DNA Hydroxymethylation in Hepatocellular Carcinoma

Francois Collin

2020-08-30

Contents

Preamble 5

License 5

1 Introduction 7

2 Preprocessing 9

2.1 Load the data 9

2.2 Differential representation analysis 13

2.3 UP TO HERE 19

2.4 Analysis of coverage variability 19

3 Baseline Model 25

4 Fitted Model Suite 27

5 Conclusions 29

```
  # file rmarkdown file management options: cache, figures
figures_DIR <- file.path('Static', 'figures/')
suppressMessages(dir.create(figures_DIR, recursive=T))
```

```
## Warning in dir.create(figures_DIR, recursive = T): 'Static/figures' already
## exists
```

```
knitr::opts_chunk$set(fig.path=paste0(figures_DIR))
```

Preamble

This vignette offers some exploratory data analyses of data available from the NCBI GEO web site.

License

Chapter 1

Introduction

The goal of detecting cancer at the earliest stage of development with a non-invasive procedure has busied many groups with the task of perfecting techniques to support what has become commonly known as a liquid biopsy - the analysis of biomarkers circulating in fluids such as blood, saliva or urine. Epigenetic biomarkers present themselves as good candidates for this application (Gai and Sun (2019) [1]). In particular, given their prevalence in the human genome, close correlation with gene expression and high chemical stability, DNA modifications such as 5-methylcytosine (5mC) and 5-hydroxymethylcytosine (5hmC) are DNA epigenetic marks that provide much promise as cancer diagnosis biomarkers that could be profitably analyzed in liquid biopsies [2–5].

Li et al. (2017) [3] used a sensitive and selective chemical labeling technology to extract genome-wide 5hmC profiles from circulating cell-free DNA (cfDNA) as well as from genomic DNA (gDNA) collected from a cohort of 260 patients recently diagnosed with colorectal, gastric, pancreatic, liver or thyroid cancer and normal tissues from 90 healthy individuals They found 5hmC-based biomarkers of circulating cfDNA to be highly predictive of some cancer types. Similar small sample size findings were reported in Song et al. (2017) [4].

Focusing on hepatocellular carcinoma, Cai et al. (2019) [2] assembled a sizable dataset to demonstrate the feasibility of using features derived from 5-hydroxymethylcytosines marks in circulating cell-free DNA as a non-invasive approach for the early detection of hepatocellular carcinoma. The data that are the basis of that report are available on the NCBI GEO web site (Series GSE112679). The data have also been bundled in a R data package which can be installed from github:

```
if (!requireNamespace("devtools", quietly = TRUE))
    install.packages("devtools")
devtools::install_github("12379Monty/GSE112679")
```

An important question in the early development of classifiers of the sorts that are the basis of any liquid biopsy diagnostic tool is how many samples should be collected to make properly informed decisions. In this report we will explore the GSE112679 data to shed some light on the relationship between sample size and model performance in the context classifying samples based on 5hmC data.

In Section 2 we preprocess the data that we will use for the classification analysis and perform some light QC analyses. In Section 3 we fit some models to discriminate between early stage HCC and control samples and examine their performance. In Section 4 we examine the results of fitting a suite of models to investigate the effect of sample size on model performance.

Chapter 2

Preprocessing

2.1 Load the data

The data that are available from NCBI GEO Series GSE112679 can be conveniently accessed through an R data package. Attaching the GSE112679 package makes the count data tables available as well as a gene annotation table and a sample description table. See GSE112679 R Data Package page. For the Cai et al. [2] model fitting and analysis, samples were separated into `Train` and `Val-1` subsets. `Val-2` was an external validation set.

```r
if (!("GSE112679" %in% rownames(installed.packages()))) {
  if (!requireNamespace("devtools", quietly = TRUE)) {
    install.packages("devtools")
  }
  devtools::install_github("12379Monty/GSE112679")
}
library(GSE112679)
sampDesc$DxStage <- with(sampDesc, ifelse(outcome=='HCC',
   paste0(outcome,':',stage), outcome))

with(
  sampDesc %>% dplyr::filter(sampType == "blood"),
  table(DxStage, trainValGroup, exclude = NULL)
)
```

```
##             trainValGroup
## DxStage      Train Val-1 Val-2
##   Benign       253   132     3
##   CHB          190    96     0
##   Cirrhosis    73    33     0
##   HCC:Early   335   220    24
##   HCC:Late      0   442    13
##   HCC:NA        0   147    23
##   Healthy     269   124   177
```

For this analysis, we will consider early stage cancer samples and healthy or benign samples from the `Train` or `Val-1` subsets. The appropriate outcome variable will be renamed or aliased `group`

```
sampDescA <-
  sampDesc %>%
  dplyr::filter(sampType == "blood" &
                (trainValGroup %in% c("Train", "Val-1")) &
    ((outcome2 == "BenignHealthy") |
     (outcome2 == "HCC" & stage == "Early"))) %>%
  dplyr::rename(group = outcome2) %>%
  dplyr::arrange(group, sampID)
# Recode group
sampDescA$group <- with(sampDescA,
   ifelse(group == "BenignHealthy", "Control", group))
# set groupCol for later
groupCol <- c("#F3C300", "#875692")
names(groupCol) <- unique(sampDescA$group)

with(sampDescA, table(group, exclude = NULL))
```

```
## group
## Control     HCC
##     778     555
```

The features are counts of reads captured by chemical labeling, and indicate the level of 5-hydroxymethylcytosines within each gene body. Cai et al. (2019), Li et al. (2017) and Song et al. (2017) [2–4] all analyze 5hmC gene body counts using standard RNA-Seq methodologies, and we will do the same here.

