Case Study 1

Najeeb Khan, Hafik Arhan Kamac, Swaraj Oturkar, Elif Erbil

12/4/2018

Introduction

In this case study we have examined the relations between a genome, gene, expression rate and growth rate in a set of yeast segregants and environments. We used the data given to us to answer questions such as the relation between genotype and environment on the growth rate of the yeast, which can also be related to the expression of the genes resulting in their translation and production of proteins. In order to analyze our data, we first tidied each table so that each column has a unique type of variable and each row has an observation. We will be presenting the methods we used to tidy the data and the analysis we made in this report.

Tidying Data

Methods introduced in the lecture such as melting the data to combine columns that depict the same variable and casting the data to form multiple columns out of a given column are extensively used in for tidying up the data.

```
## Reading Datasets
gene_file <- 'Data/eqtl/gene.txt'
expression_file <- 'Data/eqtl/expression.txt'
genotype_file <-'Data/eqtl/genotype.txt'
growth_file <- 'Data/eqtl/growth.txt'
marker_file <- 'Data/eqtl/marker.txt'

gene <- as.data.table(read.delim(gene_file))
expression <- as.data.table(read.delim(expression_file, comment.char = "#"))
genotype <- as.data.table(read.delim(genotype_file))
growth <- as.data.table(read.delim(growth_file))
marker <- as.data.table(read.delim(marker_file))

name <- gene[,.(name)]
type <- gene[,.(type)]
expression[, gene := name][, gene_type := type]

# Tidying expression.txt file
expression_tidy <- gather(expression, medium_strand, expression_rate, c("YPD.seg_01B":"YPMalt.seg_45C"))
expression_tidy <- as.data.table(expression_tidy)
expression_tidy[, medium_strand := .(gsub("_", "", medium_strand))]
head(expression_tidy, n = 10)
```

```
##          gene gene_type medium_strand expression_rate
##  1: SY_A0001W       SUT    YPD.seg01B     -2.02576611
##  2:    SUT001       SUT    YPD.seg01B     -2.02716467
##  3: SY_A0003W       SUT    YPD.seg01B     -0.44516559
##  4: SY_A0004W       SUT    YPD.seg01B      1.05709276
##  5: SY_A0005W       SUT    YPD.seg01B     -2.02912397
##  6:   YAL062W     ORF-T    YPD.seg01B     -0.05015662
```

```
##  7:   YAL062W     ORF-T    YPD.seg01B     -1.22395849
##  8: SY_A0008W       SUT    YPD.seg01B     -1.49782194
##  9:   YAL061W     ORF-T    YPD.seg01B      1.96513834
## 10: SY_A0010W       SUT    YPD.seg01B      0.46947972
```

```
expression_tidy <- as.data.table(separate(as.data.frame(expression_tidy), medium_strand, into = c("medi

# Tidied version of expression data
head(expression_tidy)
```

```
##         gene gene_type medium strand expression_rate
## 1: SY_A0001W       SUT    YPD seg01B     -2.02576611
## 2:    SUT001       SUT    YPD seg01B     -2.02716467
## 3: SY_A0003W       SUT    YPD seg01B     -0.44516559
## 4: SY_A0004W       SUT    YPD seg01B      1.05709276
## 5: SY_A0005W       SUT    YPD seg01B     -2.02912397
## 6:   YAL062W     ORF-T    YPD seg01B     -0.05015662
```

```
# Tidying growth.txt file
growth_tidy <- melt(growth, id.vars = 'strain', variable.name = 'env', value.name = 'rate')
head(growth_tidy)
```

```
##     strain env     rate
## 1: seg_01B YPD 12.60399
## 2: seg_01C YPD 10.79114
## 3: seg_01D YPD 12.81727
## 4: seg_02B YPD 10.29921
## 5: seg_02C YPD 11.13278
## 6: seg_02D YPD 13.91084
```

```
# Tidying genotype.txt file
names <- c("strain")
for(i in 1:1000)
  names <- c(names, paste("mrk", i, sep = "_"))
colnames(genotype) <- names
print(colnames(genotype)[1:10])
```

```
##  [1] "strain" "mrk_1"  "mrk_2"  "mrk_3"  "mrk_4"  "mrk_5"  "mrk_6"
##  [8] "mrk_7"  "mrk_8"  "mrk_9"
```

```
# Tidying markers.txt file
marker[, id := colnames(genotype)[2:1001]]
print(marker[, id][1:10])
```

```
##  [1] "mrk_1"  "mrk_2"  "mrk_3"  "mrk_4"  "mrk_5"  "mrk_6"  "mrk_7"
##  [8] "mrk_8"  "mrk_9"  "mrk_10"
```

Both `growth.txt` and `expression.txt` contained data points where the value of growth and expression respectively was depicted in different columns. The tables were tidied to the form shown above. There exists 1000 markers in the dataset (as can be seen in `markers.txt` as well as `genotype.txt`). The names of the markers were changed so that they have a coherent name ranging from mrk_1 to mrk_1000

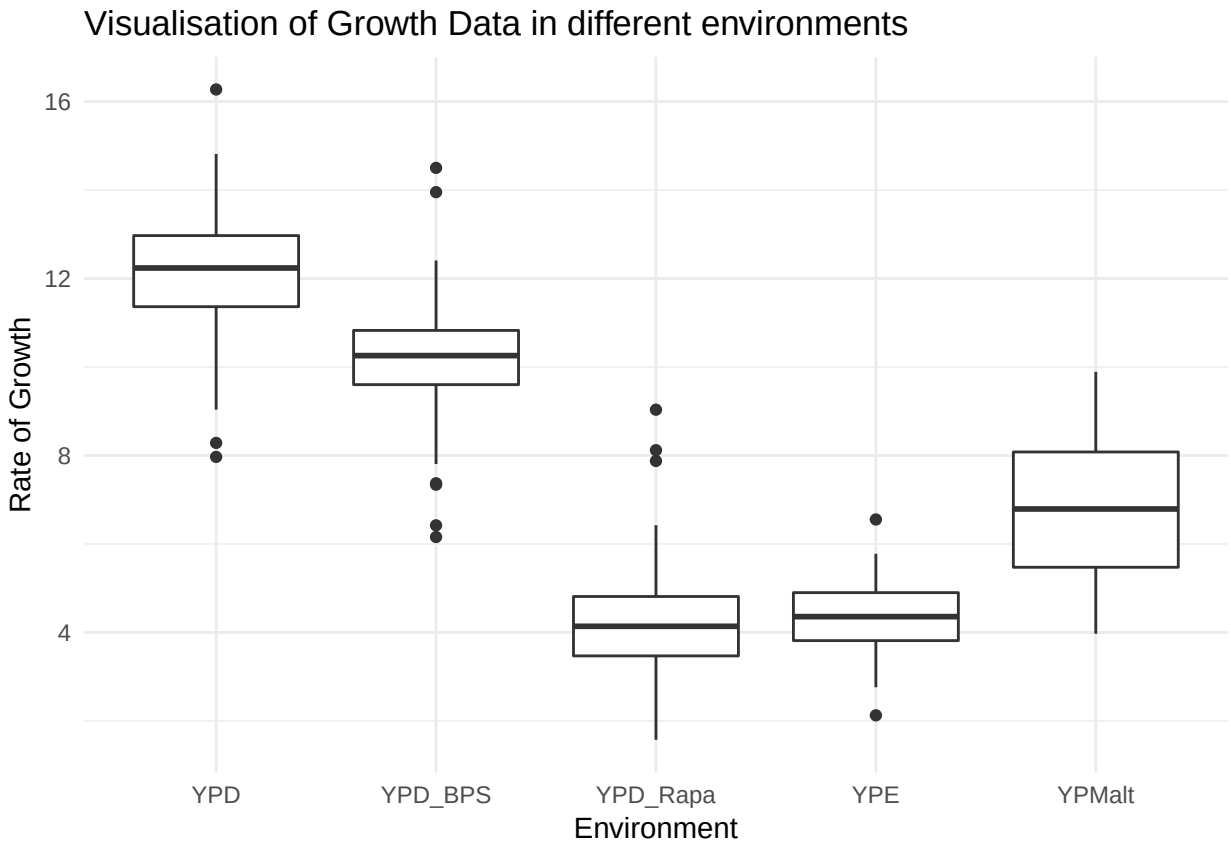
Analysis

Effect of Environment on Growth

We investigate how the environment affect growth rate of the segregants. Since there exists multiple number of segregants, a box plot showing the distribution of the growth of these segregants in different mediums can help us in identifying outliers and which medium has the highest median growth attached to it.

```
# Plotting boxplots to recognize any outliers
ggplot(growth_tidy, aes(env, rate)) + geom_boxplot() + theme_minimal() +
  labs(title = "Visualisation of Growth Data in different environments") +
  xlab("Environment") +
  ylab("Rate of Growth")
```

```
## Warning: Removed 42 rows containing non-finite values (stat_boxplot).
```



In the generated box-plot we see that the distribution of the growth rates varies in every environment. Since we have the growth data of the same strain in different environments, the box-plot shows that growth rate is affected by the environment and most of the strains were able to grow more in the environment YPD than any of the other environments.

