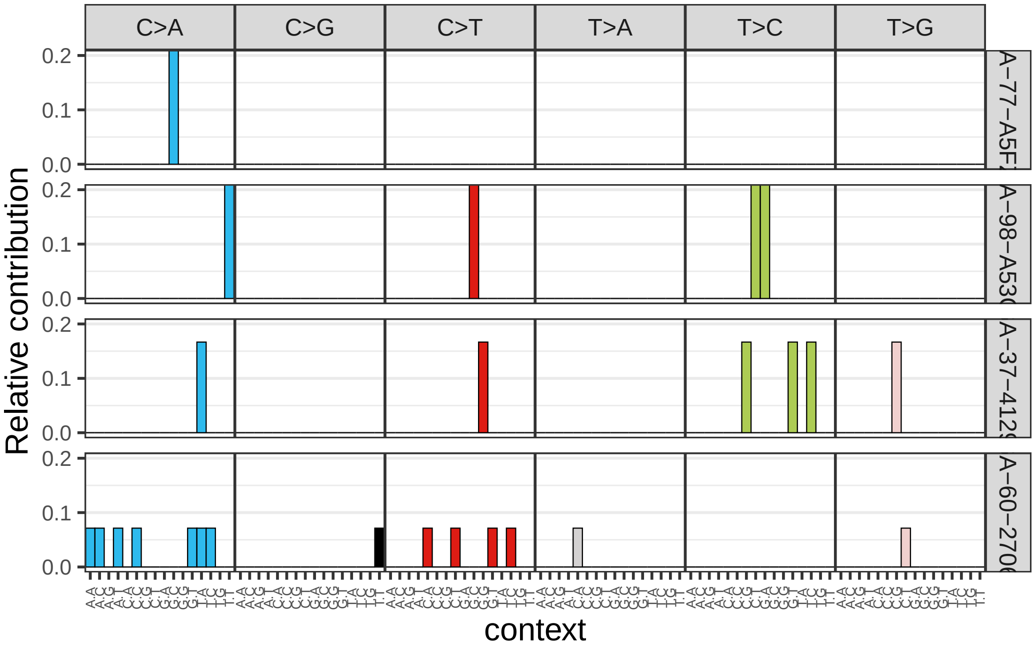
```r
# Generate mutational profiles (top 4 mutated samples and top 4 less mutated)
mutations_in_samples = colSums(mm_coad)
mutations_in_samples = sort(mutations_in_samples, decreasing = T)
samples_to_plot = names(mutations_in_samples)[1:4]
```

```
plot_96_profile(mm_coad[,samples_to_plot], condensed = T)
```



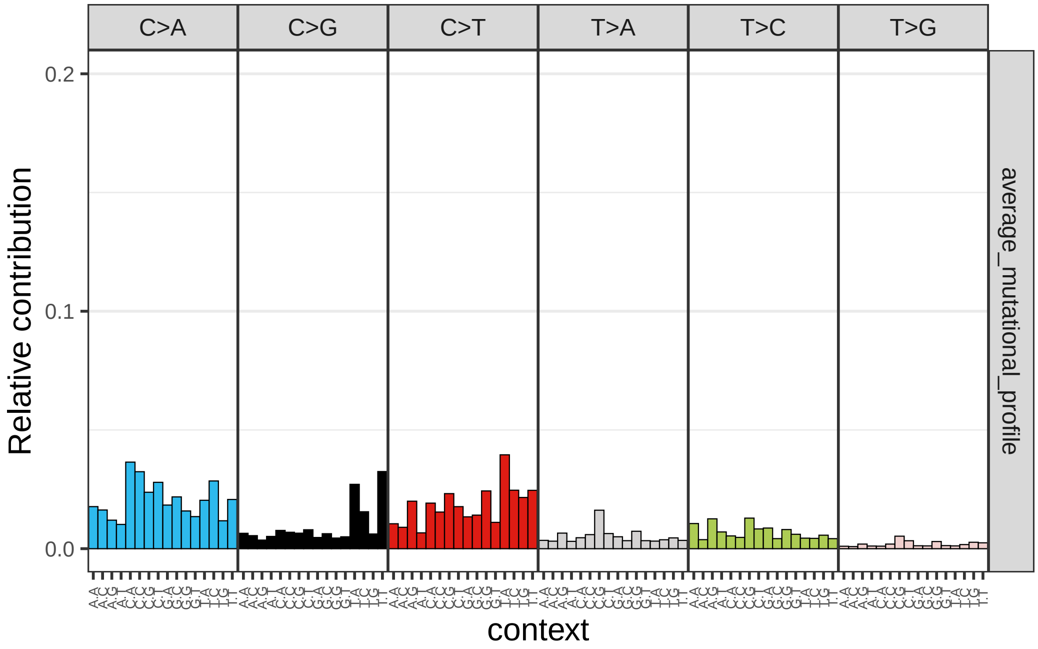
Sorting profiles:

```
mutations_in_samples = sort(mutations_in_samples, decreasing = F)
samples_to_plot = names(mutations_in_samples)[1:4]
plot_96_profile(mm_coad[,samples_to_plot], condensed = T)
```