Note that before conducting any substantive analyses, the data would normally be very carefully examined for any sign of quality variation between groups of samples. This analysis would integrate sample meta data - where and when were the blood samples collected - as well as library preparation and sequencing metrics in order to detect any sign of processing artifacts that may be present in the dataset. This is particularly important when dealing with blood samples as variable DNA quality degradation is a well known challenge that is encountered when dealing with such samples [6]. Although blood specimen handling protocols can be put in place to minimize quality variation [7], variability can never be completely eradicated, especially in the context of blood samples collected by different groups, working in different environments. The problem of variable DNA quality becomes paricularly pernicuous when it is compounded with a confounding factor that sneaks in when the control sample collection events are separated in time and space from the cancer sample collection events; an all too common occurence.

As proper data QC requires an intimate familiarity with the details of data collection and processing, such a task cannot be untertaken here. We will simply run a *minimal set of QC sanity checks* to make sure that there are no apparent systematic effects in the data.

```
  featureCountsA <- cbind(Train_featureCount,
                          Val1_featureCount,
                          Val2_featureCount)[,rownames(sampDescA)]
```

We first look at coverage - make sure there isn't too much disparity of coverage across samples. To detect shared variability, samples can be annotated and ordered according to sample features that may be linked to sample batch processing. Here we the samples have been ordered by group and sample id (an alias of geoAcc).

```
par(mar = c(1, 3, 2, 1))
boxplot(log2(featureCountsA + 1),
  ylim = c(3, 11), ylab='log2 Count',
  staplewex = 0,       # remove horizontal whisker lines
  staplecol = "white", # just to be totally sure :)
  outline = F,         # remove outlying points
  whisklty = 0,        # remove vertical whisker lines
  las = 2, horizontal = F, xaxt = "n",
  border = groupCol[sampDescA$group]
)
legend("top", legend = names(groupCol), text.col = groupCol,
  ncol = 2, bty = "n")
# Add reference lines
SampleMedian <- apply(log2(featureCountsA + 1), 2, median)
abline(h = median(SampleMedian), col = "grey")
axis(side = 4, at = round(median(SampleMedian), 1),
  las = 2, col = "grey", line = -1, tick = F)
```

Coverage level looks fairly comparable across samples. It is sometimes helpful to keep track of the actual coverage which can be adequetely tracked by distribution quantiles.

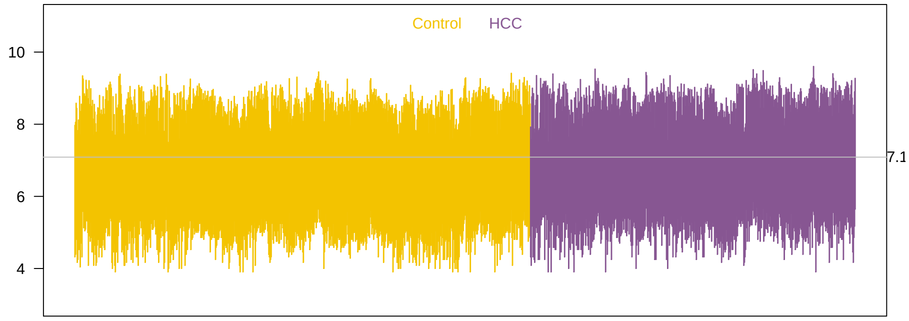

Figure 2.1: Sample log2 count boxplots

Table 2.1: Coverage Summary - Columns are sample coverage quantiles and total coverage Rows are quartiles across samples

	10%	25%	50%	75%	totCovM
25%	0	24	111	321	5.5
50%	0	30	135	391	6.7
75%	0	35	162	468	8.0

```
featureCountsA_quant <- apply(featureCountsA, 2, function(CC)
 c(quantile(CC, prob=c(.1,(1:3)/4)), totCovM=sum(CC)/1e6))

featureCountsA_quant2 <- apply(featureCountsA_quant, 1, function(RR)
  quantile(RR, prob=(1:3)/4))

knitr::kable(featureCountsA_quant2, digits=1,
caption=paste("Coverage Summary - Columns are sample coverage quantiles and total cove
"\nRows are quartiles across samples"))
```

From this table, we see that 25% of the samples have total coverage exceeding 8, 25% of samples have a 10 percentile of coverage lower than 0, etc.

