Group Project: Differential Isoform Usage & Identification of NMD sensitive RNAs and isoforms with long-read direct RNA sequencing(DRS)

Gonzalo Cardenal Antolin (GonzaloCardenalAl), Michael Cibien (MCibien), Tobia Ochsner (ochsneto)

2024-01-12

Introduction

Biological Background – NMD

Nonsense-mediated mRNA decay (NMD) is a eukaryotic pathway that is responsible for degradation of not only aberrant, but also some endogeneous mRNA, that would result in physiologically active proteins. This pathway therefore plays a pivotal role in post-transcriptional gene regulation by eliminating mRNA transcripts that could otherwise lead to the synthesis of truncated or malfunctioning proteins. The underlying mechanisms by which NMD selectively targets mRNAs for degradation are not fully elucidated, however, key proteins implicated in this process have been identified, including UPF1, an RNA helicase; SMG1, a phosphatidylinositol-kinase-related kinase; SMG6, an endonuclease; and SMG5 and SMG7, which function as adaptor proteins. [1] [2]

Historically, it was posited that the primary role of NMD was to target mRNAs containing premature stop codons. Recent research, however, indicates that the scope of NMD is broader, encompassing additional features of mRNAs. To explore these additional features, particularly in endogenous mRNAs, researchers have employed strategies such as the knockdown of key NMD proteins. This approach increases the intracellular concentration of NMD-targeted mRNA isoforms, facilitating their study through subsequent RNA sequencing (RNA-seq) experiments.

Gene and isoform expression analysis from direct RNA sequencing (DRS)

In conventional differential gene expression analysis short-read sequencing is widely used. However, short-read RNA sequences have limitations in identifying and quantifying gene transcript isoforms due to RNA fragmentation, bias introduction in the reverse transcription into cDNA and the subsequent amplification by PCR. DRS could act as a potential remedy to this problem, since it sequences full-length transcripts, while also characterizing RNA modifications, polyA tails and not needing prior PCR amplification.[3] [4]

It has been shown by Josie Gleeson et al [5] that DRS has the potential to quantify both genes and transcript isoforms in an unbiased manner allowing for a multitude of analyses: gene and isoform quantification; differential gene expression analysis; differential isoform usage analysis (DUI); discovery of novel isoforms. However, the key challenges of long-read RNA sequencing – namely sequencing depth – remains and impacts the sensitivity of isoform identification with a high probability.[5]

Goal of this Study

We were lucky to receive DRS data from a follow up project on their paper by Dr. Evan Karousis. They assessed endogenous targets of NMD using a combination of long-read and short-read sequencing and found that the juxtaposition of short-read sequencing with long-read sequencing enabled the identification of novel NMD-targeted

mRNA isoforms. [1]

By comparing the results of their previous work to the results of this study, where we use DRS, we aim to answer the question of whether DRS alone possesses sufficient resolution to identify NMD-sensitive mRNAs in human cells.

Data Background

The data was kindly provided by Dr. Evangelos Karousis, who generated the data while working as post-doctoral researcher in the laboratory of Prof. Dr. Oliver Mühlemann, at the University of Bern in 2019.

Knockdown experiments on HeLa cells were performed by RNA interference (RNAi). For this, plasmids expressing shRNAs against SMG6 and SMG7 were transfected into the "dKD" cells, whereas in the "Scr" (control) cells plasmids targeting nothing were transfected. polyA+ mRNA was then isolated, and direct RNA-seq was performed using the Flowcell SQK-RNA002 from Oxford Nanopore. The MinKNOW instrument software generated fast5 files, on which basecalling was performed using GUPPY (from Oxford Nanopore Technologies).

The data was divided into 6 folders, NP05_Scr1, NP06_dKD1, NP07_dKD2, NP07B_dKD2, NP08_Scr2, NP08B_Scr2, where "Scr" refers to the controls and "dKD" to the double-knockdown conditions. The two conditions were performed in a biological duplicate, however the sequencing run of dKD2 and Scr2 samples were interrupted, so thats why there are "B" samples. These folders were merged after quality control (before alignment and subsequent analyses) and therefore conditions are referred to as Scr1, Scr2, dKD1 and dKD2.

Preprocessing

Quality Control (QC)

To validate the quality control and the filtering heuristics that were performed by [1] with GUPPY, two scripts were written and run. We confirmed that the filtering criterion was a q-score lower than 7. (The scripts are available on the github repository under filter.r and checkGuppyFilter.r). We then used NanoPlot, which generated a report of different sequencing quality statistics like mean and median read quality, read lengths etc. These reports are also available here. The results provided by NanoPlot confirmed that our samples had mean read quality greater than 7 and displayed coherent read length distribution, sequencing error rates, and number of reads.

The sequencing of DRS is more difficult than sequencing cDNA, since mRNA contains dynamic base modifications, for example one of the most prevalent base modification is N6-methyladenosine (m6A) [6]. For such modified bases the current produced when passing through the ONT MinION pore are atypical and thus are read as different nucleotides. Therefore higher error rates are expected and if we were to investigate mRNA modifications a lower quality score threshold should be contemplated.

Mapping and quantification

To map long reads, the current gold standard is using minimap2. We ran two mappings, one to the genome and another one to the transcriptome. In the genomic mapping, Homo_sapiens.GRCh38.dna.primary_assembly.fa was used as the reference genome for each sample. In the transcriptome mapping cDNA reference from the Ensmbl database was used.

To quantify the genes Feature Counts was used, which provides a mode "-L" tailored for long reads. The annotation file was obtained from the Ensembl database. With this mode the raw counts were obtained, which were needed by Deseq2 for the differential gene expression analysis. For the transcript and the different isoform quantification NanoCount was utilized. NanoCount is a recently released tool specifically developed to quantify Oxford Nanopore direct RNA sequencing (DRS) reads. [5]

The scripts for mapping and quantification are available here.