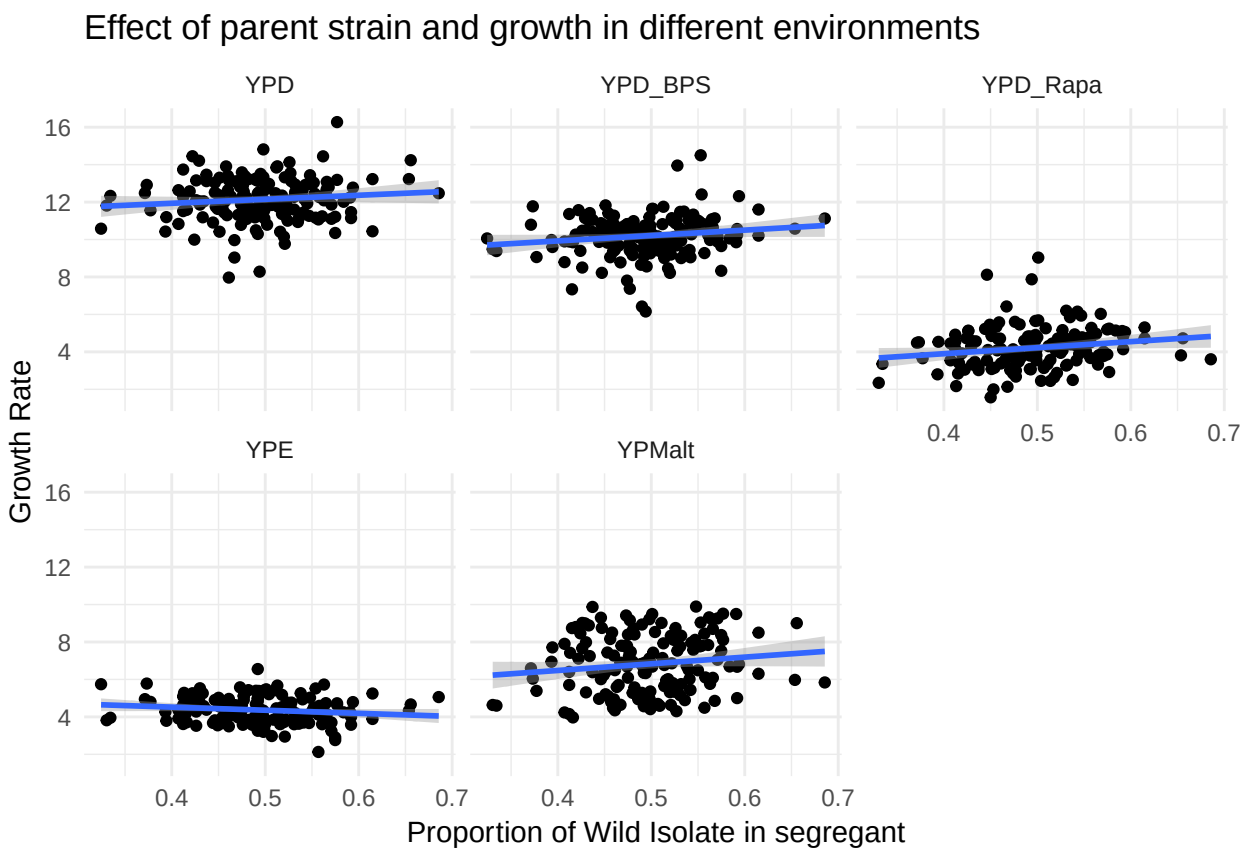
Getting insight into how proportion of each parent strain affects the growth

There exists multiple number of parent strain in each of the segregants as can be observed in genotype.txt table. We devise a single number metric (the proportion of each of the parent strain) and associate it with each of the segregants. Using this metric, in this section we investigate how the proportion of parent strain (from Wild Isolate and Lab Strain) is responsible for effecting the growth in a particular medium.

```
# Calculating the proportion of Lab strain
# We walk down each of the segregants and calculate the number of occurences of each strain
genotype[, labProp := apply(genotype, MARGIN = 1, FUN = function(row) mean(grepl(unname(unlist(row)), pa

# Calculating the proportion of Wild Isolate
# Same as above except now we do it for the Wild Isolate
genotype[, wildProp := apply(genotype, MARGIN = 1, FUN = function(row) mean(grepl(unname(unlist(row)),

# Merging the two datasets on the basis of segregants
# We don't need the strain at each marker since we already have the proportion
growth_geno <- merge(genotype[, c(1, 1002, 1003)], growth)

# Melting the merged table to make it tidy and easy to plot
growth_geno_melt <- melt(growth_geno, id.vars = c('strain', 'labProp', 'wildProp'), variable.name = 'env

# PLotting the data points as scatter distributed over environments
ggplot(growth_geno_melt, aes(x = wildProp, y=rate)) + geom_point() + theme_minimal() +
  facet_wrap(~env) +
  geom_smooth(method = 'lm') +
  labs(title = 'Effect of parent strain and growth in different environments') +
  xlab('Proportion of Wild Isolate in segregant') +
  ylab('Growth Rate')
```

```
## Warning: Removed 42 rows containing non-finite values (stat_smooth).
```

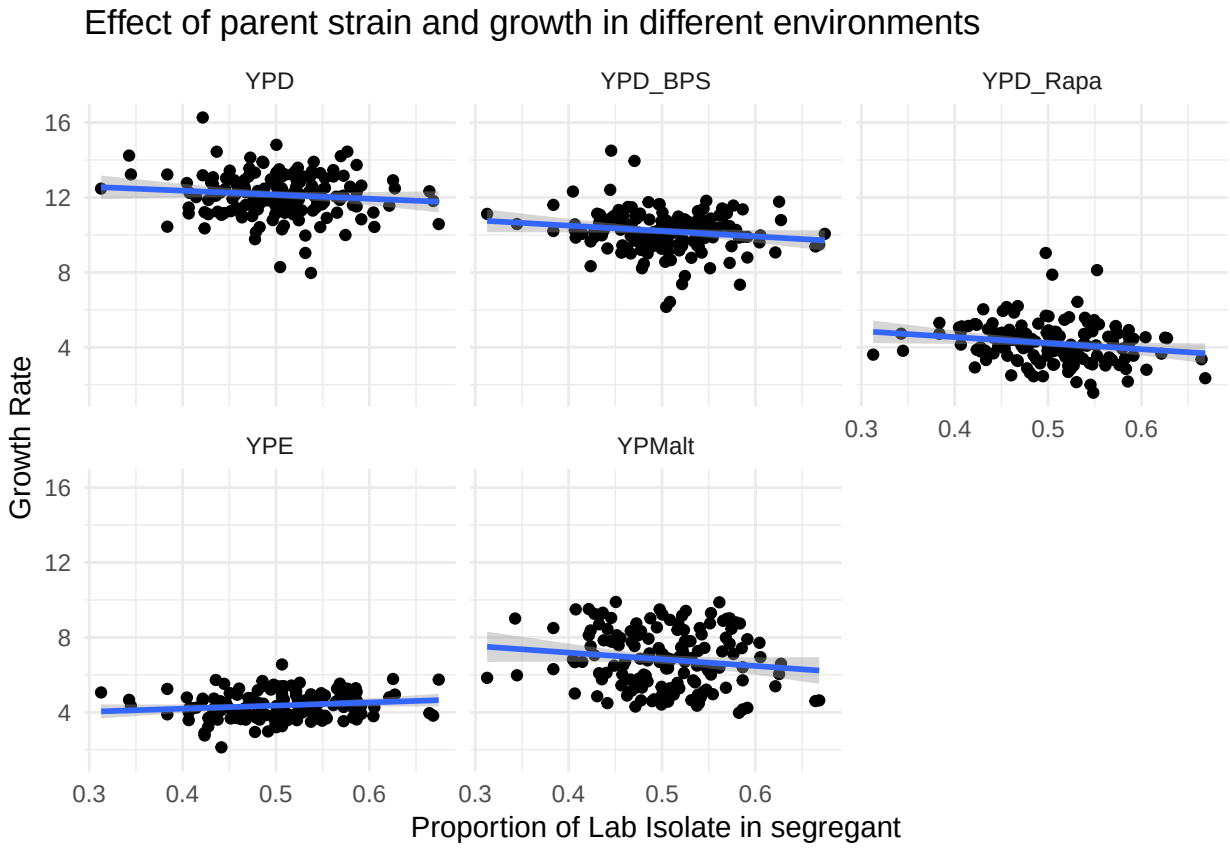
```
## Warning: Removed 42 rows containing missing values (geom_point).
```



```
ggplot(growth_geno_melt, aes(x = labProp, y=rate)) + geom_point() + theme_minimal() +
  facet_wrap(~env) +
  geom_smooth(method = 'lm') +
  labs(title = 'Effect of parent strain and growth in different environments') +
  xlab('Proportion of Lab Isolate in segregant') +
  ylab('Growth Rate')
```

```
## Warning: Removed 42 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 42 rows containing missing values (geom_point).
```



