

# doomsayer\_\_diagnostics

*jed c.*

*2017-07-01*

```
# knitr::opts_chunk$set(echo = TRUE)

#specify the packages of interest
packages = c("dplyr","tidyr","ggplot2","yaml", "devtools")

#use this function to check if each package is on the local machine
#if a package is installed, it will be loaded
#if any are not, the missing package(s) will be installed and loaded
package.check <- lapply(packages, FUN = function(x) {
  if (!require(x, character.only = TRUE)) {
    install.packages(x, dependencies = TRUE)
    library(x, character.only = TRUE)
  }
})

# cmd_args <- commandArgs(TRUE)
# yaml_cfg <- as.double(cmd_args[1])

# static path for development
# yaml_cfg <- "/mnt/c/Users/jedidiah/Dropbox/Github/doomsayer/demo/output/config.yaml"
yaml_cfg <- params$yaml_cfg

# debug--paths generated from doomsayer.py refer are formatted for bash on Windows
# the sys.info check updates these paths with Windows format for proper loading in R
if(Sys.info()['sysname']=="Windows"){
  yaml_cfg <- gsub("/mnt/c", "C:", yaml_cfg)
}

yaml_args <- yaml.load_file(yaml_cfg)
if(Sys.info()['sysname']=="Windows"){
  yaml_args <- lapply(yaml_args, function(x) gsub("/mnt/c", "C:", x))
}

attach(yaml_args)

## The following objects are masked from yaml_args (pos = 3):
##
##      drop_path, H_path, keep_path, M_path, M_path_rates, W_path
## The following objects are masked from yaml_args (pos = 4):
##
##      drop_path, H_path, keep_path, M_path, M_path_rates, W_path
## The following objects are masked from yaml_args (pos = 5):
##
##      drop_path, H_path, keep_path, M_path, M_path_rates, W_path
## The following objects are masked from yaml_args (pos = 6):
##
```

