# Normalization

```
suppressPackageStartupMessages({
  library(HIPCMatrix)
  library(Biobase)
  library(ImmuneSpaceR)
  library(vsn)
})
```

### Download original raw and normalized matrices

```
con <- CreateConnection("")
# illumina_raw <- con$getGEMatrix("SDY63_PBMC_Young_Geo", outputType = "raw")

affy_raw <- con$getGEMatrix("SDY80_PBMC_Cohort2_geo", outputType = "raw")
#> Reading local matrix
#> Downloading Features..
#> Constructing ExpressionSet
affy_norm <- con$getGEMatrix("SDY80_PBMC_Cohort2_geo", outputType = "norm")
#> Reading local matrix
#> Returning latest annotation from cache
#> Constructing ExpressionSet
# affy_raw_dt <- data.table(exprs(affy_raw))
# affy_raw_dt[, feature_id := rownames(exprs(affy_raw))]

illumina_raw <- con$getGEMatrix("SDY212_WholeBlood_Older_Geo", outputType = "raw")
#> Reading local matrix
#> Downloading Features..
#> Constructing ExpressionSet
illumina_norm <- con$getGEMatrix("SDY212_WholeBlood_Older_Geo", outputType = "norm")
#> Reading local matrix
#> Returning latest annotation from cache
#> Constructing ExpressionSet
# illumina_raw_dt <- data.table(exprs(illumina_raw))
# illumina_raw_dt[, feature_id := rownames(exprs(illumina_raw))]

rna_raw <- con$getGEMatrix("SDY1256_WholeBlood_EPIC001_geo", outputType = "raw")
#> Reading local matrix
#> Downloading Features..
#> Constructing ExpressionSet
rna_norm <- con$getGEMatrix("SDY1256_WholeBlood_EPIC001_geo", outputType = "norm")
#> Reading local matrix
#> Returning latest annotation from cache
#> Constructing ExpressionSet
# rna_raw_dt <- data.table(exprs(rna_raw))
# rna_raw_dt[, feature_id := rownames(exprs(rna1289_raw))]
```

## Changes in normalization methods

### RNA-seq

1. Update to `DESeq2` package
2. Use `DESeq2::vst` which internally does `estimateSizeFactors` and `estimateDispersions`.

Old normalization method for RNA-seq

```
library(DESeq)
# newCountDataSet does not take duplicated column names, so assign temporary unique names
orginal_colnames <- colnames(em)
colnames(em) <- seq_len(ncol(em))

cds <- newCountDataSet(countData = em, conditions = colnames(em))
cds <- estimateSizeFactors(cds)
cdsBlind <- estimateDispersions(cds, method = "blind" )
vsd <- varianceStabilizingTransformation(cdsBlind)
norm_exprs <- exprs(vsd)
colnames(norm_exprs) <- orginal_colnames
```

New normalization method for RNA-seq

```
normalize_rnaseq <- function(counts_mx,
                             verbose = FALSE) {

  if (verbose) message(" --- normalize_rnaseq --- ")
  if (verbose) message("Normalizing counts data using variance stabilizing transformation...")

  # newCountDataSet does not take duplicated column names, so assign temporary unique names
  original_colnames <- colnames(counts_mx)
  colnames(counts_mx) <- seq_len(ncol(counts_mx))

  dds <- DESeq2::DESeqDataSetFromMatrix(countData = counts_mx,
                                        colData = data.frame(sample = original_colnames),
                                        design = ~ 1)
  vsd <- DESeq2::vst(dds)

  norm_exprs <- SummarizedExperiment::assay(vsd)
  colnames(norm_exprs) <- original_colnames

  norm_exprs
}
```

### Microarray

1. Always perform log-2 transformation before quantile normalization
    1. Use smarter logic for when to perform log2 transformation, as Affymetrix data read in using RMA and two-color-array data are already in log-2 scale, and add messages and warnings to ensure that log-2 transformation is performed when it should (and not otherwise)
2. Perform log-2 transformation on `(exprs + 1)` instead of `pmax(exprs, 1)`

1. Do not floor matrix at 1.

Old normalization method for microarray

```
cnames <- colnames(em)
norm_exprs <- preprocessCore::normalize.quantiles(em)
colnames(norm_exprs) <- cnames
norm_exprs <- pmax(norm_exprs, 1)
if (max(norm_exprs) > 100) {
  norm_exprs <- log2(norm_exprs)
}
```

New normalization method for microarray

```
normalize_microarray <- function(exprs_mx,
                                 log2_transform = TRUE,
                                 force = FALSE,
                                 verbose = FALSE) {
  if (verbose) message(" --- normalize_microarray --- ")

  # normalize.quantiles removes row and column names
  cnames <- colnames(exprs_mx)
  rnames <- rownames(exprs_mx)

  # Do log2 transformation BEFORE normalization.
  if ( log2_transform ) {
    if ( max(exprs_mx) < 100 )
      if ( !force ) {
        stop("max(exprs_mx) < 100. ",
             "It is likely already in log2 scale. ",
             "Run with force=TRUE if you still want to log2 transform")
      } else if ( verbose ) {
        message("max(exprs_mx) < 100. Forcing log2 transform... ")
      }
    if (verbose) message("log2-transforming exprs_mx")
    exprs_mx <- log2(exprs_mx + 1)
  }
  if (verbose) message("Performing quantile normalization...")
  norm_exprs <- preprocessCore::normalize.quantiles(exprs_mx)

  colnames(norm_exprs) <- cnames
  rownames(norm_exprs) <- rnames

  norm_exprs
}
```

## Run new normalization method on raw matrix

```
affy_norm_new <- normalize_microarray(exprs(affy_raw),
                                      log2_transform = FALSE)
illumina_norm_new <- normalize_microarray(exprs(illumina_raw),
```

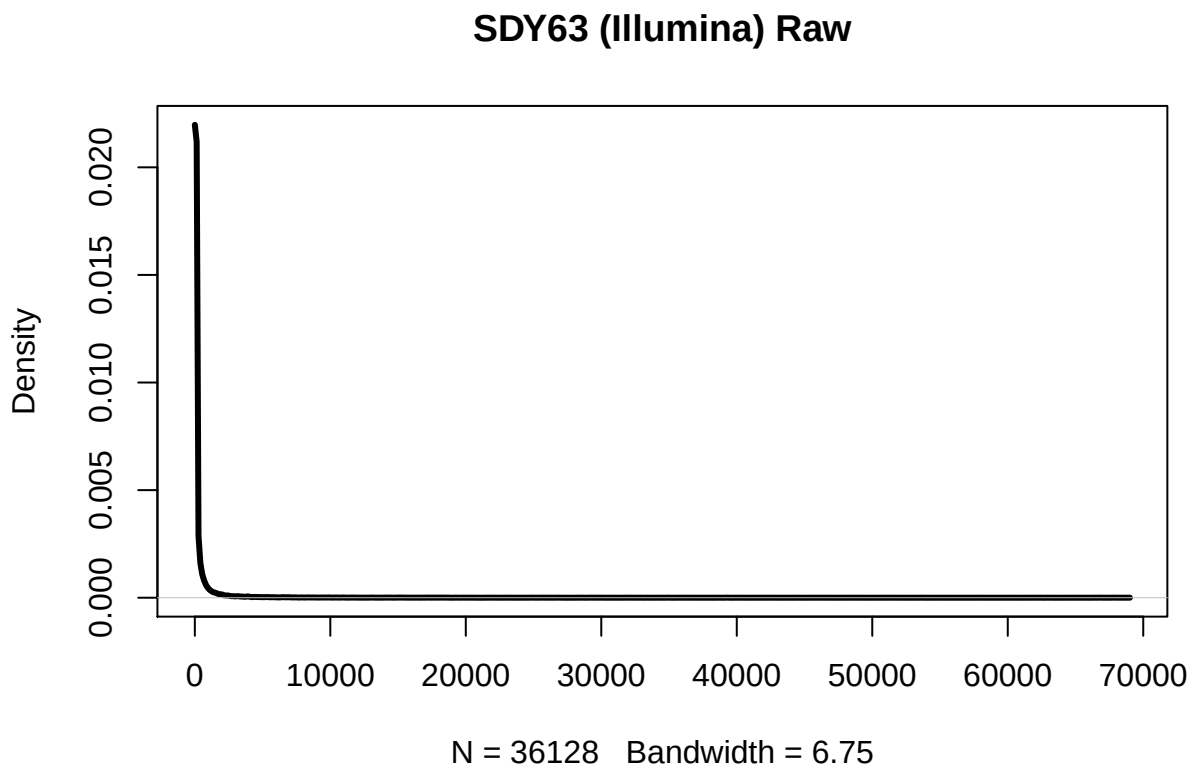
```
                                          log2_transform = TRUE)
rna_norm_new <- normalize_rnaseq(exprs(rna_raw))
```