We next look at relative log representation (RLR) (in the context of measuring the density of 5hmC marks in genes, we refer to `representation` as opposed to `expression`; the two can be used interchangibly) - make sure the shapes of the distributions are not widely different.

```
lcpm_mtx <- edgeR::cpm(featureCountsA, log = T)
median_vec <- apply(lcpm_mtx, 1, median)
RLR_mtx <- sweep(lcpm_mtx, 1, median_vec, "-")
```

```
par(mar = c(1, 3, 2, 1))
boxplot(RLR_mtx,
  xlab = "", ylab='Relative Log Representation', ylim = c(-.6, .6),
  staplewex = 0, # remove horizontal whisker lines
  staplecol = "white", # just to be totally sure :)
  outline = F, # remove outlying points
  whisklty = 0, # remove vertical whisker lines
  las = 2, horizontal = F, xaxt = "n",
  border = groupCol[sampDescA$group]
)
legend("top", legend = names(groupCol),
  text.col = groupCol, ncol = 2, bty = "n")
# Add group Q1, Q3
for (GRP in unique(sampDescA$group)) {
  group_ndx <- which(sampDescA$group == GRP)
  group_Q1Q3_mtx <- apply(RLR_mtx[, group_ndx], 2,
     quantile, prob = c(.25, .75))
  abline(h = apply(group_Q1Q3_mtx, 1, median),
     col = groupCol[GRP], lwd = 2)
}
```

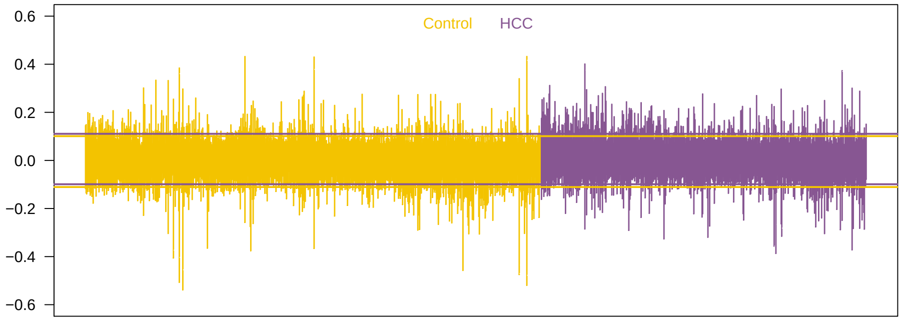

Figure 2.2: Sample RLR

We note that the HCC samples have slightly more variable coverage distribution. A few samples are quite different.

2.2 Differential representation analysis

- word on GC content

In the remainder of this section, we will process the data and perform differential expression analysis as outlined in Law et al. (2018) [8]. The main analysis steps are:

- remove lowly expressed genes
- normalize gene expression distributions
- remove heteroscedascity
- fit linear models and examine DE results

It is good practice to perform this differential expression analysis prior to fitting models to get an idea of how difficult it will be to discriminate between samples belonging to the different subgroups. The pipeline outlined in Law et al. (2018) [8] also provides some basic quality assessment opportunities.

Remove lowly expressed genes

Genes that are not expressed at a biologically meaningful level in any condition should be discarded to reduce the subset of genes to those that are of interest, and to reduce the number of tests carried out downstream when looking at differential expression. Carrying un-informative genes may also be a hindrance to classification and other downtream analyses.

To determine a sensible threshold we can begin by examining the shapes of the distributions.

```r
par(mar = c(4, 3, 2, 1))
plot(density(lcpm_mtx[, 1]),
  col = groupCol[sampDescA$group[1]],
  lwd = 2, ylim = c(0, .25), las = 2, main = "", xlab = "log2 CPM"
)
abline(v = 0, col = 3)
# After verifying no outliers, can plot a random subset
for (JJ in sample(2:ncol(lcpm_mtx), size = 100)) {
  den <- density(lcpm_mtx[, JJ])
  lines(den$x, den$y, col = groupCol[sampDescA$group[JJ]], lwd = 2)
} # for(JJ
legend("topright", legend = names(groupCol),
  text.col = groupCol, bty = "n")
```

As is typically the case with RNA-Seq data, we notice many weakly represented genes in this dataset. A cpm value of 1 appears to adequatly separate the expressed from the un-expressed genes, but we will be slightly more strict here and require a CPM threshold of 3 . Using a nominal CPM value of 3, genes are deeemed to be `represented` if their expression is above this threshold,

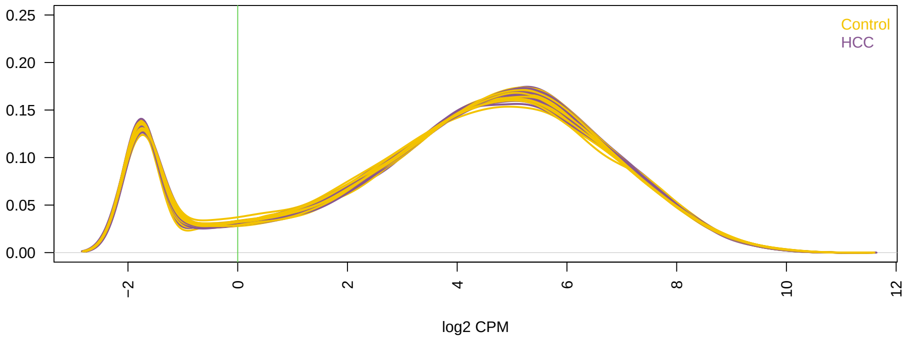

Figure 2.3: Sample log_2 CPM densities

and not represented otherwise. For this analysis we will require that genes be `represented` in at least 25 samples across the entire dataset to be retained for downstream analysis. Here, a CPM value of 3 means that a gene is represented if it has at least 9 reads in the sample with the lowest sequencing depth (library size 2.9 million). Note that the thresholds used here are arbitrary as there are no hard and fast rules to set these by. The voom-plot, which is part of analyses done to remove heteroscedasticity, can be examined to verify that the filtering performed is adequate.