```

##      drop_path, H_path, keep_path, M_path, M_path_rates, W_path
## The following objects are masked from yaml_args (pos = 9):
##
##      drop_path, H_path, keep_path, M_path, M_path_rates, W_path
## The following objects are masked from yaml_args (pos = 10):
##
##      drop_path, H_path, keep_path, M_path, M_path_rates, W_path
## The following objects are masked from yaml_args (pos = 11):
##
##      drop_path, H_path, keep_path, M_path, M_path_rates, W_path
## The following objects are masked from yaml_args (pos = 12):
##
##      H_path, M_path, M_path_rates, W_path
## The following objects are masked from yaml_args (pos = 13):
##
##      H_path, M_path, W_path
## The following objects are masked from yaml_args (pos = 14):
##
##      H_path, M_path, W_path
## The following objects are masked from yaml_args (pos = 15):
##
##      H_path, M_path, W_path
## The following objects are masked from yaml_args (pos = 16):
##
##      H_path, M_path, W_path
## The following objects are masked from yaml_args (pos = 17):
##
##      H_path, M_path, W_path

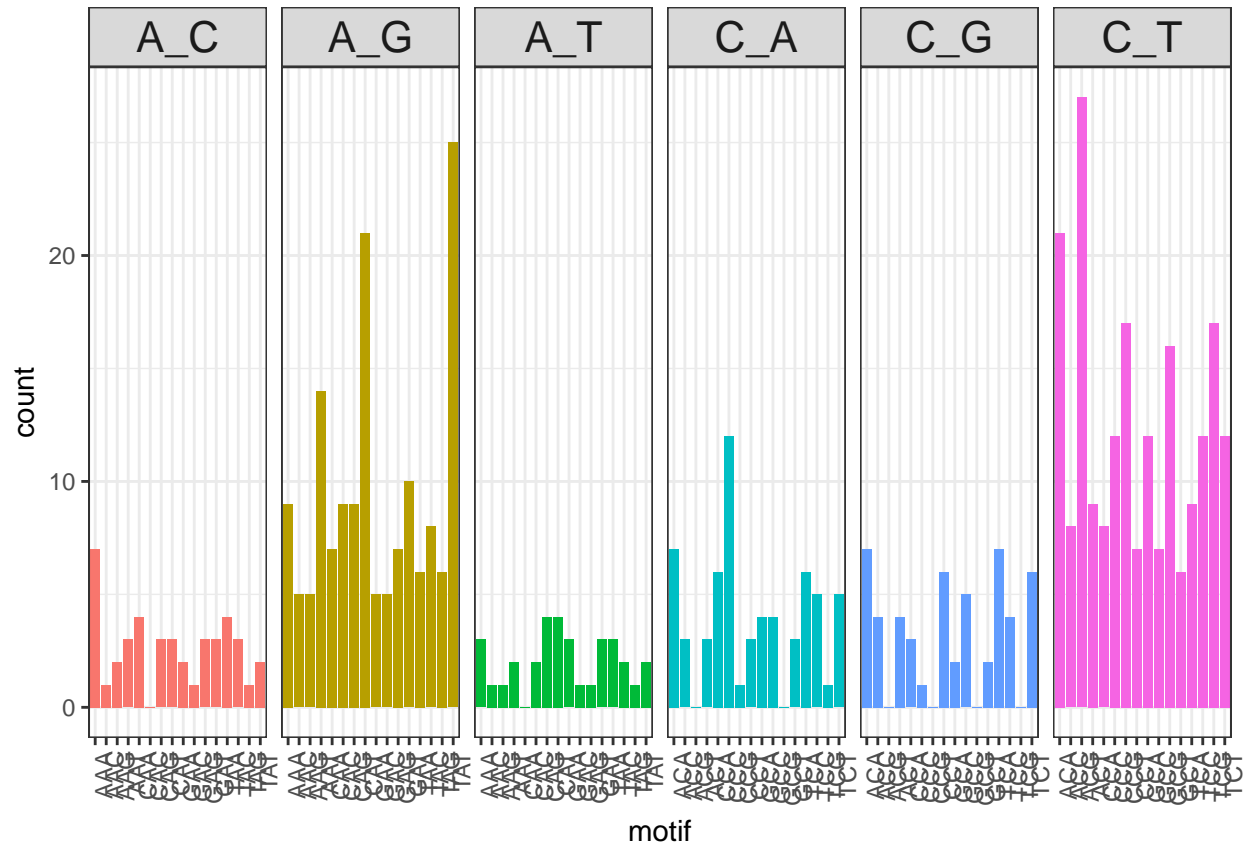
# read tables
keep_ids <- read.table(keep_path, header=F, stringsAsFactors=F)
# drop_ids <- read.table(drop_path, header=F, stringsAsFactors=F)
spectra <- read.table(M_path, header=T, stringsAsFactors=F)