In the above plots it can be observed that as the proportion of Wild isolate or Lab Strain increases there is not any considerable change in the rate of growth in different media. We can filter out some of the markers and change the proportion by getting only those markers that have a gene between them. All those markers that does not have a gene between them can be discarded. We perform this analysis in the following session:

```
# Finding those markers that have a gene between them either completely or partially
gene_marker <- merge(marker,gene,by = "chrom",allow.cartesian = T)
setnames(gene_marker,"start.x", "marker_start")
setnames(gene_marker,"start.y", "gene_start")
setnames(gene_marker,"end.x", "marker_end")
setnames(gene_marker,"end.y", "gene_end")
gene_marker_filtered <- subset(gene_marker,!(marker_start > gene_end | marker_end < gene_start))
head(gene_marker_filtered, n=2)
```

```
##    chrom    id marker_start marker_end gene_start gene_end strand
## 1: chr01 mrk_2        29161      29333      29136    29934      +
## 2: chr01 mrk_2        29161      29333      28385    29514      -
```

```
##         name commonName type source novel
## 1: SY_A0005W  SY_A0005W  SUT genenv  TRUE
## 2:    SUT433     SUT433  SUT   XU09 FALSE
```

We have determined the markers which contains one or more gene by checking the indices of markers and genes and the ones that overlapped has those genes contained in them. The genotype through these genes can be used for analysing the growth and also expression.

```
# Melting genotype data
genotype_melt <- melt(genotype, id.vars="strain", variable.name = "id", value.name = "parent_strain")
```

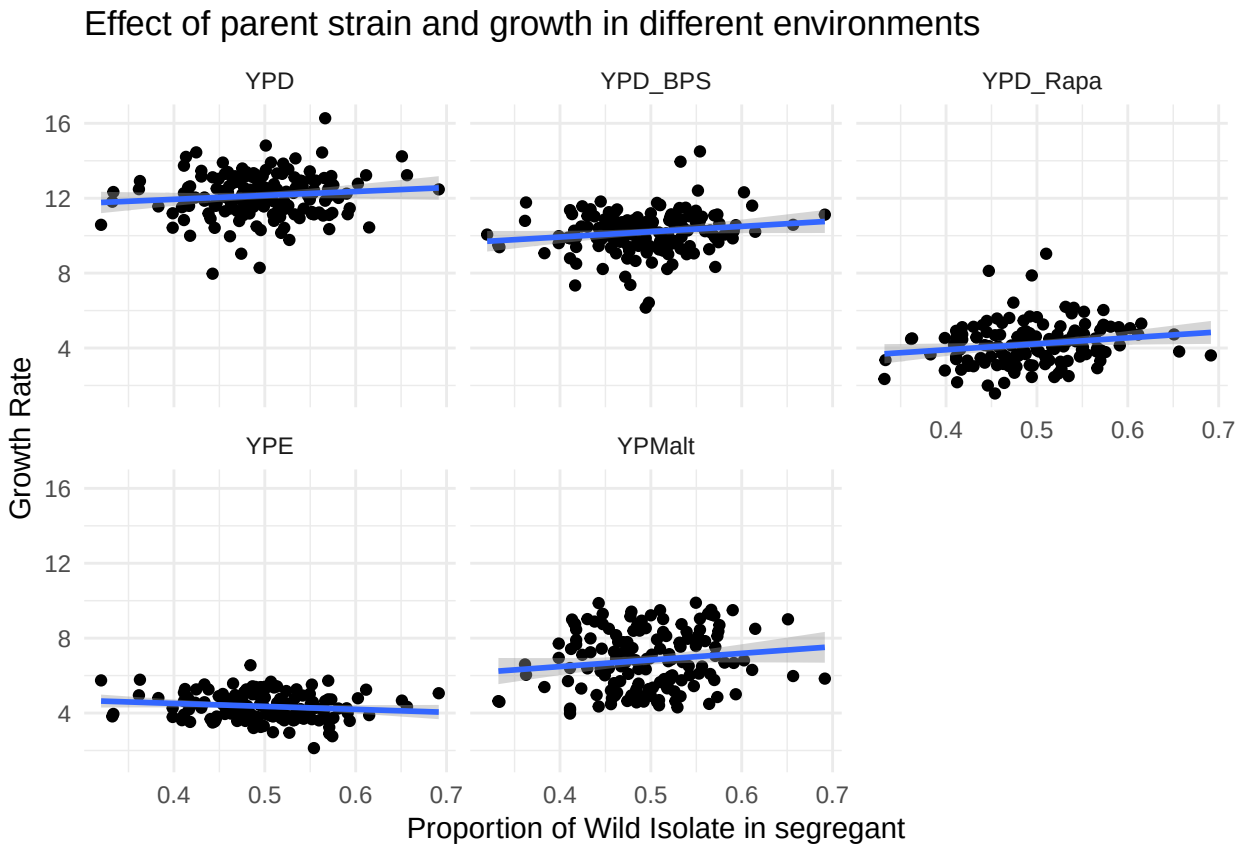
```
## Warning in melt.data.table(genotype, id.vars = "strain", variable.name =
## "id", : 'measure.vars' [mrk_1, mrk_2, mrk_3, mrk_4, ...] are not all of
## the same type. By order of hierarchy, the molten data value column will be
## of type 'character'. All measure variables not of type 'character' will be
## coerced to. Check DETAILS in ?melt.data.table for more on coercion.
```

The melting is performed to filter the genotype table with filtered markers table. For each of strain there exists 1000 markers. We filter the genotype table such that only those markers that are present in the filtered table are present in the filtered table.

```
# Subsetting the table
filtered_genotype <- subset(genotype_melt, id %in% unique(gene_marker_filtered[, id]))

# Casting it back into the form to count proportions of each different type of parent strain
filtered_genotype <- dcast(filtered_genotype, ... ~ id, value.var =  "parent_strain")

# Counting proportions and plotting
# We do it only for wild isolate because the proportions are correlated
# Lab isolate = 1 - wild isolate (in proportion)
filtered_genotype[, wildProp := apply(filtered_genotype, MARGIN = 1, FUN = function(row) mean(grepl(unna

# Merging with the growth
filtered_genotype <- merge(growth, filtered_genotype[, .(strain, wildProp)], by = "strain")

# Melting the data
filtered_genotype <- melt(filtered_genotype, id.vars = c('strain', 'wildProp'), variable.name = 'env', v

# PLotting the data points as scatter distributed over environments
ggplot(filtered_genotype, aes(x = wildProp, y=rate)) + geom_point() + theme_minimal() +
  facet_wrap(~env) +
  geom_smooth(method = 'lm') +
  labs(title = 'Effect of parent strain and growth in different environments') +
  xlab('Proportion of Wild Isolate in segregant') +
  ylab('Growth Rate')
```

```
## Warning: Removed 42 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 42 rows containing missing values (geom_point).
```



From the plots it can be observed that there is no significant change in the relation between the proportion of Wild Isolate and growth rate in different medium. The growth rate does not seem to be affected by how much of Wild Isolate and Lab strain is present in a given segregant.

In the above analysis, we calculated the percentage of lab and wild strains of each segregant to observe if a segregant with a majority of a certain genotype has a higher growth rate in a specific environment. Our findings showed that in none of the environments, there exists a relation between increasing genotype percentage (of either type) and growth rate. Since each gene has a different expression rate and just by knowing the proportion of a genotype, it cannot be concluded that it has any effect on the growth rate.

In the next part we will be examining the relation between gene expression and growth rate with respect to whether the genotype of a each strain effects its expression rate.

