

## Visualizations LDA coefficients

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
##
```

```
## Attaching package: 'dplyr'
```

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

```
##
```

```
## filter, lag
```

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

```
##
```

```
## intersect, setdiff, setequal, union
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
library("ggpubr")
```

```
## Warning: package 'ggpubr' was built under R version 4.0.5
```

```
library(LDATS)
```

```
## Warning: package 'LDATS' was built under R version 4.0.5
```

```
library(ggVennDiagram)
```

```
## Warning: package 'ggVennDiagram' was built under R version 4.0.5
```

```
library(stringr)
```

```
library(abind)
```

```
## Warning: package 'abind' was built under R version 4.0.3
```

```
library(patchwork)
```

```
## Warning: package 'patchwork' was built under R version 4.0.3
```

```
source("utils.R")
```

```
## Warning: package 'hash' was built under R version 4.0.5
```

```
## hash-2.2.6.1 provided by Decision Patterns
```

```
## Warning: package 'reticulate' was built under R version 4.0.5
```

```
## Warning: package 'berryFunctions' was built under R version 4.0.5
```

```
##
```

```
## Attaching package: 'berryFunctions'
```

```
## The following object is masked from 'package:ggVennDiagram':
```

```
##
```

```
## circle
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
## between
```

```
## Warning: package 'purrr' was built under R version 4.0.3
```

```
## Warning: package 'reshape2' was built under R version 4.0.3
```

```
##
```

```
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
```

```
##
```

```
## smiths
```

```
base_dir <- "../data/data_train_val_c10"
```

```
repls <- 0:29
```

```
classes <- 10
```

```
nets_outputs <- load_network_outputs(base_dir, repls)
```

```
ens_outputs <- load_ensemble_outputs(base_dir, repls)
```

```
sort_ind <- function(lst)
```

```
{
```

```
  return(sort(lst, index.return=TRUE, decreasing=TRUE)$ix)
```

```
}
```

```
nets_test_top_indices <- apply(X=nets_outputs$test_outputs, MARGIN=c(1, 2, 3), FUN=sort_ind)[1, , ]
```

```
ens_tt_top_indices <- apply(X=ens_outputs$train_training, MARGIN=c(1, 2, 3), FUN=sort_ind)[1, , ]
```

```

ens_vt_top_indices <- apply(X=ens_outputs$val_training, MARGIN=c(1, 2, 3), FUN=sort_ind)[1, , , ]
r_n <- length(repls)
samples_n <- dim(nets_outputs$test_labels)[2]
nets_n <- length(nets_outputs$networks)
ens_n <- length(ens_outputs$methods)
test_labs <- nets_outputs$test_labels + 1
dim(test_labs) <- c(r_n, 1, samples_n)
test_labs_nets <- aperm(abind(array(rep(aperm(test_labs, perm=c(2, 1, 3))), nets_n), c(r_n, samples_n, n
test_labs_ens <- aperm(abind(array(rep(aperm(test_labs, perm=c(2, 1, 3))), ens_n), c(r_n, samples_n, ens
nets_test_cor_preds <- test_labs_nets == nets_test_top_indices
ens_tt_cor_preds <- test_labs_ens == ens_tt_top_indices
ens_vt_cor_preds <- test_labs_ens == ens_vt_top_indices

```

```

for (ri in 1:1)
{
  nets_cor_list <- list()
  incor <- 1:samples_n
  for (ni in 1:nets_n)
  {
    cor_list <- which(nets_test_cor_preds[ri, ni, ])
    nets_cor_list[[nets_outputs$networks[ni]]] = cor_list
    incor <- setdiff(incor, cor_list)
  }
  incor_n <- length(incor)

  for (ei in 1:ens_n)
  {
    cor_list_ens_tt <- which(ens_tt_cor_preds[ri, ei, ])
    cor_list_ens_vt <- which(ens_vt_cor_preds[ri, ei, ])

    cur_cor_list_tt <- nets_cor_list
    cur_cor_list_tt[[ens_outputs$methods[ei]]] = cor_list_ens_tt

    cur_cor_list_vt <- nets_cor_list
    cur_cor_list_vt[[ens_outputs$methods[ei]]] = cor_list_ens_vt