spectra_rates <- read.table(M_path_rates, header=T, stringsAsFactors=F)
sig_contribs <- read.table(W_path, header=T, stringsAsFactors=F)
sig_loads <- read.table(H_path, header=T, stringsAsFactors=F)

# overall distribution
spectra2 <- spectra %>%
  gather(subtype, count, 2:ncol(spectra)) %>%
  separate(subtype, c("category", "motif"), sep = "[.]") %>%
  group_by(category, motif) %>%
  summarise(count=sum(count))

ggplot(spectra2, aes(x=motif, y=count, fill=category))+
  geom_bar(stat="identity")+
  facet_wrap(~category, ncol=6, scales="free_x")+
  theme_bw()+
  theme(axis.text.x=element_text(angle=90, hjust=1),

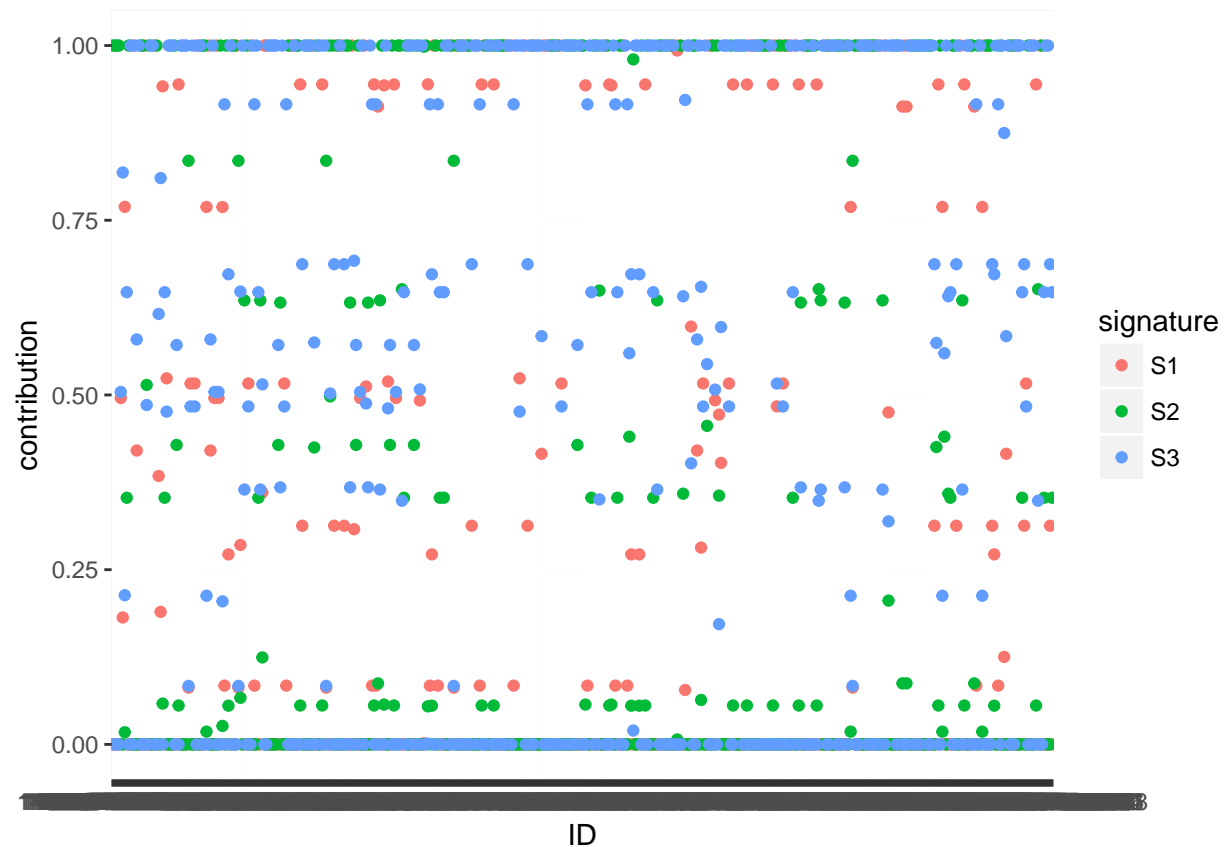
```

```
strip.text=element_text(size=16),
legend.position="none")
```



```
# signature contributions across samples
sig_contri2 <- sig_contri[complete.cases(sig_contri),]
sig_contri2 <- sig_contri2 %>%
  gather(signature, contribution, S1:S3)

ggplot(sig_contri2, aes(x=ID, y=contribution, colour=signature))+
  geom_point()
```



```
# signature loadings
sig_loads_long <- sig_loads %>%
  gather(subtype, loading, 2:ncol(sig_loads)) %>%
  separate(subtype, c("category", "motif"), sep = "[.]")

ggplot(sig_loads_long, aes(x=motif, y=loading, fill=Sig))+
  geom_bar(stat="identity")+
  facet_grid(Sig~category)+
  theme_bw()+
  theme(axis.text.x=element_text(angle=90, hjust=1),
        strip.text=element_text(size=16),
        legend.position="none")
```

