

PLOT FUN

Ethan Allavarpu (UID: 405287603)

11/30/2020

```
training <- read.csv("training.csv", stringsAsFactors = FALSE)
dim(training)

## [1] 7242 260
any(is.na(training))

## [1] FALSE

var_types <- vapply(training, class, character(1))
names(training) [ -which(var_types %in% c("integer", "numeric"))]

## [1] "PublishedDate"

dates <- training$PublishedDate
head(dates)

## [1] "4/17/2020 10:38" "8/31/2020 9:56"  "8/16/2020 12:15" "8/22/2020 9:00"
## [5] "8/22/2020 14:18" "7/24/2020 21:16"

split_dates <- strsplit(dates, "[/:]")
head(split_dates)

## [[1]]
## [1] "4"      "17"     "2020"   "10"     "38"
##
## [[2]]
## [1] "8"      "31"     "2020"   "9"      "56"
##
## [[3]]
## [1] "8"      "16"     "2020"   "12"     "15"
##
## [[4]]
## [1] "8"      "22"     "2020"   "9"      "00"
##
## [[5]]
## [1] "8"      "22"     "2020"   "14"     "18"
##
## [[6]]
## [1] "7"      "24"     "2020"   "21"     "16"

split_dates <- lapply(split_dates, as.numeric)
years <- function(date) {
  date[3]
}

yrs <- lapply(split_dates, years)
```

```

unique(yrs) # Only 2020

## [1]
## [1] 2020

new_time <- function(old_date) {
  old_date <- as.numeric(old_date)
  month <- old_date[1]
  day <- old_date[2]
  hr <- old_date[4]
  minute <- old_date[5]
  month_days <- c(31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31)
  complete_months <- month - 1
  if (complete_months > 0) {
    complete_month_days <- sum(month_days[1:complete_months])
  } else {
    complete_month_days <- 0
  }
  total_days <- complete_month_days + day
  pre_hrs <- total_days * 24
  total_hrs <- pre_hrs + hr
  pre_min <- total_hrs * 60
  final_time <- pre_min + minute
  final_time
}

processed_dates <- vapply(split_dates, new_time, numeric(1))
training$PublishedDate <- processed_dates

subs <- training[, 248:250]
sub_final <- 3 - 3 * subs[[1]] - 2 * subs[[2]] - subs[[3]]
views <- training[, 251:253]
views_final <- 3 - 3 * views[[1]] - 2 * views[[2]] - views[[3]]
growth <- training[, 254:256]
growth_final <- 3 - 3 * growth[[1]] - 2 * growth[[2]] - growth[[3]]
vids <- training[, 257:259]
vid_final <- 3 - 3 * vids[[1]] - 2 * vids[[2]] - vids[[3]]
growth_2_6 <- training$growth_2_6
names(training)[c(1, 248:260)]

## [1] "id"                               "Num_Subscribers_Base_low"
## [3] "Num_Subscribers_Base_low_mid"   "Num_Subscribers_Base_mid_high"
## [5] "Num_VIEWS_Base_low"              "Num_VIEWS_Base_low_mid"
## [7] "Num_VIEWS_Base_mid_high"        "avg_growth_low"
## [9] "avg_growth_low_mid"            "avg_growth_mid_high"
## [11] "count_vids_low"                "count_vids_low_mid"
## [13] "count_vids_mid_high"           "growth_2_6"

training <- training[, -c(1, 248:260)]
training <- data.frame(training, Num_Subscribers_Base = sub_final,
                       Num_VIEWS_Base = views_final, avg_growth = growth_final,
                       count_vids = vid_final, growth_2_6 = growth_2_6)

# See which variables have no variation (useless)
var_sds <- vapply(training, sd, numeric(1))
useless_vars <- names(training)[var_sds == 0]

```

```

useless_vars

## [1] "cnn_35"          "cnn_37"          "min_red"         "min_green"
## [5] "min_blue"        "punc_num_..17"   "punc_num_..19"   "punc_num_..23"
## [9] "punc_num_..25"   "punc_num_.."    "punc_num_..29"   "punc_num_..30"
training <- training[, !(names(training) %in% useless_vars)]

# Remove highly correlated variables
cor_mtx <- round(cor(training[, -ncol(training)]), 2)
library(reshape2)
library(ggplot2)
melted_cor_mtx <- melt(cor_mtx)
cor_vars <- which(abs(cor_mtx) > 0.9, arr.ind = TRUE)
for (i in 1:nrow(cor_vars)) {
  if (cor_vars[i, 1] >= cor_vars[i, 2]) {
    cor_vars[i, ] <- rep(NA, 2)
  }
}

all_na <- function(data) {
  all(is.na(data))
}
unique_cor_vars <- apply(cor_vars, 1, all_na)
cor_vars <- cor_vars[!unique_cor_vars, ]
corr_vars <- unique(rownames(cor_vars))

library(dplyr)

##
## 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

training <- training[-which(training$views_2_hours > 2000000), ]
training <- training[-which(training$hog_0 > 0.5), ]
training <- training[-which(training$hog_1 > 0.35), ]
training <- training[-which(training$hog_152 > 0.4), ]
training <- training[-which(training$hog_182 > 0.4), ]
training <- training[-which(training$hog_350 > 0.5), ]
training <- training[-which(training$hog_364 > 0.6), ]
training <- training[-which(training$hog_512 > 0.5), ]

training <- training[which(training$hog_522 <= 0.5), ]
training <- training[which(training$hog_584 <= 0.5), ]
training <- training[which(training$hog_643 <= 0.8), ]
training <- training[which(training$hog_649 <= 0.5), ]
training <- training[which(training$hog_658 <= 0.5), ]
training <- training[which(training$hog_705 <= 0.5), ]

```

```

training <- training[which(training$hog_724 <= 0.5), ]
training <- training[which(training$hog_774 <= 0.5), ]
training <- training[which(training$hog_818 <= 0.4), ]
training <- training[which(training$hog_828 <= 0.6), ]
training <- training[which(training$hog_831 <= 0.5), ]
training <- training[which(training$hog_832 <= 0.6), ]
training <- training[which(training$hog_855 <= 0.5), ]
training <- training[which(training$cnn_0 == 0), ]
training <- training[which(training$cnn_36 == 0), ]
training <- training[which(training$cnn_65 == 0), ]
training <- training[which(training$punc_num_..18 == 0), ]
training <- training[which(training$punc_num_..21 <= 1.5), ]
training <- training[which(training$punc_num_..26 == 0), ]
training <- training[which(training$punc_num_..27 == 0), ]

```

```

new_var_sd <- vapply(training, sd, numeric(1))
useless_vars <- names(training)[new_var_sd == 0]
useless_vars

```

```

## [1] "cnn_0"           "cnn_36"          "cnn_65"          "punc_num_..18"
## [5] "punc_num_..26"   "punc_num_..27"
training <- training[, !(names(training) %in% useless_vars)]

```

```

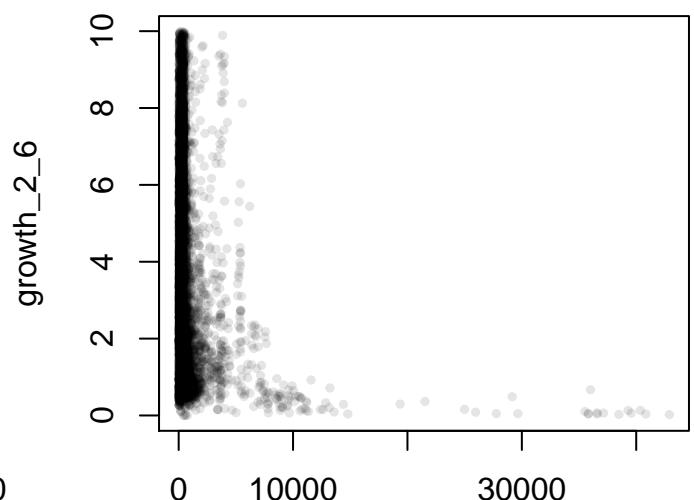
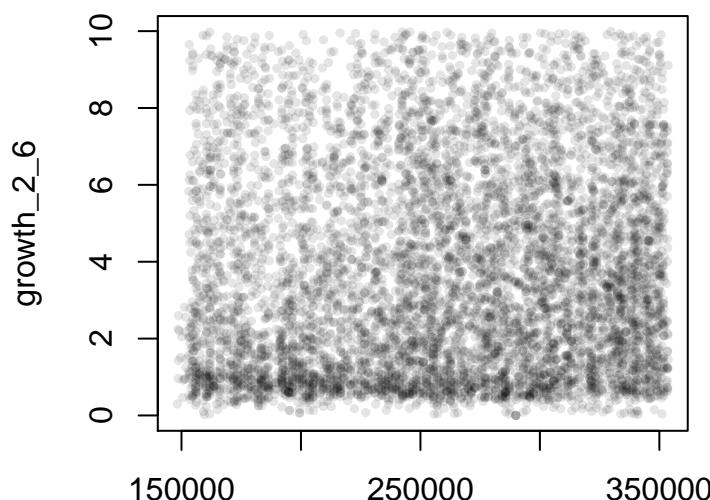
factor_vars <- training[, 229:232]
factor_vars <- data.frame(lapply(factor_vars, factor))
training[, 229:232] <- factor_vars
training <- training[, !(names(training) %in% corr_vars)]

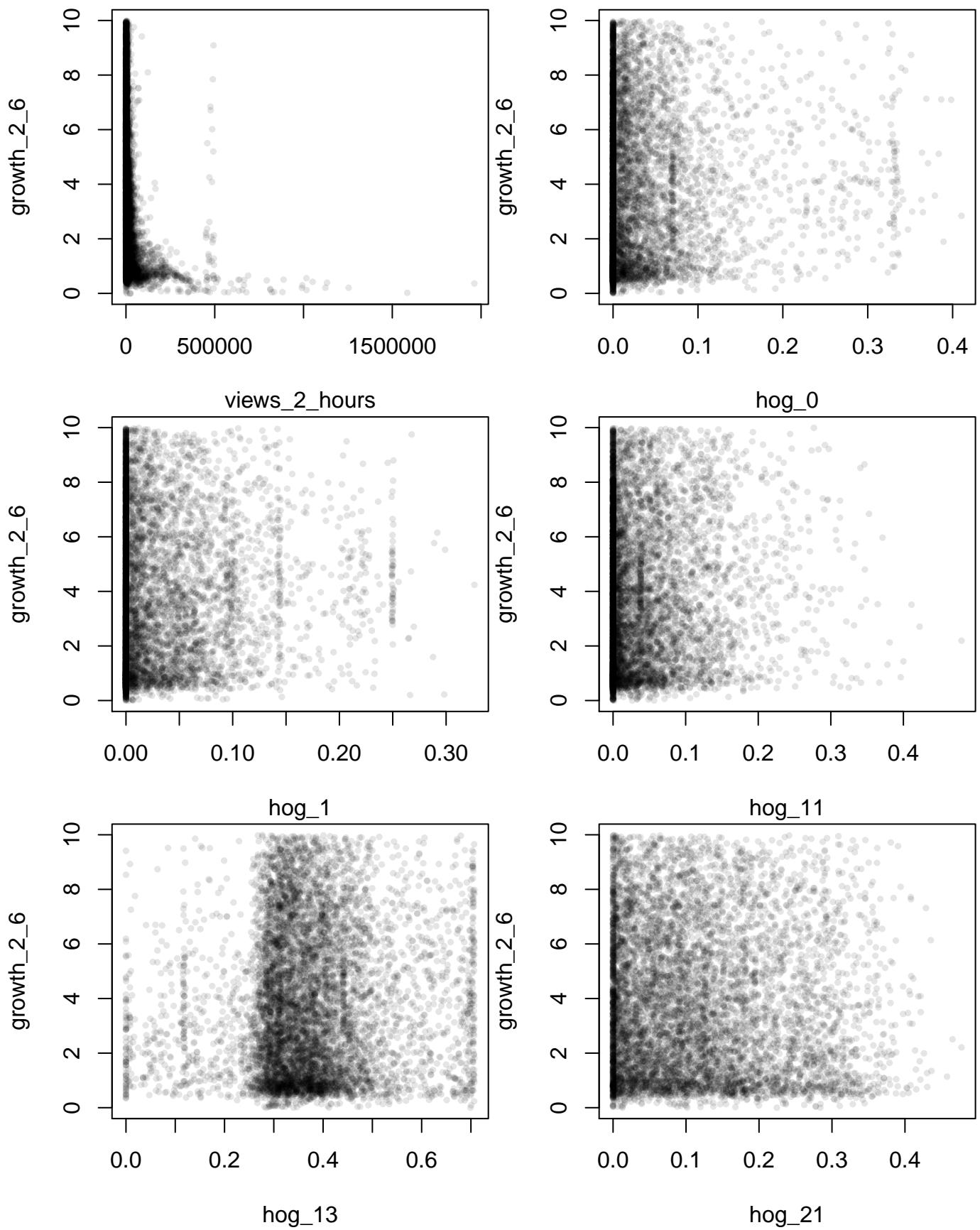
```