The most important quality control metrics were summarized in Figure 1. Reads had a median quality score of 8.1 - 8.2 across all samples and mean redian lengths were also comparable. The only metric that differed significantly between the samples was the number of reads, however this is partly due to the aforementioned interruptions of the DRS runs.

	Scr1	Scr2		dKD1	dKD2	
	NP05_Scr1	**NP08_Scr2**	**NP08B_Scr2**	**NP06_dKD1**	**NP07_dKD2**	**NP07B_dKD2**
Number of reads	1'549'683	1'776'660	481'888	1'592'867	915'205	1'340'688
Median read length (nt)	946	950	923	1003	979	912
Median read quality score	8.1	8.2	8.1	8.1	8.2	8.1
Longest alignment length (nt)	288'042	121'433	46'021	166'309	59'725	41'680
Reads aligned to genome (primary) (%)	41.5	41.7		43.5	42.9	
Reads aligned to genome (%)	76.2	76.3		72.3	72.7	

Figure 1: Direct RNA sequencing quality scores and genome alignment metrics.

Statistical Analysis

Transcript Coverage

To assess the coverage of the reads, coverage fractions for the samples were computed using BamSlam scripts. Coverage fraction is defined as the ratio between the transcript length and the theoretical full length of their respective isoforms.

```
Rscript BamSlam.R rna NP05-sorted-transcriptomic-aln.bam NP05_BamSlam_output
Rscript BamSlam.R rna NP06-sorted-transcriptomic-aln.bam NP06_BamSlam_output
Rscript BamSlam.R rna NP07-sorted-transcriptomic-aln.bam NP07_BamSlam_output
Rscript BamSlam.R rna NP08-sorted-transcriptomic-aln.bam NP08_BamSlam_output
```

The median coverage fraction is of special interest to us, as we would expect a 5-10% increase in full-length transcripts upon SMG6 knockdown, since its nuclease activity is suppressed. In our case, the median coverage fraction for Scr1 and Scr2 are 81.90% and 80.99% (average = 81.45%), while in the knockdown samples they are 80.30% and 78.69% for dKD1 and dKD2 respectively (average = 79.50%). This demonstrates that in both conditions most reads covered most of the original RNA transcript lengths. However, we cannot observe a significant change of the median transcript coverage between the two conditions. Additionally, the number of full-length reads does not change significantly either between the two conditions, as can be seen in the Figure 2.

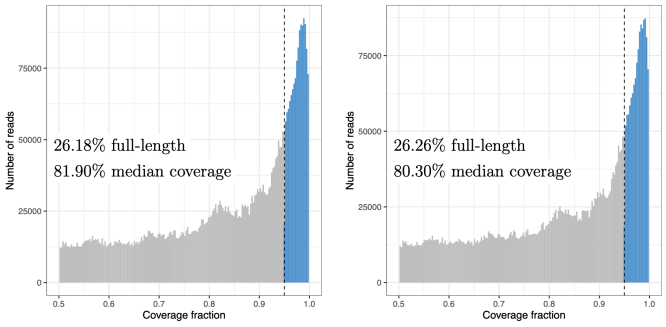
When assessing the relationship between coverage fraction and known transcript length, it becomes apparent that longer transcripts were less likely to be aligned at full length. This can be seen in all conditions and there does not seem to be an observable difference between them.

Import Data

The code for importing the data is not shown for brevity purposes. The full code including the sections omitted here and in subsequent sections can be found in the .qmd file in the repository.

We are working with direct RNA long-reads. The output from NanoCount are estimated counts (est_count) and transcripts per million (tpm). These metrics are not normalised by transcript length as it is usually done with Illumina data because in DRS one read is supposed to represent a single transcript molecule starting from the polyA tail, even if the fragment does not extend to the 5' end.

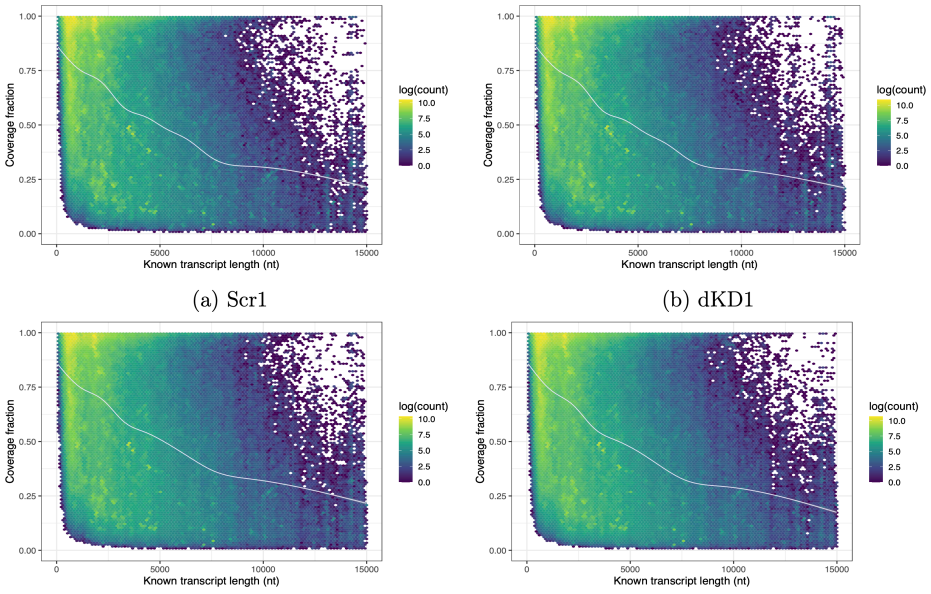
Est_count is obtained by multiplying the raw abundance by the number of primary alignments, the latter is the estimated counts obtained by multiplying the raw abundance by 10^6. It was decided to use tpm for the subsequent analyses.