Averaging profiles:

```
# Generate average mutational profiles
relative_mutational_profile = apply(mm_coad, 2, prop.table) # obtained relative
                                                            # mutational matrix
average_mutational_profile = rowMeans(relative_mutational_profile)
average_mutational_profile = data.frame(average_mutational_profile)
plot_96_profile(average_mutational_profile, condensed = T)
```



4. COSMIC reference mutational signatures

5. Assigning reference mutational signatures

```
# Mutational signature assignment
cosmic_signatures = get_known_signatures(source = 'COSMIC_v3.2')
fit_res = fit_to_signatures(mm_coad, cosmic_signatures)

# Top contributing signatures
contributions = fit_res$contribution

top_contributing_signatures_abs = rowMeans(contributions)
top_contributing_signatures_abs = sort(top_contributing_signatures_abs,
                                       decreasing = T)[1:4]

## Top 4 contributing signatures (absolute values)
top_contributing_signatures_abs
```

```
    SBS4    SBS24    SBS39    SBS13
104.9223  30.2159  29.8143  26.5996
```

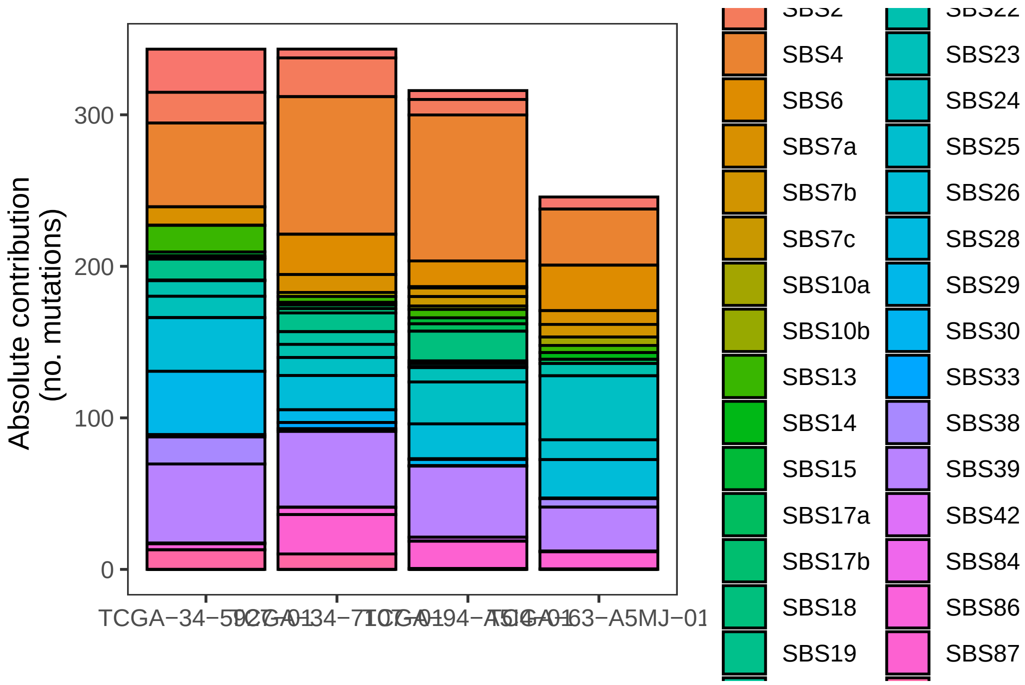
```
relative_contributions = apply(contributions,2,prop.table)
top_contributing_signatures_rel = rowMeans(relative_contributions)
top_contributing_signatures_rel = sort(top_contributing_signatures_rel,
                                       decreasing = T)[1:4]

## Top 4 contributing signatures (relative values)
top_contributing_signatures_rel
```

```
      SBS4      SBS24      SBS39      SBS13 
0.23615327 0.08534249 0.08064435 0.06469107 
```