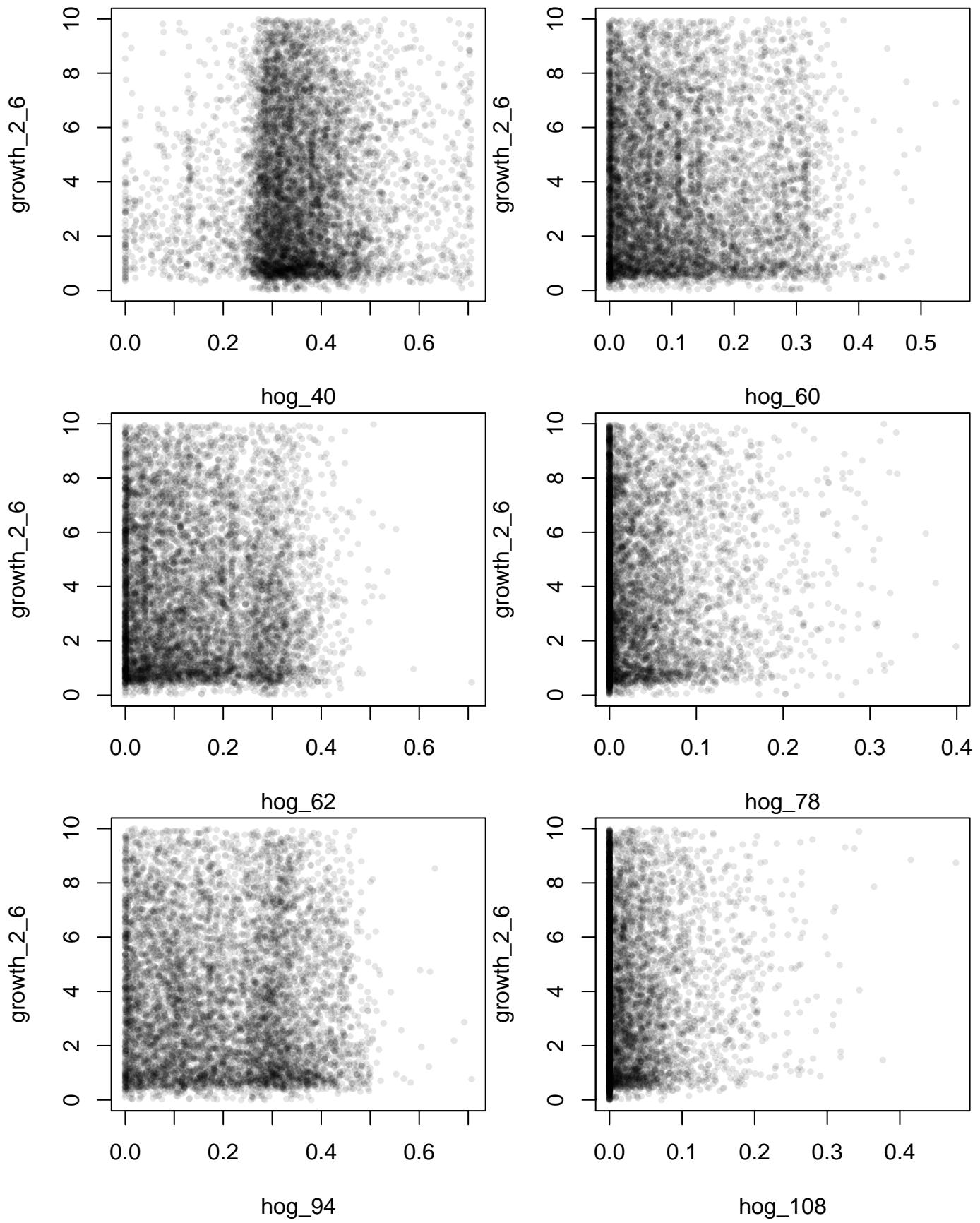
```

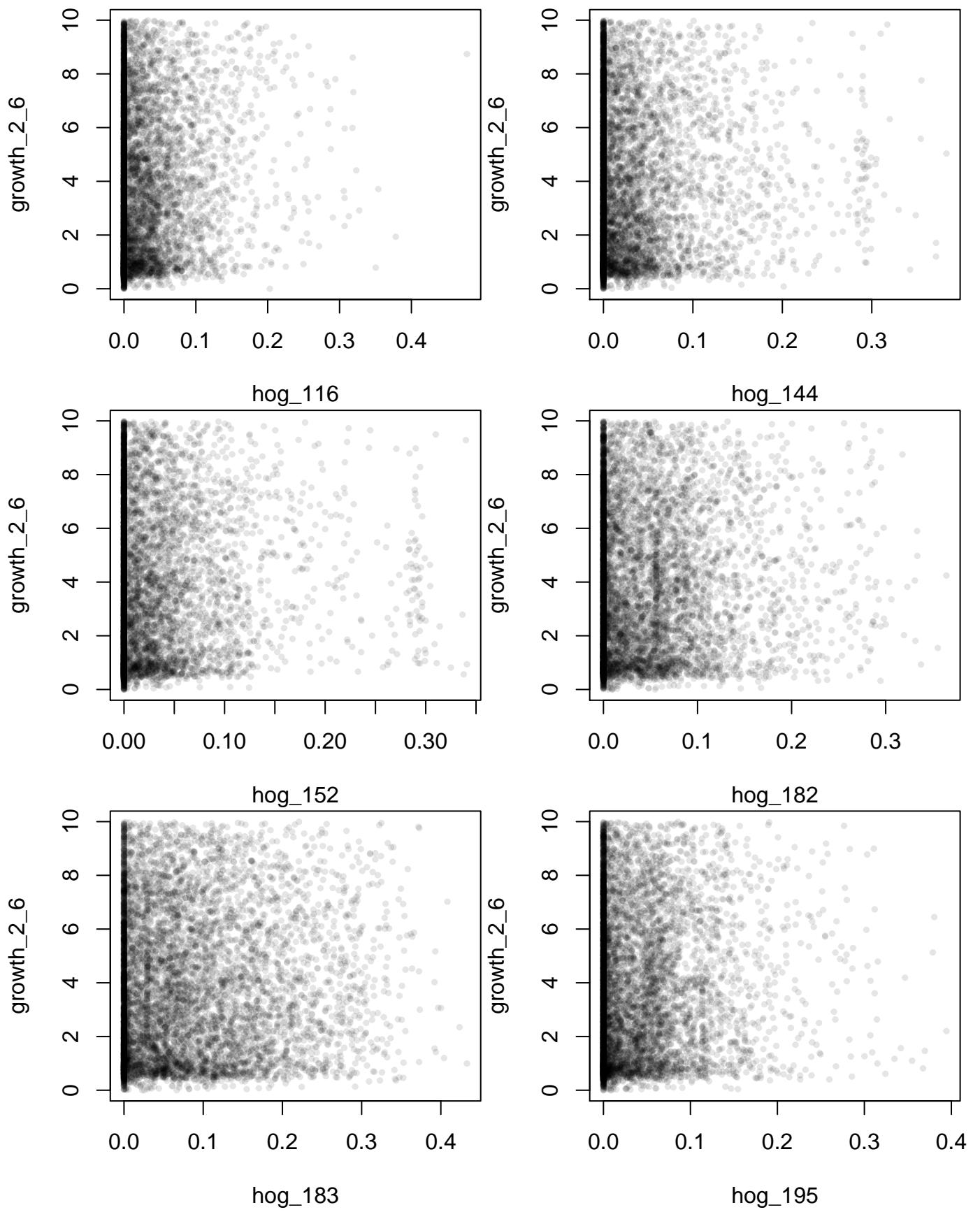
library(dplyr)
plot(growth_2_6 ~ ., data = training,
      cex = 0.5, pch = 19, col = rgb(0, 0, 0, 0.1))

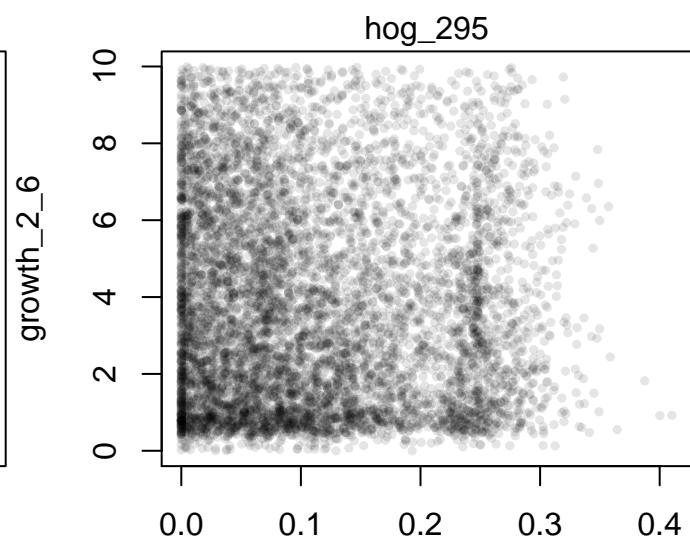
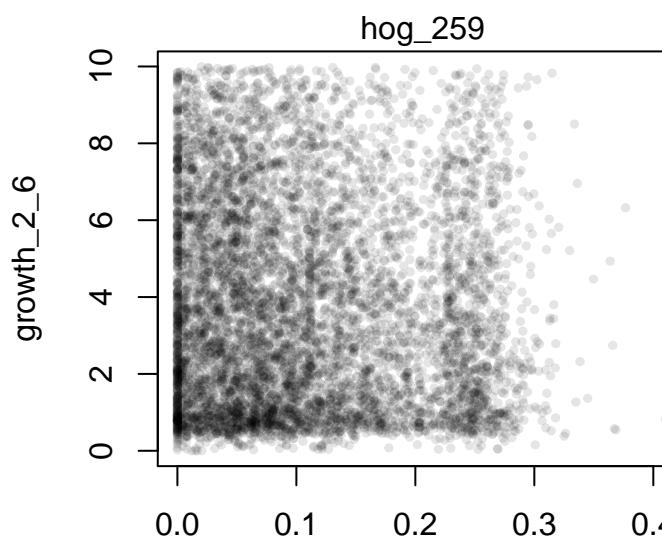
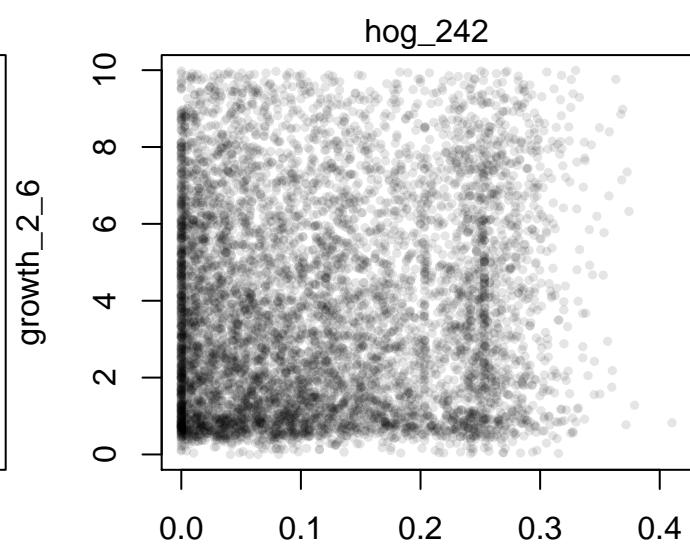
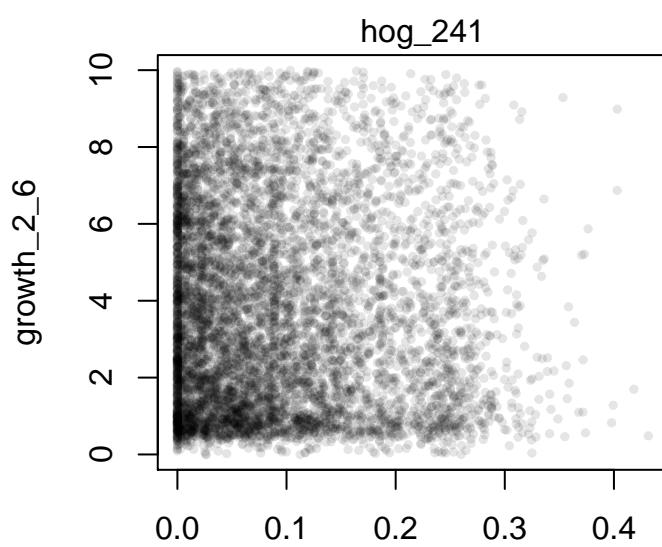
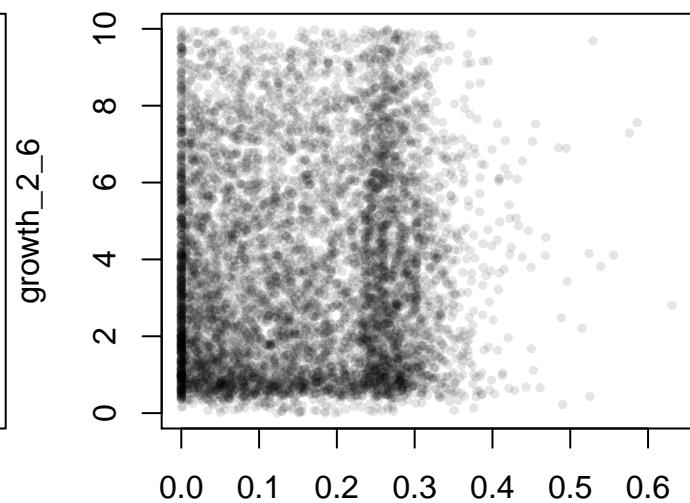
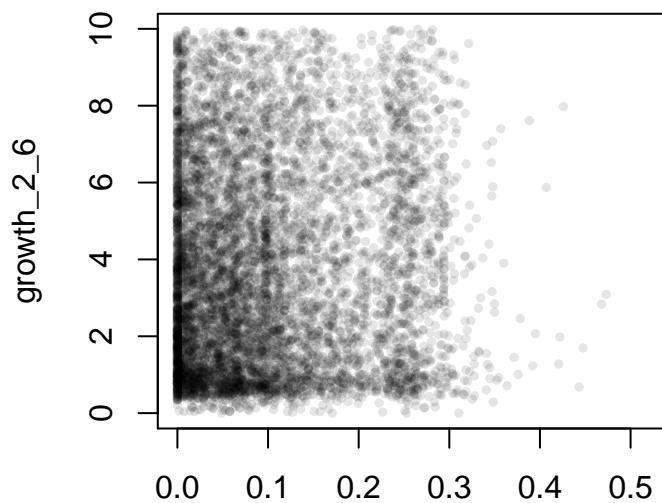
```





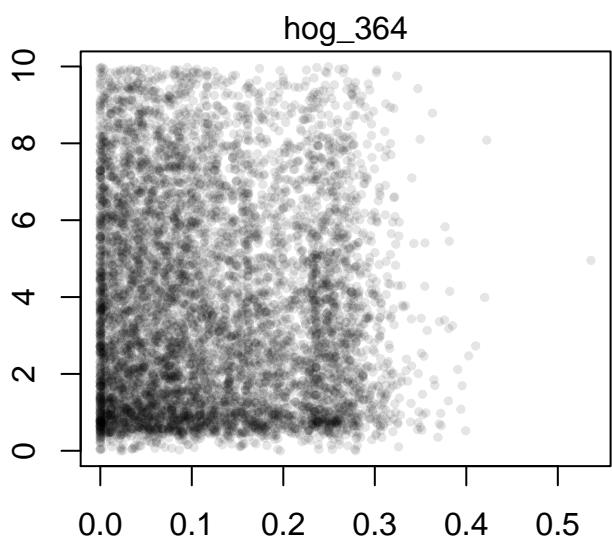
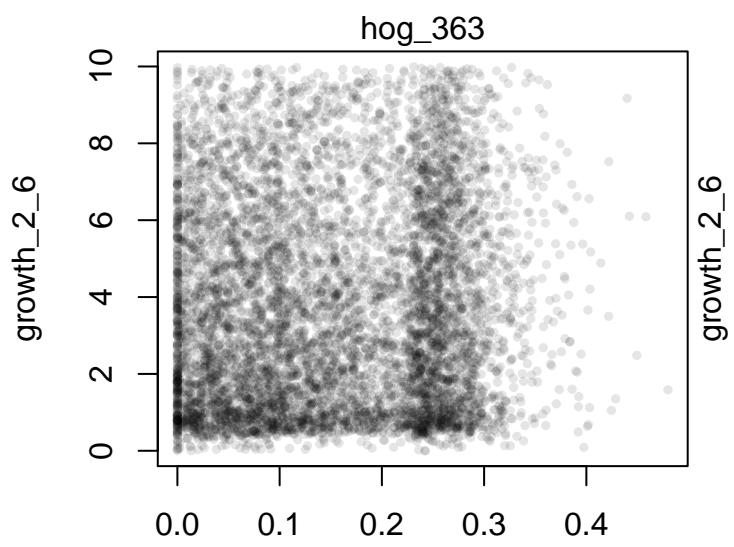
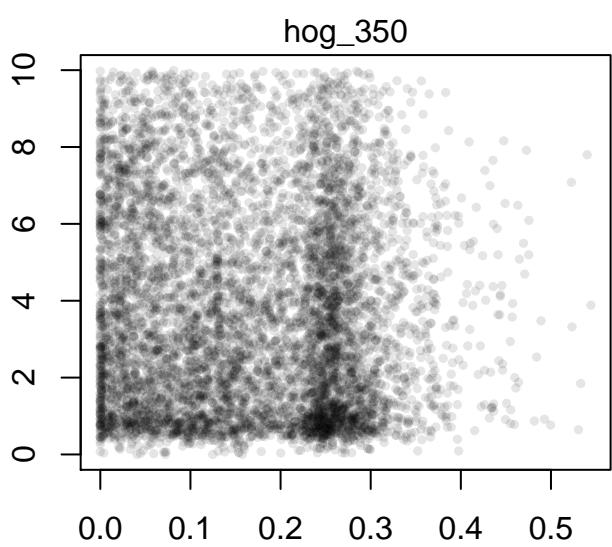
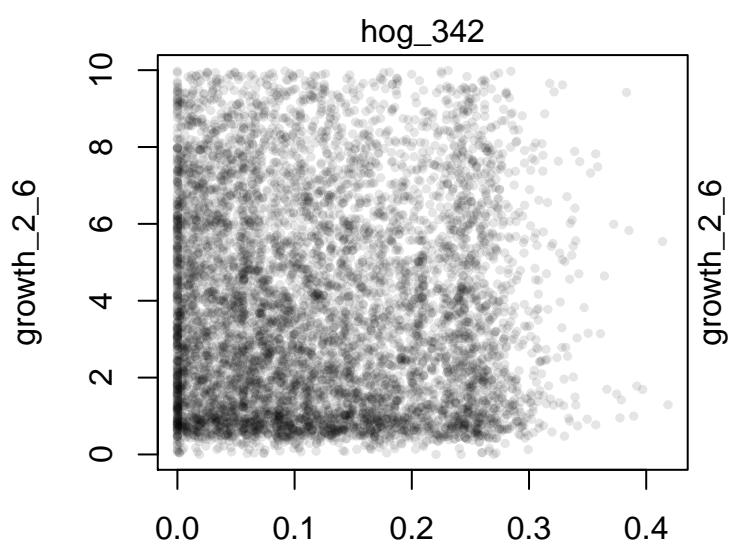
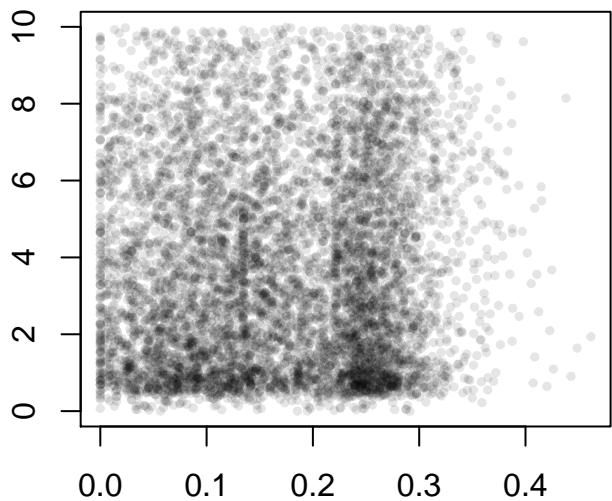
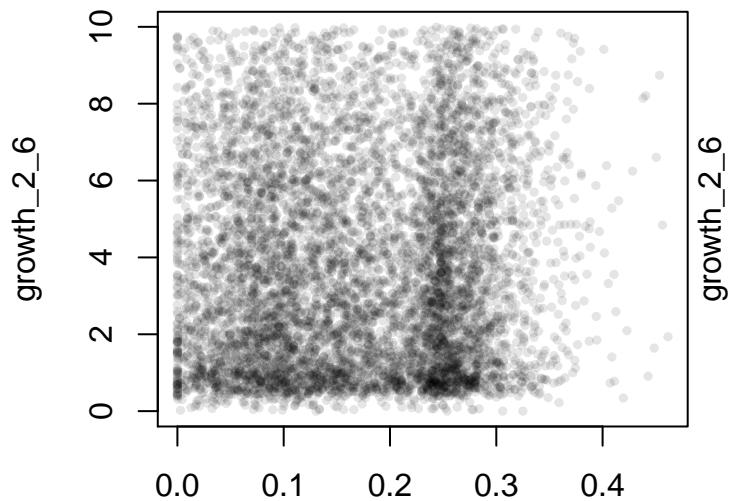