(a) Scr1 (b) dKD1

(c) Scr2 (d) dKD2

Figure 2: Distribution of transcript coverage fraction. Dotted line represents 95% cutoff for full-length reads. Full-length reads are shaded in blue.



(a) Scr1 (b) dKD1

(c) Scr2 (d) dKD2

Figure 3: Fraction of known transcript length covered by each read (coverage fraction) compared to known transcript length. Trend line was plotted using a generalized additive model, an extension of a generalised linear model where the linear form is replaced by sum of smooth functions.

Exploratory Analysis

Once the gene and transcript counts were obtained, an initial exploratory analysis was performed.

```
  colSums(raw_genomic)
```

```
NP05-Scr1 NP06-dKD1 NP07-dKD2 NP08-Scr2
   882803    877859   1239173   1323920
```

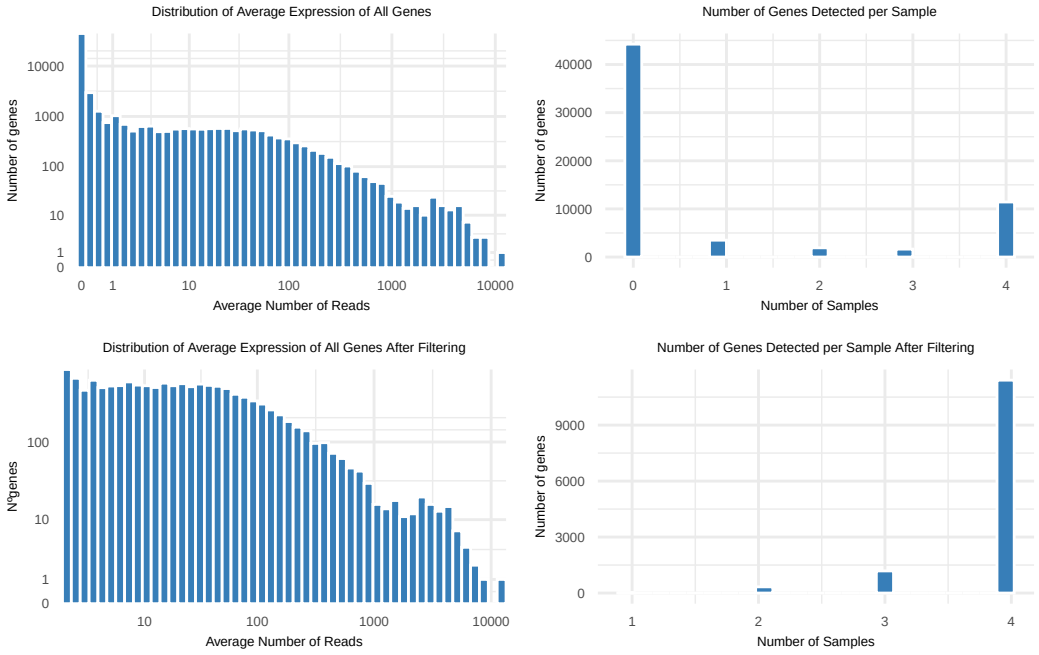
```
  colSums(tpm_transcriptome)
```

```
tpm_np05 tpm_np06 tpm_np07 tpm_np08
   1e+06    1e+06    1e+06    1e+06
```

The samples exhibit the same number of transcript counts and number of total counts, whereas the raw genomic counts differ from the number of total counts.

```
  avg_genomic <- rowMeans(raw_genomic)
  layout <- matrix(c(1, 2, 3, 4), nrow = 2, byrow = TRUE)

  theme_custom <- theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5, size = 5),
      axis.text = element_text(size = 5),  # Adjust the size as needed
      axis.title = element_text(size = 5)  # Optional: Adjust the size of axis titles
    )

  plot_1 <- ggplot(data = as.data.frame(avg_genomic), mapping =
                     aes(x = avg_genomic)) +
    geom_histogram(
      color = "white",
      fill = brewer.pal(n = 3, name = "Set1")[2],
      bins = 50
    ) +
    scale_x_continuous(
      breaks = c(0, 1, 10, 100, 1000, 10000, 20000),
      trans = "log1p",
      expand = c(0, 0)
    ) +
    scale_y_continuous(breaks = c(0, 1, 10, 100, 1000, 10000),
                       expand = c(0, 0),
                       trans = "log1p") +
    labs(title = "Distribution of Average Expression of All Genes",
         x = "Average Number of Reads",
         y = "Number of genes") +
      theme_custom


  num_detected_genes <- rowSums(raw_genomic > 0)

  plot_2 <- ggplot(data = as.data.frame(num_detected_genes), mapping =
                     aes(x = num_detected_genes)) +
    geom_histogram(color = "white",fill = brewer.pal(n = 3, name = "Set1")[2],
    bins = 23  ) +labs(title = "Number of Genes Detected per Sample",
                       x = "Number of Samples", y = "Number of genes") +theme_custom
```

```
# filter expressed genes
# threshold: genes must be detected in at least half of the samples
#            or the average counts must be >= 1

expressed<-rowSums(raw_genomic)>=5
num_filtered_expressed_genes<-rowSums(raw_genomic[expressed,]>0)

avg_genomic <- data.frame(avg_genomic)
plot_3 <- ggplot(data = as.data.frame(avg_genomic[expressed,]), mapping =
                   aes(x = avg_genomic[expressed,])) +
  geom_histogram(color = "white",fill = brewer.pal(n = 3, name = "Set1")[2],
    bins = 50) +
  scale_x_continuous(breaks = c(0, 1, 10, 100, 1000, 10000, 20000),
    trans = "log1p",expand = c(0, 0)) +
  scale_y_continuous(breaks = c(0, 1, 10, 100, 1000, 10000),expand = c(0, 0),trans = "log1p") +
  labs(title = "Distribution of Average Expression of All Genes After Filtering",
       x = "Average Number of Reads",
       y = "N°genes") + theme_custom

plot_4 <- ggplot(data = as.data.frame(num_filtered_expressed_genes), mapping =
                   aes(x = num_filtered_expressed_genes)) +
  geom_histogram(color = "white",fill = brewer.pal(n = 3, name = "Set1")[2],bins = 23) +
  labs(title = "Number of Genes Detected per Sample After Filtering",
      x = "Number of Samples",y = "Number of genes") +theme_custom

grid.arrange(plot_1, plot_2, plot_3, plot_4, layout_matrix = layout)
```