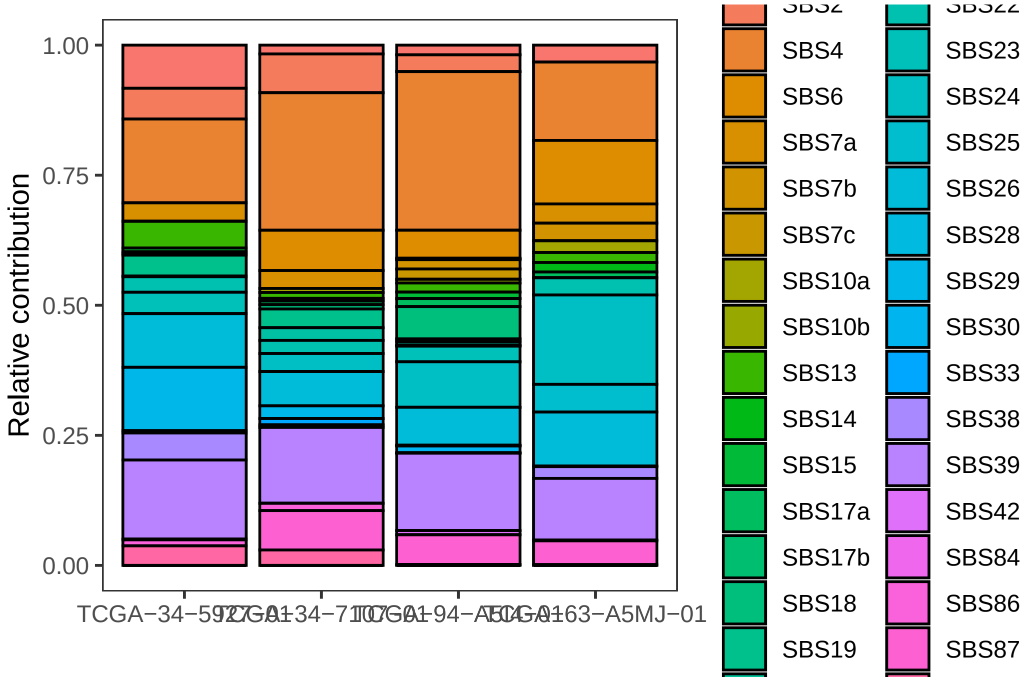
```
# Mutational signature assignment strict
fit_res_strict = fit_to_signatures_strict(mm_coad, cosmic_signatures)
fit_res_strict = fit_res_strict$fit_res
contributions_strict = fit_res_strict$contribution
```