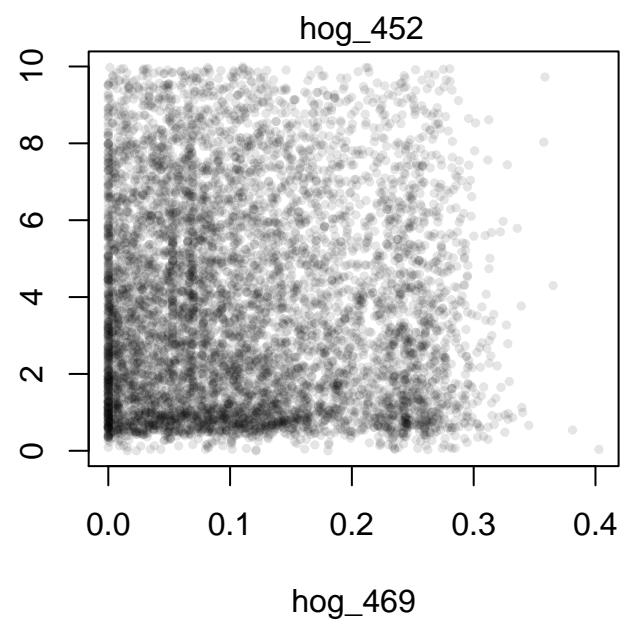
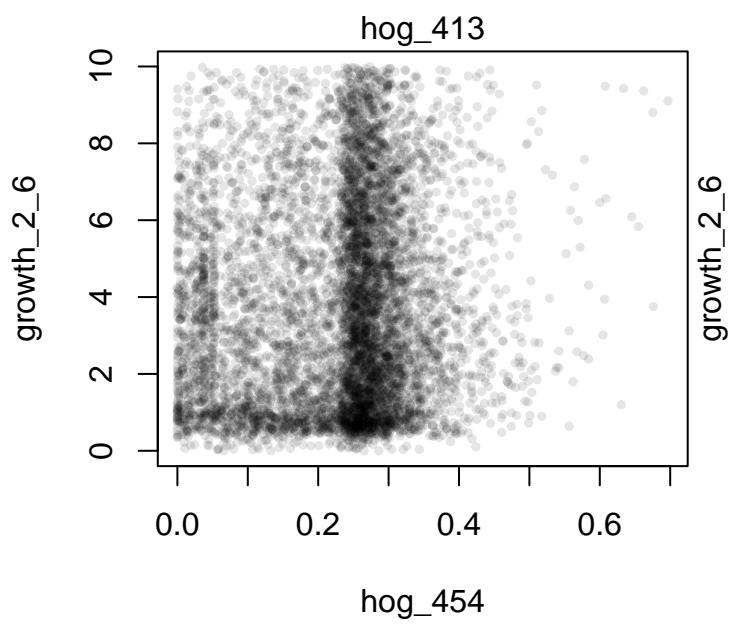
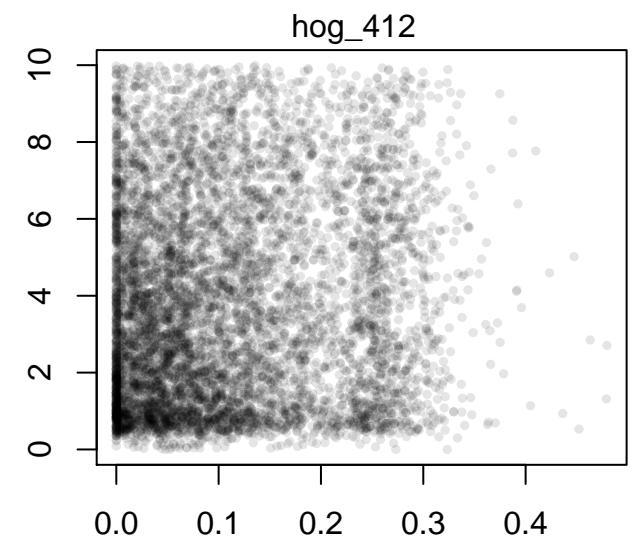
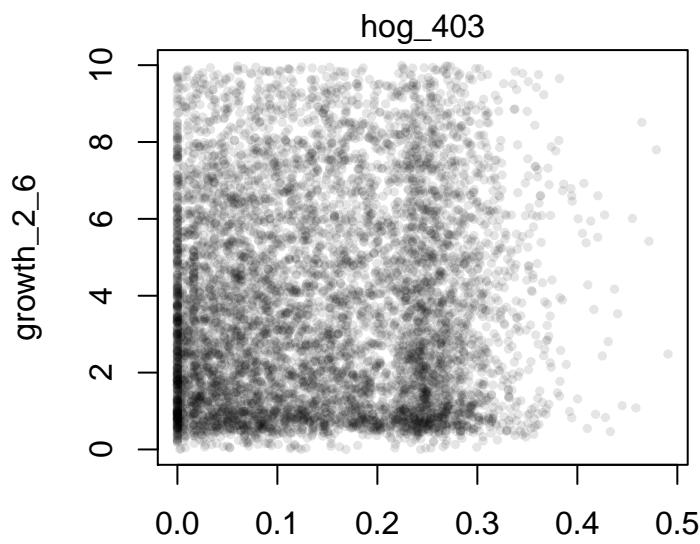
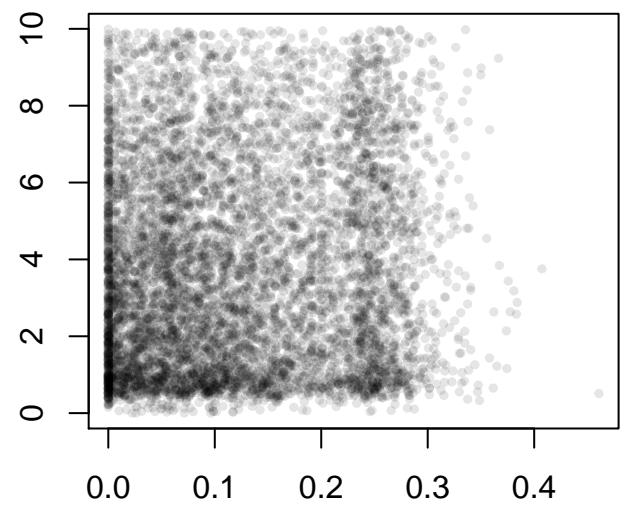
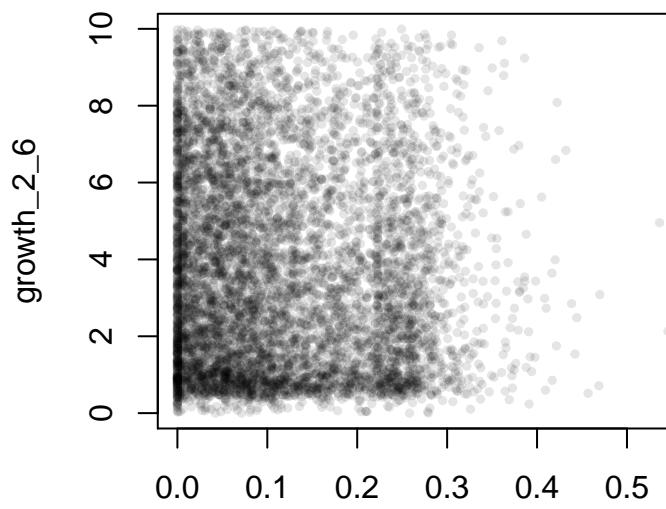




`hog_304`

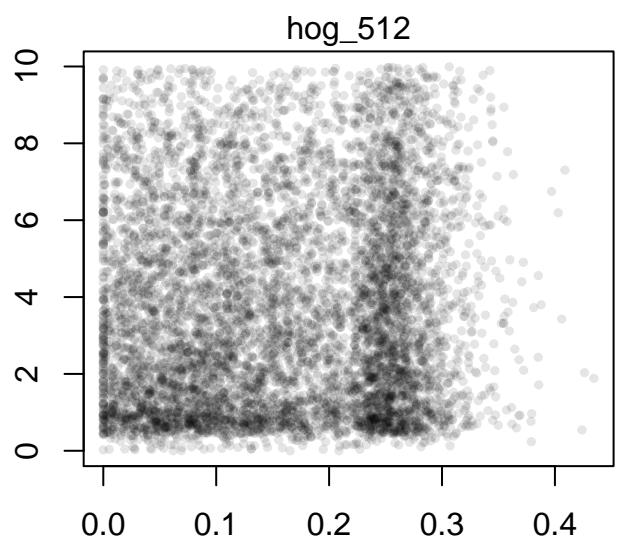
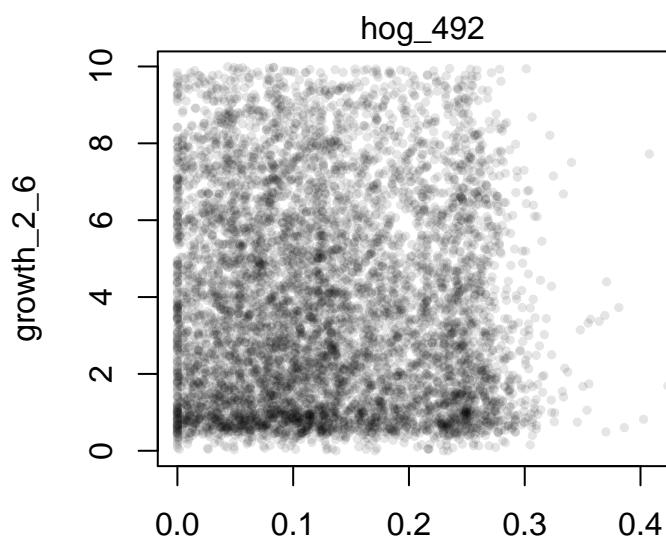
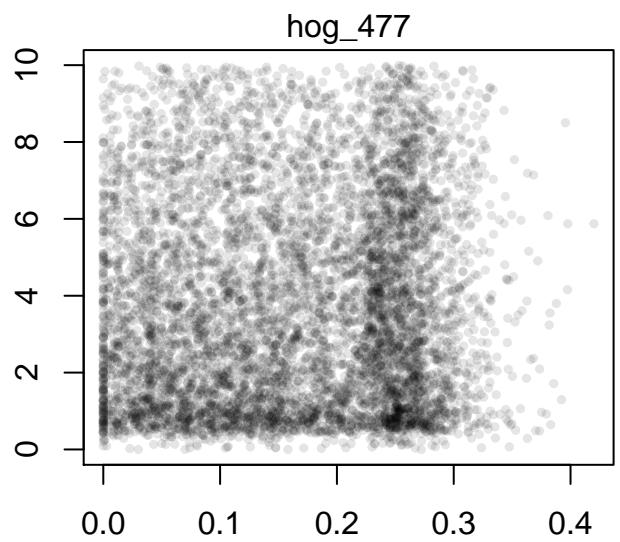
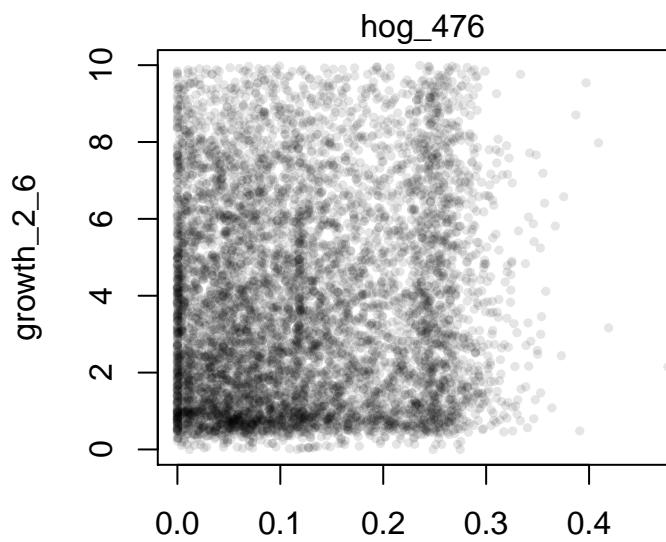
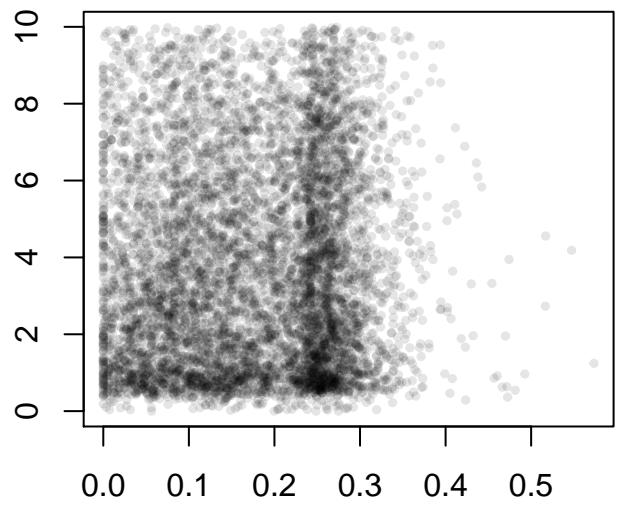
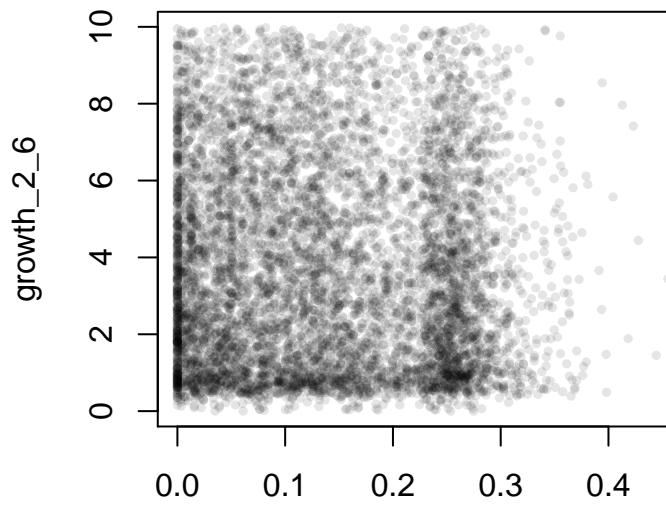
`hog_330`