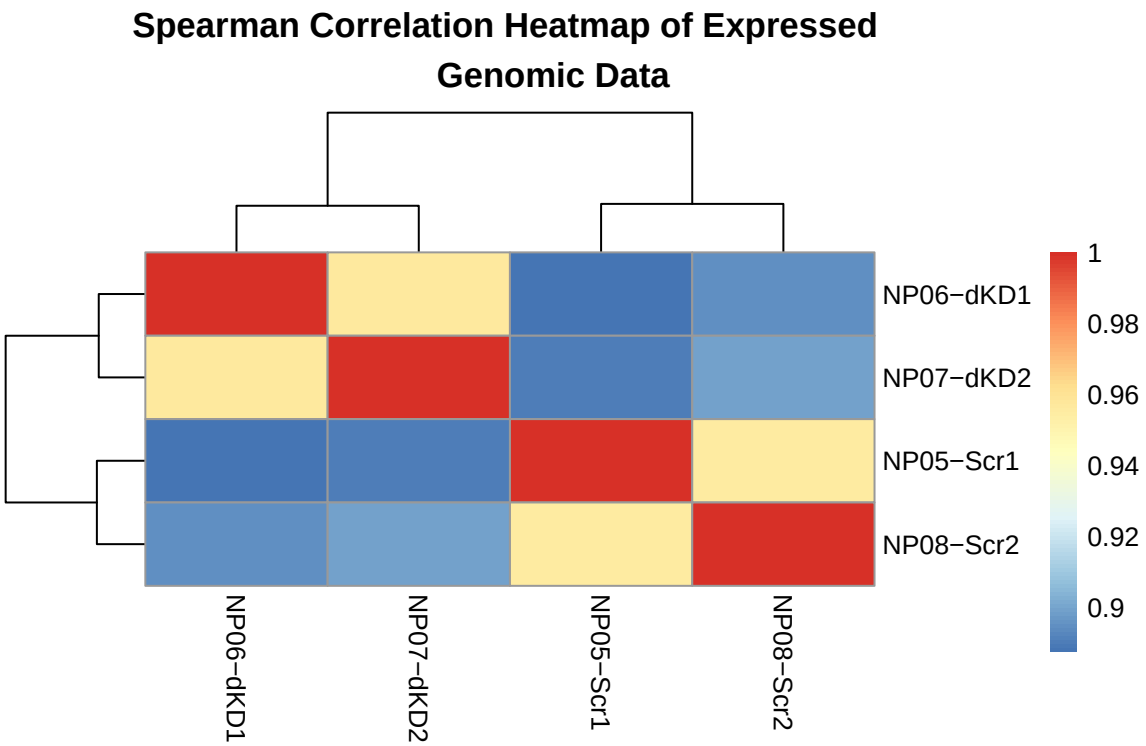


In the plot to the left we can see the number of genes distributed by their average number of counts in logarithmic scale. Because the quantification of the genes in long reads are one read equivalent to one count, these two histograms display the number of genes distributed by the number of reads. The histograms to the left show that DRS is able to quantify certain genes with up to 10000 average counts (4 orders of magnitude), which indicates a good sensitivity

for the quantification of gene expression.

The genes were then filtered by removing those with a sum over the samples lower than 5 counts. By looking at the plots on the right side of the figure, we can observe that our filtering criteria filtered not only all the genes that were not expressed in any of the samples, but also most of the genes only expressed in one individual sample.

```
#Plot heatmap
corr_pearson <- cor(log1p(raw_genomic[expressed,]), method = "spearman")
pheatmap(corr_pearson, main="Spearman Correlation Heatmap of Expressed
         Genomic Data",  cex.axis = 0.5, cex.lab = 0.5, cex.main = 0.5)
```