6. Visualizing mutational signature assignment results

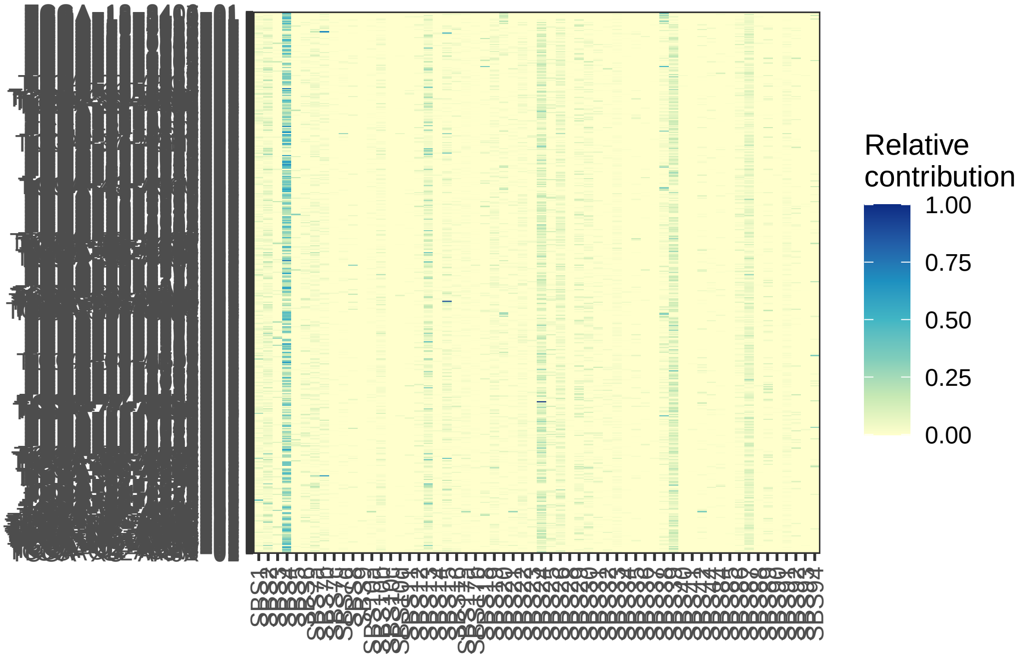
```
# Visualization of signature assignment results (fit_to_signatures)
set.seed(11111)
samples_to_plot = sample(1:ncol(mm_coad),4)

plot_contribution(contributions[,samples_to_plot], mode = "absolute")
```



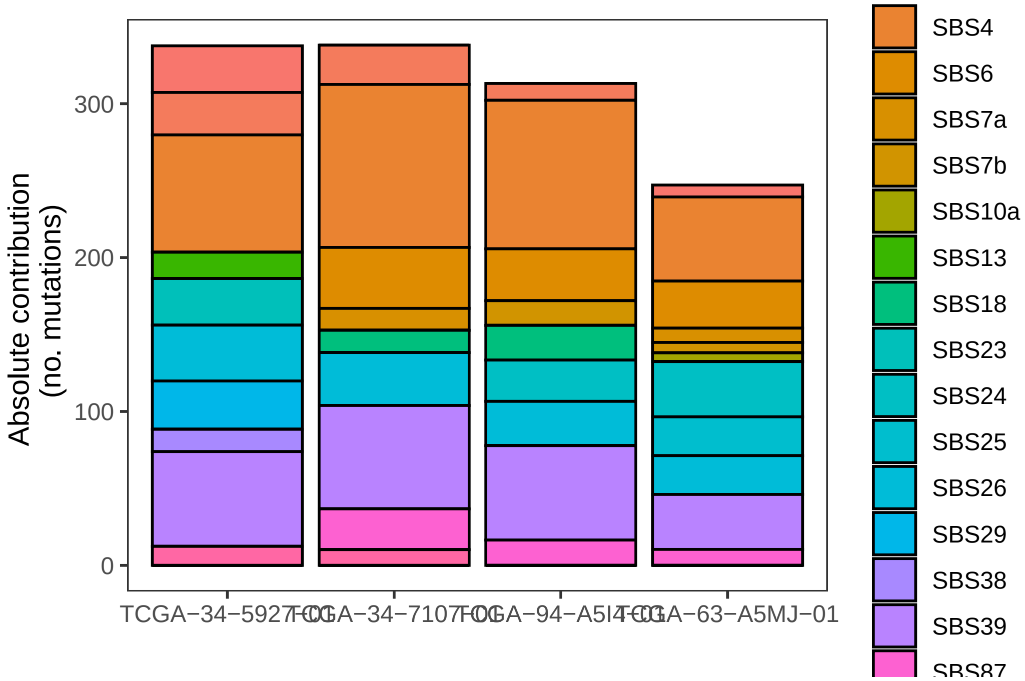
```
plot_contribution(contributions[,samples_to_plot], mode = "relative")
```