## Explore datasets

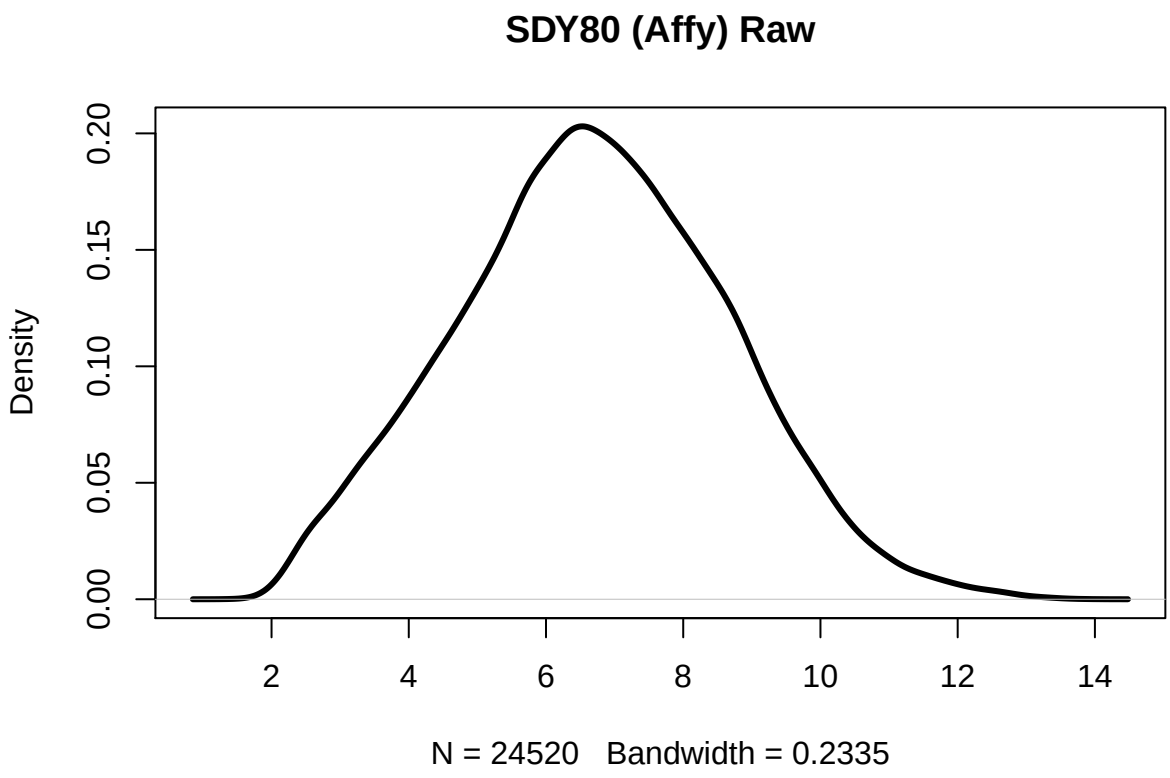
### Raw (non-normalized) data

Note that affy data is already in log-2 scale, as a result of the RMA function.

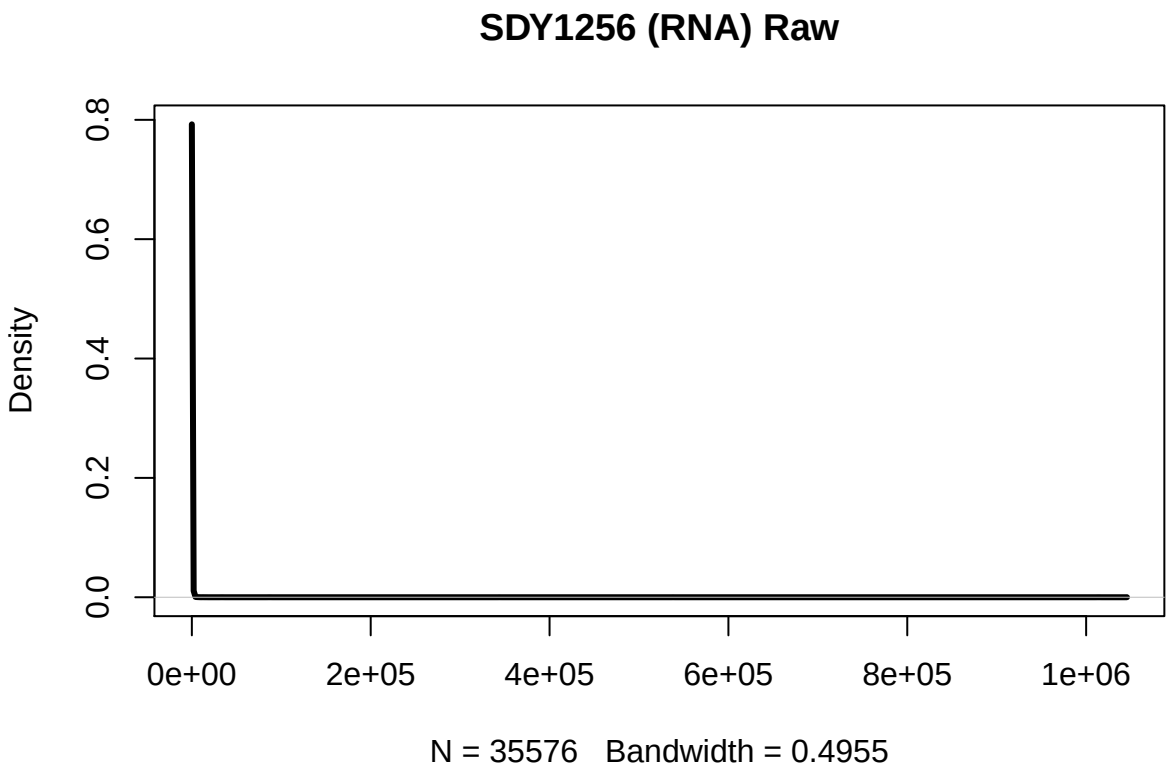
```
plot(density(exprs(illumina_raw)[,1]),
     lwd=3,
     main = "SDY63 (Illumina) Raw")
```



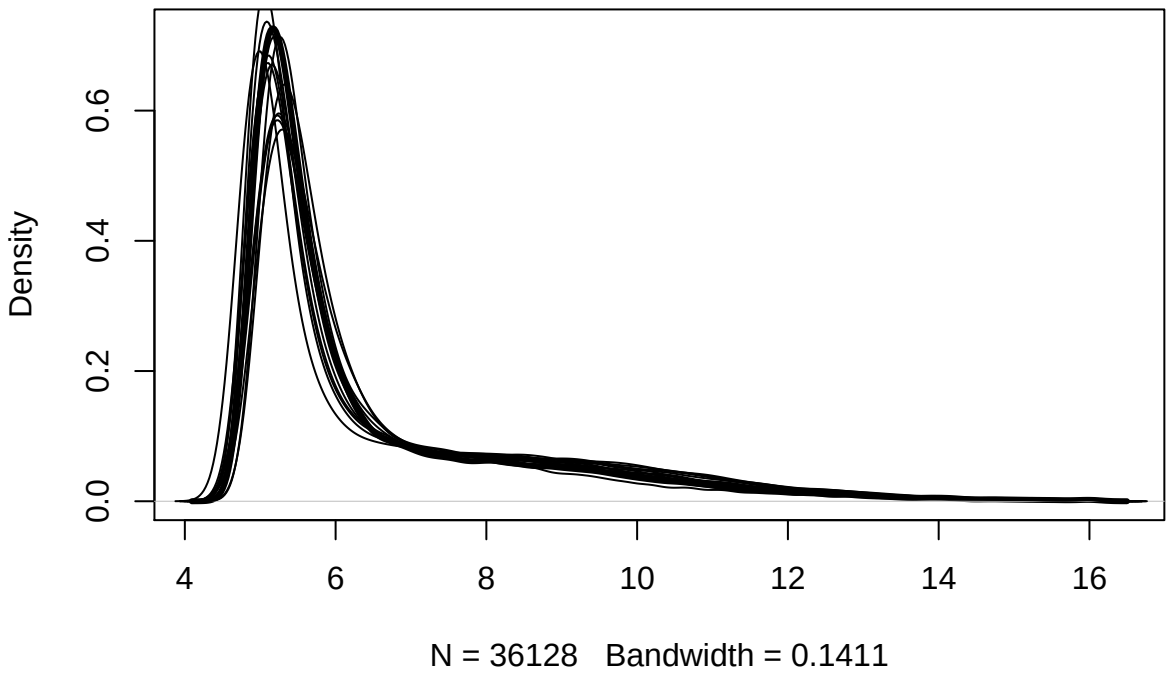

```
plot(density(exprs(affy_raw)[,1]),
     lwd=3,
     main = "SDY80 (Affy) Raw")
```