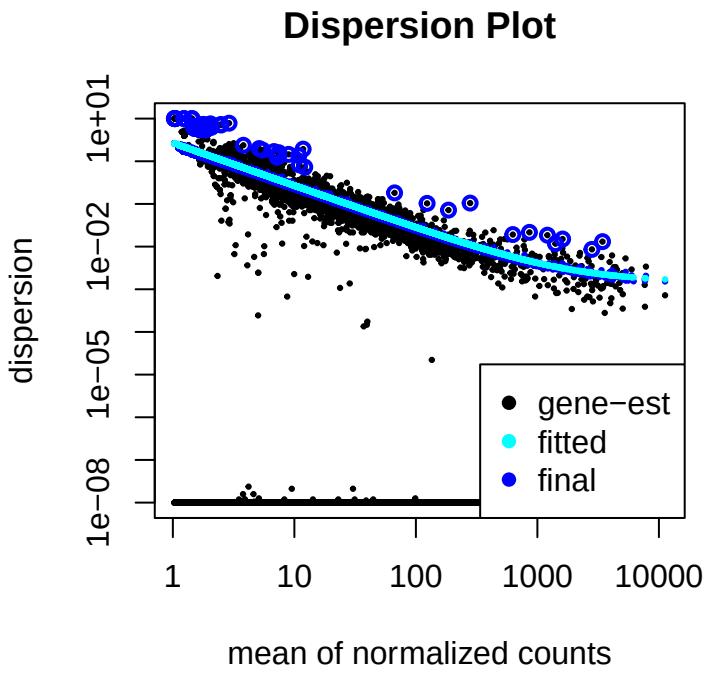
The heatmap clearly shows a correlation between the two control samples, as well as the knockdown samples, which indicates that there is a differential expression between the knockdown and the control. This again indicates that the NMD knockdown is causing a differential gene expression.

Differential Gene Expression Analysis

```
#DESeq input
raw_genomic<- as.matrix(raw_genomic[expressed,])
condition <- factor(c("control","knockdown","knockdown","control")) #NP05-Scr1, NP06-dKD1, NP07-dKD2, NP08-Scr2
coldata <- data.frame(row.names=colnames(raw_genomic), condition)

# Make DESeq dataset
dds <- DESeqDataSetFromMatrix(countData=raw_genomic, colData=coldata,
                              design=~condition)

# Run DESeq2 pipeline
```

```
dds <- DESeq(dds)
res <- DESeq2::results(dds)

#DESeq2 results
res <- res[order(res$padj), ]

# Merge with normalized count data
resdata <- merge(as.data.frame(res), as.data.frame(counts(dds, normalized=TRUE)),
                 by="row.names", sort=FALSE)
names(resdata)[1] <- "Gene"
resdata$DE <-resdata$padj<0.05
```

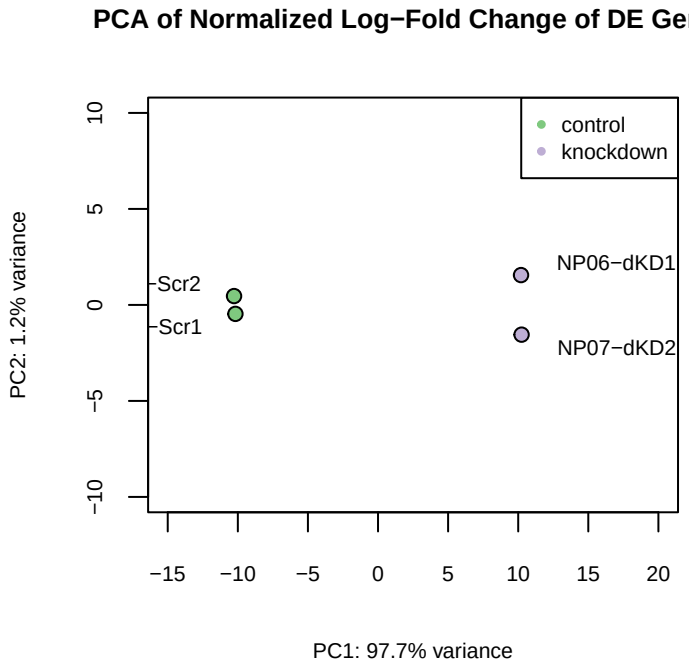
```
# Plot dispersions
plotDispEsts(dds, main="Dispersion Plot", genecol = "black", fitcol = "cyan", finalcol = "blue", legend = TRUE)
```



A trend can be observed that dispersion decreases as the mean of normalized counts increases, which is typical in RNA-seq data due to biological variability being more pronounced in genes with low expression levels. The model (fitted line) captures the overall trend of the dispersion estimates well, as indicated by the fitted points that closely follow the line.

```
rld <- rlogTransformation(dds) #applies a regularized log transformation to the dds(DESeqDataSet)

#Set colours for plotting
mycols <- brewer.pal(8, "Accent")[1:length(unique(condition))]

# PCA
rld_pca <- function(rld, intgroup = "condition", ntop = 500, colors = NULL, legendpos = "topright",
                    main = "PCA of Normalized Log-Fold Change of DE Genes", textcx = 1, ...) {
```