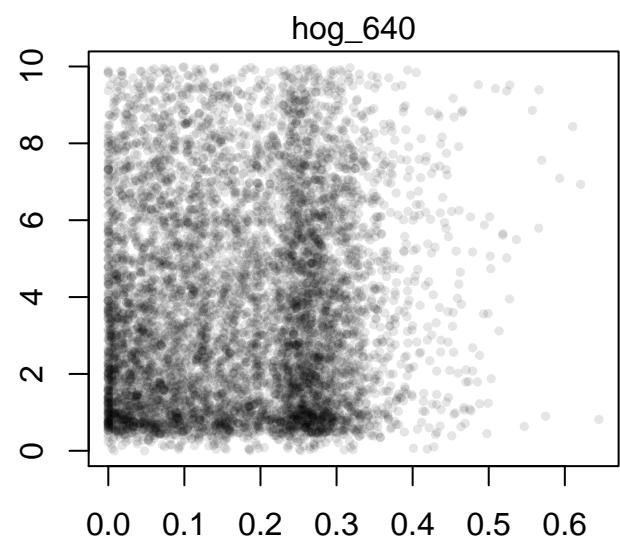
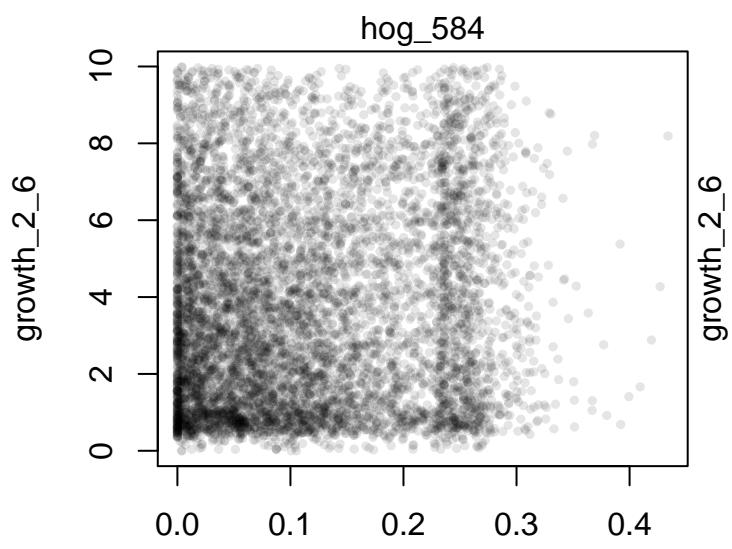
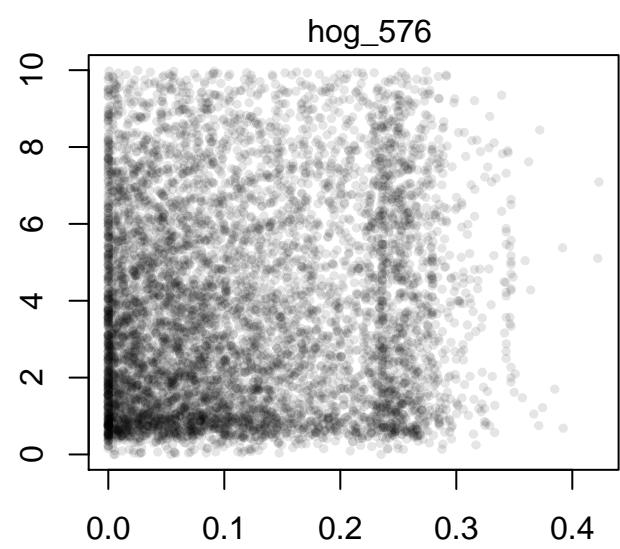
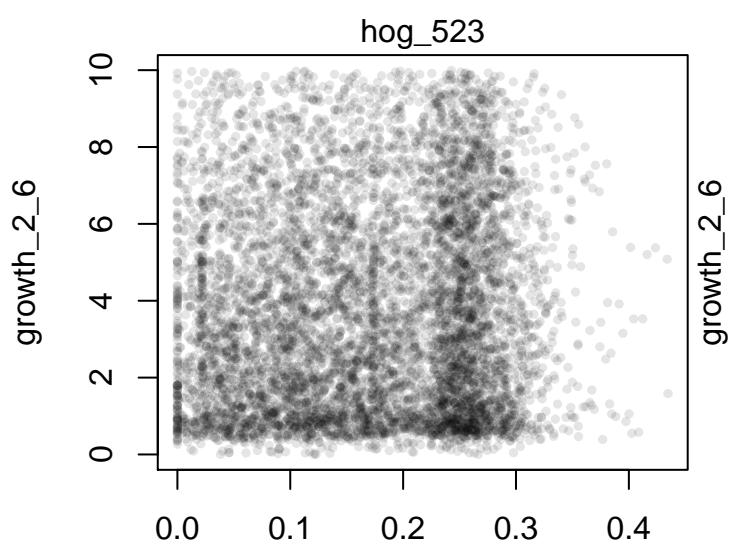
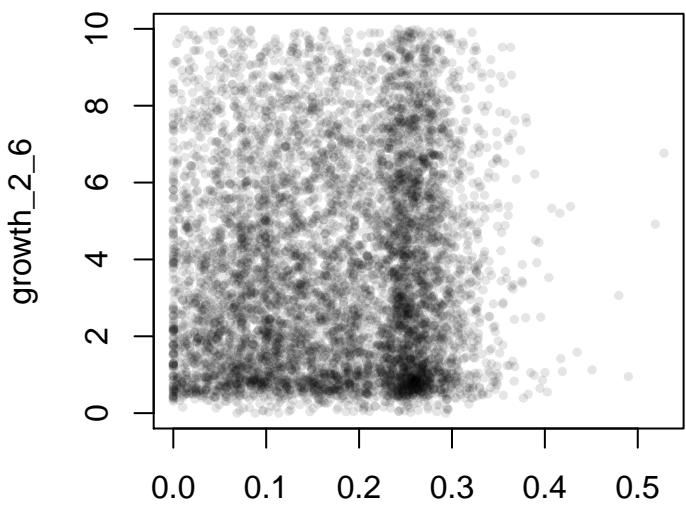
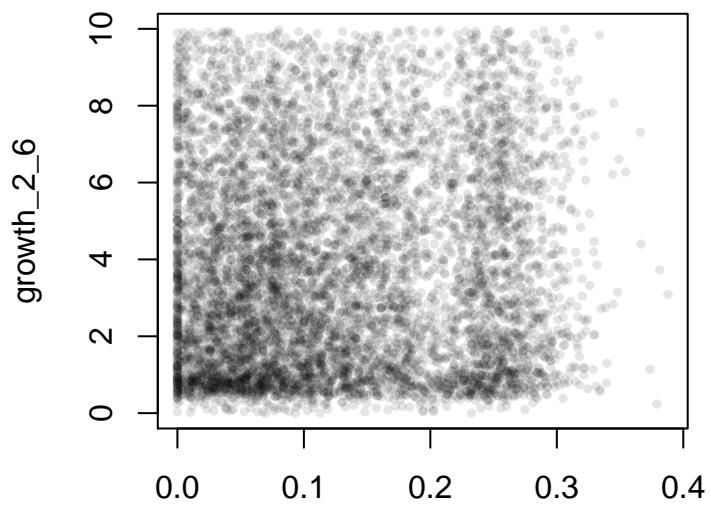
`hog_454`

`hog_469`



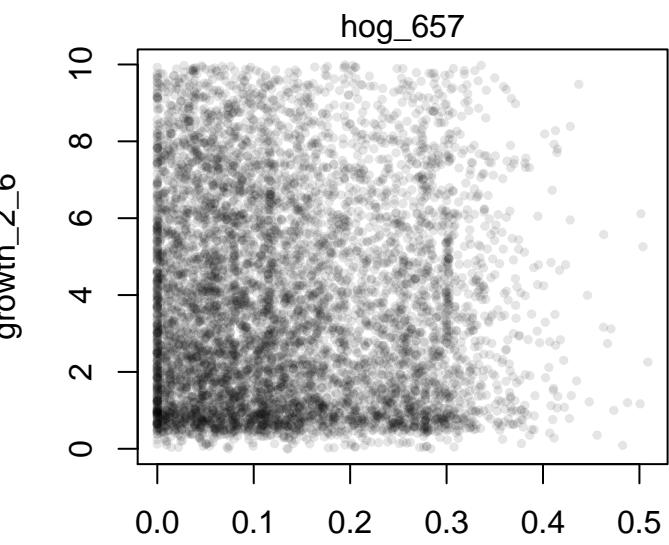
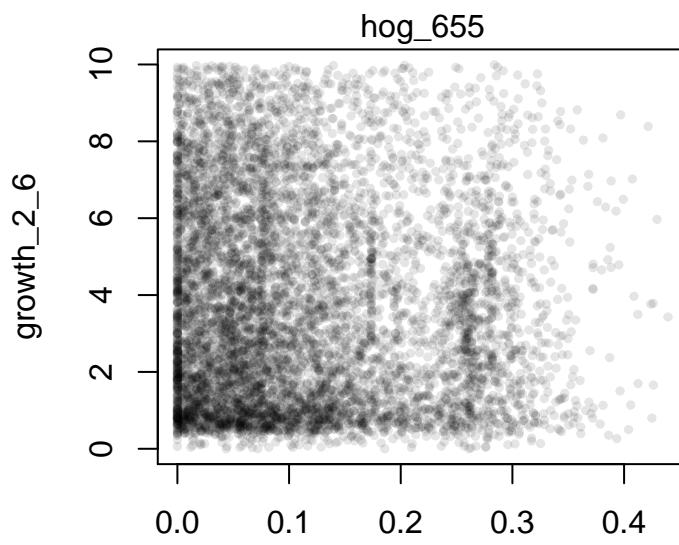
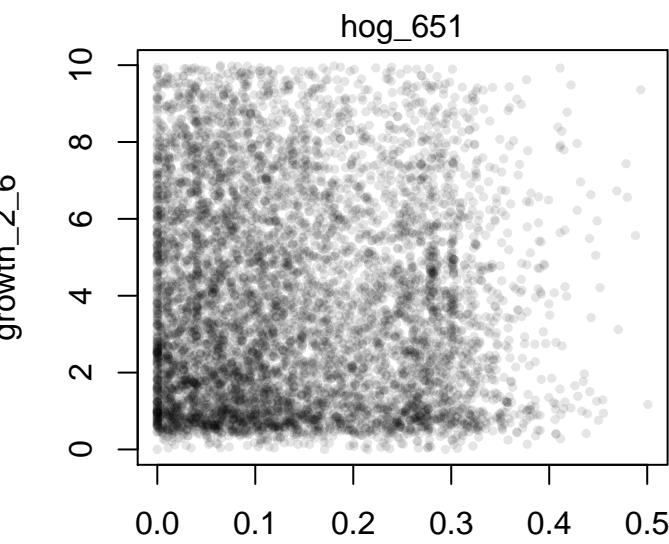
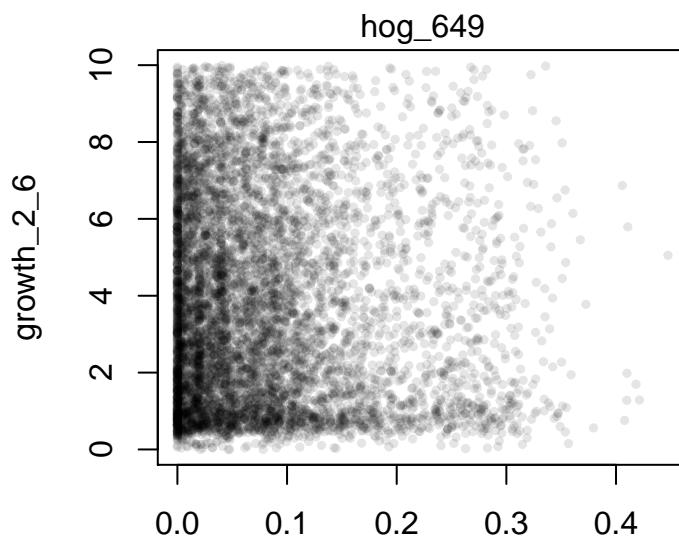
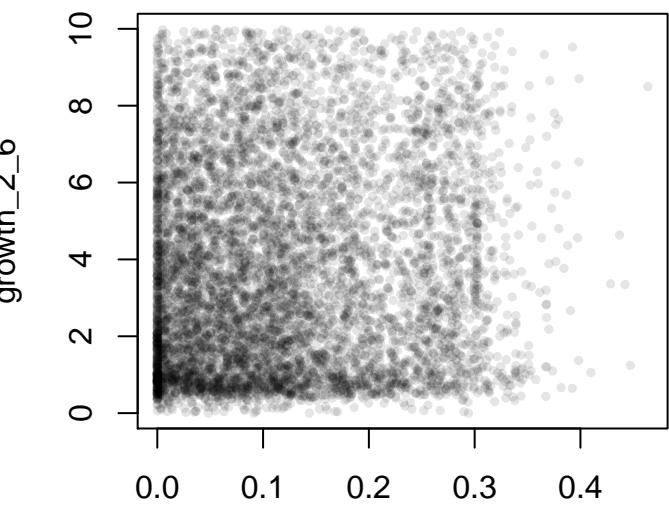
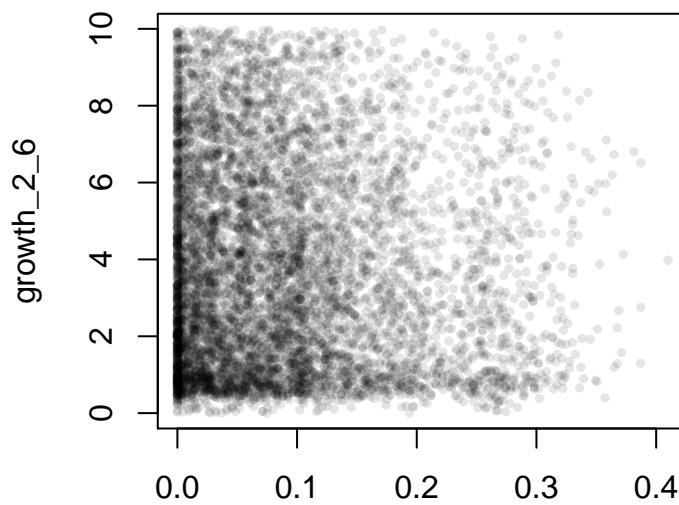
hog_514

hog_522



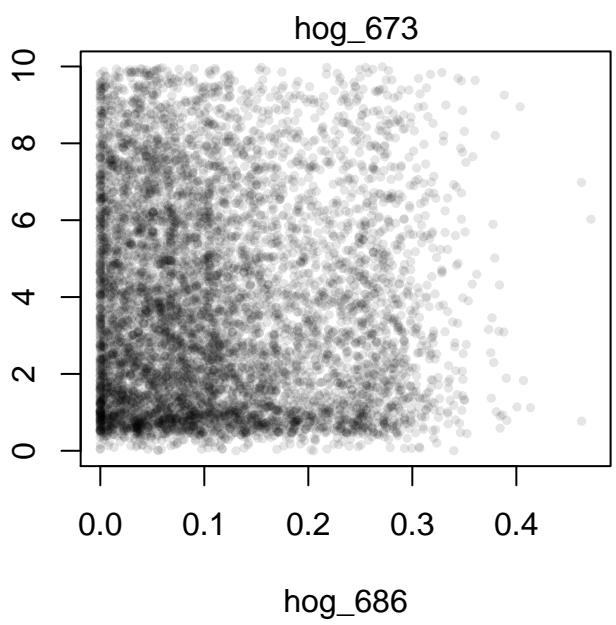
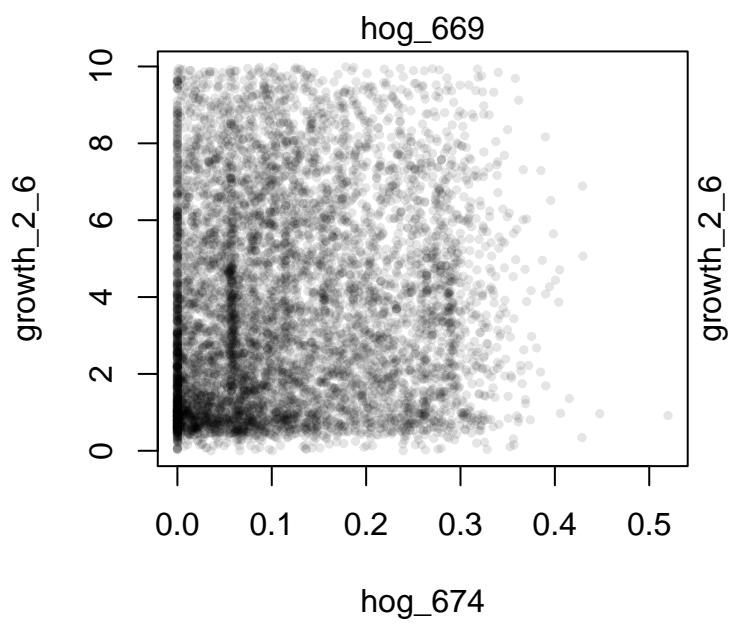
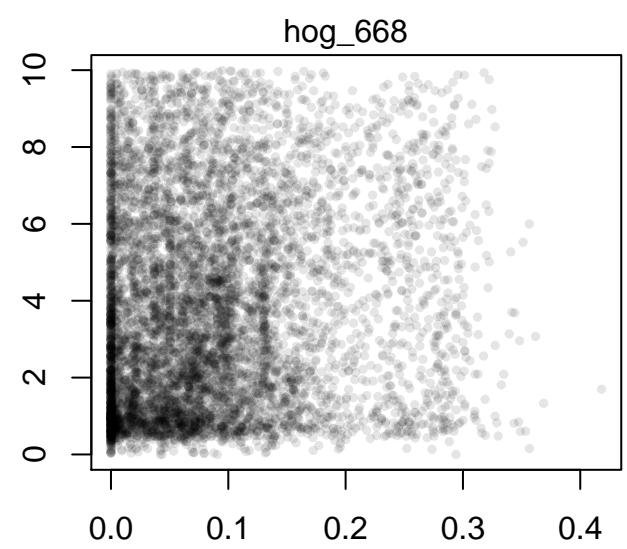
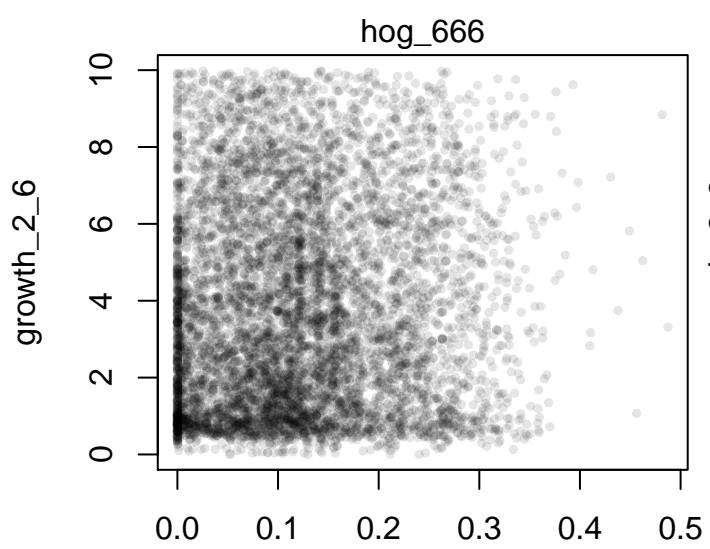
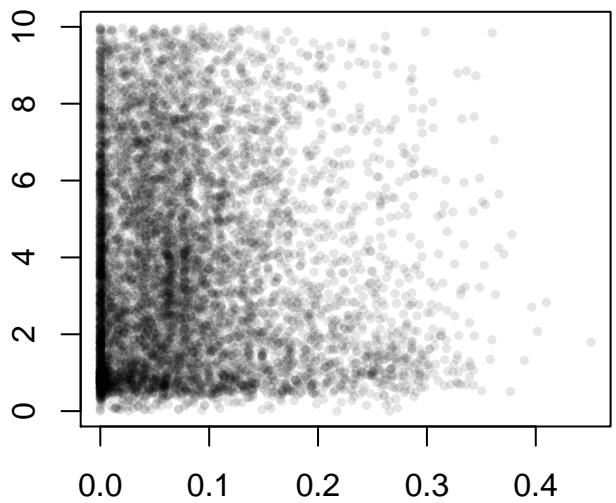
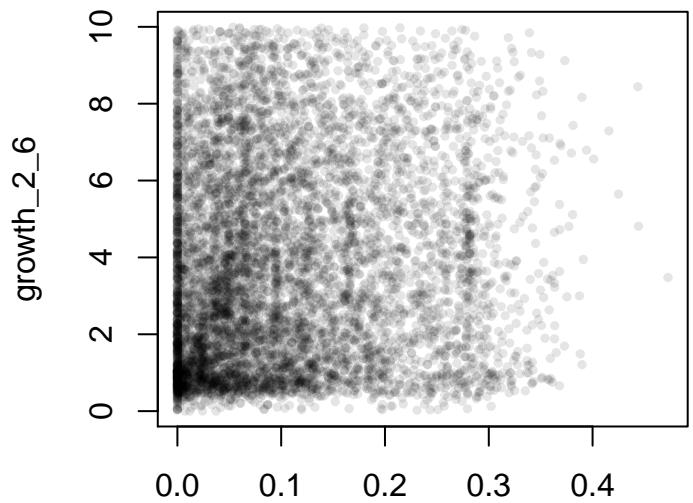
`hog_641`