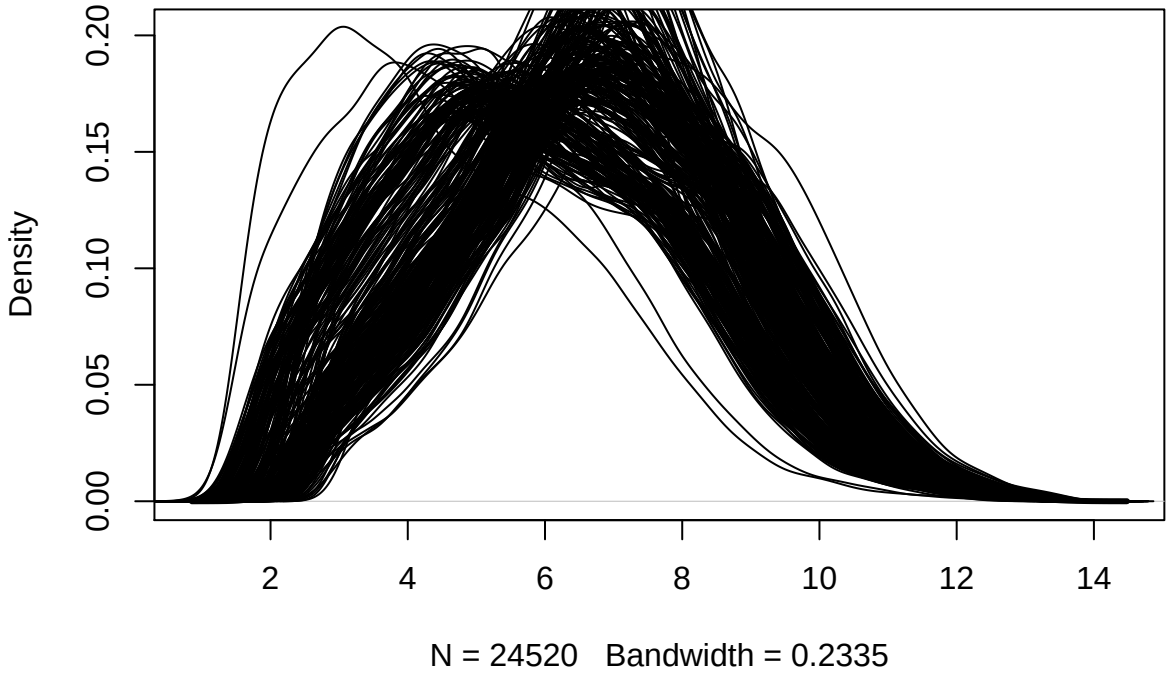

```
plot(density(exprs(rna_raw)[,1]),
     lwd=3,
     main = "SDY1256 (RNA) Raw")
```




```
# Plot all samples
plot(density(log2(exprs(illumina_raw) + 1)[,1]),
     lwd=3,
     main = "SDY63 (Illumina) Raw: log2")
apply(log2(exprs(illumina_raw) + 1), 2, function(x) lines(density(x)))
```

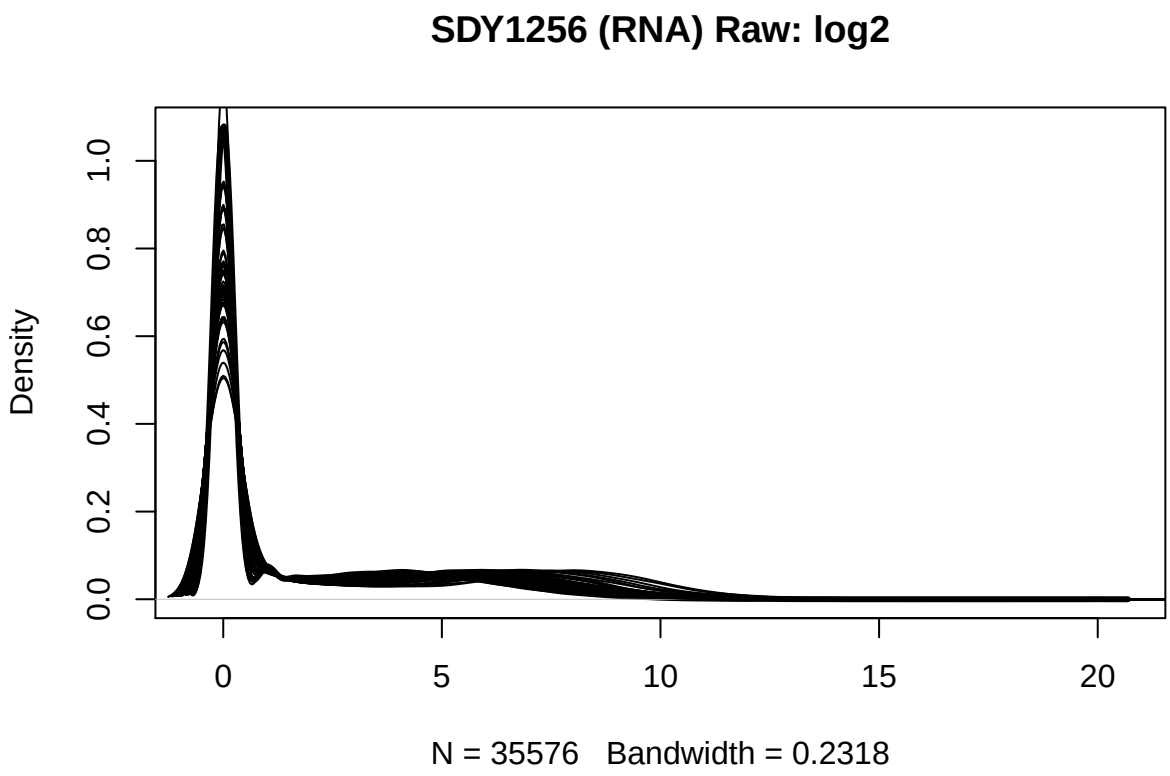



```
#> NULL

plot(density(exprs(affy_raw)[,1]),
     lwd=3,
     main = "SDY80 (Affy) Raw")
apply(exprs(affy_raw), 2, function(x) lines(density(x)))
```




```
#> NULL
```