Investigating the genotype affect on the growth rate of a segregant

In this section we employ a different strategy for finding out whether a particular type of genotype has effect on the growth. We calculate the median difference between growth rate of Wild Isolate and Lab strain at each marker position and check whether the markers have positive or negative difference. If there exists more markers with a positive difference, then it is safe to assume that the Wild Isolate has more growth rate as compared to Lab isolate and vice versa.

```
# Function for finding out the difference of median growth for both strain
# The function also calculates these values in different environment
getMeasure <- function(strt, chr, genotype, growth, df)
{
  cols <- which(df$chrom == chr & df$start == strt)

  if(cols == 1)
```

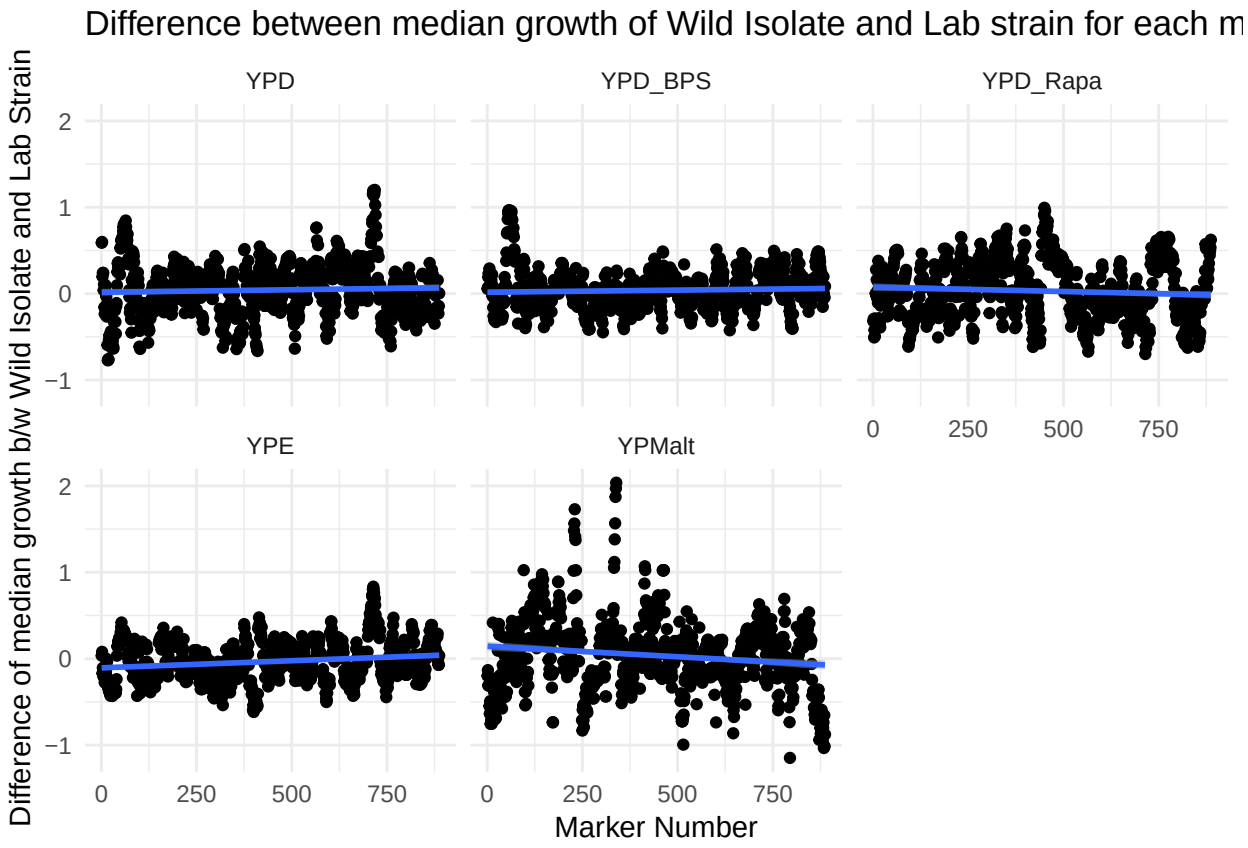
```
    cols = cols + 1
  mygeno <- genotype[, cols]

  # Creating temporary database for Lab strain
  temp <- as.data.table(melt(growth[mygeno == "Lab strain", ], id.vars = "strain"))
  med_lab <- temp[, median(value, na.rm = T), by=variable]

  # Creating temporary database for Wild Isolate
  temp <- as.data.table(melt(growth[mygeno == "Wild isolate", ], id.vars = "strain"))
  med_wild <- temp[, median(value, na.rm = T), by=variable]

  return(med_wild[, V1] - med_lab[, V1])
}

# We have extended the implementation of Warm Up Exercise which expects data frame
genotype_df <- as.data.frame(genotype)
growth_df <- as.data.frame(growth)
marker_df <- as.data.frame(marker)
difference <- list()

# Operating only over those markers that have a gene on them
filtered_markers <- unique(gene_marker_filtered[, .(chrom, id, marker_start)])
num_filtered_markers <-  dim(filtered_markers)[1]

# There can exist a more optimized implementation but this is more explanatory
for(i in 1:num_filtered_markers){

  c <- as.character(filtered_markers[i, 1]$chrom)
  s <- as.integer(filtered_markers[i, 3]$marker_start)
  x <- getMeasure(s, c, genotype_df, growth_df, marker_df)

  # The result is a list so convert to a data table for easy exploration
  dim(x) <- c(1,5)
  x <- as.data.table(x)

  difference <- rbindlist(list(difference, x))

  # Phew! That was a long one.
}

# Adding colnames according to the environment
colnames(difference) <- colnames(growth)[2:6]

# Adding marker number to make table nice
difference[, mrk_no := c(1:887)]
print(head(difference, n=2))
```

```
##          YPD    YPD_BPS   YPD_Rapa        YPE     YPMalt mrk_no
## 1: 0.5893653 0.05584207 -0.3190480 0.03462993 -0.1988258      1
## 2: 0.5979048 0.07324015 -0.2916687 0.07872441 -0.1327372      2
```

```
# Melting and Plotting the differences for each of the environment
difference <- melt(difference, id.vars="mrk_no", variable.name = "env", value.name = "diff")
print(head(difference, n=2))
```

```
##    mrk_no env      diff
## 1:      1 YPD 0.5893653
## 2:      2 YPD 0.5979048
```

```
ggplot(difference, aes(mrk_no, diff)) + geom_point() +
  facet_wrap(~env) +
  theme_minimal() +
  geom_smooth(method = "lm") +
  labs(title = "Difference between median growth of Wild Isolate and Lab strain for each marker.") +
  xlab("Marker Number") +
  ylab("Difference of median growth b/w Wild Isolate and Lab Strain")
```



The above plots show that for each of the medium there does not exist a relation such that there are more number markers with positive difference than negative difference. Thus neither Wild Isolate nor Lab Isolate has an effect over the growth rate which was also depicted by the proportion analysis.

Determining the dependency of gene expression on environments

We investigate how the expression rate of each gene and gene type varies in different environments.