Remove weakly represented genes and replot densities.

Removing 17.5% of genes…

```
featureCountsA <- featureCountsA[!weak_flg, ]
genes_annotA <- genes_annot[rownames(featureCountsA), ]
lcpm_mtx <- edgeR::cpm(featureCountsA, log = T)
dim(lcpm_mtx)
```

```
## [1] 15752  1333
```

```
par(mar = c(4, 3, 2, 1))
plot(density(lcpm_mtx[, 1]),
  col = groupCol[sampDescA$group[1]],
  lwd = 2, ylim = c(0, .25), las = 2, main = "", xlab = "log2 CPM"
)
#abline(v = 0, col = 3)
# After verifying no outliers, can plot a random subset
for (JJ in sample(2:ncol(lcpm_mtx), size = 100)) {
  den <- density(lcpm_mtx[, JJ])
  lines(den$x, den$y, col = groupCol[sampDescA$group[JJ]], lwd = 2)
} # for(JJ
legend("topright", legend = names(groupCol),
  text.col = groupCol, bty = "n")
```

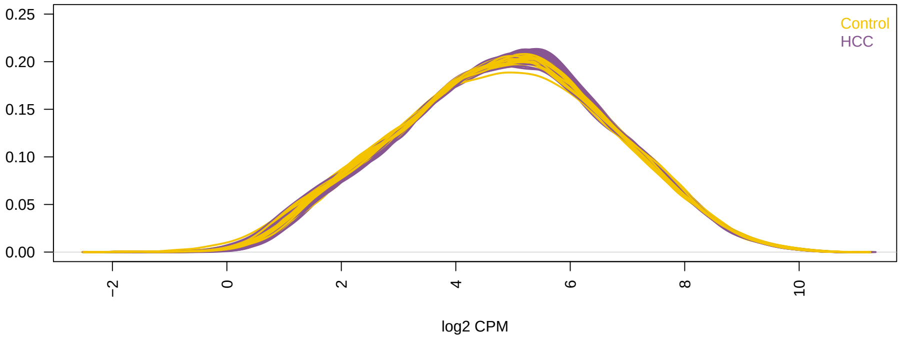

Figure 2.4: Sample log_2 CPM densities after removing weak genes

As another sanity check, we will look at a multidimensional scaling plot of distances between gene expression profiles. We use `plotMDS` in limma package [9]), which plots samples on a two-dimensional scatterplot so that distances on the plot approximate the typical log2 fold changes between the samples.

Before producing the MDS plot we will normalize the distributions. We will store the data into s `DGEList` object as this is convenient when running many of the analyses implemented in the edgeR and limma packages.

```
filteredCountsA_dgel <- edgeR::DGEList(
 counts   = featureCountsA,
 genes    = genes_annotA,
 samples = sampDescA,
 group    = sampDescA$group)
filteredCountsA_dgel <- edgeR::calcNormFactors(filteredCountsA_dgel)
filteredCountsA_lcmp_mtx <- edgeR::cpm(filteredCountsA_dgel, log=T)
```

Verify that the counts are properly mormalized.

```
par(mar = c(1, 3, 2, 1))
boxplot(filteredCountsA_lcmp_mtx,
  ylim = c(1, 8), ylab='Normalized Log CPM',
  staplewex = 0,       # remove horizontal whisker lines
  staplecol = "white", # just to be totally sure :)
  outline = F,         # remove outlying points
  whisklty = 0,        # remove vertical whisker lines
  las = 2, horizontal = F, xaxt = "n",
  border = groupCol[sampDescA$group]
)
legend("top", legend = names(groupCol), text.col = groupCol,
  ncol = 2, bty = "n")
# Add reference lines
SampleMedian <- apply(filteredCountsA_lcmp_mtx, 2, median)
abline(h = median(SampleMedian), col = "grey")
axis(side = 4, at = round(median(SampleMedian), 1),
  las = 2, col = "grey", line = -1, tick = F)
```