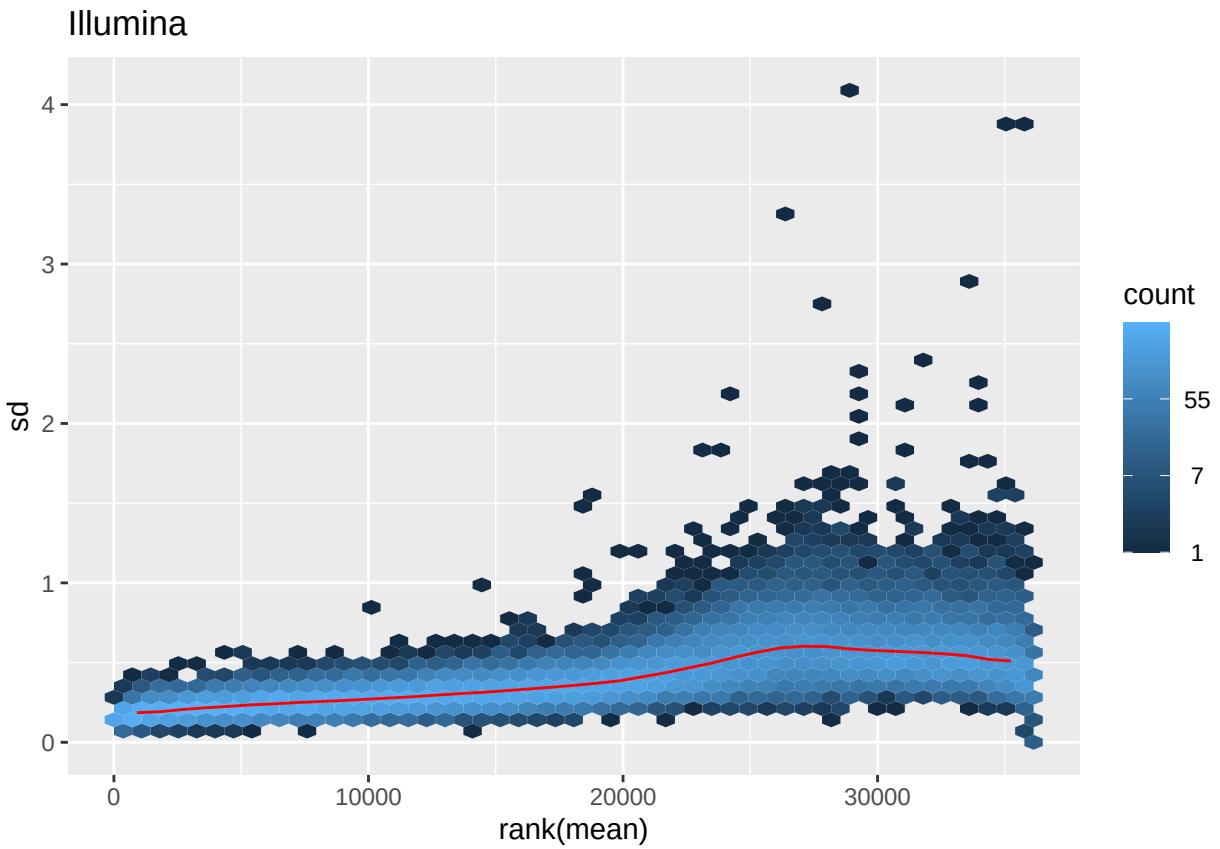
```
plot(density(log2(exprs(rna_raw)[,1] + 1)),
     lwd=3,
     main = "SDY1256 (RNA) Raw: log2")
apply(log2(exprs(rna_raw) + 1), 2, function(x) lines(density(x)))
```



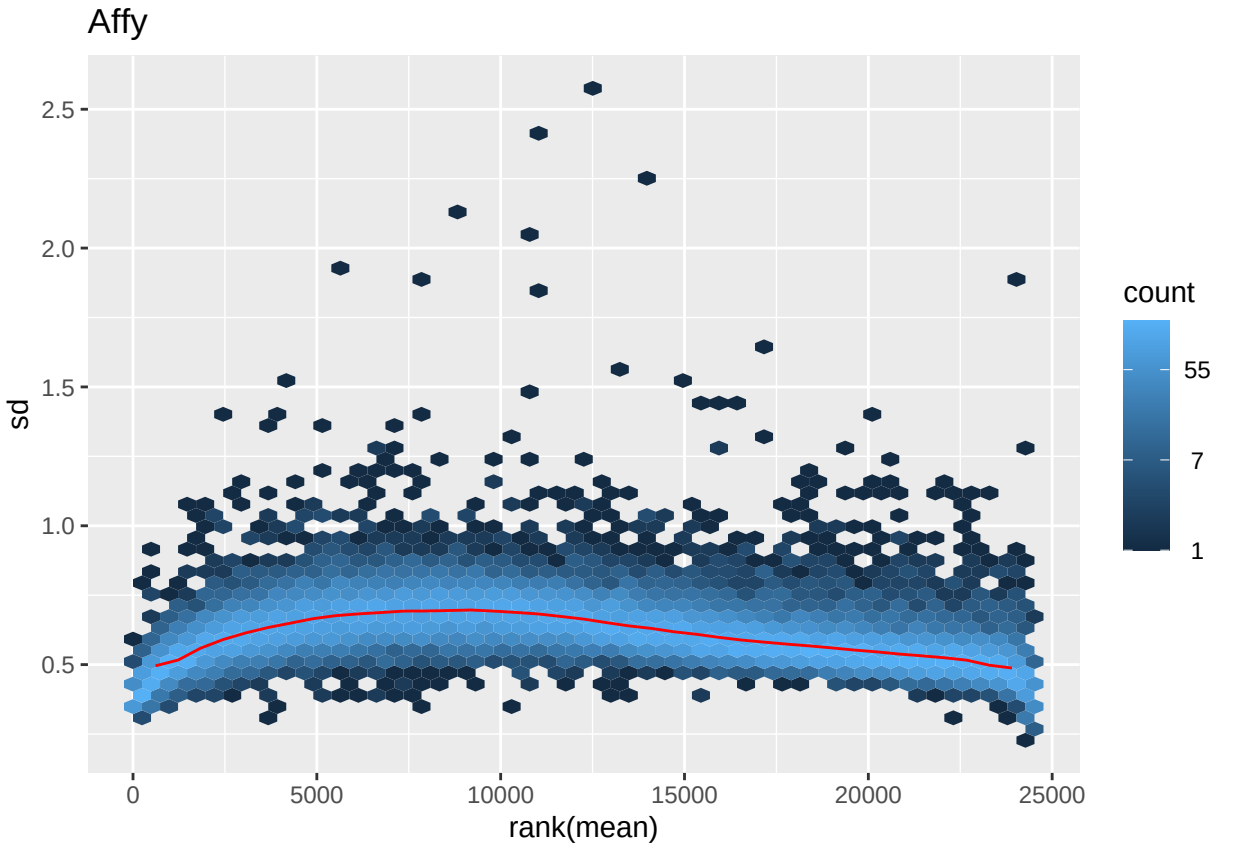

```
#> NULL
```

Plot mean vs sd by gene. Note that for rna-seq data, genes with higher means also generally have larger variance.

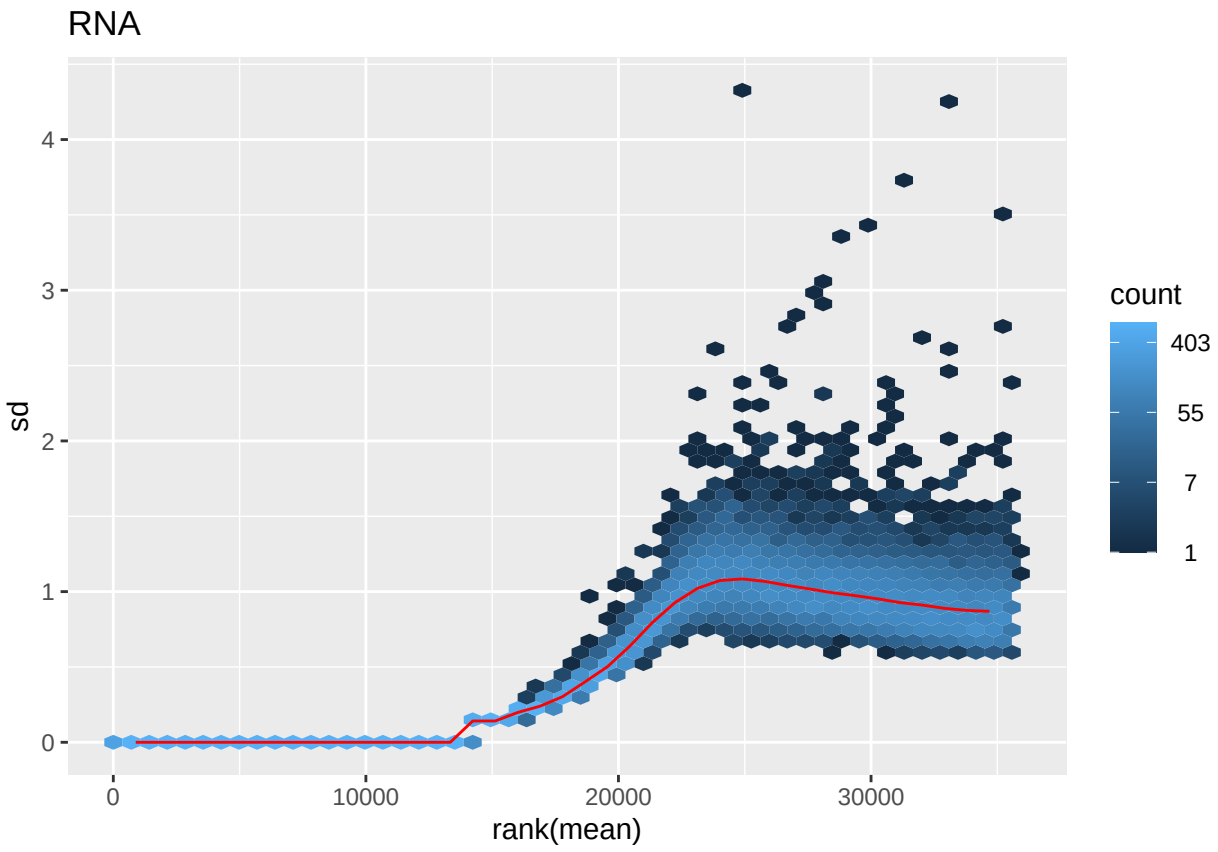
```
# Plot row standard deviations vs row means on log2-scaled data
# Note that for rna-seq data,
meanSdPlot(log2(exprs(illumina_raw) + 1), plot = FALSE)$gg + ggplot2::ggtitle("Illumina")
```




```
meanSdPlot(exprs(affy_raw), plot = FALSE)$gg + ggplot2::ggtitle("Affy")
```