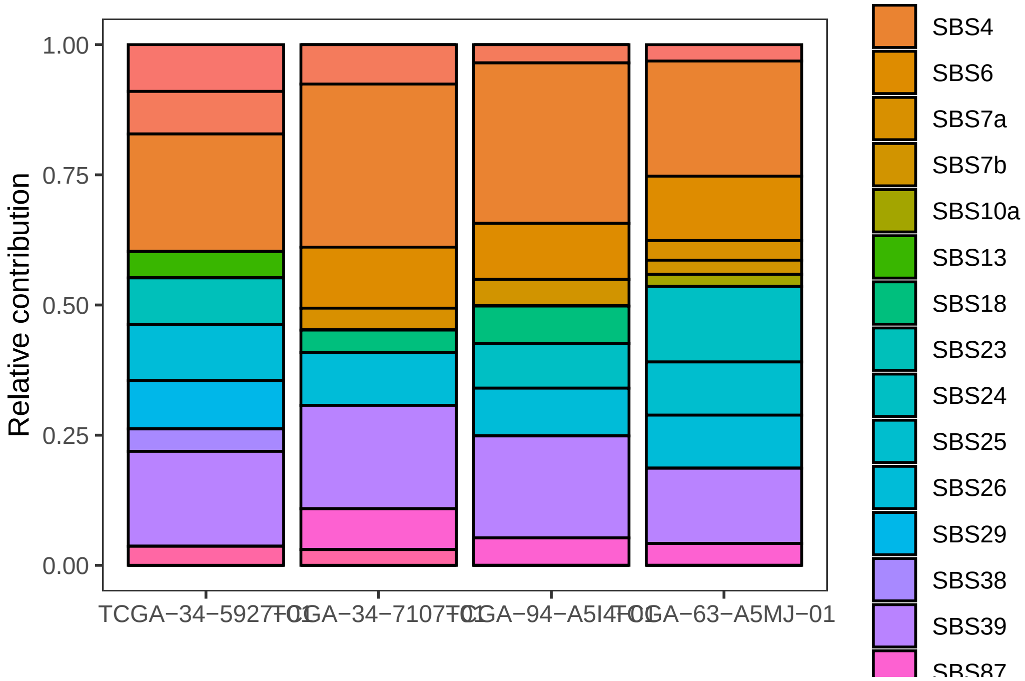
```
plot_contribution_heatmap(contributions, cluster_samples = F)
```