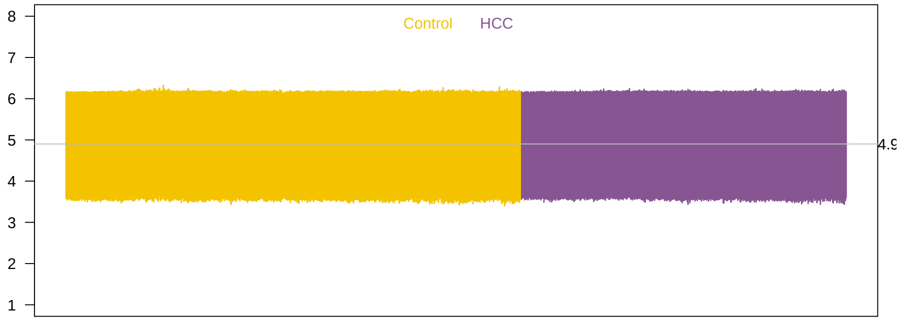

Figure 2.5: Sample log2 count boxplots

Proceed wit MDS plot.

```
 par(mfcol=c(1,2), mar=c(4,4,2,1), xpd=NA, oma=c(0,0,2,0))

# without loss of generality or sensitivity, sample 500 samples
# this is simply a matter of convenience and to save time
 set.seed(1)
 samp_ndx <- sample(1:ncol(filteredCountsA_lcmp_mtx), size=500)
 MDS.out <- limma::plotMDS(filteredCountsA_lcmp_mtx[,samp_ndx],
  col=groupCol[sampDescA$group[samp_ndx]], pch=1)
 legend("topleft", legend = names(groupCol),
  text.col = groupCol, bty = "n")

 MDS.out <- limma::plotMDS(filteredCountsA_lcmp_mtx[,samp_ndx],
  col=groupCol[sampDescA$group[samp_ndx]], pch=1,
     dim.plot=3:4)
```

The MDS plot, which is analogous to a PCA plot adapted to gene exression data, does not indicate strong clustering of samples. The fanning pattern observed in the first two dimensions indicates that a few samples are drifting way from the

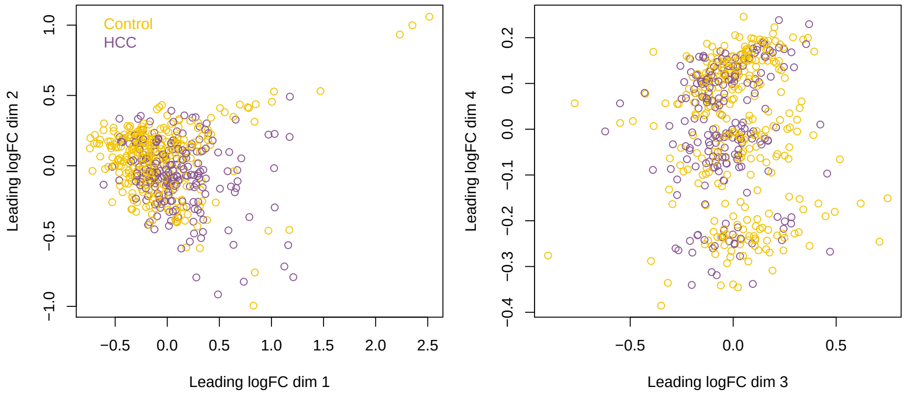

Figure 2.6: MDS plots of log-CPM values

core set, but in no particular direction. There is some structure in the 3rd and 4th dimension plot which should be investigated.
glMDSPlot from package Glimma provides an interactive MDS plot that can extremely usful for exploration

```
# NOTE - changes made to objects here are not saved.


set.seed(1)
samp_ndx <- sample(1:ncol(filteredCountsA_lcmp_mtx), size=500)

Glimma::glMDSPlot(filteredCountsA_dgel[, samp_ndx],
          groups=filteredCountsA_dgel$samples[samp_ndx,
             c("group", "trainValGroup", "sampType", "tissue","title", "stage")],
          main=paste( "MDS plot: filtered counts"), ####, Excluding outlier samples")
          path = '.', folder = figures_DIR,
          html = paste0("GlMDSplot"), launch = F)
```

Link to glMDSPlot: Here

No obvious factor links the samples in the 3 clusters observed on the 4th MDS dimensions. The percent of variance exaplained by this dimension or ≈ 4%. The glMDSPlot indicates further segregation along the 6th dimension. The percent of variance exaplained by this dimension or ≈ 2%. Tracking down this source of variability may be quite challenging, especially without having the complete information about the sample attributes and provenance.

Unwanted variability is a well-documented problem in the analysis of RNA-Seq data (see Peixoto et al. (2015) [10]), and many procedures have been proposed to reduce the effect of unwanted variation on RNA-Seq analsys results ([[11];[10];Risso:2014aa]). There are undoubtedly some similar sources of systematic variation in the 5hmC data, but it is beyond the scope of this work to investigate this in this particular dataset. Given that the clustering of samples occurs in MDS dimensions that explain a small fraction of variability, and that these is no assocation with the factor of interest, HCC vs Control, these sources of variability should not interfere too much with our classification analysis. It would be interesting to see if downstream results can be improved by removing this variability.

2.3 UP TO HERE

Create Design Matrix and Contrasts

Removing heteroscedascity from count data

Fitting linear models for comparisons of interest

Examining the DE genes

2.4 Analysis of coverage variability

We will use the methods described in Hart et al. (2013) [12] to characterize coverage variability in these data. These methods do not take multiple comparisons into account. Other tools for sample size calculation in RNA-Seq studies include Bi and Liu (2016) [13], Baccarella (2018) [14], Guo (2014) [15], Yu (2017) [16], and Zhao (2018) [17]. Poplawski (2018) [18] evaluated RNA-seq sample size tools identified from a systematic search. They found the six evaluated tools provided widely different answers, which were strongly affected by fold change.