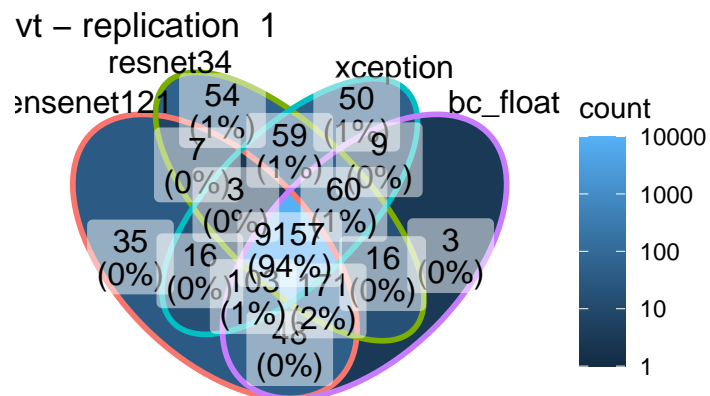
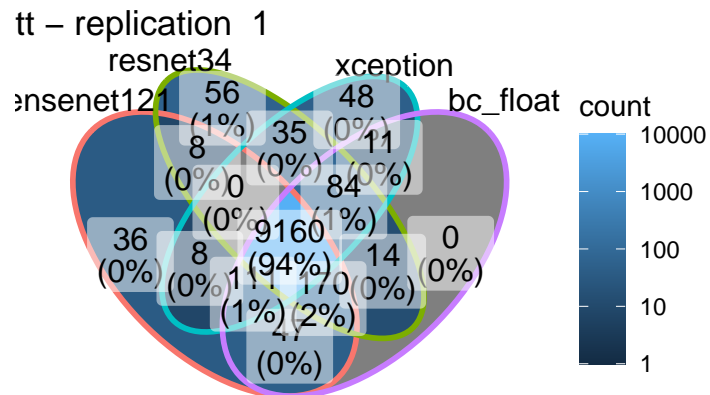
    venn_diag_tt <- ggVennDiagram(cur_cor_list_tt) + scale_fill_gradient(trans="log10", name="count", l
    ggtitle(paste("tt - replication ", ri)) +
      scale_x_continuous(limits=c(0, 1))

    venn_diag_vt <- ggVennDiagram(cur_cor_list_vt) + scale_fill_gradient(trans="log10", name="count", l
    ggtitle(paste("vt - replication ", ri)) +
      scale_x_continuous(limits=c(0, 1))

