Linear models for Kidera factors

```
library(data.table)
library(dplyr)
```

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

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

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

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(reshape2)
```

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

```
## The following objects are masked from 'package:data.table':
##
##     dcast, melt
```

```
library(ggplot2)
library(parallel)
library(RColorBrewer)
select = dplyr::select
```

Load data

```
pairs = fread('zcat tcr_ab_pairs.txt.gz') %>%
  mutate(cdr3 = aaSeqCDR3, gene = substr(allVGenes, 1, 3)) %>%
  select(sample, clone, gene, cdr3) %>%
  filter(startsWith(cdr3, "C"),
         endsWith(cdr3, "F") | endsWith(cdr3, "W"),
         gene %in% c("TRA", "TRB")) %>%
  as.data.table
```

```
##
Read 71.3% of 589472 rows
Read 589472 rows and 13 (of 13) columns from 0.058 GB file in 00:00:03
```

```
# filter upaired
paired = pairs %>%
  group_by(sample, clone) %>%
  summarise(count = n()) %>%
  filter(count == 2) %>%
  select(sample, clone) %>%
  ungroup %>%
  as.data.table
```

```
pairs = pairs %>%
  merge(paired)
```

Compute Kidera factors for CDR3 sequences. Note that we don't divide by CDR3 length L, I've tried this and actually got larger absolute values of C_L coefficient for all linear models (except for $L \sim ... + L$). This means that by dividing by length we rather introduce additional dependency instead of removing it.

```
kidera = fread("prop_kidera.txt") %>%
  mutate(len = 1) %>%
  melt
```

```
## Using aa as id variables
```

```
cdr_flat = pairs$cdr3 %>%
  unique %>%
  strsplit("") %>%
  mclapply(function(x)
    data.table(cdr3 = paste0(x, collapse = ""), aa = x),
    mc.cores = 80) %>%
  rbindlist

cdr_kidera = cdr_flat %>%
  merge(kidera, by = "aa", allow.cartesian = T) %>%
  group_by(variable, cdr3) %>%
  summarise(value = sum(value)) %>%
  as.data.table
```

Append pairing info

```
pairs_kidera = pairs %>%
  merge(cdr_kidera, by = "cdr3", allow.cartesian = T) %>%
  dcast(sample+clone~gene+variable, value.var = "value")
```

Linear regression

```
vars = colnames(pairs_kidera)
vars = vars[grep("TR", vars)]

compute_mdl = function(response) {
  gene = substr(response, 1, 3)
  predictors = vars[!grepl(gene, vars)]

  eq = response %>%
    paste(predictors %>% paste0(collapse = " + "),
          sep = " ~ ")

  res = lm(as.formula(eq), pairs_kidera) %>% summary
  coef = res$coefficients
  predictor = rownames(coef)

  out = coef %>%
    as.data.table
  colnames(out) = c("coef", "coef.SD", "coef.T", "coef.P")

  out %>%
    mutate(response.gene = gene,
           response = response,
```

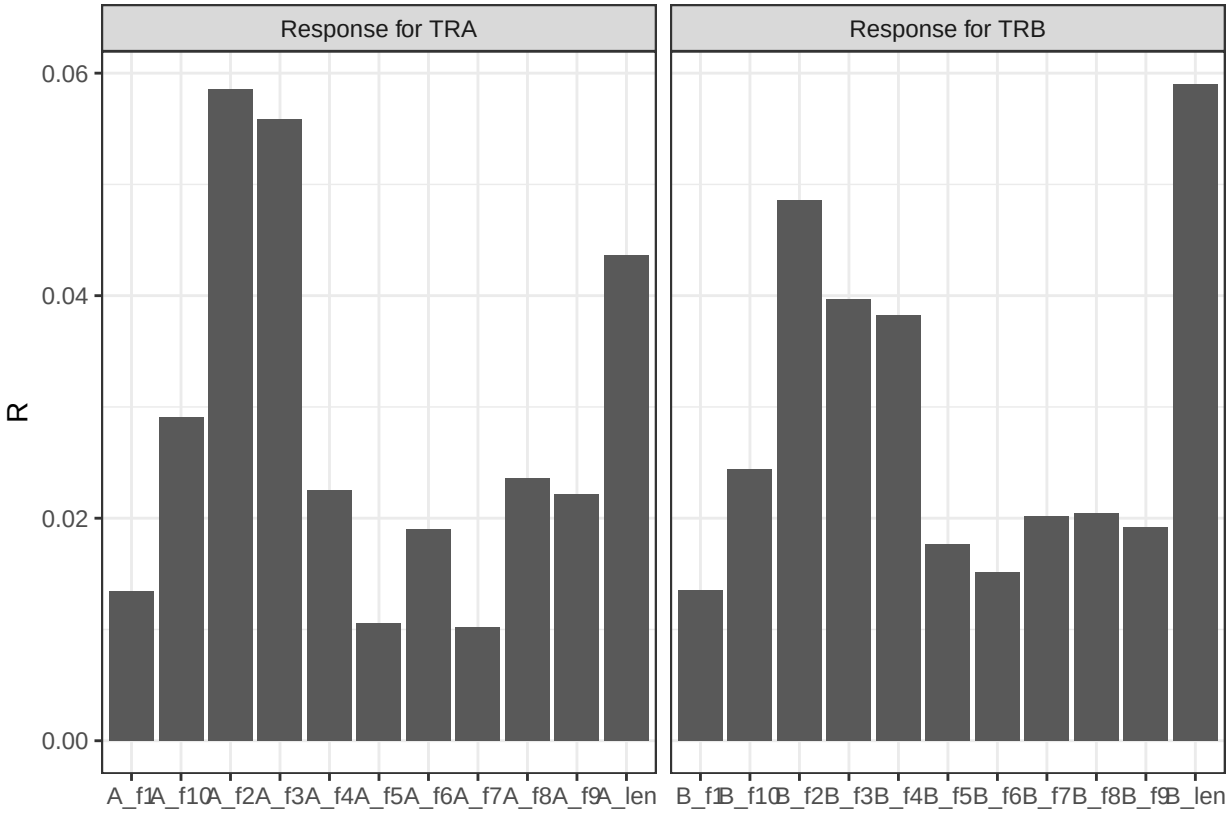
```
            predictor = predictor,
            R = sqrt(res$r.squared)) %>%
    as.data.table
}

lmres = vars %>%
  lapply(function(x) compute_mdl(x)) %>%
  rbindlist

lmres$set = paste("Response for", lmres$response.gene)
```

Plot results

```
ggplot(lmres %>% select(set, response, R) %>% unique,
       aes(x = gsub("TR", "", response),
           y = R)) +
  geom_bar(stat = "identity") +
  xlab("") + ylab("R") +
  facet_wrap(~set, scales = "free_x") +
  theme_bw()
```




```
ggplot(lmres %>% filter(predictor != "(Intercept)"),
       aes(x = gsub("TR", "", response),
           y = gsub("TR", "", predictor),
           fill = coef)) +
  geom_tile() +
  geom_tile(data = lmres %>% filter(predictor != "(Intercept)", coef.P < 0.05),
            color = "black") +
  xlab("") + ylab("") +
```

```
  scale_fill_gradientn(colors=colorRampPalette(rev(brewer.pal(11, 'RdYlBu')))(32),
                       limits = c(-0.06, 0.06)) +
  facet_wrap(~set, scales = "free") +
  theme_bw() +
  theme(legend.position = "bottom")
```