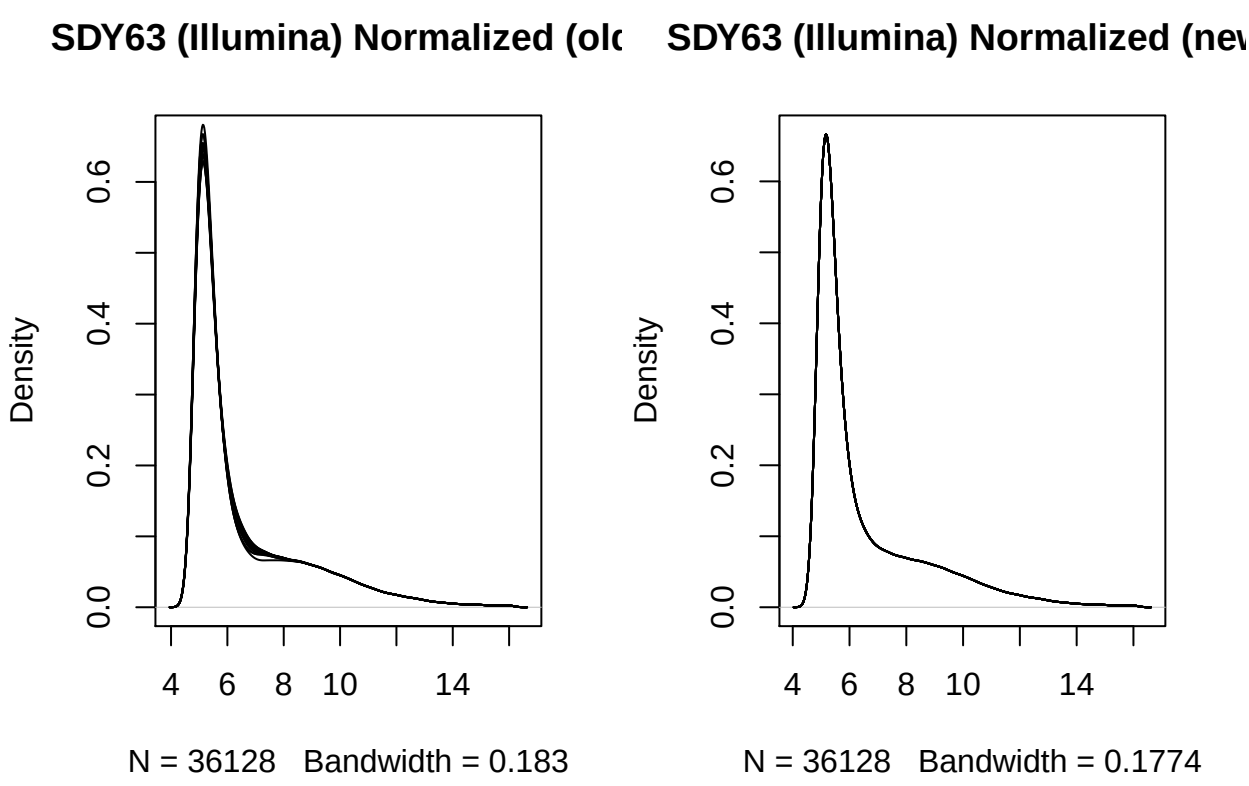

```
meanSdPlot(log2(exprs(rna_raw) + 1), plot = FALSE)$gg + ggplot2::ggtitle("RNA")
```