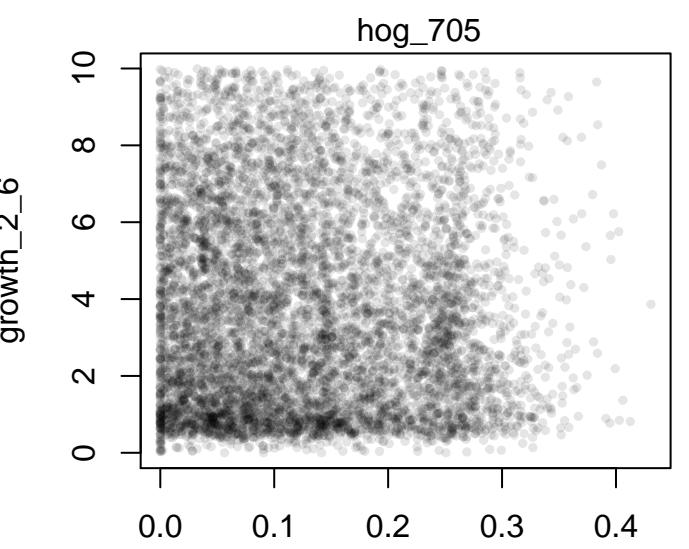
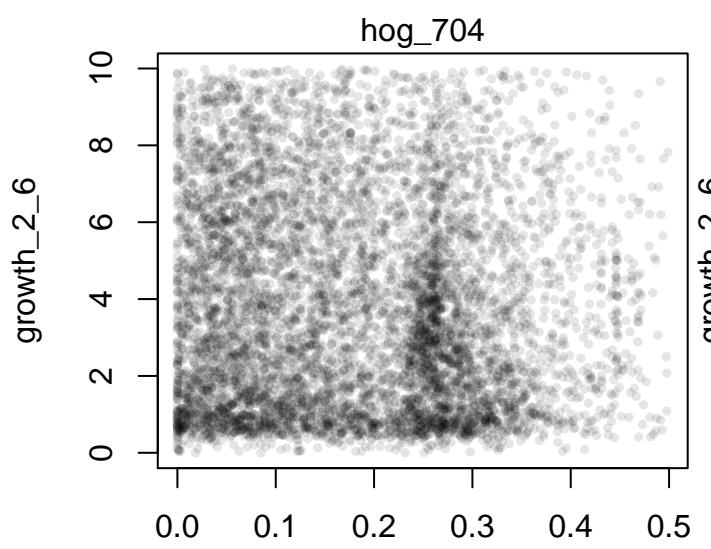
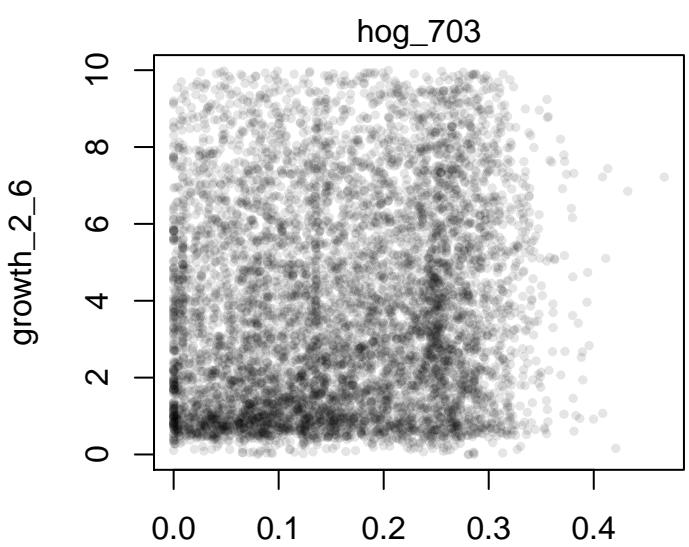
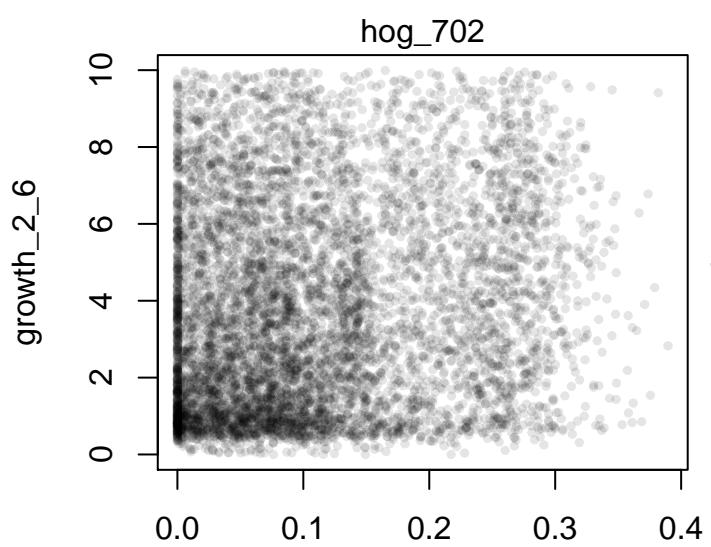
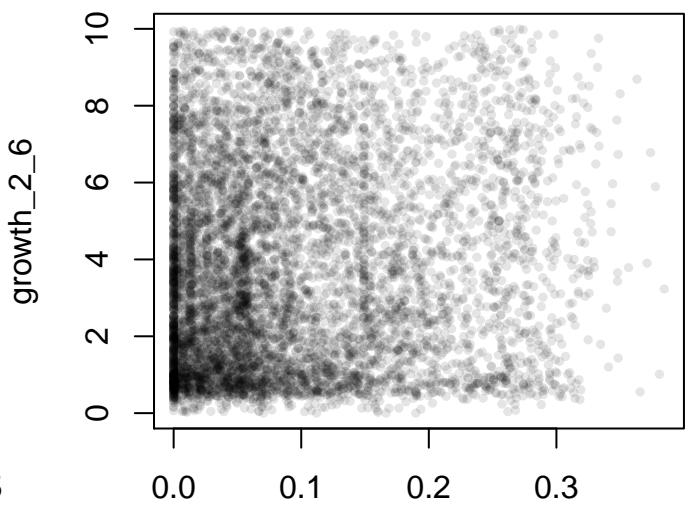
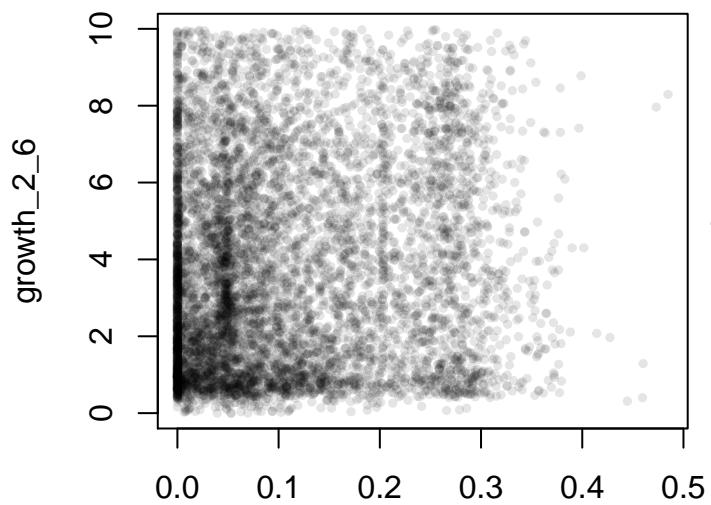
`hog_643`



hog_658

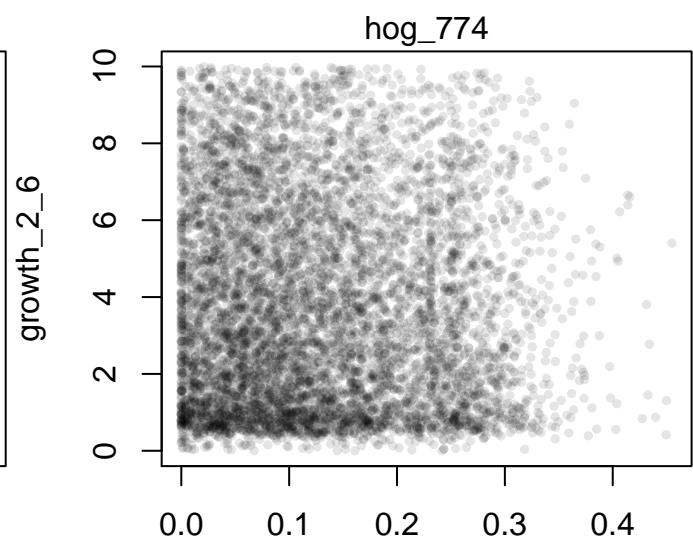
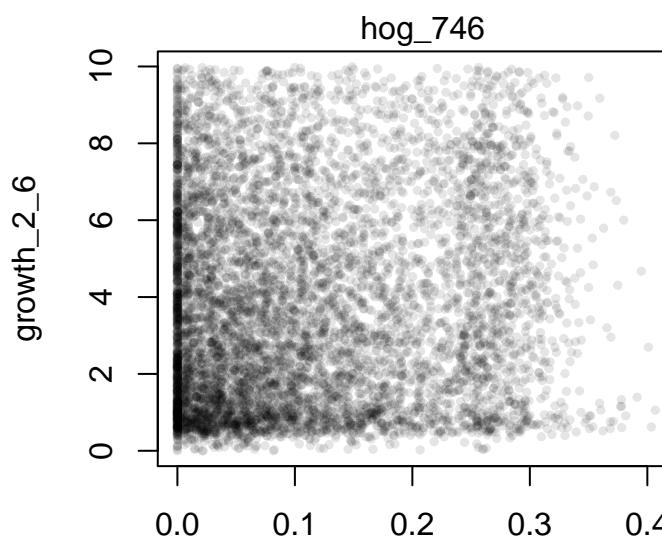
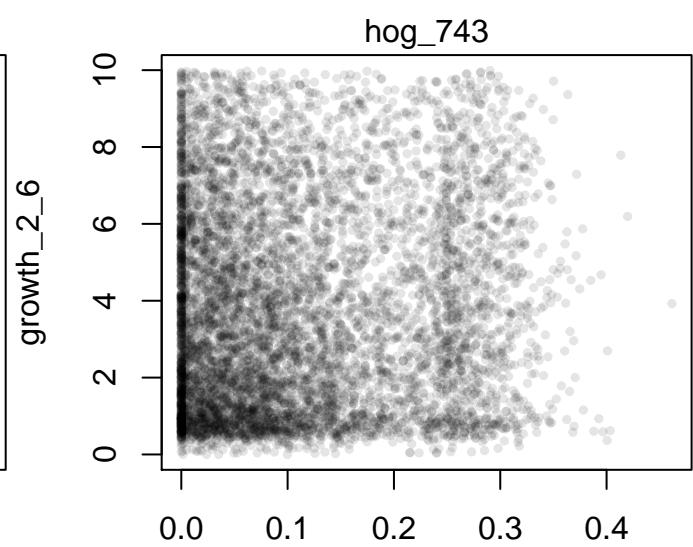
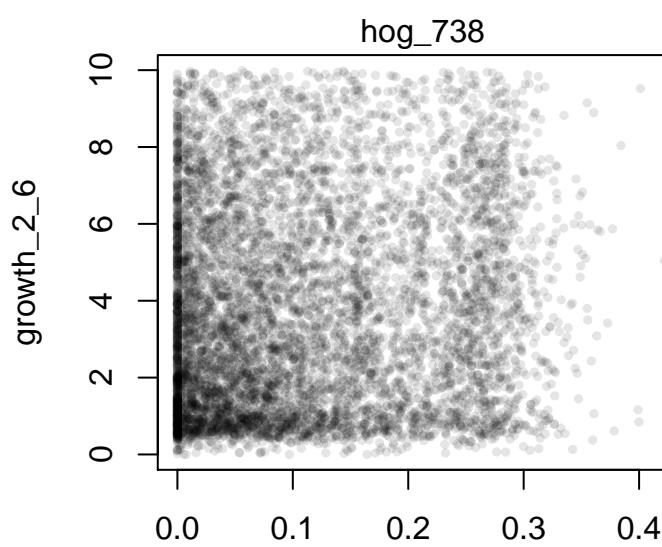
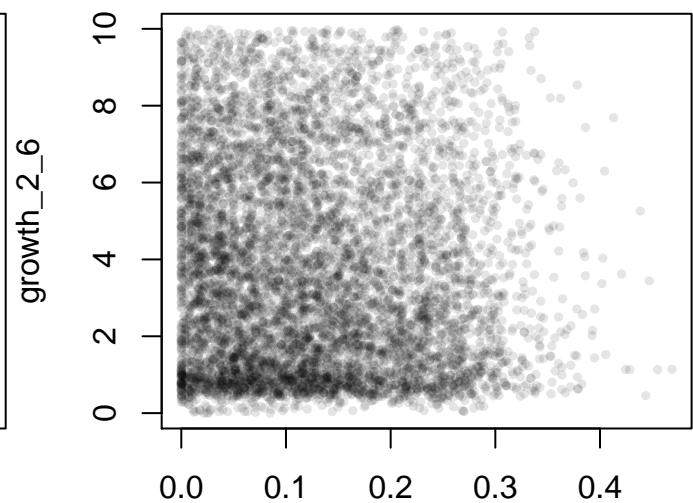
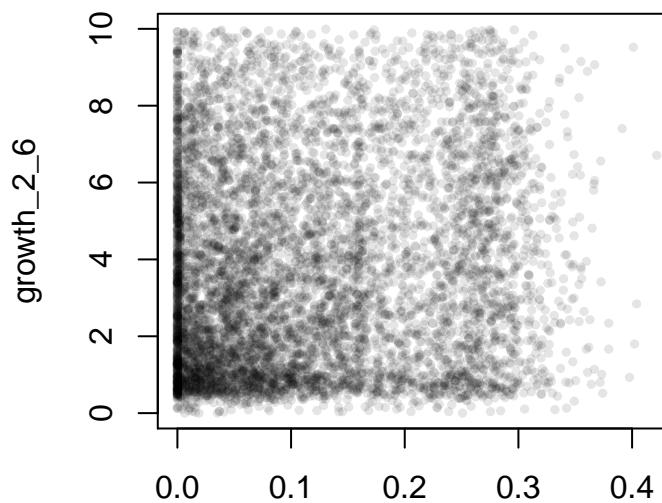
hog_665