The references listed above aim at providing guidance for RNA-Seq experimental design. There is much discussion and a wide range of opinion on sample size requirements to ensure reproducibility in RNA-Seq results. At one end of the spectrum, Ein-Dor et al. (2006) [19] argue that thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. At the other end, Dobbin et al. (2007, 2008) [20,21] claim that sample sizes in the range of 20–30 per class may be adequate for building a good predictor in many cases. Part of the disparity in sample size requirement recomendation comes from differences of opinion in terms of what constitutes `reproducible results`. In the context of sample classification, if we focus on the predicted probabilities for individual samples, we may find good reproducibility across studies with moderate samples sizes. If, on the other hand, we closely inspect the gene signatures reported across studies, much greater sample sizes may be required to achieve concordance. Kim (2009) [22], like Ein-Dor et al., also find issues in RNA-Seq research in terms of the instability of identified prognostic gene signatures, few overlap between independently developed prognostic gene signatures, and poor inter-study applicability of gene signatures. Fan et al. (2006) [23], on the other hand, found good concordance among gene-expression–based predictors for breast cancer. We will return to this question when we examine the relationship between classification model results and sample size in this dataset later on this paper.

For two groups comparisons, the basic formula for the required number of samples per group is:

$$n = 2(z_{1-\alpha/2} + z_{\beta})^2 \frac{(1/\mu + \sigma^2)}{ln(\Delta^2)}$$

- The parameters α and β are **size** and **power** of the test.
- Δ is the targeted **effect size**.
- μ and σ are the **mean** and **coefficient of variation** of the distribution of measurement, gene representation indices in this case.

These three parameters will be fixed across genes or a given study, and are often dictated by external requirements. Typical values might be an effect size of $\Delta = 1.5$ (a.k.a fold change), corresponding to detection of a 50% change in gene expression between the two groups. $z_{1-.05/2} = 1.96$, corresponding to a two sided test at size $\alpha = 0.05$; and $z_{.90} = 1.28$ corresponding to 90% power. The other two variables will be gene and experiment dependent: the normalized depth of coverage μ of the gene, and the coefficient of variation σ in this gene between biological replicates. The technical variation of the comparison is inversely proportional to the number of sequenced reads for the gene and therefore decreases with sequencing depth. The biological variation is a property of the particular gene/model system/condition under study. One would expect it to be smaller for uniform systems such as cell culture and/or products that are under tight regulatory control, and larger for less uniform replicates such as human subject samples. The dataset under study in this report is the first of its kind to give us an idea of variability levels in 5hmC representation.

As in Hart et al. (2013) [12] , we estimate the biological coefficient of variation (CV) in expression across samples in the data set using a negative binomial model.

```
BCV_mtx <- do.call('cbind', lapply(unique(sampDescA$group),
function(GRP) {
 GRP_dgel <-
```

```
    edgeR::DGEList(counts=featureCountsA[, sampDescA$group==GRP])
  GRP_dgel <- edgeR::estimateDisp(GRP_dgel)

  sqrt(GRP_dgel$tagwise.dispersion)
  }))
```

```
## Design matrix not provided. Switch to the classic mode.
## Design matrix not provided. Switch to the classic mode.
```

```
colnames(BCV_mtx) <- unique(sampDescA$group)
plot(spatstat::CDF(density(BCV_mtx[,1])),
  col=groupCol[colnames(BCV_mtx)[1]],
  lwd=2, ylab='Prob(BCV<x)',
  xlim=c(0, 0.5))
for(JJ in 2:ncol(BCV_mtx))
plot(spatstat::CDF(density(BCV_mtx[,JJ])),
  col=groupCol[colnames(BCV_mtx)[JJ]],
  lwd=2, add=T, xlim=c(0, 2.0))
legend('bottomright', legend=names(groupCol), col=groupCol, lwd=2)
BCV_90perc_vec <- round(apply(BCV_mtx,2,quantile, prob=0.90), 2)
BCV_50perc_vec <- round(apply(BCV_mtx,2,quantile, prob=0.50), 2)
for(JJ in 1:length(BCV_90perc_vec))
rug(BCV_90perc_vec[JJ], lwd=2, ticksize = 0.05,
   col=groupCol[names(BCV_90perc_vec)[JJ]])
```