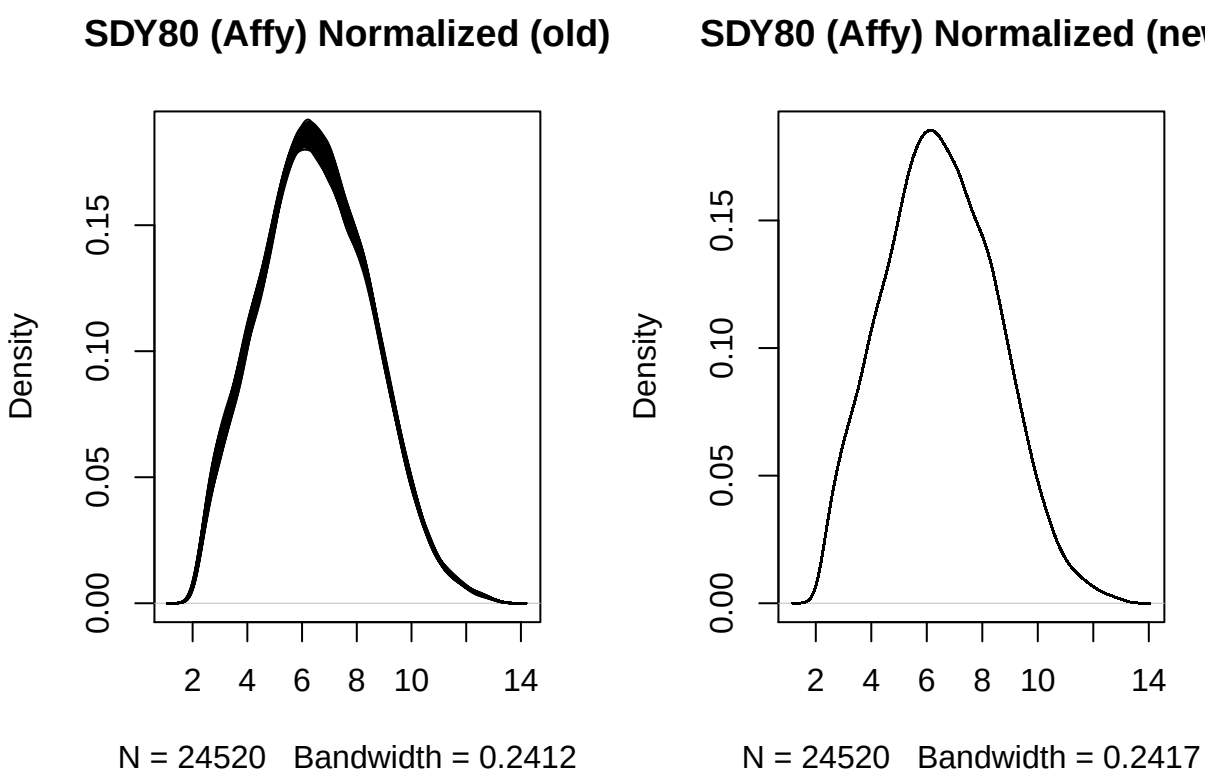

**Normalized data**

First, compare density plots for old vs new normalization

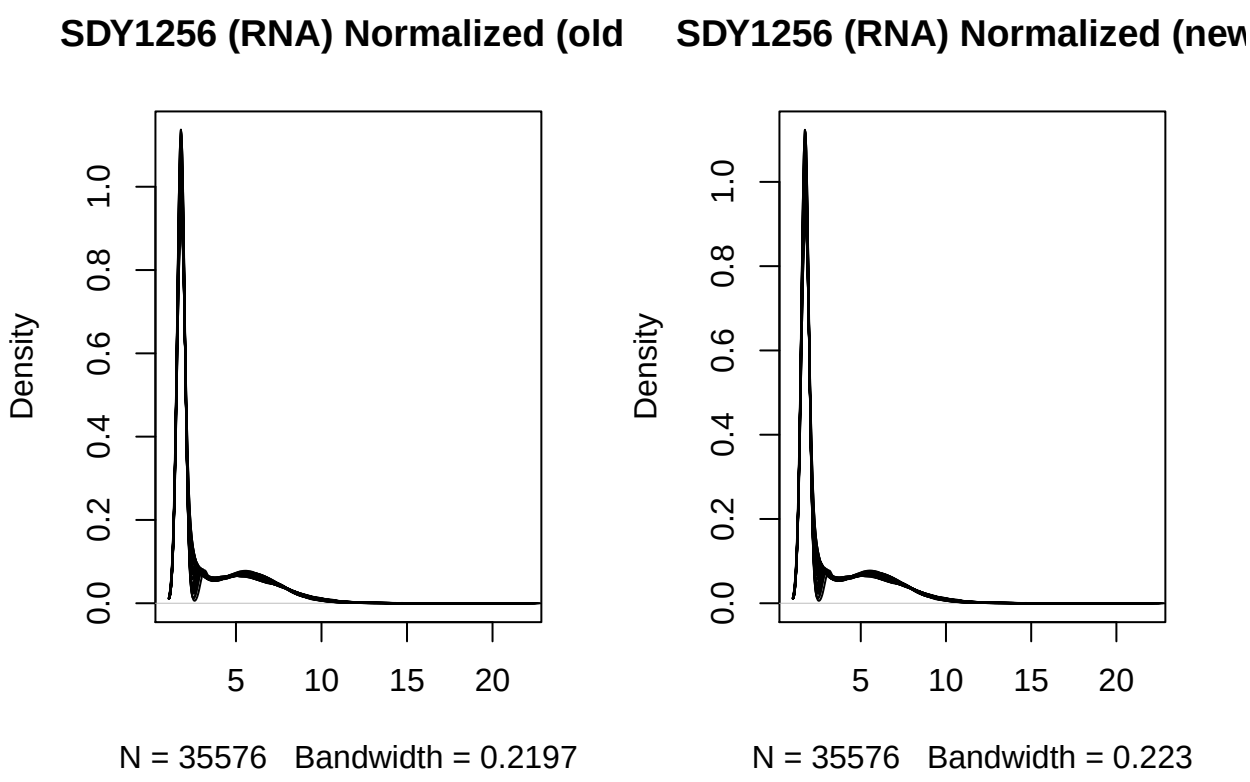
```
par(mfcol = c(1, 2))
plot(density(exprs(illumina_norm)[,1]),
     main = "SDY63 (Illumina) Normalized (old)")
apply(exprs(illumina_norm), 2, function(x) lines(density(x)))
#> NULL
plot(density(illumina_norm_new[,1]),
     main = "SDY63 (Illumina) Normalized (new)")
apply(illumina_norm_new, 2, function(x) lines(density(x)))
```




```
#> NULL

plot(density(exprs(affy_norm)[,1]),
     main = "SDY80 (Affy) Normalized (old)")
apply(exprs(affy_norm), 2, function(x) lines(density(x)))
#> NULL
plot(density(affy_norm_new[,1]),
     main = "SDY80 (Affy) Normalized (new)")
apply(affy_norm_new, 2, function(x) lines(density(x)))
```



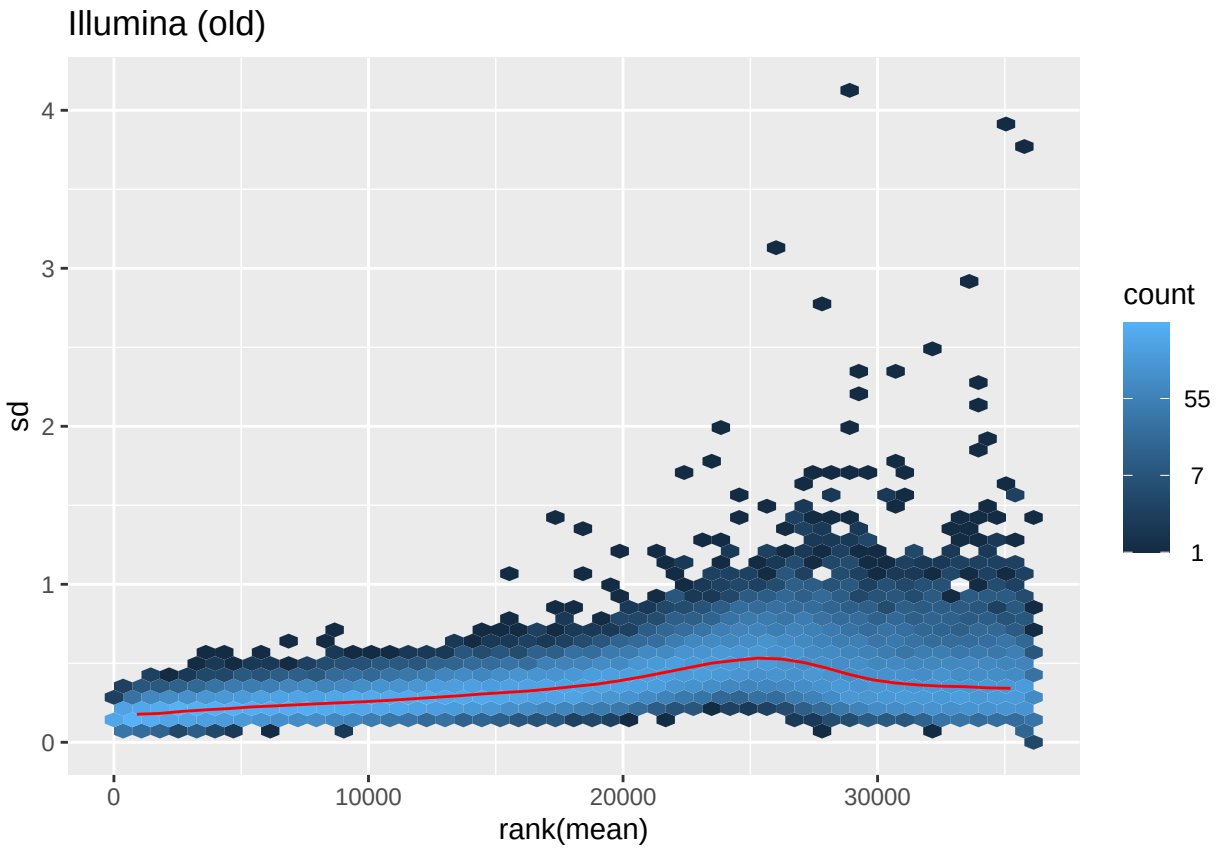

```
#> NULL

plot(density(exprs(rna_norm)[,1]),
     main = "SDY1256 (RNA) Normalized (old)")
apply(exprs(rna_norm), 2, function(x) lines(density(x)))
#> NULL
plot(density(rna_norm_new[,1]),
     main = "SDY1256 (RNA) Normalized (new)")
apply(rna_norm_new, 2, function(x) lines(density(x)))
```