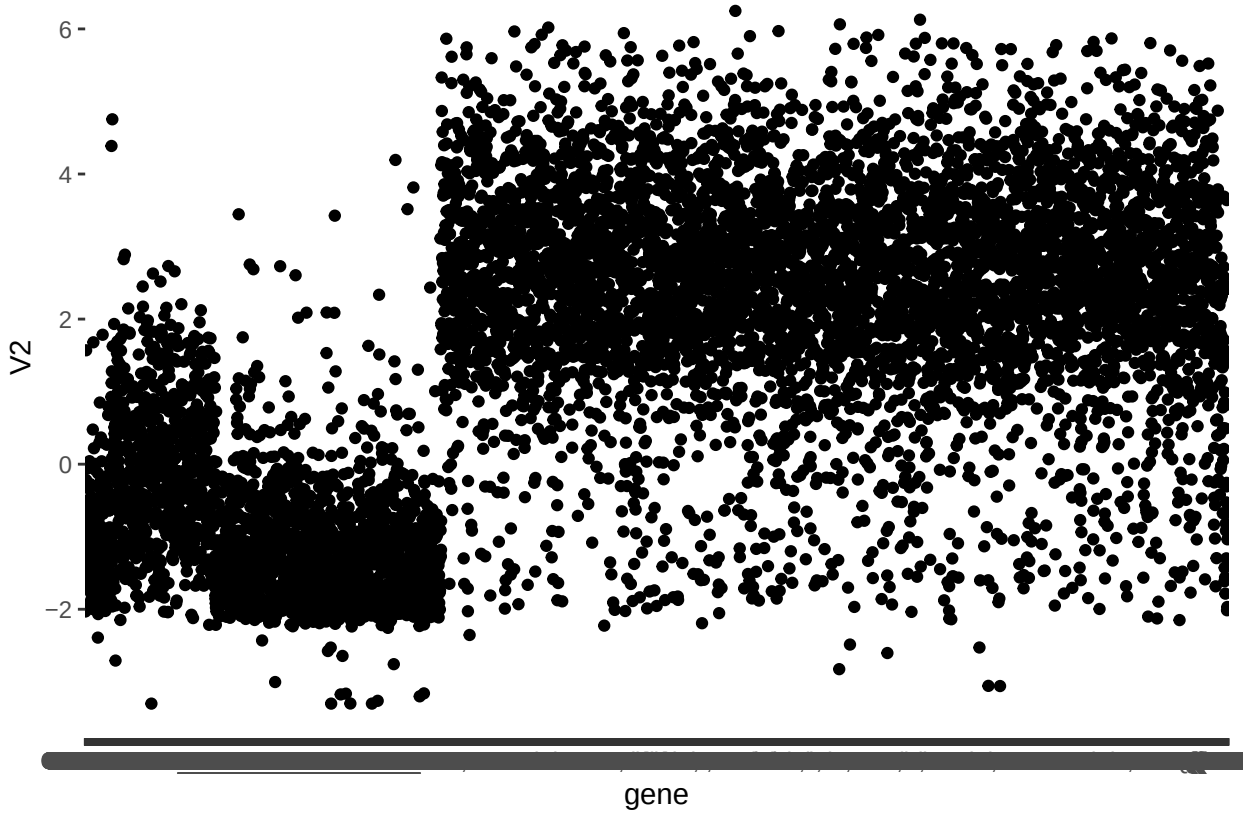
```
# Mean expression rate of each gene in each environments
mean_expr_gene <- unique(expression_tidy[, .(gene_type, mean_expr = .SD[, mean(expression_rate)]), by =

expression_tidy <- gather(expression, medium_strand, expression_rate, c("YPD.seg_01B":"YPMalt.seg_45C"))
expression_tidy <- as.data.table(expression_tidy)
expression_tidy[, medium_strand := .(gsub("_", "", medium_strand))]
head(expression_tidy, n = 10)
```

```
##          gene gene_type medium_strand expression_rate
##  1: SY_A0001W       SUT    YPD.seg01B     -2.02576611
##  2:    SUT001       SUT    YPD.seg01B     -2.02716467
##  3: SY_A0003W       SUT    YPD.seg01B     -0.44516559
##  4: SY_A0004W       SUT    YPD.seg01B      1.05709276
##  5: SY_A0005W       SUT    YPD.seg01B     -2.02912397
##  6:   YAL062W     ORF-T    YPD.seg01B     -0.05015662
##  7:   YAL062W     ORF-T    YPD.seg01B     -1.22395849
##  8: SY_A0008W       SUT    YPD.seg01B     -1.49782194
##  9:   YAL061W     ORF-T    YPD.seg01B      1.96513834
## 10: SY_A0010W       SUT    YPD.seg01B      0.46947972
```

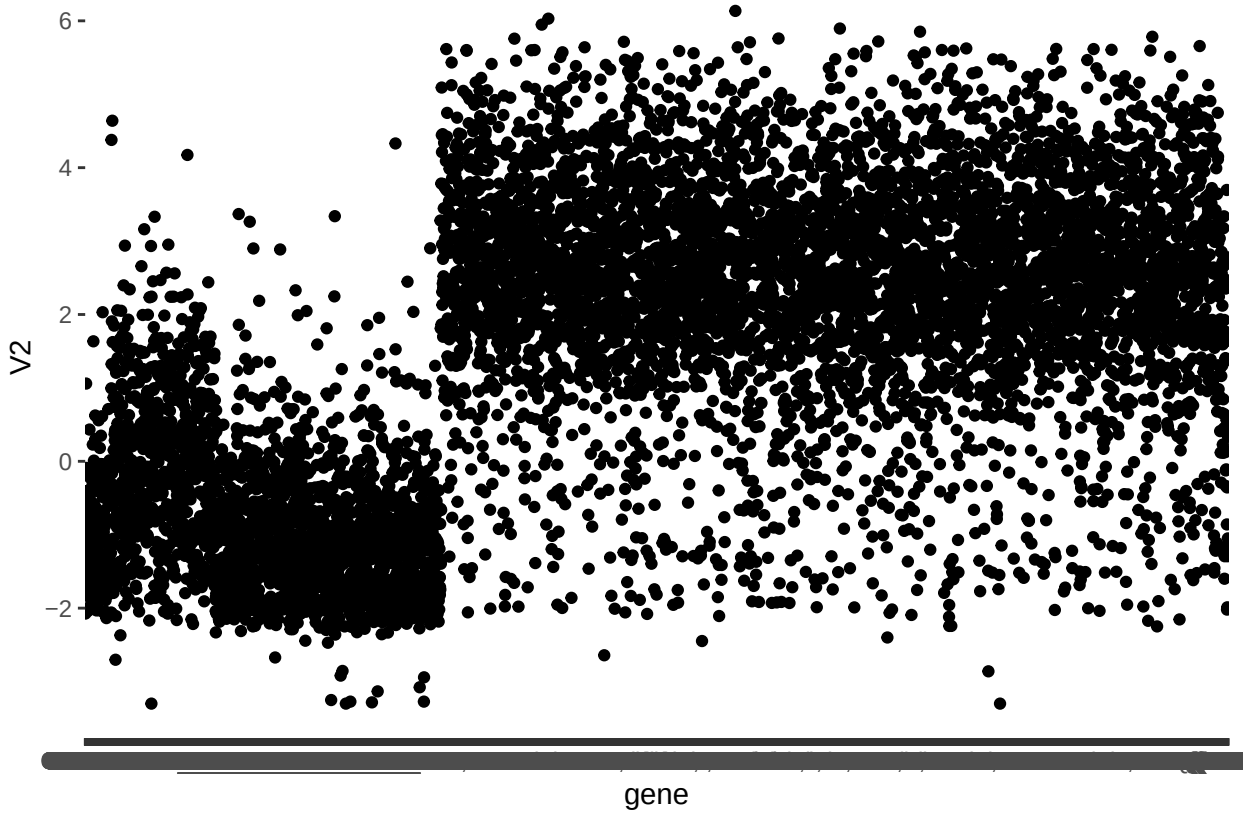
```
expression_tidy <- as.data.table(separate(as.data.frame(expression_tidy), medium_strand, into = c("mediu
expression_tidy
```

```
##               gene gene_type medium strand expression_rate
##       1: SY_A0001W       SUT    YPD seg01B      -2.0257661
##       2:    SUT001       SUT    YPD seg01B      -2.0271647
##       3: SY_A0003W       SUT    YPD seg01B      -0.4451656
##       4: SY_A0004W       SUT    YPD seg01B       1.0570928
##       5: SY_A0005W       SUT    YPD seg01B      -2.0291240
##      ---
## 1533902:   YPR195C     ORF-T YPMalt seg45C       1.3293799
## 1533903:    SUT846       SUT YPMalt seg45C       0.6267752
## 1533904:    SUT847       SUT YPMalt seg45C       1.5945409
## 1533905:   YPR199C     ORF-T YPMalt seg45C       2.5568570
## 1533906:   YPR200C     ORF-T YPMalt seg45C      -0.6287938
```