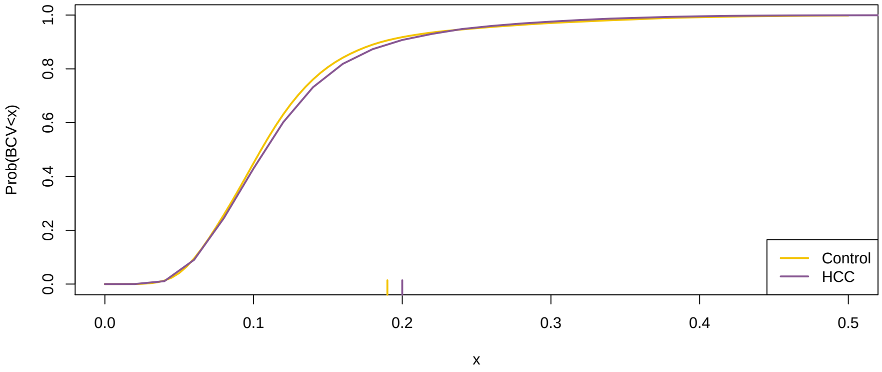

Figure 2.7: Cumulative Distribution of CV - rug = 90th percentile

We can now look at sample size estimates to required to detect various effect sizes. The effect sizes examined here are selected based on the differential representation analysis in Section 2.2 below.

```r
par(mfrow = c(2, 1), mar = c(3, 3, 2, 1), oma = c(2, 2, 2, 0))
for (EFFECT in c(1.10, 1.20)) {
  plot(
    x = 10:100,
    y = RNASeqPower::rnapower(depth = 10:100, cv = max(BCV_90perc_vec),
          effect = EFFECT, alpha = .05, power = .80),
    lwd = 2, ylab = "", xlab = "", type='l'
  )
  title(paste("Effect =", EFFECT, "cv =",
     max(BCV_90perc_vec), "Alpha=0.05, Beta=0.80"))
}
mtext(side = 1, outer = T, "Gene Coverage")
mtext(side = 2, outer = T, "Samples Needed")
mtext(side = 3, outer = T, cex = 1.25,
    "Negative Binomial 2 Group Sample Size Curves")
```

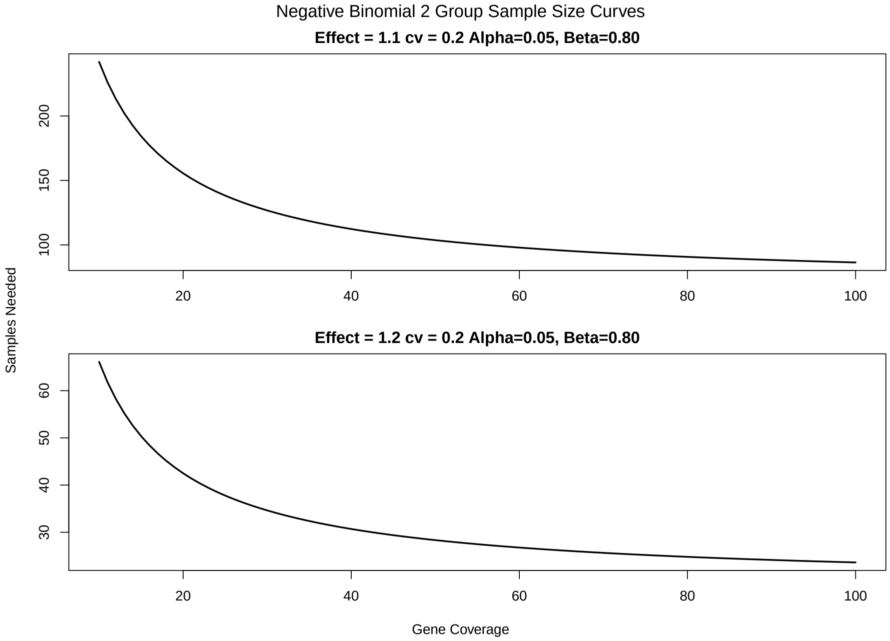

Figure 2.8: Sample Size Estimates

Note that with these data, moderate sample sizes are adequate to detect genes with an effect size of 1.2 (20% fold change). This is due to the fact that the biological variability in gene body 5hmC density is quite low. For human samples, RNA-Seq within group biological variability is typically in the 0.4-1.0 range [12].

Chapter 3

Baseline Model

In the section we look at the baseline model fit:

- What is the accuracy?

- Look at individual points and store some sample scores
- Baseline model
  - how separable are the data
  - individual sample quality

Chapter 4

Fitted Model Suite

We examine the results of fitting a suite of models to investigate the effect of sample size on model performance.

Chapter 5

Conclusions

We have found that ...

Other questions ...

1. Gai, W., and Sun, K. Epigenetic biomarkers in cell-free dna and applications in liquid biopsy. Genes *10*, 32. Available at: https://pubmed.ncbi.nlm.nih.gov/30634483.

2. Cai, J., Chen, L., Zhang, Z., Zhang, X., Lu, X., Liu, W., Shi, G., Ge, Y., Gao, P., and Yang, Y. *et al.* Genome-wide mapping of 5-hydroxymethylcytosines in circulating cell-free dna as a non-invasive approach for early detection of hepatocellular carcinoma. Gut, gutjnl–2019–318882. Available at: http://gut.bmj.com/content/early/2019/07/28/gutjnl-2019-318882.abstract.

3. Li, W., Zhang, X., Lu, X., You, L., Song, Y., Luo, Z., Zhang, J., Nie, J., Zheng, W., and Xu, D. *et al.* DNA 5-hydroxymethylcytosines from cell-free circulating dna as diagnostic biomarkers for human cancers. bioRxiv, 163204. Available at: http://biorxiv.org/content/early/2017/07/13/163204.abstract.

4. Song, C.-X., Yin, S., Ma, L., Wheeler, A., Chen, Y., Zhang, Y., Liu, B., Xiong, J., Zhang, W., and Hu, J. *et al.* (2017). 5-hydroxymethylcytosine signatures in cell-free dna provide information about tumor types and stages. Cell Research *27*, 1231–1242. Available at: https://doi.org/10.1038/cr.2017.106.

5. Collin, F., Ning, Y., Phillips, T., McCarthy, E., Scott, A., Ellison, C., Ku, C.-J., Guler, G.D., Chau, K., and Ashworth, A. *et al.* Detection of early stage pancreatic cancer using 5-hydroxymethylcytosine signatures in circulating cell free dna. bioRxiv, 422675. Available at: http://biorxiv.org/content/early/2018/09/26/422675.abstract.