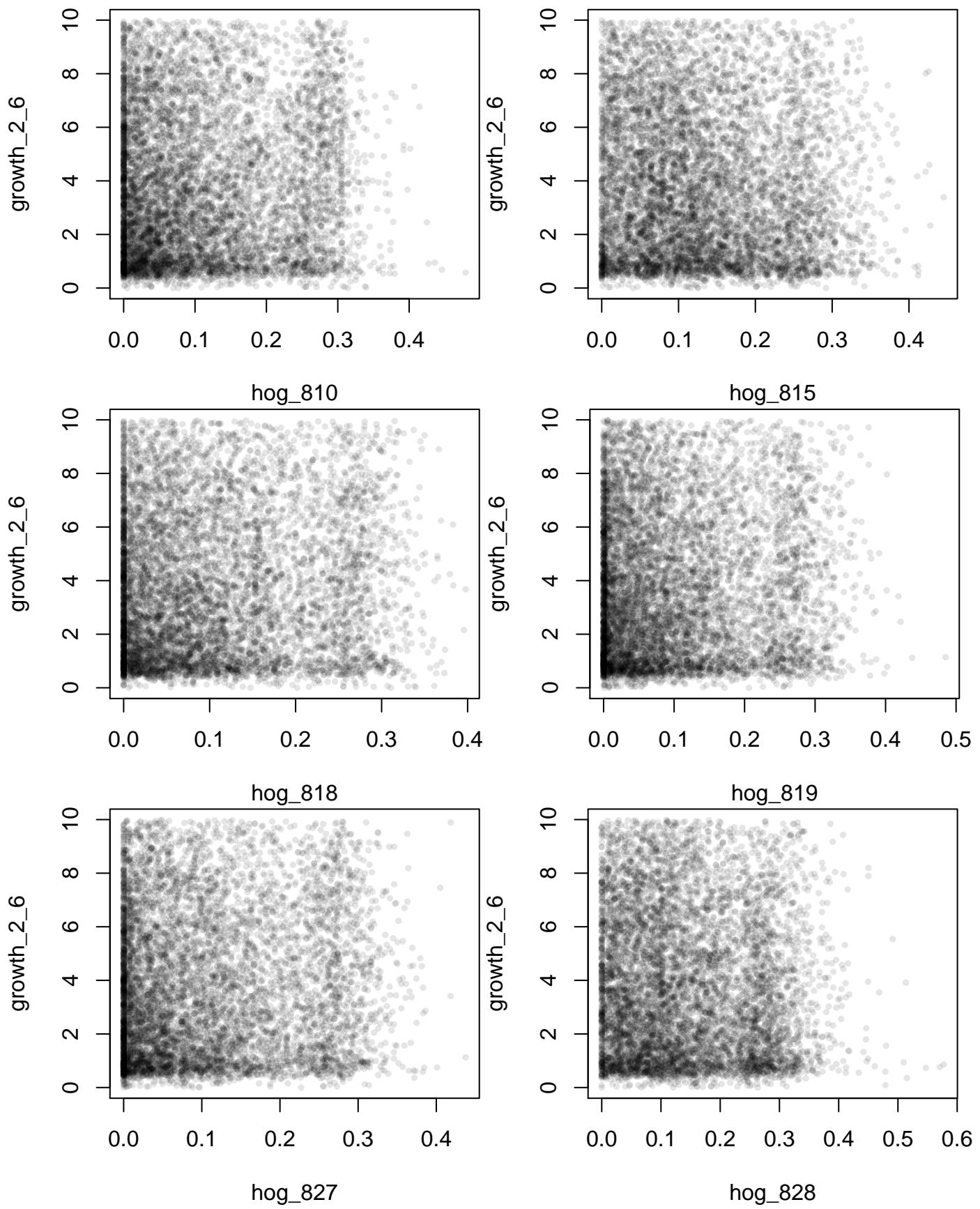
`hog_724`

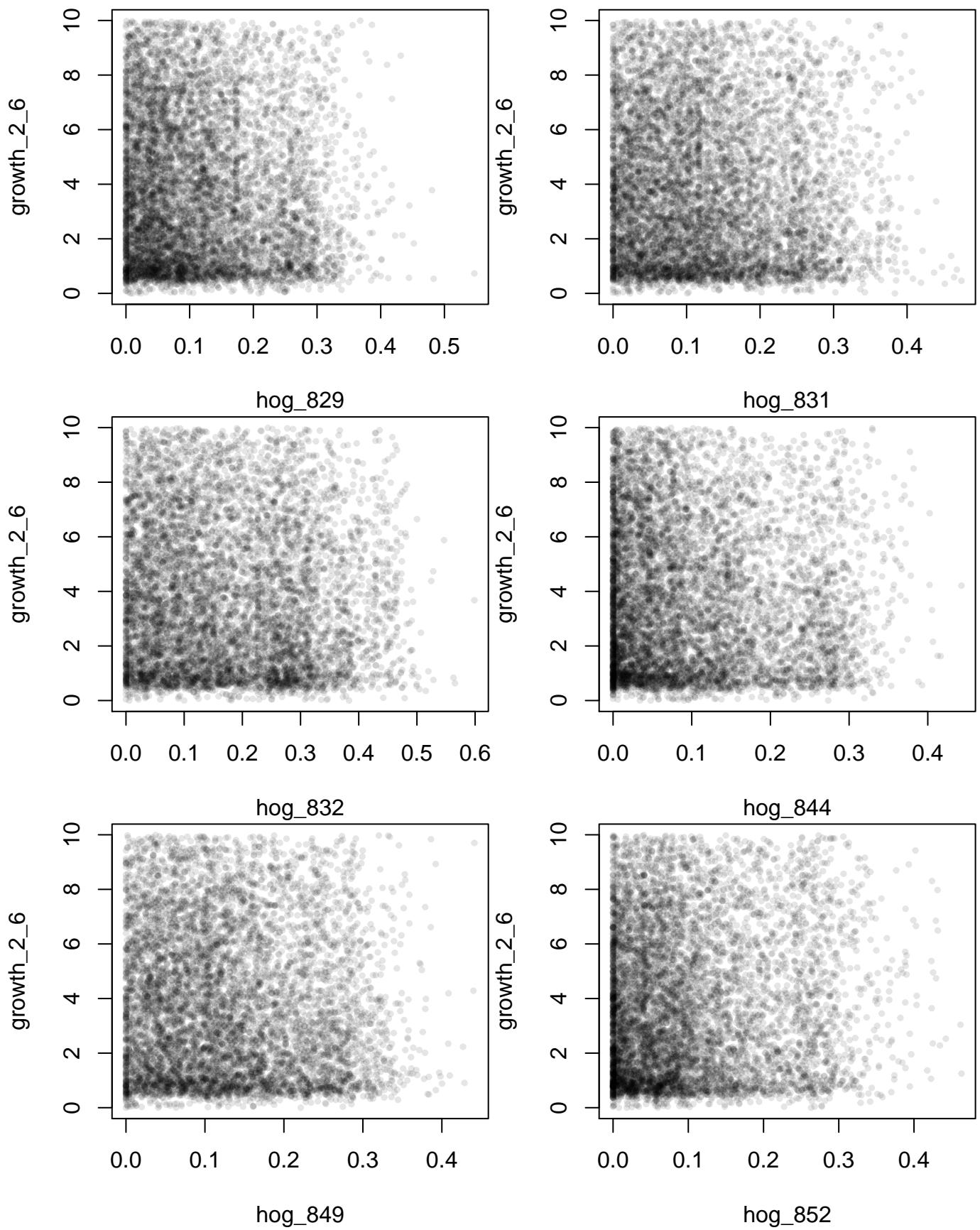
`hog_725`

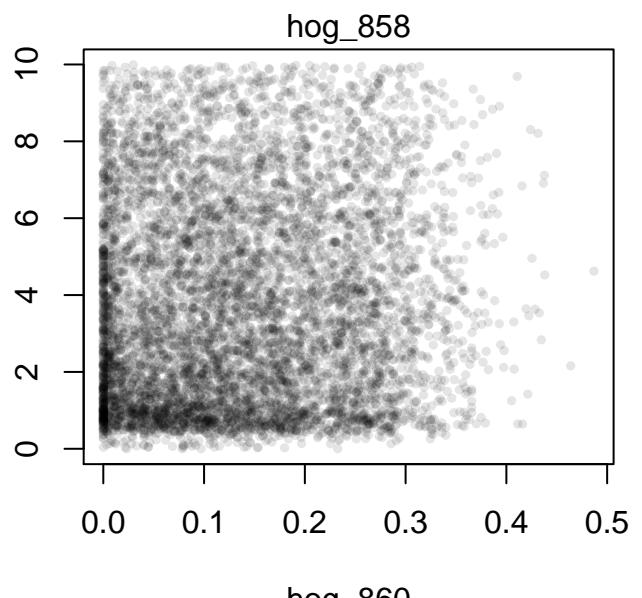
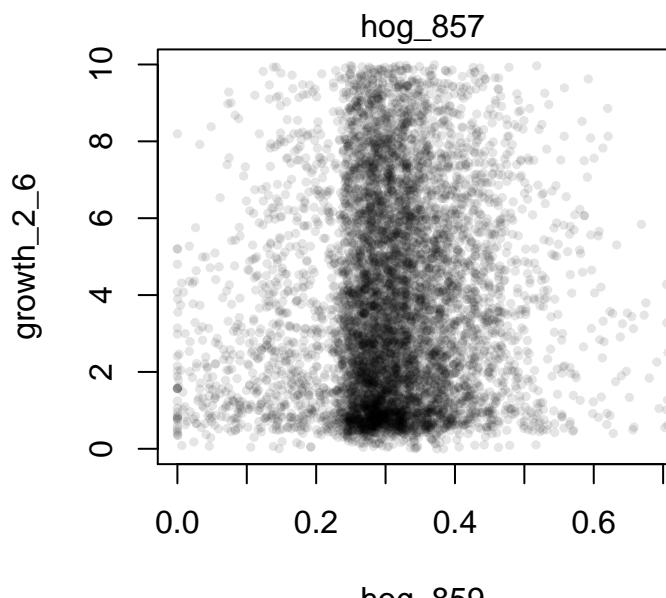
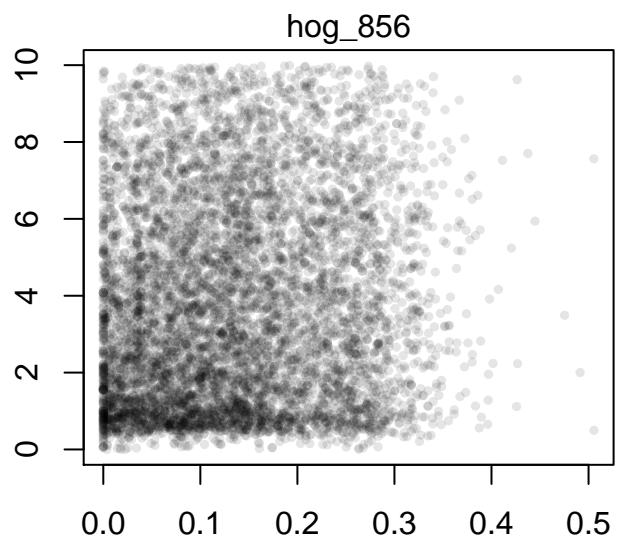
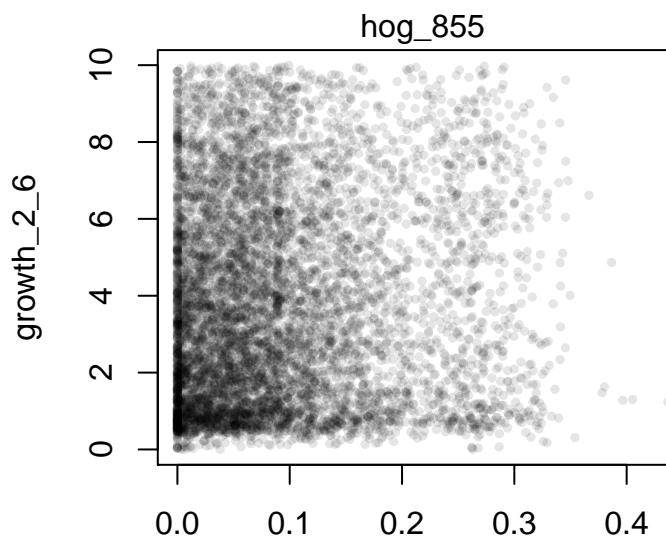
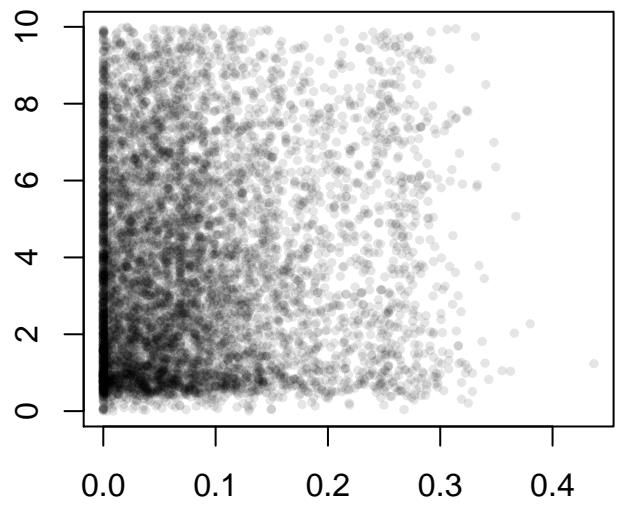
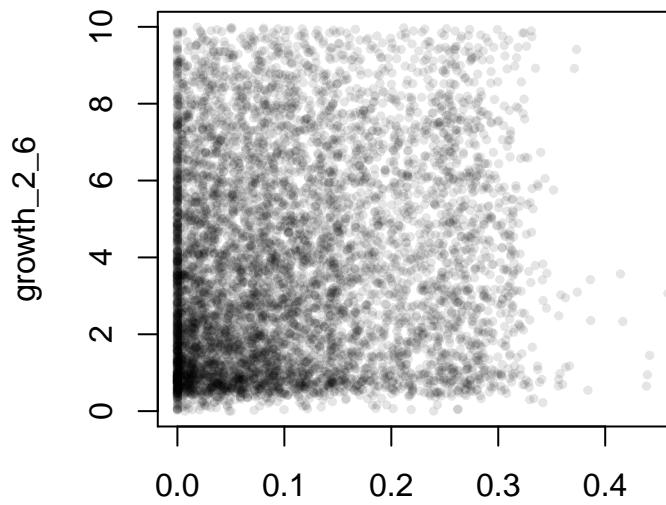