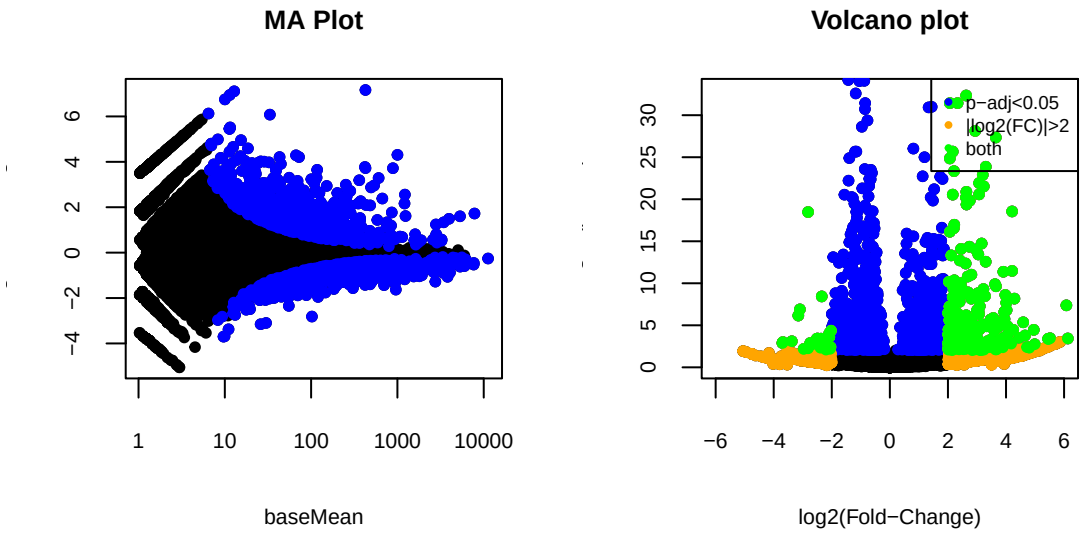
```
  rv <- rowVars(assay(rld))
  select <- order(rv, decreasing = TRUE)[seq_len(min(ntop, length(rv)))]
  pca <- prcomp(t(assay(rld)[select, ]))
  fac <- factor(apply(as.data.frame(colData(rld)[, intgroup, drop = FALSE]), 1, paste, collapse = " : "))

  pc1var <- round(summary(pca)$importance[2, 1] * 100, digits = 1)
  pc2var <- round(summary(pca)$importance[2, 2] * 100, digits = 1)
  pc1lab <- paste0("PC1: ", as.character(pc1var), "% variance")
  pc2lab <- paste0("PC2: ", as.character(pc2var), "% variance")


  plot(PC2 ~ PC1, data = as.data.frame(pca$x), bg = colors[fac], pch = 21, xlab = pc1lab,
       ylab = pc2lab, main = main, cex.axis = 0.7, cex.lab = 0.7, ...)

  with(as.data.frame(pca$x), textxy(PC1, PC2, labs = rownames(pca$x), cex = textcx))
  legend(legendpos, legend = levels(fac), col = colors, cex = 0.7, pch = 20)
}

rld_pca(rld, colors = mycols, intgroup = "condition", xlim = c(-15, 20), ylim = c(-10, 10),
        cex.main = 0.8, textcx = 0.7)
```



PCA further confirms that there is little variability between the biological replicates but high variability between the gene expression of control and knockdown samples. This indicates that the DRS long-reads are capturing the variability of the NMD knockdown at a gene expression level.

```
# MA Plot
maplot <- function (res, thresh = 0.05, xlab = "Base Mean", ylab = "log2(Fold-Change)", labelsig = FALSE,
                    textcx = 1, ...) {
  with(res, plot(baseMean, log2FoldChange, pch = 20, cex.axis = 0.7, cex.lab = 0.7, log = "x", ...))
  with(subset(res, padj < thresh), points(baseMean, log2FoldChange, col = "blue", pch = 20, cex = 1))
```

```
  if (labelsig) {
    require(calibrate)
    with(subset(res, padj < thresh), textxy(baseMean, log2FoldChange, labs = Gene, cex = textcx, col = 2))
  }
}

# Volcano Plot
volcanoplot <- function (res, lfcthresh = 2, sigthresh = 0.05, xlab = "log2(Fold-Change)",
                         legendpos = "topright", labelsig = FALSE, textcx = 1.5, ...) {
  with(res, plot(log2FoldChange, -log10(pvalue), pch = 20, xlab = xlab, cex.axis = 0.7,
                 cex.lab = 0.7, ...))  # Adjust cex.axis and cex.lab
  with(subset(res, padj < sigthresh), points(log2FoldChange, -log10(pvalue), pch = 20, col = "blue", ...))
  with(subset(res, abs(log2FoldChange) > lfcthresh), points(log2FoldChange, -log10(pvalue), pch = 20,
                                                            col = "orange", ...))
  with(subset(res, padj < sigthresh & abs(log2FoldChange) > lfcthresh), points(log2FoldChange,
                                                               -log10(pvalue), pch = 20, col = "green", ...))
  legend(legendpos, xjust = 1, yjust = 1, legend = c(paste("p-adj<", sigthresh, sep = ""),
                                                     paste("|log2(FC)|>", lfcthresh, sep = ""), "both"),
         cex = 0.6, pch = 20, col = c("blue", "orange", "green"))  # Adjust cex
}

# Set up the layout
par(mfrow = c(1, 2), mar = c(4, 3, 3, 2) + 0.1)

# Plot MA Plot
maplot(resdata, main = "MA Plot", cex.main = 0.8)

# Plot Volcano Plot
volcanoplot(resdata, lfcthresh = 2, sigthresh = 0.05, xlim = c(-6, 6), ylim = c(0, 33),
            legendpos = "topright", main="Volcano plot", cex.main = 0.8)
```




```
# Reset the par settings to default after plotting
par(mfrow = c(1, 1), mar = c(4, 3, 3, 2) + 0.1)
```