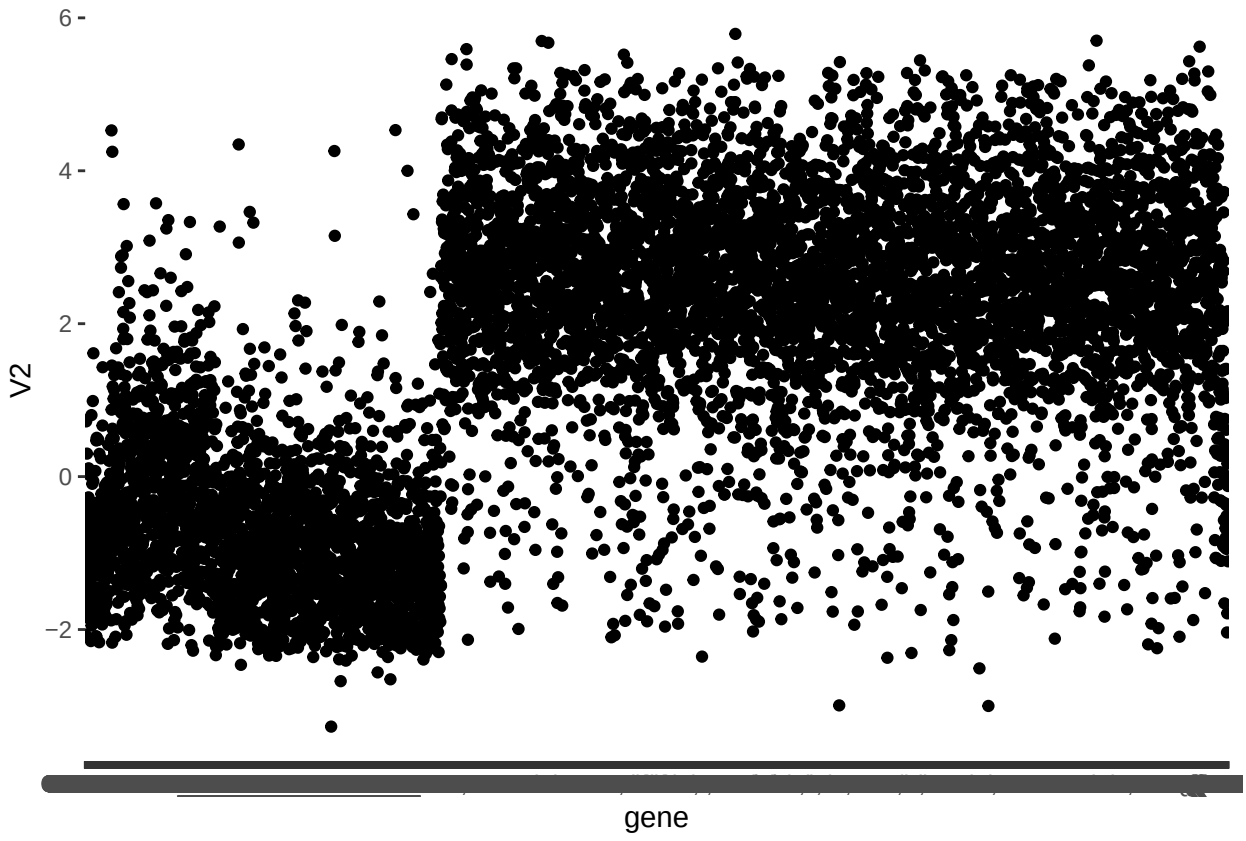
```
gene_expr <- unique(expression_tidy[, mean_expr := .SD[, mean(expression_rate)], by = c("gene","medium"]

# Expression rates in each medium
YDP_expr <- unique(expression_tidy[medium == "YPD", .SD[,.(medium, mean(expression_rate))], by = gene])
YDPBPS_expr <-unique(expression_tidy[medium == "YPDBPS", .SD[,.(medium, mean(expression_rate))], by = g
YDPRapa_expr <- unique(expression_tidy[medium == "YPDRapa", .SD[,.(medium, mean(expression_rate))], by =
YPE_expr <- unique(expression_tidy[medium == "YPE", .SD[,.(medium, mean(expression_rate))], by = gene])
YPMalt_expr <- unique(expression_tidy[medium == "YPMalt", .SD[,.(medium, mean(expression_rate))], by = g

ggplot(YDP_expr, aes(gene, V2)) + geom_point()
```



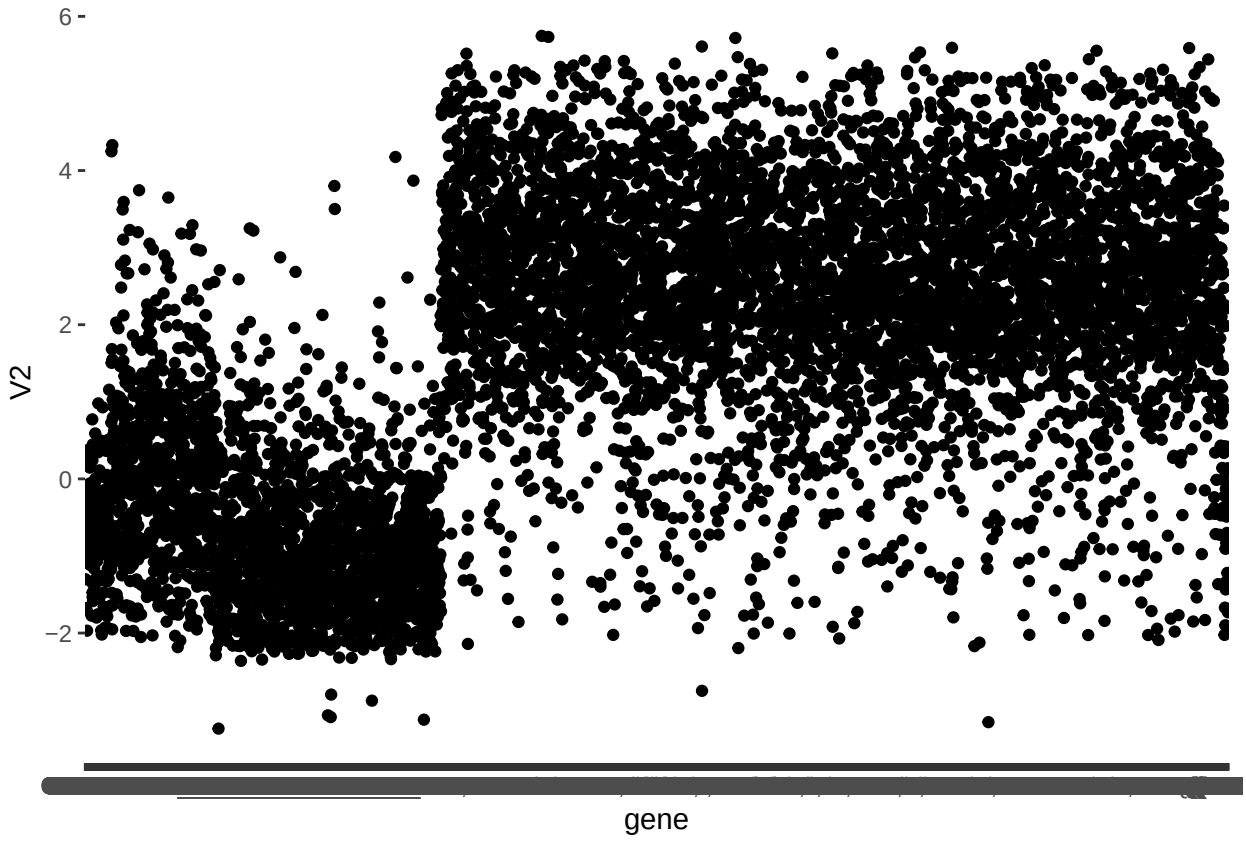

```
ggplot(YDPBPS_expr, aes(gene, V2)) + geom_point()
```



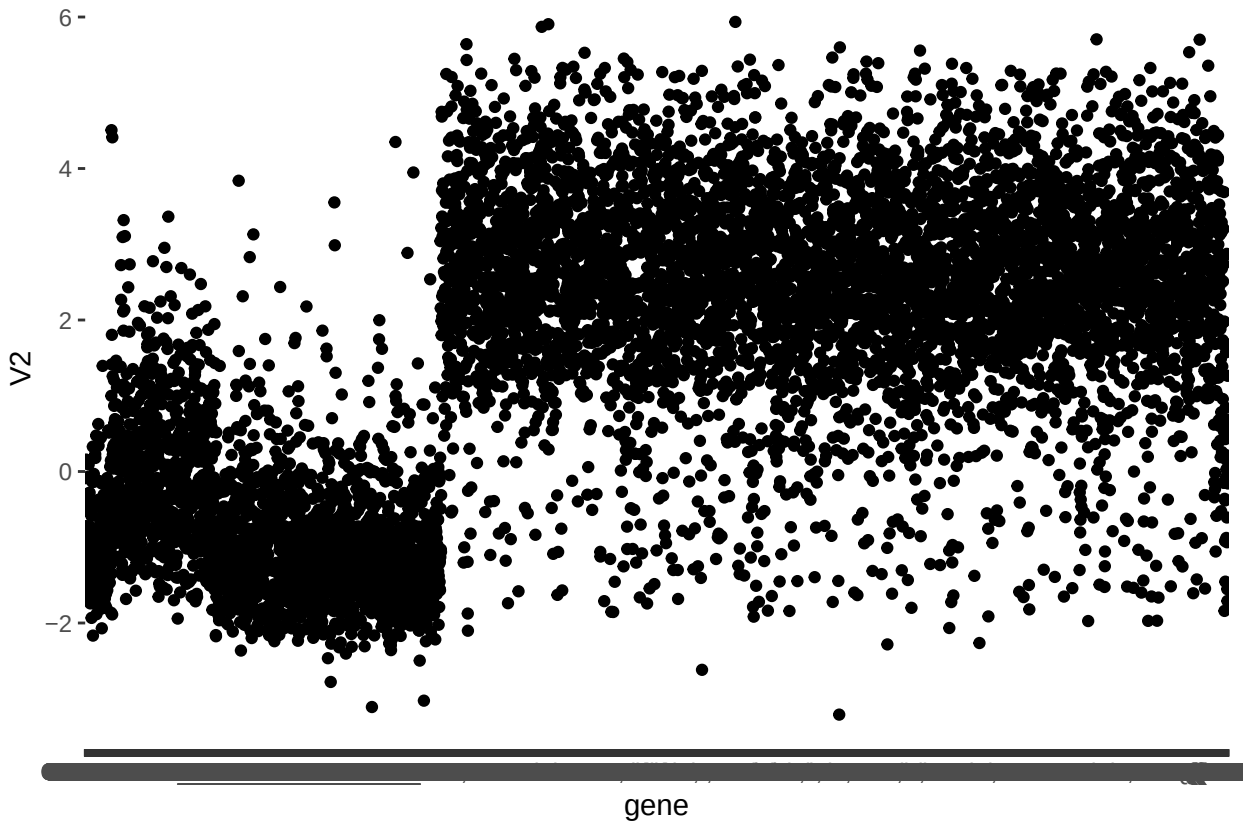

```
ggplot(YDPRapa_expr, aes(gene, V2)) + geom_point()
```