hog_782

hog_797

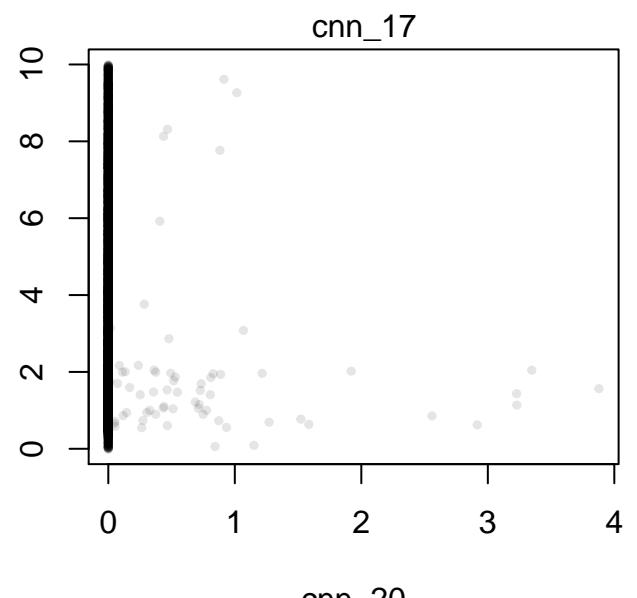
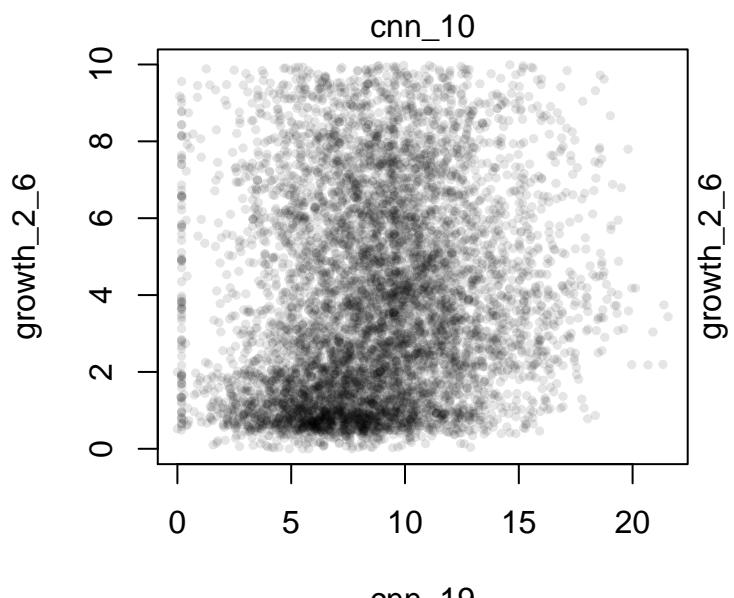
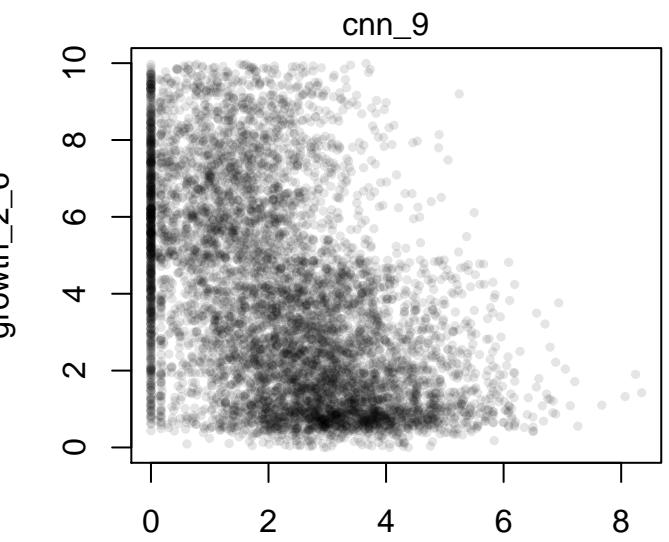
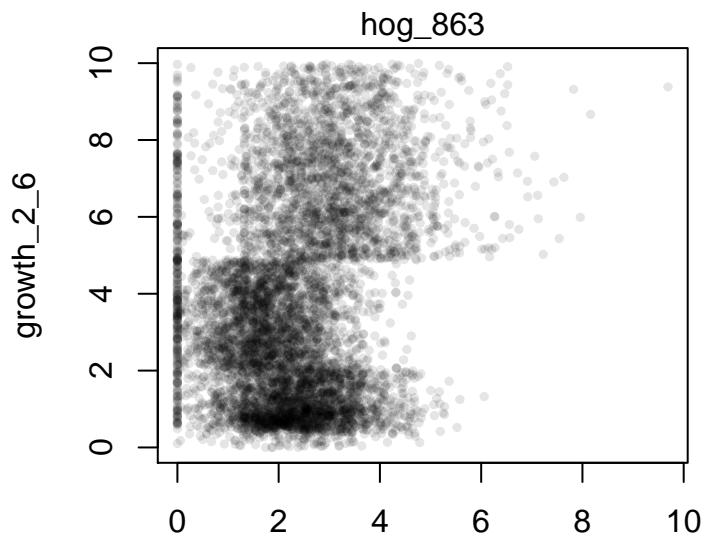
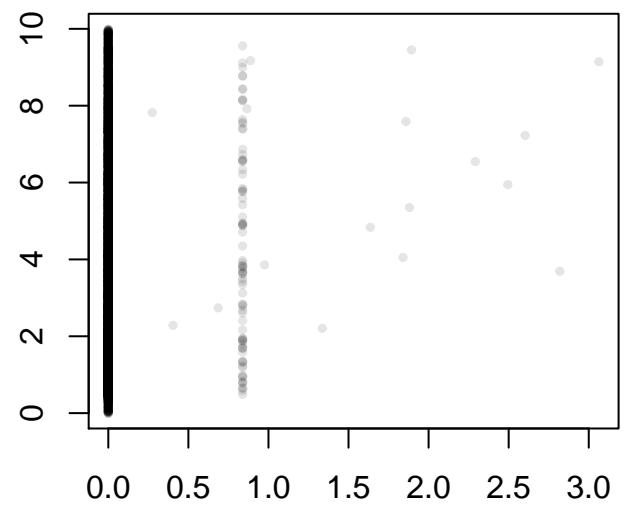
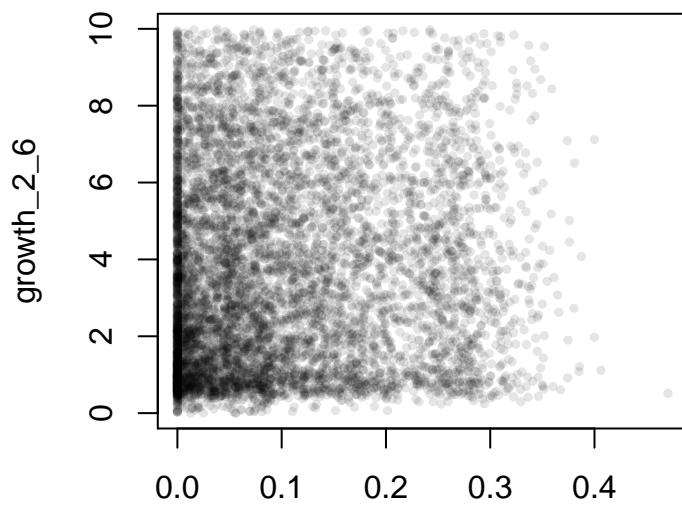