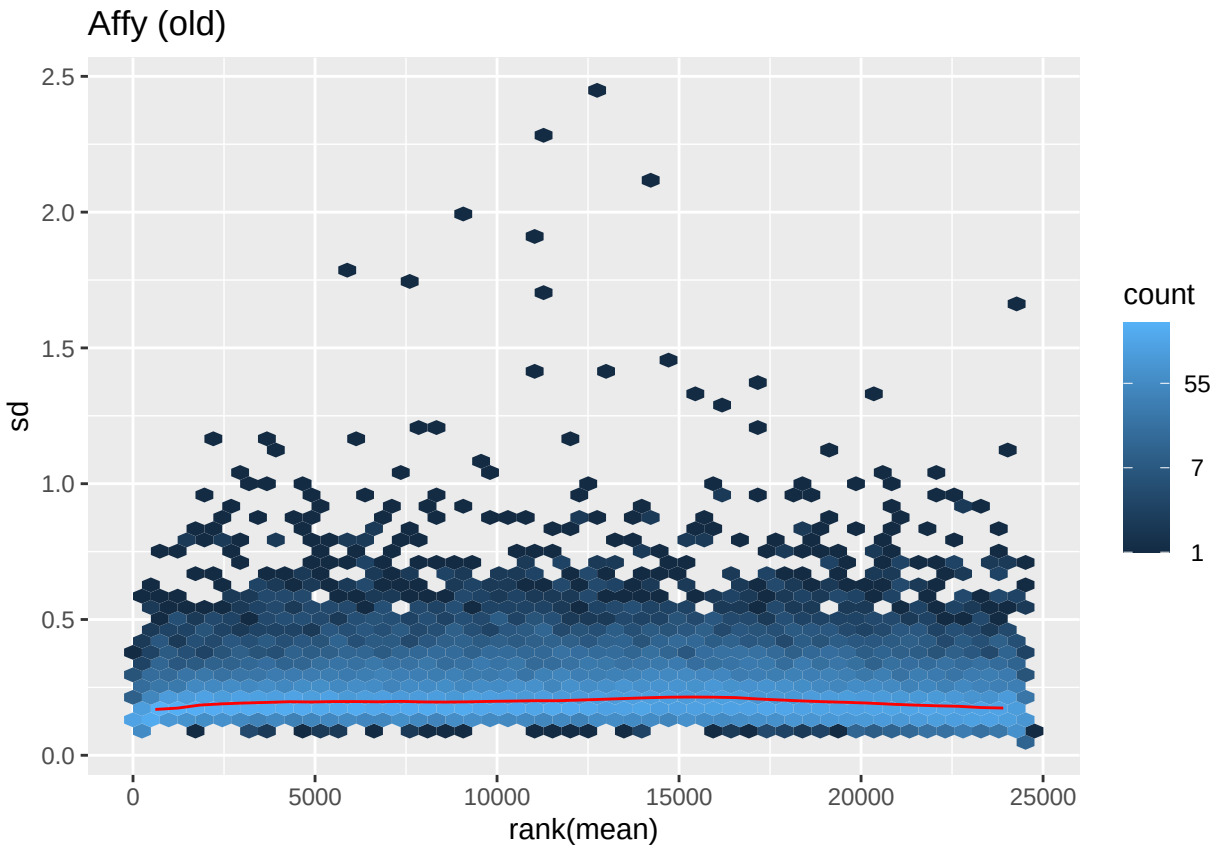

```
#> NULL
```

Mean vs sd plots for old and new normalized data

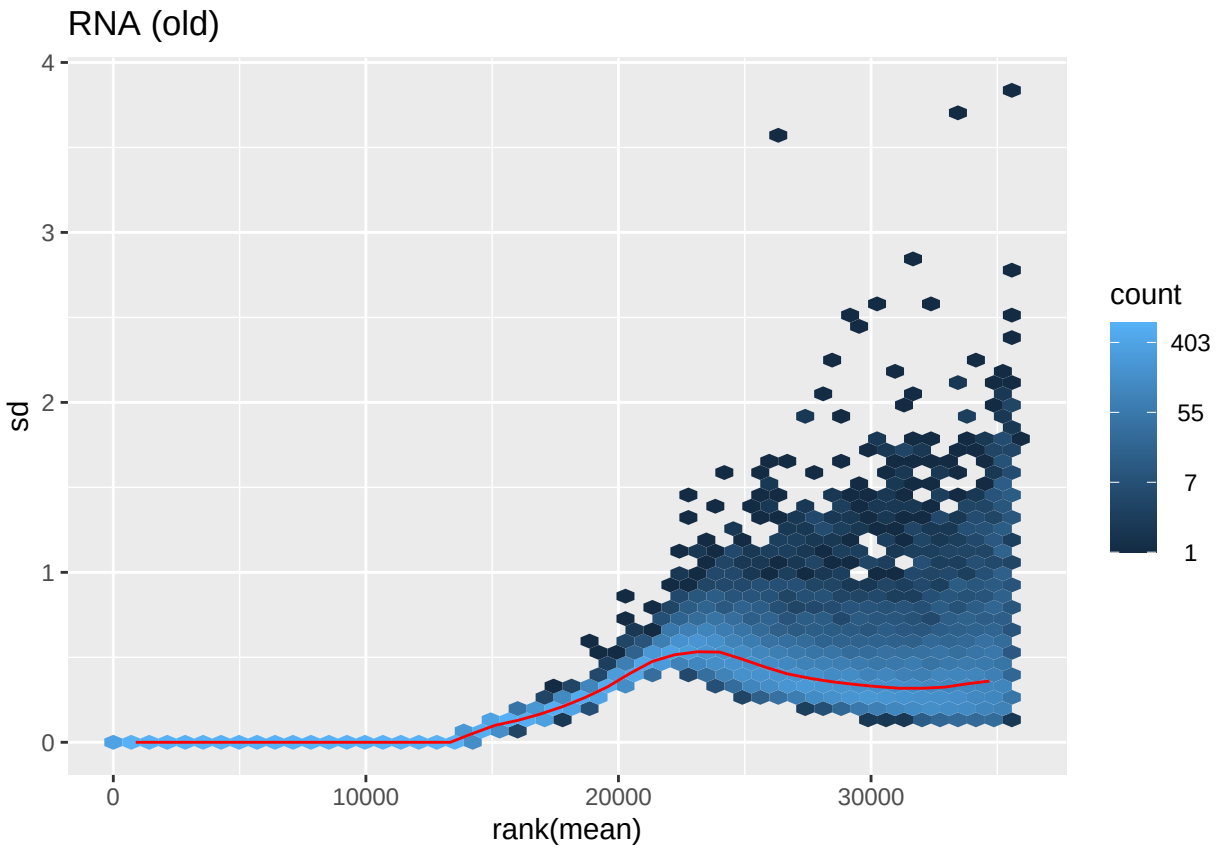
```
# Plot row standard deviations vs row means on log2-scaled data
# Note that for rna-seq data,
meanSdPlot(exprs(illumina_norm), plot = FALSE)$gg + ggplot2::ggtitle("Illumina (old)")
```



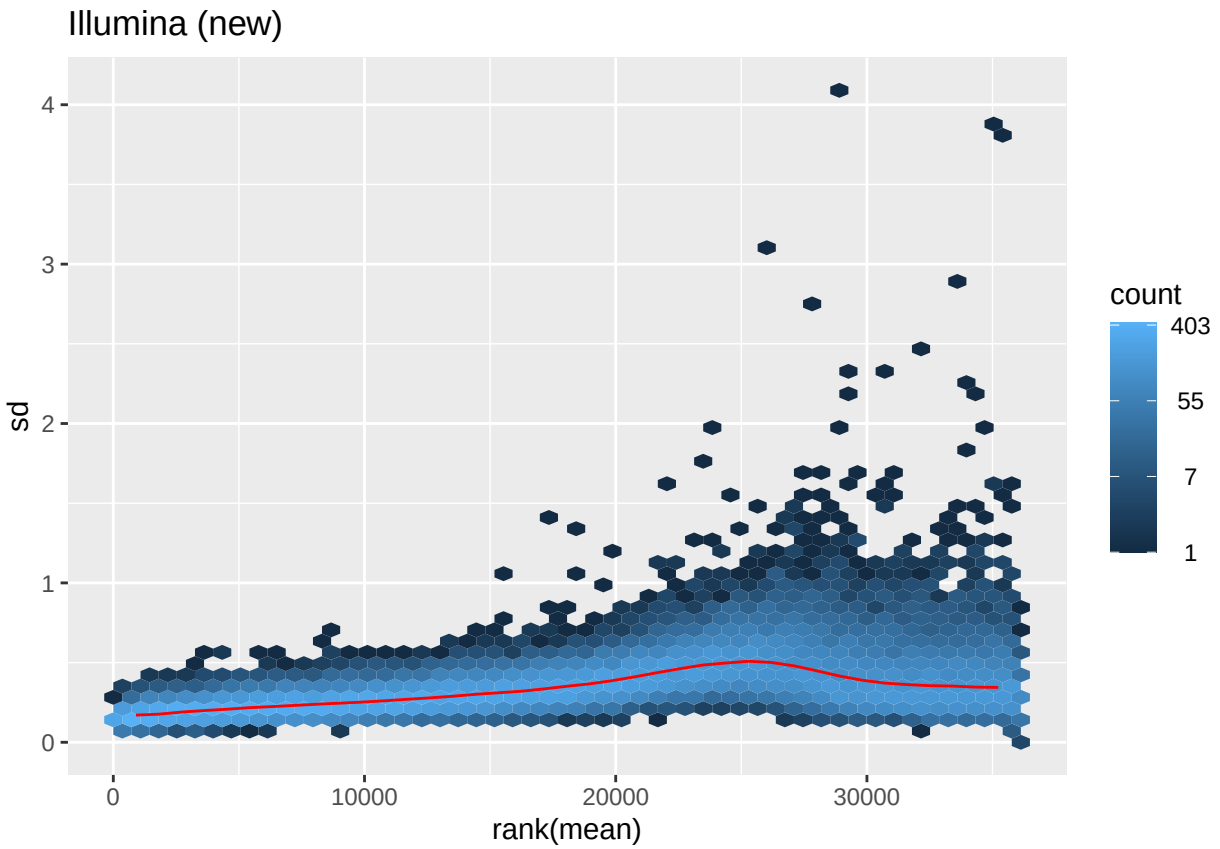

```
meanSdPlot(exprs(affy_norm), plot = FALSE)$gg + ggplot2::ggtitle("Affy (old)")
```