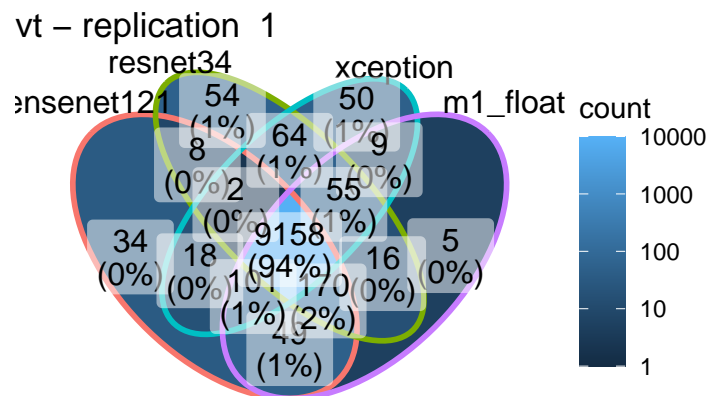
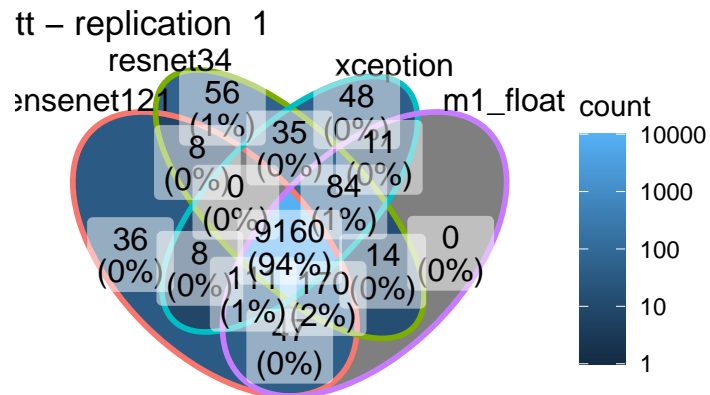
    print(venn_diag_tt / venn_diag_vt)
  }
}

```

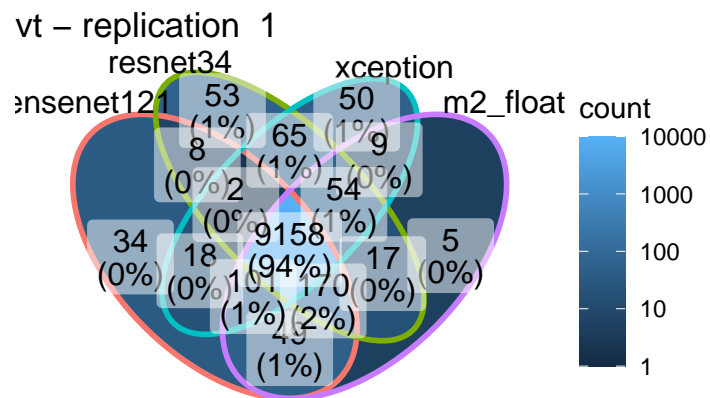
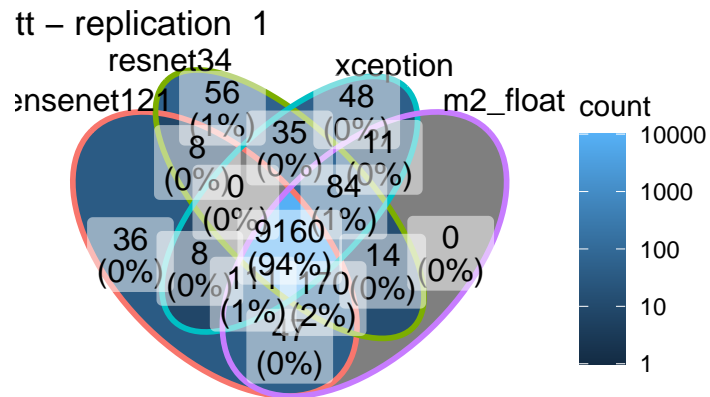
```
## Warning: Transformation introduced infinite values in discrete y-axis
```



## Warning: Transformation introduced infinite values in discrete y-axis

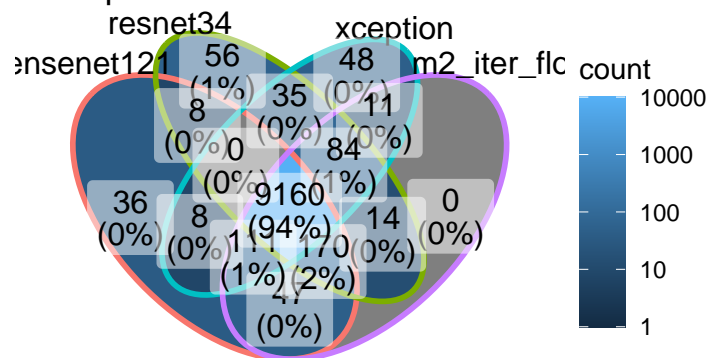


## Warning: Transformation introduced infinite values in discrete y-axis



## Warning: Transformation introduced infinite values in discrete y-axis

tt – replication 1



vt – replication 1