6. Huang, L.-H., Lin, P.-H., Tsai, K.-W., Wang, L.-J., Huang, Y.-H., Kuo, H.-C., and Li, S.-C. The effects of storage temperature and duration of blood

samples on dna and rna qualities. PloS one *12*, e0184692–e0184692. Available at: https://pubmed.ncbi.nlm.nih.gov/28926588.

7. Permenter, J., Ishwar, A., Rounsavall, A., Smith, M., Faske, J., Sailey, C.J., and Alfaro, M.P. (2015). Quantitative analysis of genomic dna degradation in whole blood under various storage conditions for molecular diagnostic testing. Molecular and Cellular Probes *29*, 449–453. Available at: http://www.sciencedirect.com/science/article/pii/S0890850815300207.

8. Law, C., Alhamdoosh, M., Su, S., Dong, X., Tian, L., Smyth, G., and Ritchie, M. (2018). RNA-seq analysis is easy as 1-2-3 with limma, glimma and edgeR [version 3; peer review: 3 approved]. F1000Research *5*. Available at: https://dx.doi.org/10.12688%2Ff1000research.9005.3.

9. Ritchie, M.E., Phipson, B., Wu, D., Hu, Y., Law, C.W., Shi, W., and Smyth, G.K. Limma powers differential expression analyses for rna-sequencing and microarray studies. Nucleic acids research *43*, e47–e47. Available at: https://pubmed.ncbi.nlm.nih.gov/25605792.

10. Peixoto, L., Risso, D., Poplawski, S.G., Wimmer, M.E., Speed, T.P., Wood, M.A., and Abel, T. How data analysis affects power, reproducibility and biological insight of rna-seq studies in complex datasets. Nucleic acids research *43*, 7664–7674. Available at: https://pubmed.ncbi.nlm.nih.gov/26202970.

11. Gandolfo, L.C., and Speed, T.P. RLE plots: Visualizing unwanted variation in high dimensional data. PloS one *13*, e0191629–e0191629. Available at: https://pubmed.ncbi.nlm.nih.gov/29401521.

12. Hart, S.N., Therneau, T.M., Zhang, Y., Poland, G.A., and Kocher, J.-P. Calculating sample size estimates for rna sequencing data. Journal of Computational Biology *20*, 970–978. Available at: https://doi.org/10.1089/cmb.2012.0283.

13. Bi, R., and Liu, P. Sample size calculation while controlling false discovery rate for differential expression analysis with rna-sequencing experiments. BMC bioinformatics *17*, 146–146. Available at: https://www.ncbi.nlm.nih.gov/pubmed/27029470.

14. Baccarella, A., Williams, C.R., Parrish, J.Z., and Kim, C.C. Empirical assessment of the impact of sample number and read depth on rna-seq analysis workflow performance. BMC bioinformatics *19*, 423–423. Available at: https://www.ncbi.nlm.nih.gov/pubmed/30428853.

15. Guo, Y., Zhao, S., Li, C.-I., Sheng, Q., and Shyr, Y. RNAseqPS: A web tool for estimating sample size and power for rnaseq experiment. Cancer informatics *13*, 1–5. Available at: https://www.ncbi.nlm.nih.gov/pubmed/25374457.

16. Yu, L., Fernandez, S., and Brock, G. (2017). Power analysis for rna-seq differential expression studies. BMC Bioinformatics *18*, 234. Available at: https://doi.org/10.1186/s12859-017-1648-2.

17. Zhao, S., Li, C.-I., Guo, Y., Sheng, Q., and Shyr, Y. RnaSeqSampleSize: Real data based sample size estimation for rna sequencing. BMC bioinformatics *19*, 191–191. Available at: https://www.ncbi.nlm.nih.gov/pubmed/29843589.

18. Poplawski, A., and Binder, H. (2017). Feasibility of sample size calculation for rna-seq studies. Briefings in Bioinformatics *19*, 713–720. Available at: https://doi.org/10.1093/bib/bbw144.

19. Ein-Dor, L., Zuk, O., and Domany, E. Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. Proceedings of the National Academy of Sciences of the United States of America *103*, 5923–5928. Available at: https://pubmed.ncbi.nlm.nih.gov/16585533.

20. Dobbin, K.K., and Simon, R.M. (2007). Sample size planning for developing classifiers using high-dimensional dna microarray data. Biostatistics *8*, 101–117. Available at: https://doi.org/10.1093/biostatistics/kxj036.

21. Dobbin, K.K., Zhao, Y., and Simon, R.M. How large a training set is needed to develop a classifier for microarray data? Clin Cancer Res *14*, 108–114.

22. Kim, S.-Y. Effects of sample size on robustness and prediction accuracy of a prognostic gene signature. BMC bioinformatics *10*, 147–147. Available at: https://pubmed.ncbi.nlm.nih.gov/19445687.

23. Fan, C., Oh, D.S., Wessels, L., Weigelt, B., Nuyten, D.S.A., Nobel, A.B., Veer, L.J. van't, and Perou, C.M. Concordance among gene-expression–based predictors for breast cancer. New England Journal of Medicine *355*, 560–569. Available at: https://doi.org/10.1056/NEJMoa052933.