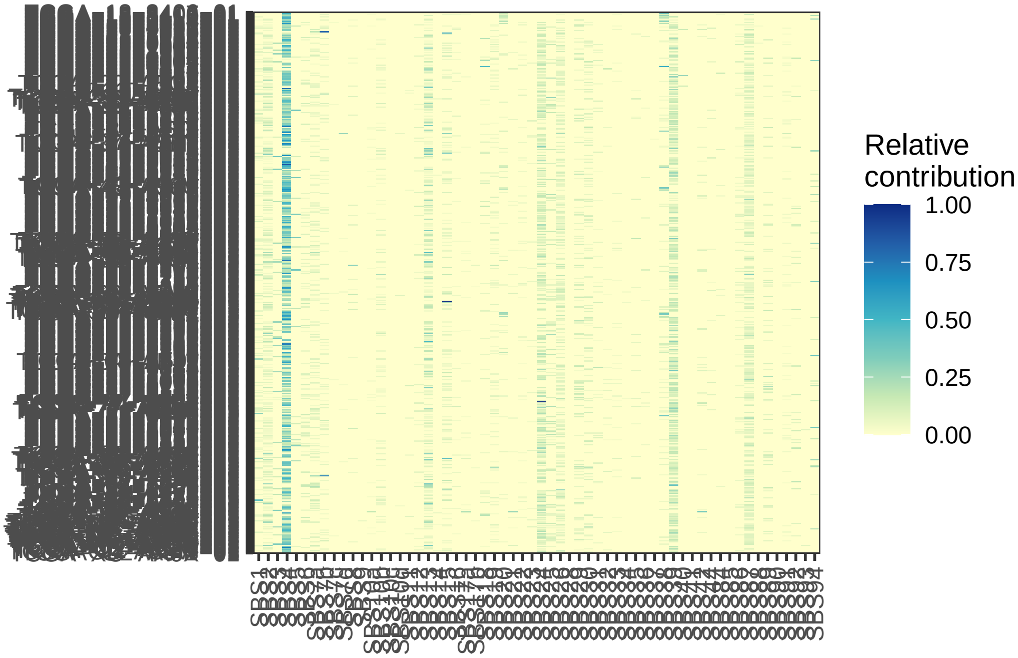
```
# Visualization of signature assignment results (strict)
plot_contribution(contributions_strict[,samples_to_plot], mode = "absolute")
```



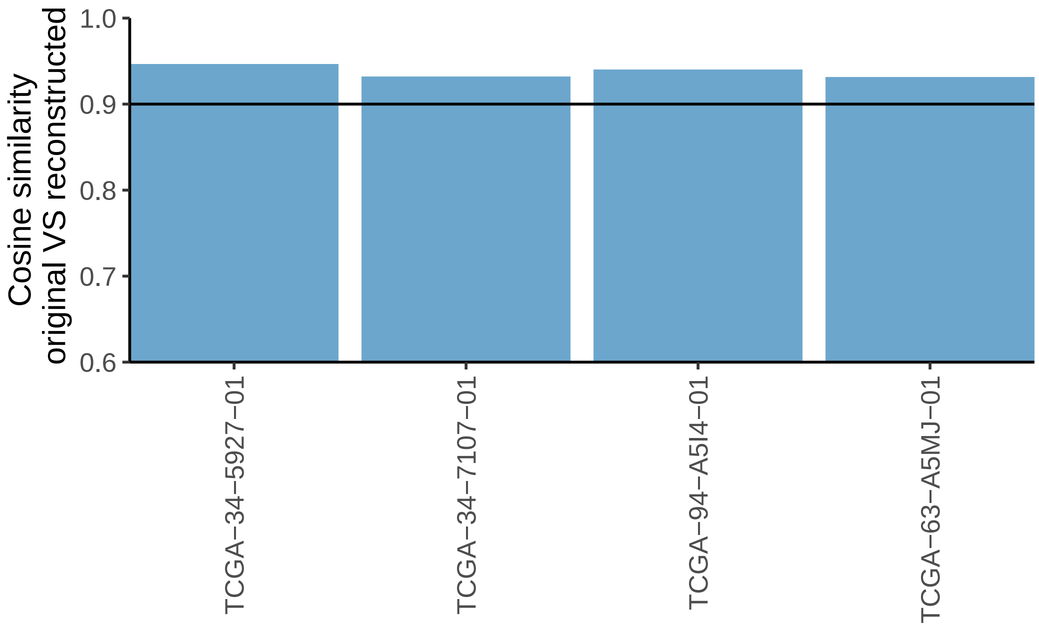
```
plot_contribution(contributions_strict[,samples_to_plot], mode = "relative")
```