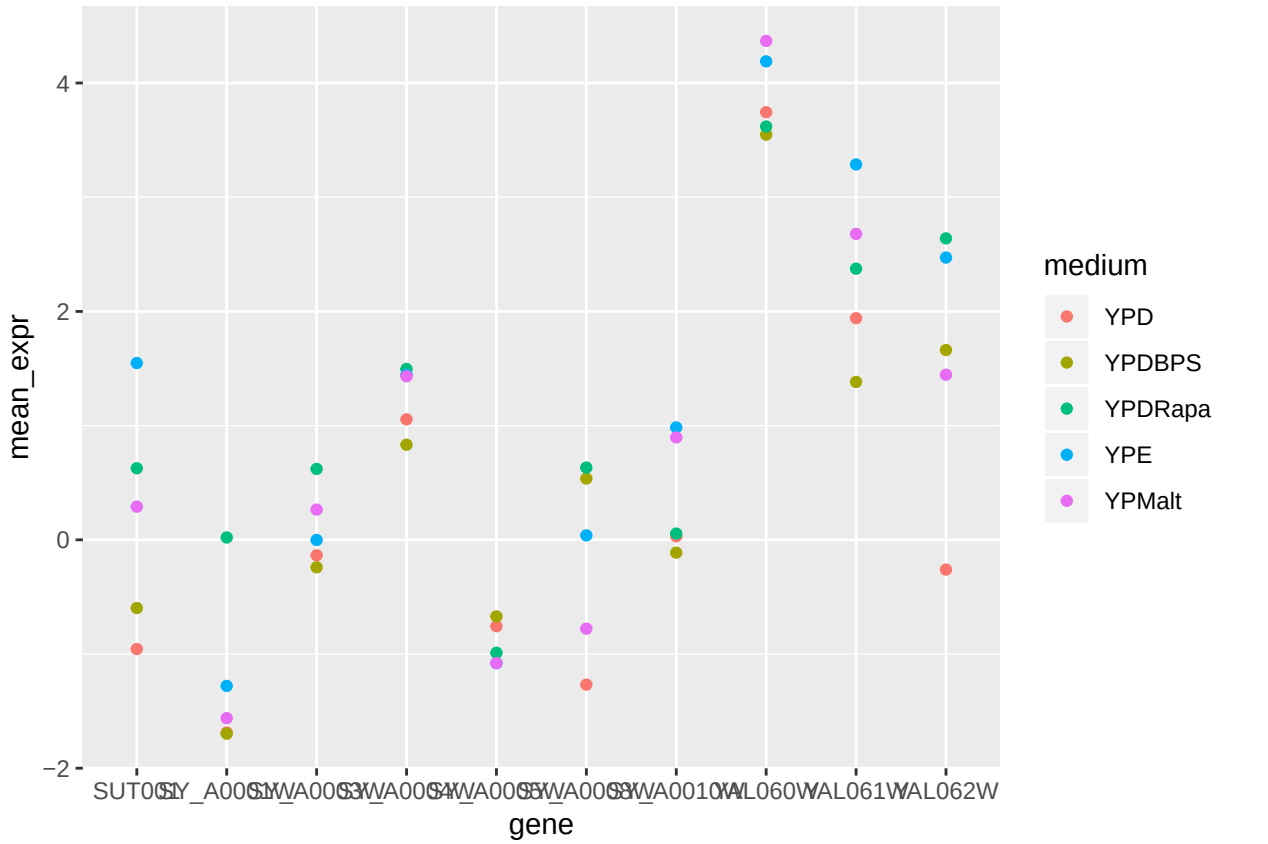

```
ggplot(YPE_expr, aes(gene, V2)) + geom_point()
```



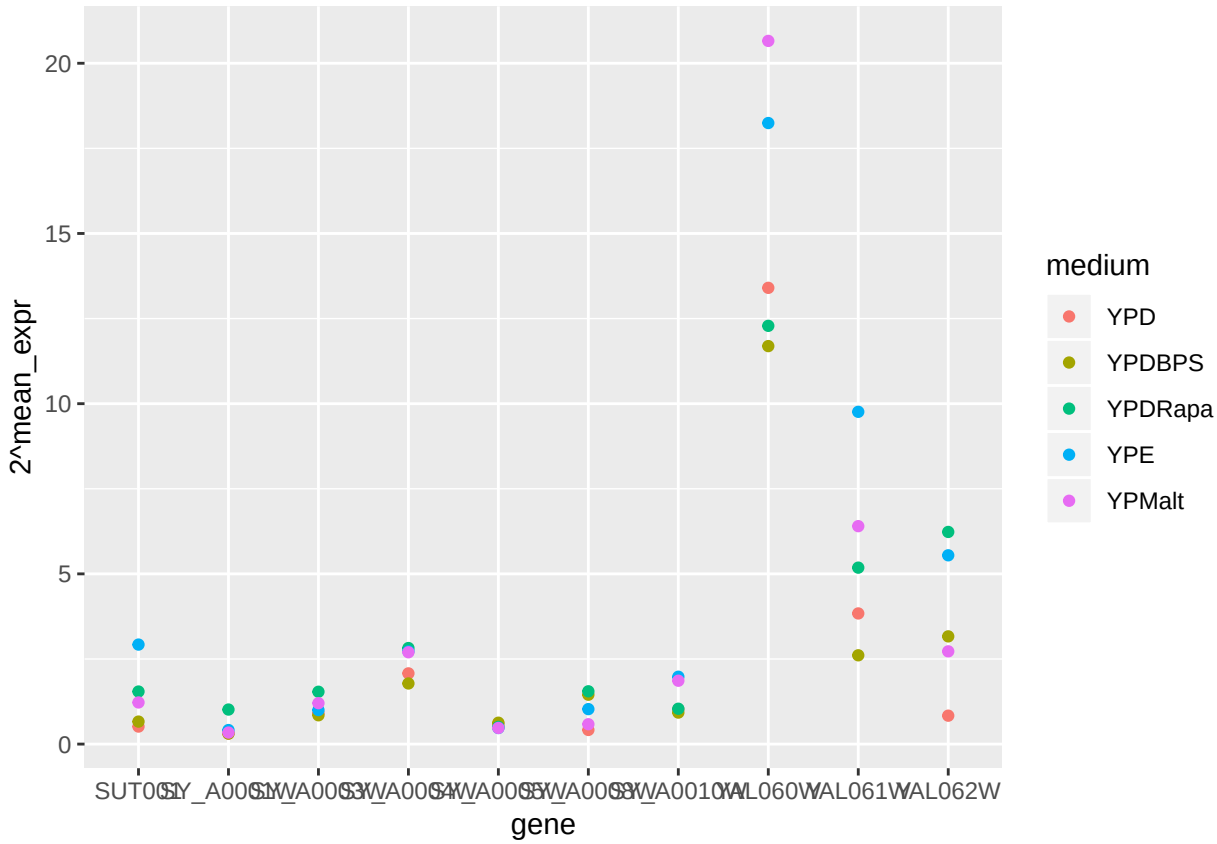

```
ggplot(YPMalt_expr, aes(gene, V2)) + geom_point()
```