Differential Isoform Usage Analysis

A differential isoform usage analysis between the control and the knockdown with IsoformSwitchAnalyzeR was run. This analysis is done to identify genes whose isoform abundances change due to the dKD of SMG6 and SMG7. Therefore, the identified isoforms should be direct or indirectly NMD sensitive. Indirectly NMD sensitive would refer to transcripts that are regulated by an NMD sensitive factor and the non-degradation of the NMD sensitive factor would lead to a change in the expression of that isoform.

```
  sampleID = c("tpm_np05","tpm_np06","tpm_np07","tpm_np08")
  myDesign = data.frame(sampleID= sampleID, condition = condition)
  tpm_transcriptome$isoform_id <- rownames(tpm_transcriptome)

  aSwitchList <-  importRdata(
    isoformCountMatrix   = tpm_transcriptome,
    isoformRepExpression = tpm_transcriptome,
    designMatrix         = myDesign,
    isoformExonAnnoation = "files/Homo_sapiens.GRCh38.110.chr_patch_hapl_scaff.gtf",
    isoformNtFasta       = "files/transcriptome_reference.fa",
    showProgress = FALSE
  )
```

```
            comparison estimated_genes_with_dtu
1 control vs knockdown               644 - 1073
```

```
  SwitchListFiltered <- preFilter(
    switchAnalyzeRlist = aSwitchList,
    geneExpressionCutoff = 5,
    isoformExpressionCutoff = 5,
    removeSingleIsoformGenes = TRUE)

  SwitchListAnalyzed <- isoformSwitchTestDEXSeq(
    switchAnalyzeRlist = SwitchListFiltered,
    reduceToSwitchingGenes=TRUE,
    reduceFurtherToGenesWithConsequencePotential = FALSE,
    alpha = 0.05,
    dIFcutoff = 0.1,
    onlySigIsoforms = FALSE
  )

  extractSwitchSummary(SwitchListAnalyzed)
```

```
            Comparison nrIsoforms nrSwitches nrGenes
1 control vs knockdown        823        520     488
```

```
  summary_isoforms <- SwitchListAnalyzed$isoformFeatures
```

The DIU identified 1676 transcripts with differential isoform usage.

From the output of the IsoformSwitchAnalyzeR, a dataset was obtained where one of the columns classifies the isoforms according to the biological function (labelled as "nonsense_mediated_decay" in $iso_biotype). This classification is based on the gencode annotation. The way in which IsoformSwitchAnalyzeR classifies transcripts as NMD-sensitive is described in the documentation: "If the coding sequence (following the appropriate reference) of a transcript finishes >50bp from a downstream splice site then it is tagged as NMD. If the variant does not cover the full reference coding sequence then it is annotated as NMD if NMD is unavoidable i.e. no matter what the exon structure of the missing portion is the transcript will be subject to NMD."

Although, all the differentially used isoforms obtained by the DIU analysis should be classified as NMD sensitive, the number of transcripts classified by the gencode annotation is only 366. This difference might arise due to the classification used in IsoformSwitchAnalyzeR, which is not necessarily correct.

For that reason, the results were also compared to a list of high confidence NMD-transcripts which were identified by Karousis et al [1].

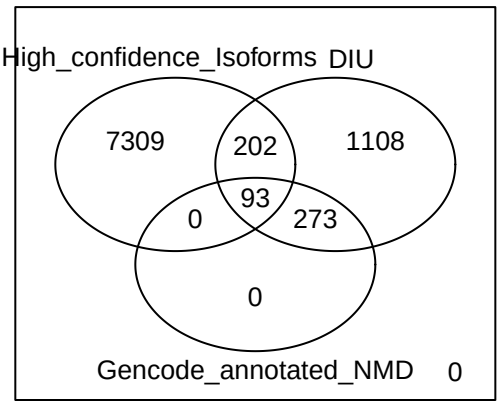
```
new_nmd = read.csv(file = 'NMD_exons_to_long_reads_transcriptome_mapping.csv',
                   sep = ',', header = TRUE)
```

In the list of high confidence NMD-transcripts, 16828 transcripts are identified. The reference transcriptome that was built and used in their study was unavailable, which is why it was not possible to identify those transcripts they identified in addition to those from ENSEMBL. These isoforms were labelled as "MSTRG*" and were thus filtered out.

```
filtered_new_nmd <- new_nmd[grepl("^ENST", new_nmd$transcript_id), ]
summary_isoforms$isoform_id <- sub("\\..*", "", summary_isoforms$isoform_id)
```

A Venn Diagram is displayed to quantify the overlap between the 3 different classifications

```
unique_transcripts <- unique(filtered_new_nmd$transcript_id)
unique_isoforms <- unique(summary_isoforms$isoform_id)

all_unique_ids <- unique(c(unique_transcripts, unique_isoforms))

merged_transcripts <- data.frame(row.names = all_unique_ids)

merged_transcripts$High_confidence_Isoforms <- FALSE
merged_transcripts$DIU <- FALSE
merged_transcripts$Gencode_annotated_NMD <- FALSE

merged_transcripts[rownames(merged_transcripts) %in% filtered_new_nmd$transcript_id,
                   "High_confidence_Isoforms"] <- TRUE

merged_transcripts[rownames(merged_transcripts) %in% summary_isoforms$isoform_id, "DIU"] <- TRUE

nonsense_mediated_decay_ids <- summary_isoforms$isoform_id[summary_isoforms$iso_biotype == "nonsense_mediated_decay
merged_transcripts[rownames(merged_transcripts) %in% nonsense_mediated_decay_ids,
                   "Gencode_annotated_NMD"] <- TRUE

x<-vennCounts(merged_transcripts)
vennDiagram(x, cex = 0.8)
```



To validate that the identification of genes is somewhat meaningful, the presence of BAG1 was corroborated, which is a gene that is known to be differentially spliced and thus produces different isoforms under dKD conditions. [1]

```
"BAG1" %in% summary_isoforms$gene_id
```

```
[1] TRUE
```

Splicing Analysis

Furthermore, SplAdder [7] was used to perform a splicing analysis similar to [1]. SplAdder first generates a splicing graph based on the RNA sequencing data and extracts alternative splicing events (compared to the reference genome). The detected splicing events correspond to single or multiple skipped exons, intron retentions, alternative 3' or 5' splicing, and mutually exclusive exons. In a second step, it runs a differential test in order to find events that occur with a significantly different frequency in the knockdown samples compared to wildtype.

```
python -m spladder.spladder build --annotation $annotationFile \
                                  --bams $wt1,$wt2,$kd1,$kd2 \
                                  --outdir $outFolder

python -m spladder.spladder test \
              --conditionA $wt1,$wt2 \
              --conditionB $kd1,$kd2 \
              --parallel 24 \
              --outdir $outFolder
```