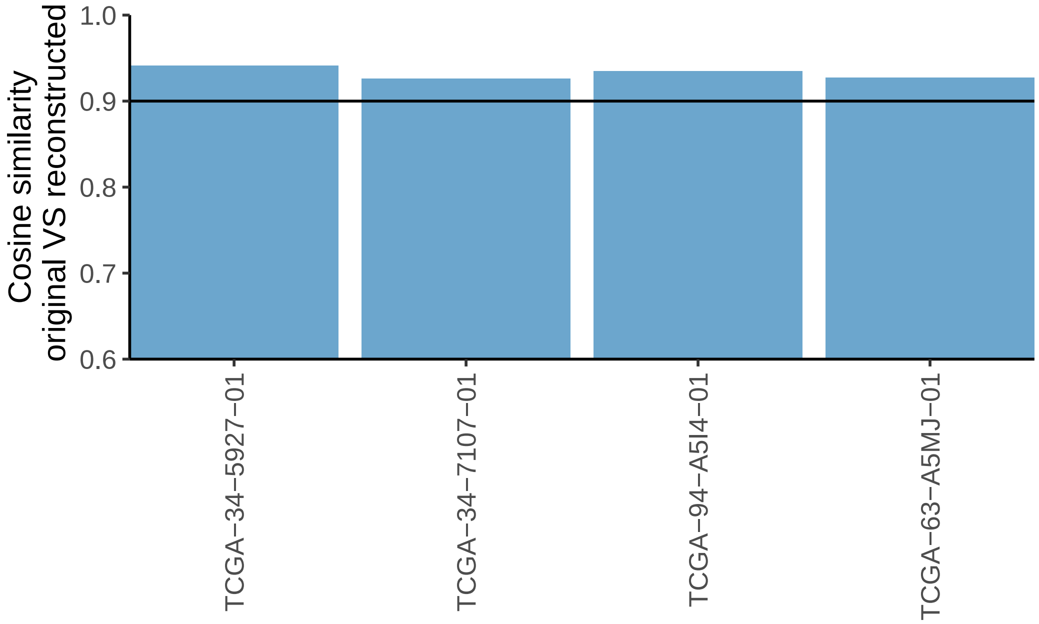
```
plot_contribution_heatmap(contributions_strict, cluster_samples = F)
```



```
# Cosine similarity reconstruction vs. original mutational profile (fit_to_signatures)
set.seed(11111)
samples_to_plot = sample(1:ncol(mm_coad),4)

plot_original_vs_reconstructed(mm_coad[,samples_to_plot],
                               fit_res$reconstructed[,samples_to_plot],
                               y_intercept = 0.90)
```



```
# Cosine similarity reconstruction vs. original mutational profile (strict)
plot_original_vs_reconstructed(mm_coad[,samples_to_plot],
                               fit_res_strict$reconstructed[,samples_to_plot],
                               y_intercept = 0.90)
```



- **Q.** Which is the etiology of the top absolute contributing signature for liver cancer? **Aristolochic acid exposure**
- **Q.** Which is the most prominent mutational context for the top contributing signature in skin cancer? **C>T**
- **Q.** The etiology of the top contributing signature for lung cancer corresponds to an endogenous cellular mechanism. **FALSE**
- **Q.** SBS4 is one of the most common signatures found in lung cancer and is associated with tobacco smoking. **TRUE**
- **Q.** SBS7d is one of the most common signatures in skin cancer and is associated with UV light exposure and high numbers of C>T mutations. **FALSE**

7. Advanced mutational signature analysis using the SigProfilerAssignment python package [OPTIONAL]

```
# Install R wrapper (SigProfilerAssignmentR)
#if (!require("devtools")){
#    install.packages("devtools")
#}
```

```
#devtools::install_github("AlexandrovLab/SigProfilerAssignmentR")
#devtools::install_github('marcos-diazg/utils.mdg')
```

```
# Generate mutational assignment analysis using SigProfilerAssignment
library(reticulate)
```

```
Attaching package: 'reticulate'


The following object is masked from 'package:rtracklayer':

    import
```

```
library(utils.mdg)
mm_coad_sigprofiler = map_MP_to_SP(mm_coad) # Order of rows is different in
                                            # SigProfiler vs. MutationalPatterns
library(SigProfilerAssignmentR)
```

```
#cosmic_fit(mm_coad_sigprofiler, './results', export_probabilities = F)
# Output results folder will be in the new "results" directory inside
# your current working directory
```