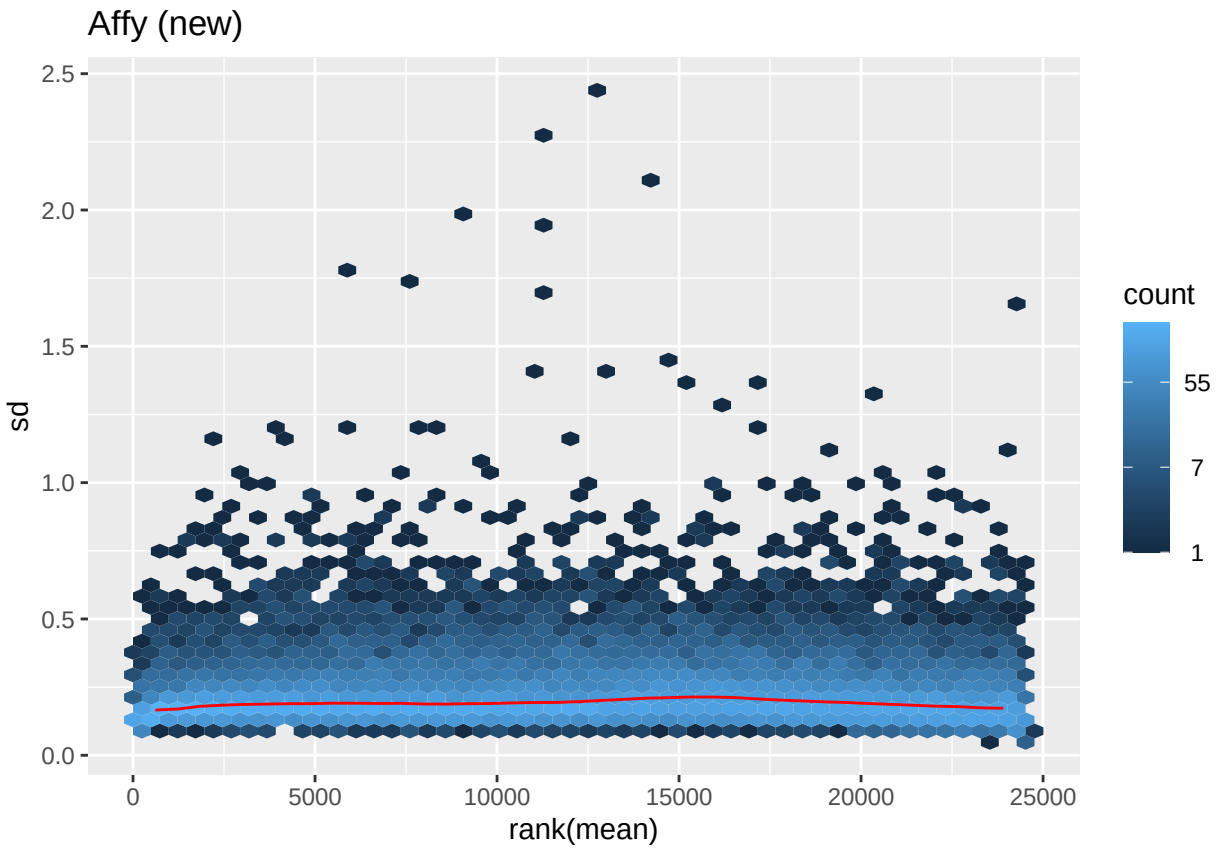

```
meanSdPlot(exprs(rna_norm), plot = FALSE)$gg + ggplot2::ggtitle("RNA (old)")
```



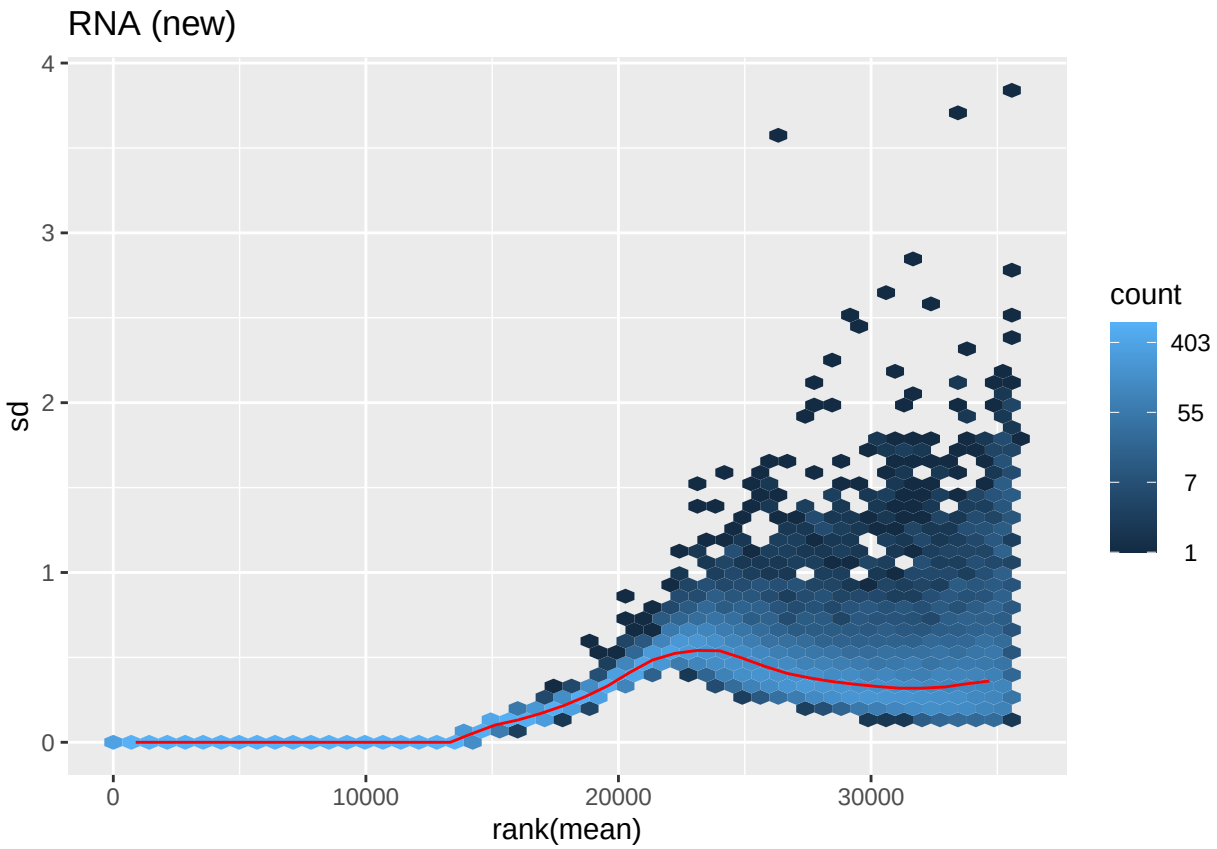

```
meanSdPlot(illumina_norm_new, plot = FALSE)$gg + ggplot2::ggtitle("Illumina (new)")
```




```
meanSdPlot(affy_norm_new, plot = FALSE)$gg + ggplot2::ggtitle("Affy (new)")
```




```
meanSdPlot(rna_norm_new, plot = FALSE)$gg + ggplot2::ggtitle("RNA (new)")
```


RNA (new)
sd
rank(mean)
0
1
2
3
4
0
10000
20000
30000
count
403
55
7
1