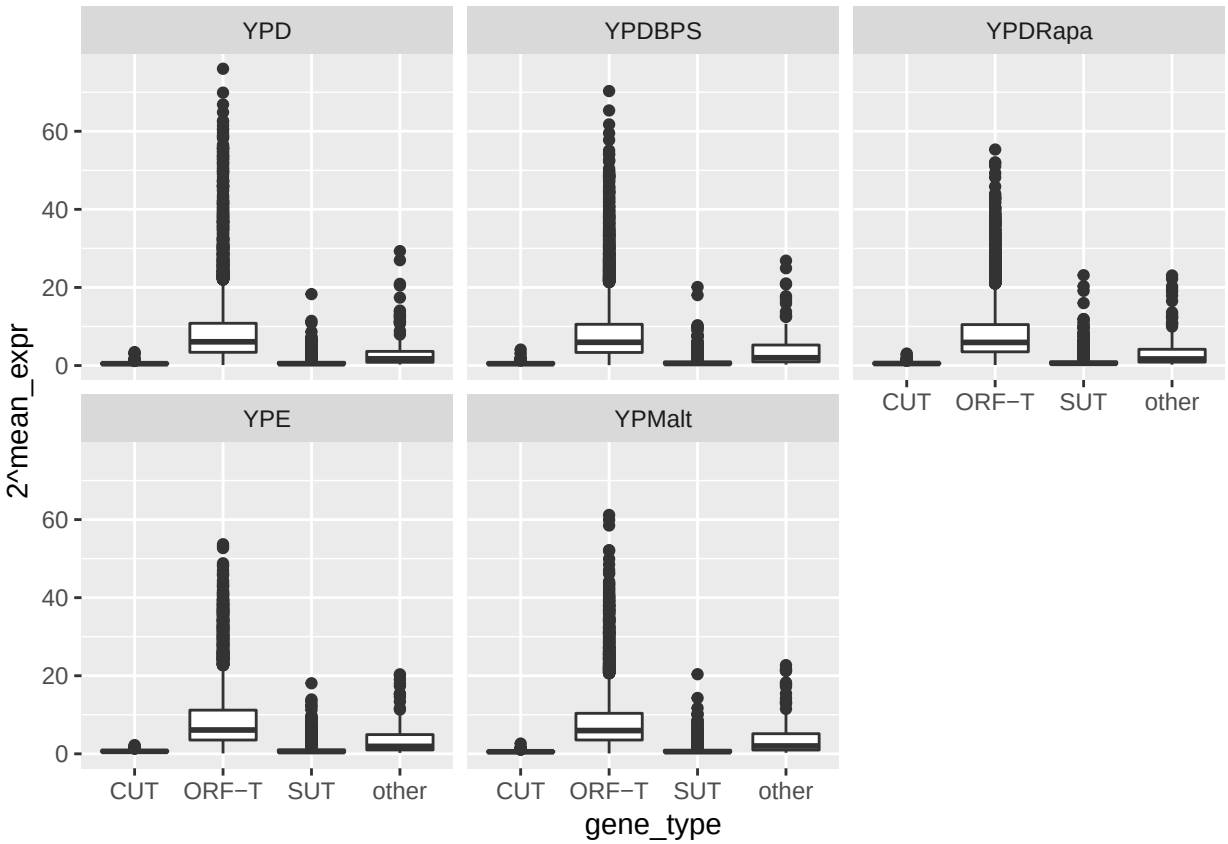

```
# Average expression of each gene in each medium in one graph
mean_expr_dt <- unique(gene_expr[,.(medium, mean_expr, gene_type), by=gene])
ggplot(mean_expr_dt[1:50], aes(gene, mean_expr)) + geom_point(aes(color = medium))
```




```
ggplot(mean_expr_dt[1:50], aes(gene, 2^mean_expr)) + geom_point(aes(color = medium))
```




```
# Average expression of each gene type in each environment
ggplot(mean_expr_dt, aes(gene_type,2^mean_expr)) + geom_boxplot() + facet_wrap(~medium)
```




```
diff_expr_gene <- mean_expr_gene[, .(mean_expr, diff_expr = (max(mean_expr) - min(mean_expr)), medium),
ordered_genes <- setorder(diff_expr_gene, diff_expr)[, .(gene, diff_expr, medium)]
ordered_genes <- unique(ordered_genes[, diff_expr, by = gene])
#the genes that are most environment independent
head(ordered_genes, n=3)[,gene]
```

```
## [1] YML035C YLR035C YPR029C
## 8157 Levels: CUT004 CUT006 CUT014 CUT016 CUT017 CUT023 CUT025 ... tT(AGU)N1
```

```
#the genes that are most environment dependent
tail(ordered_genes, n=3)[,gene]
```

```
## [1] YNL117W   YMR107W   YJR005C-A
## 8157 Levels: CUT004 CUT006 CUT014 CUT016 CUT017 CUT023 CUT025 ... tT(AGU)N1
```

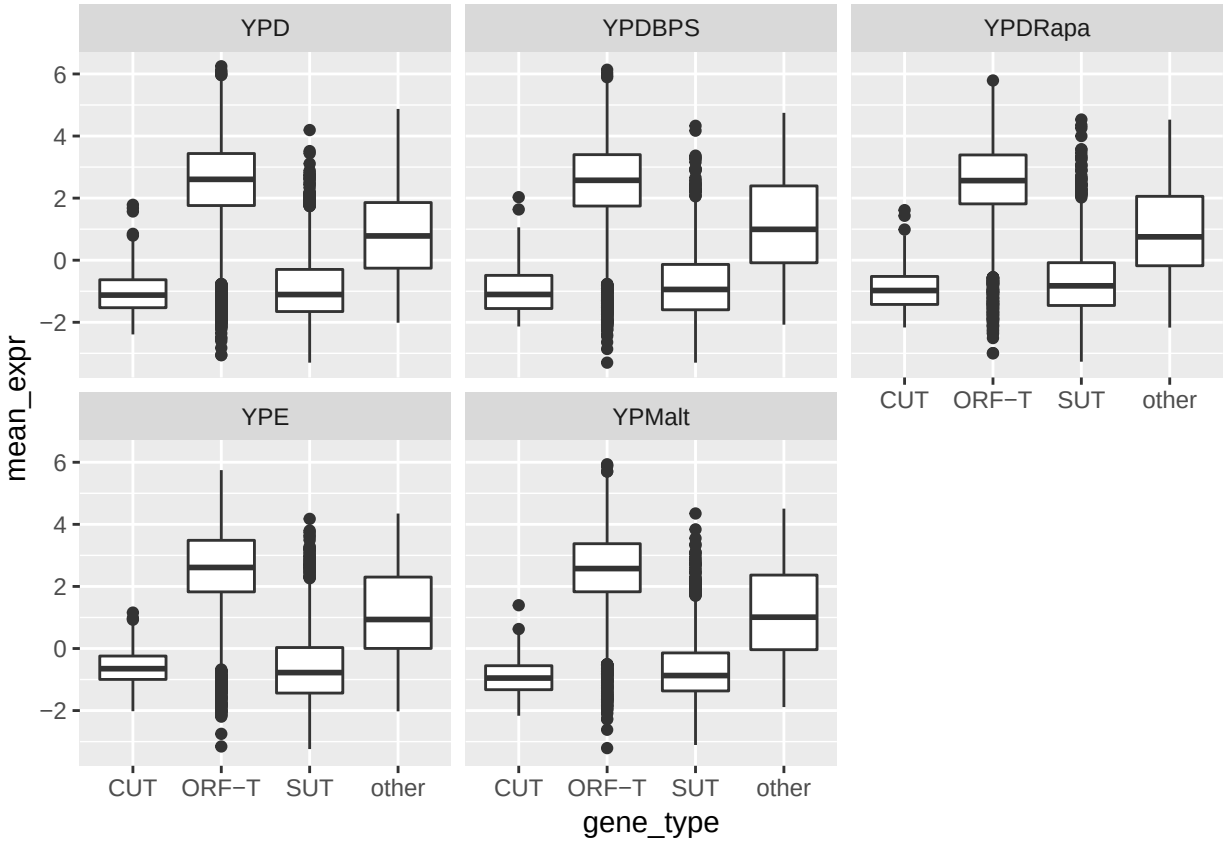
We have calculated differences between each gene's mean expression rate for each environment. We assumed that the gene with the lowest difference is the one that is not dependent on the environment since the expression rate is nearly the same at each environment. So we have ordered all genes to find out which genes are less dependent on the environment. We found out that the genes YML035C, YLR035C, YPR029C are the ones that are least depent ones and the genes YNL117W, YMR107W, YJR005C-A are the most dependant ones on the environment.

Understanding the affect of the type of the gene on the expression rate

In this section we examined how the different types of genes defer in expression rate in different environments.

```
# Expression rate of each type of genes in each environment
gene_type_data <- unique(mean_expr_gene[,.(medium, mean_expr, gene_type), by=gene])
```

```
ggplot(gene_type_data, aes(gene_type, mean_expr)) + geom_boxplot() + facet_wrap(~medium)
```



We have seen that regardless of the environment, genes with the ORF-T type are the highest expressed genes. This is partially due to the type of the gene, since ORF type genes are used in translation, it is only logical that they are the most expressed genes.

Correlation between genotype and expression rate of a gene

We have tried to find if a correlation exists between the expression rate of a gene and if the gene comes from a wild or lab isolate.

```
#TODO: Change Najeeb's code to calculate differences for each marker for comparing genotypes with expre
```

Conclusion