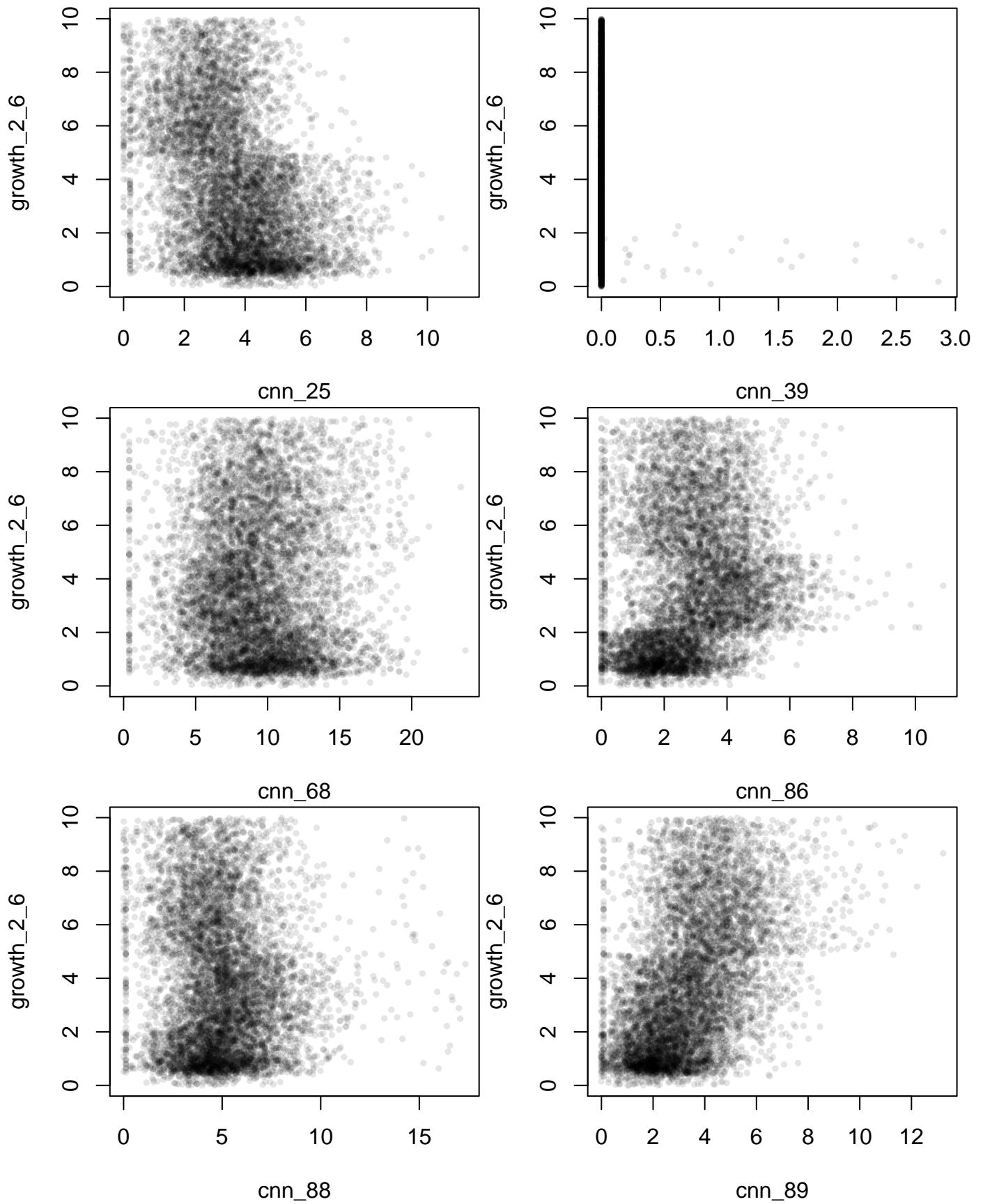


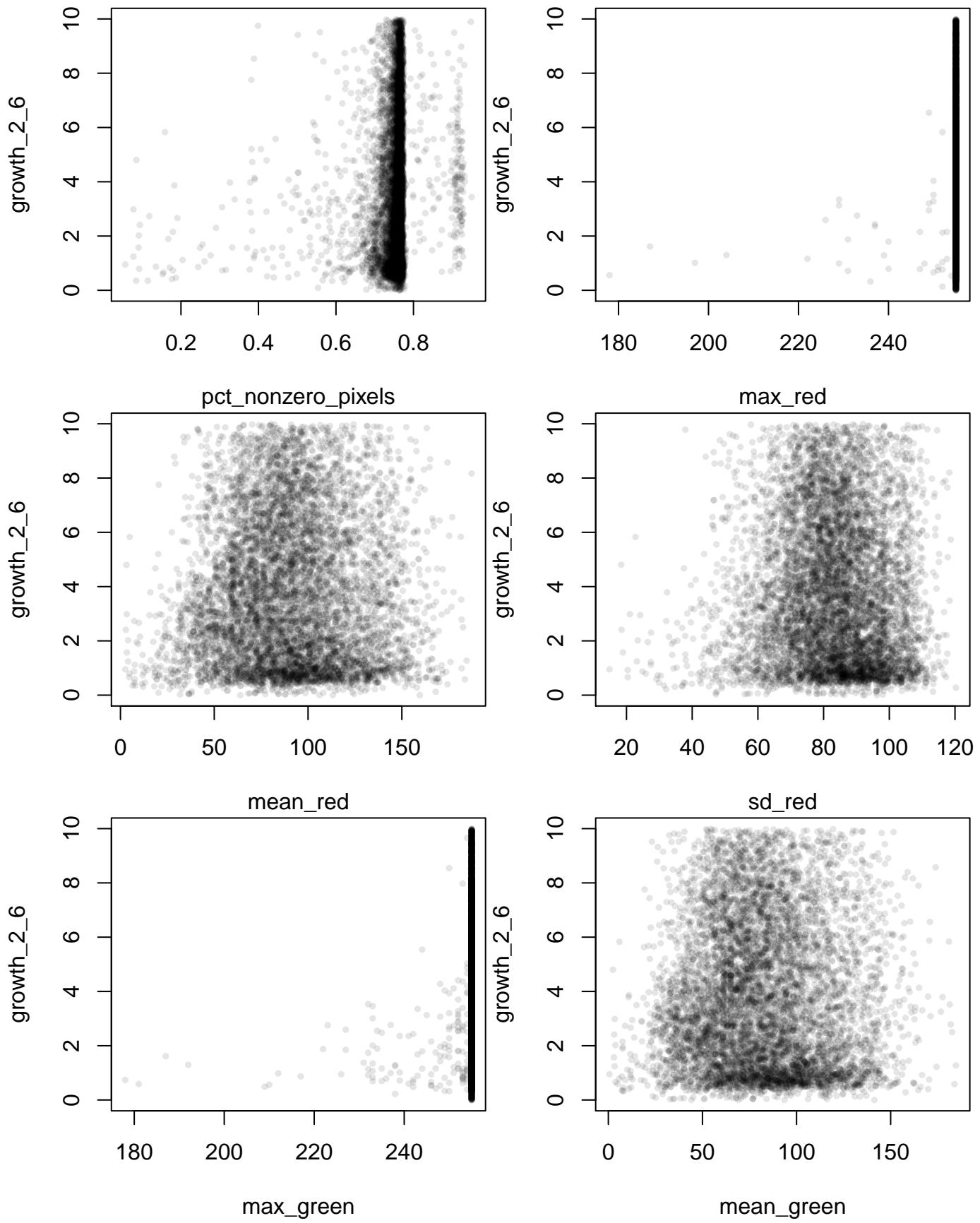


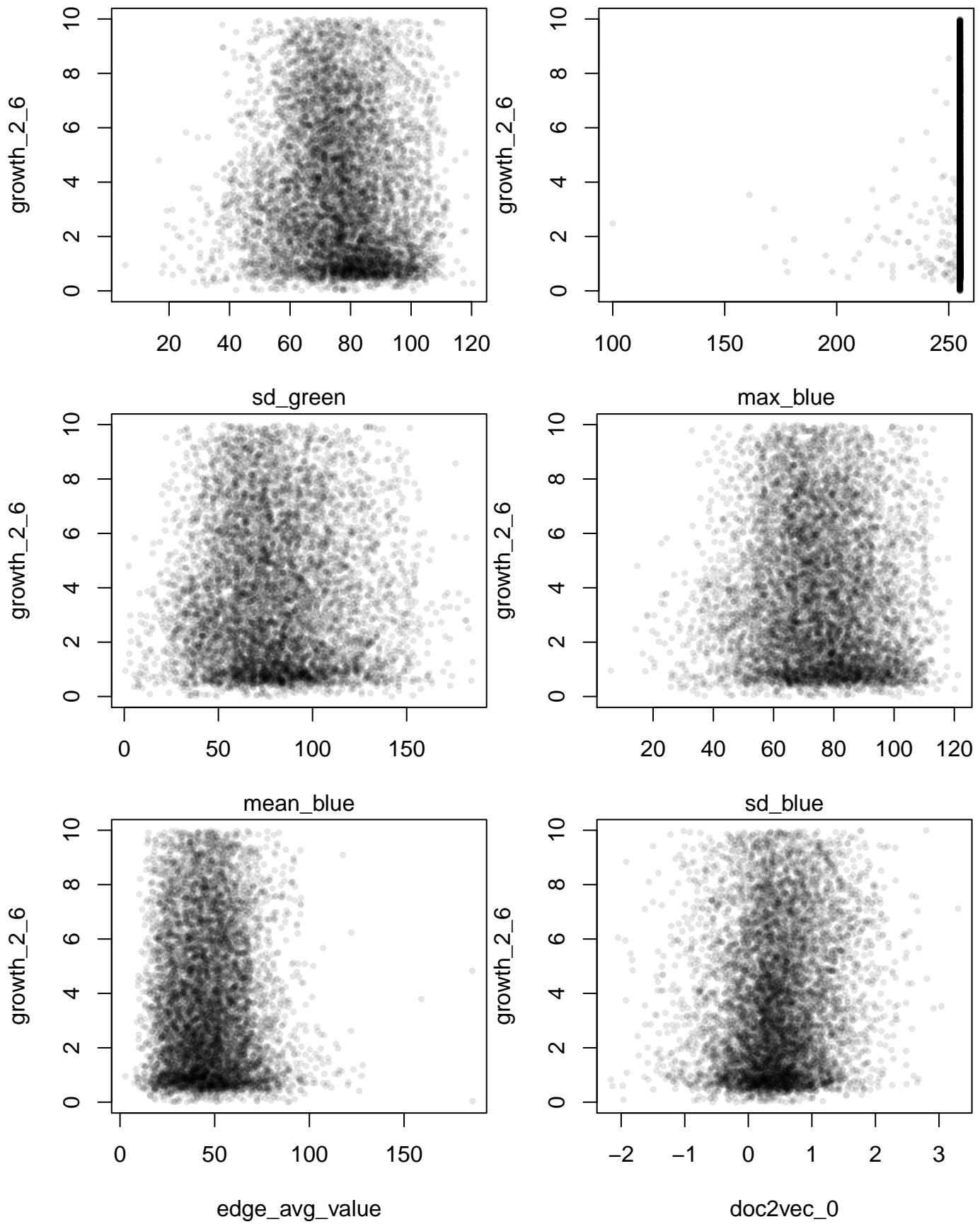
`hog_859`

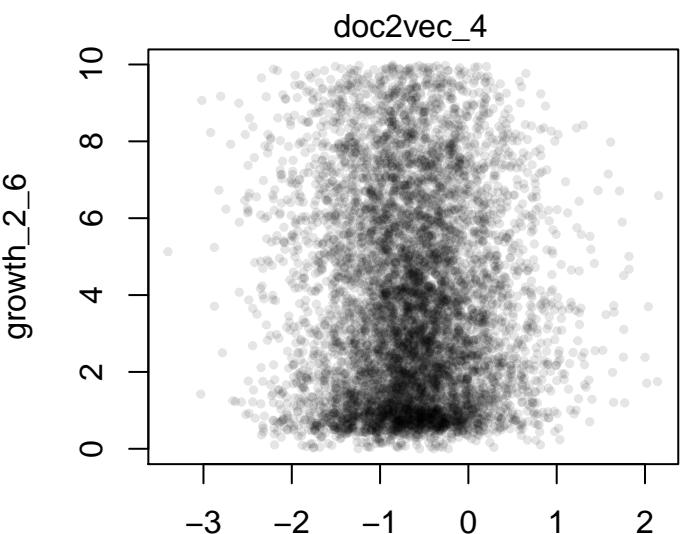
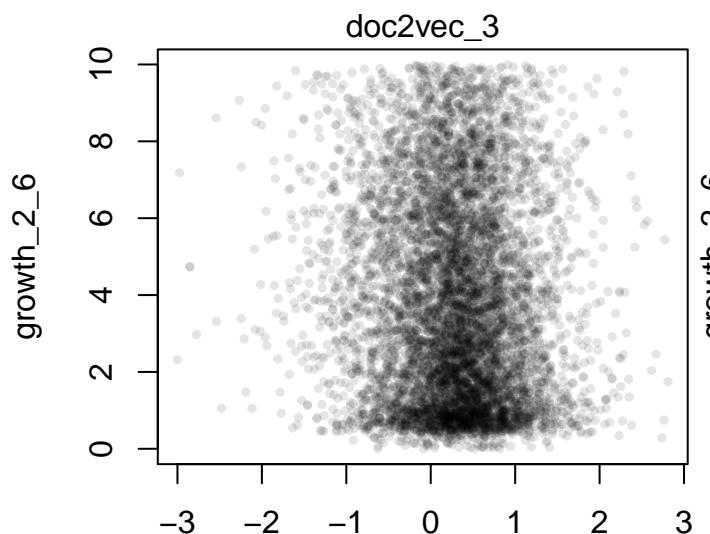
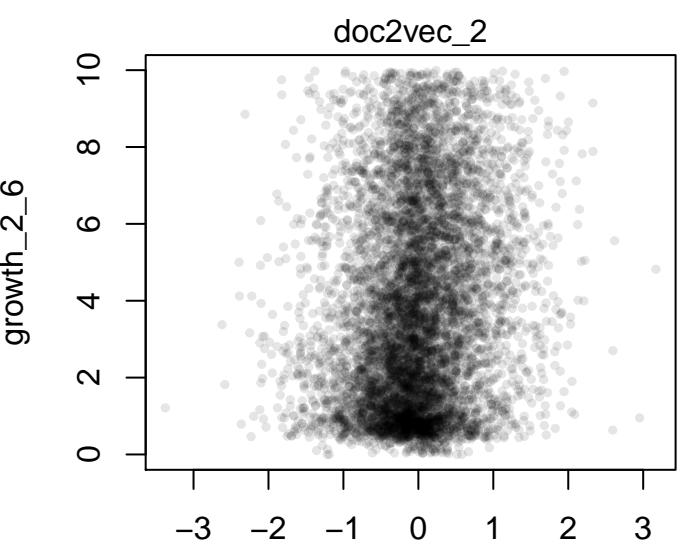
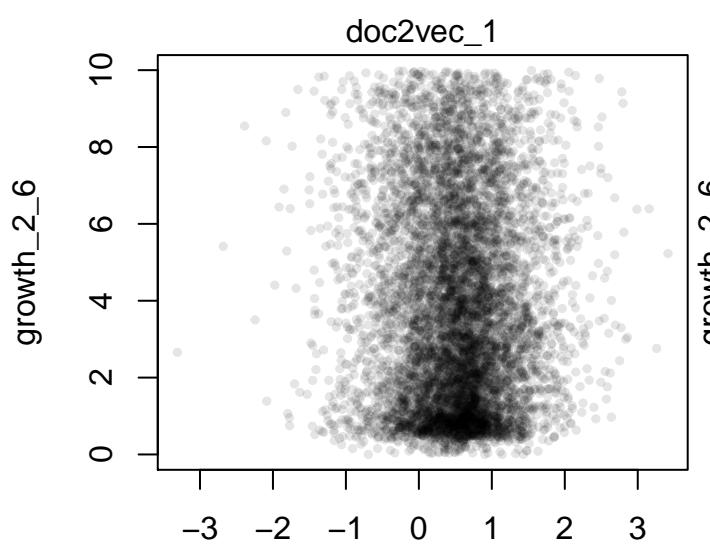
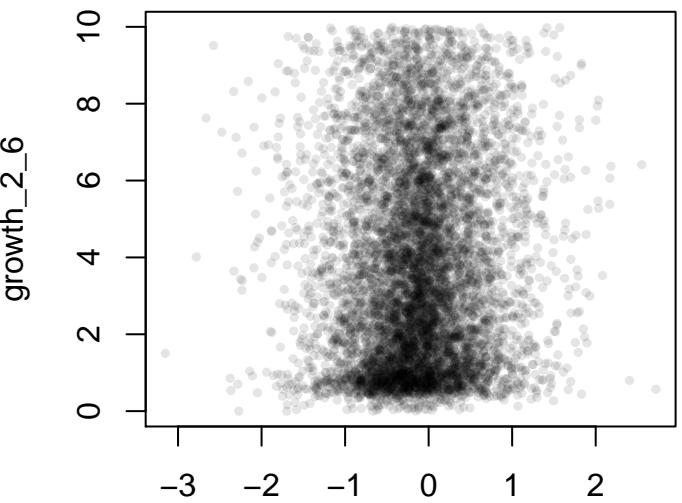
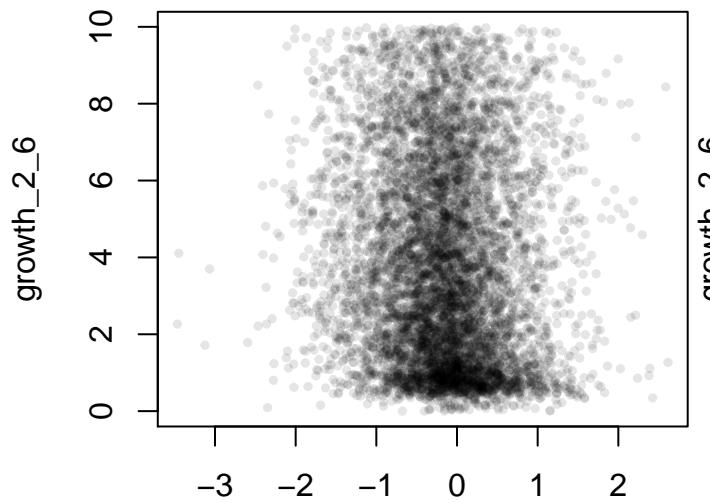
`hog_860`





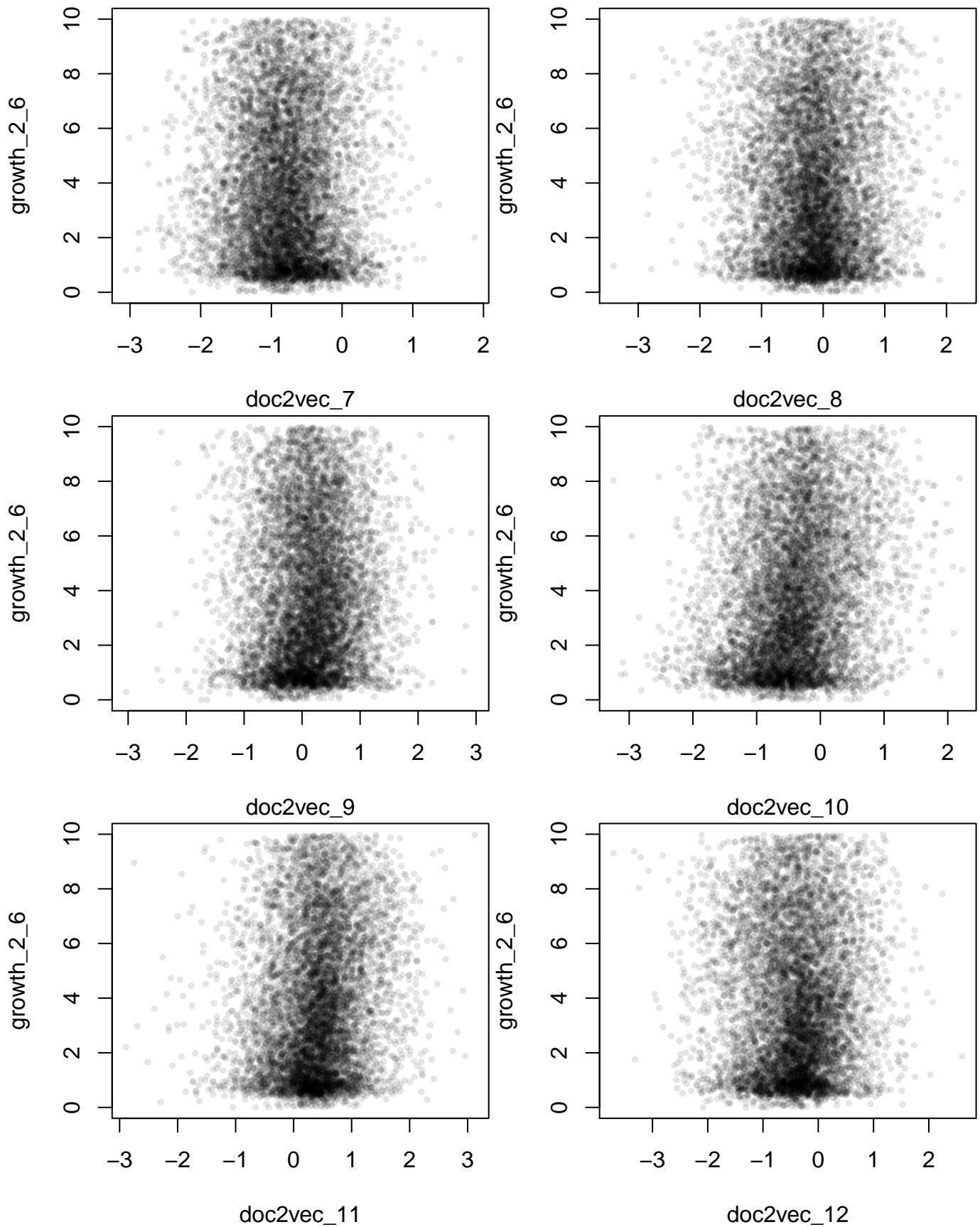


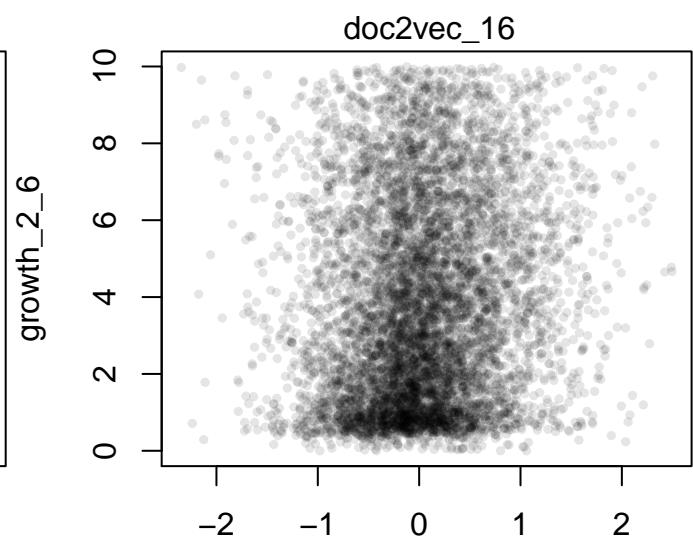
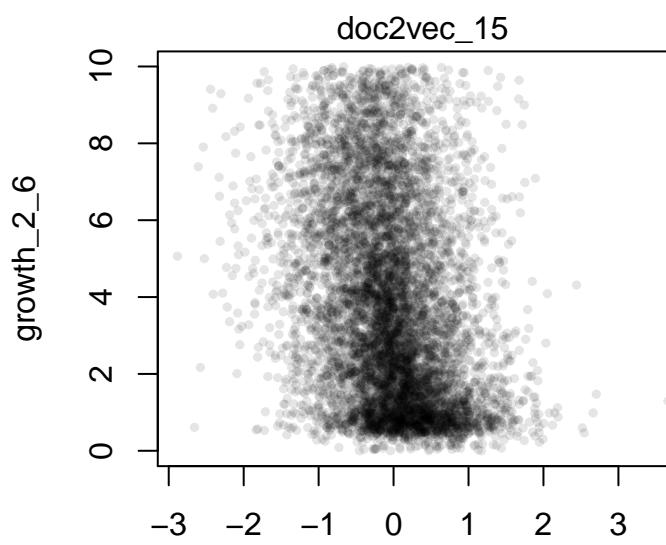
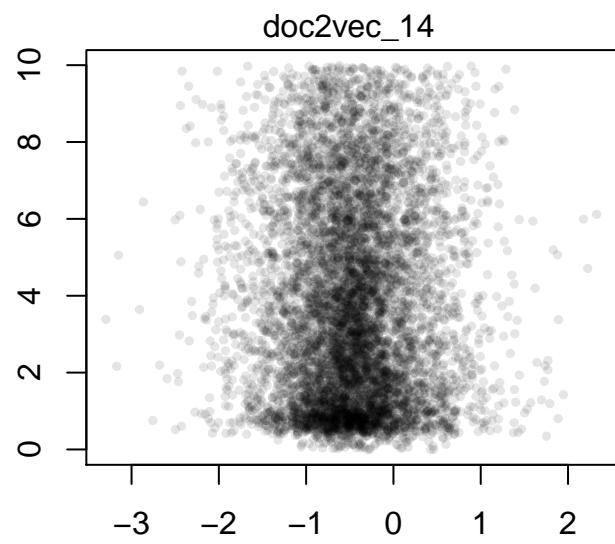
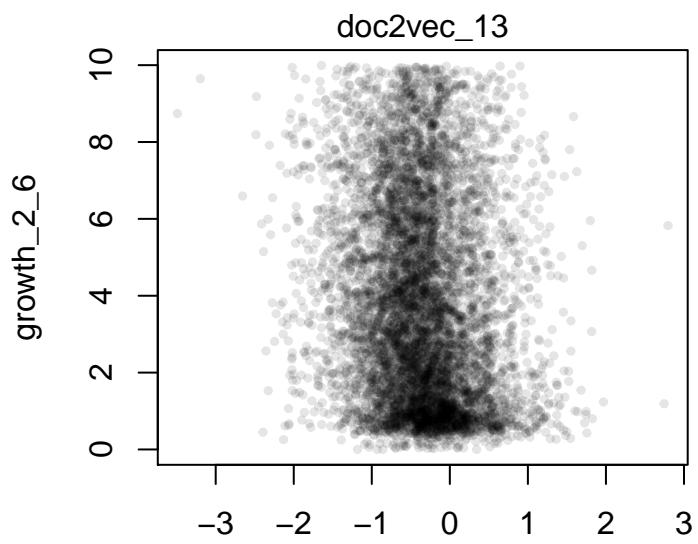
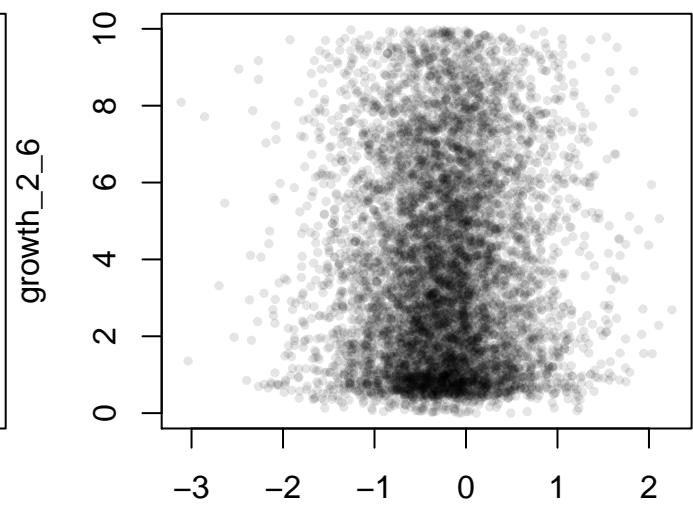
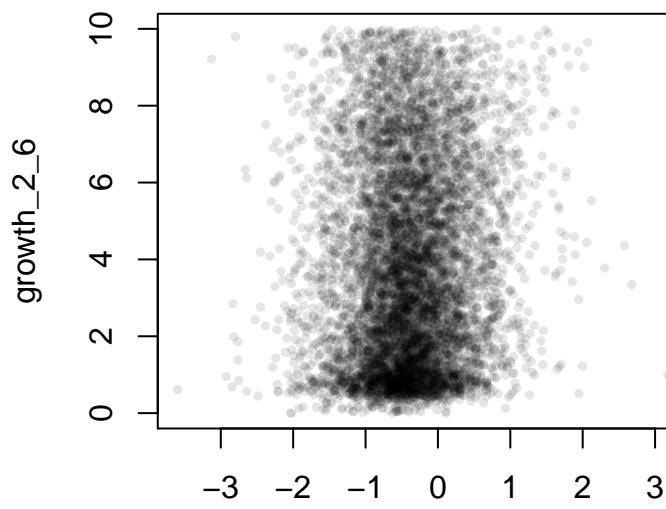




$doc2vec_5$

$doc2vec_6$





doc2vec_17

doc2vec_18