SplAdder can then be used to confirm different splicing behavior previously reported in literature. One example is the apoptosis-modulating Bcl-2-associated athanogene-1 (BAG-1). [1] reported that a BAG1 isoform with an included alternative exon is stabilized upon NMD inhibition. Using our data and our pipeline, we can confirm this.

```
python -m spladder.spladder viz \
              --range gene ENSG00000107262 \
              --track coverage wildtype:$wt1,$wt2 knockdown:$kd1,$kd2 \
              --track event exon_skip \
              -O plot --format png \
              --outdir $outFolder
```

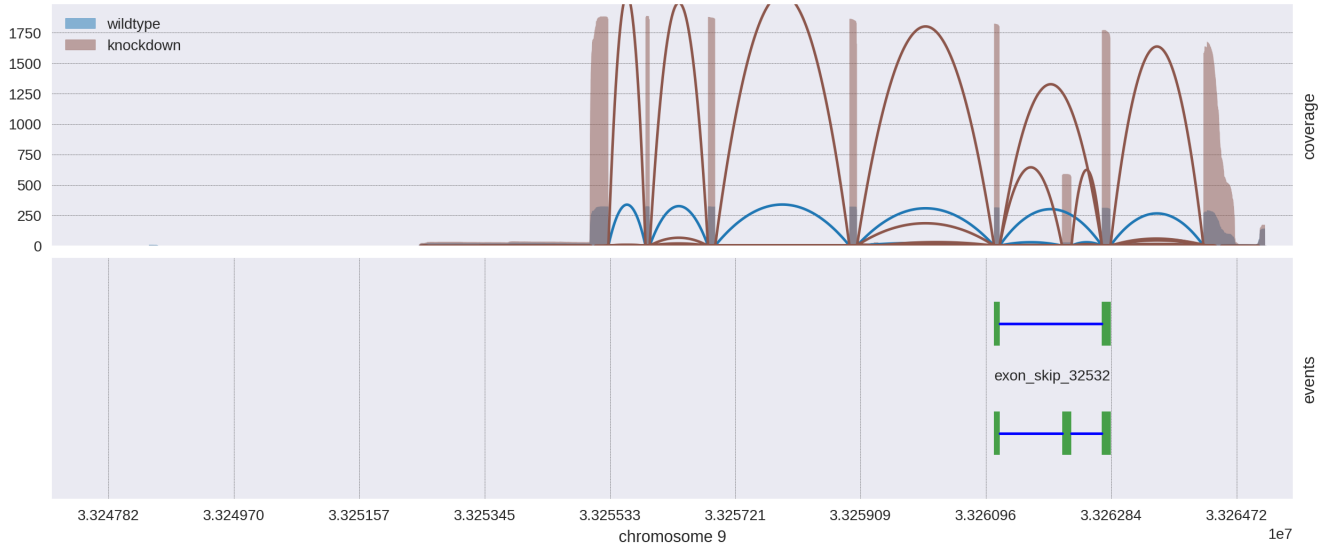


Figure 4: SplAdder Analysis of the BAG-1 Gene. One can observe that an alternative isoform is predominant in the knockdown sample, an observation that is in line with previous literature.

Discovery of New Isoforms

FLAIR [8] and gffcompare [9] were used to detect new isoforms in the sequences of the knockdown samples when compared to the reference annotation.

```
python -m flair.flair correct \
--gtf $annotationFile \
--genome $referenceFile \
--query $experimentFolder/alignments/primary-genomic-aln.bed12 \
--output $experimentFolder/flair/correct

python -m flair.flair collapse \
--gtf $annotationFile \
--reads $rawReads \
--genome $referenceFile \
--query $experimentFolder/flair/correct_all_corrected.bed \
--trust_ends \
--stringent \
--threads 16 \
--output $experimentFolder/flair/collapse

/cluster/home/ochsneto/gffcompare-0.12.6.Linux_x86_64/gffcompare -r $annotationFile \
-o $experimentFolder/flair/gffcompare -V $experimentFolder/flair/collapse.isoforms.gtf
```

Flair detects high-confidence isoforms solely based on aligned reads. As it does not rely on a reference annotation, it allows for the detection of novel isoforms. The analysis resulted in 241 novel exons as compared to the Ensembl reference transcriptome used for our mapping (see Table 1). As discussed above, DRS has the power to obtain This confirms that DRS long-read sequencing has the power to detect previously unknown transcript variations.

Table 1: Results of the FLAIR analysis comparing the isoforms in the knockdown samples with the reference transcriptome.

Type	Result
Missed exons	477423/666967 (71.6%)
Novel exons	241/81798 (0.3%)
Missed introns	282313/402495 (70.1%)
Novel introns	0/73365 (0.0%)
Missed loci	49706/57652 (86.2%)
Novel loci	228/8579 (2.7%)

Conclusion

The goal was to assess if direct RNA sequencing can be used to explore NMD-sensitive mRNAs in human cells. We wanted to assess if this approach corroborates the results obtained by the analysis of Karousis et al [1], but by using DRS instead of a combination of short-read and long-read sequencing.

The differential gene expression analysis showed differences between the dKD and the Scr samples, as well as a clear separation in the dimensionality reduction by PCA. Since NMD targets mRNAs based on their features and not their biological role, a functional enrichment analysis was not considered essential.

The efficiency of our pipeline is confirmed by the fact that we detect a signature NMD-sensitive exon that was discovered with the previous combined work. According to our analysis we detect new exons that could be further verified. However, an early filtering step reduces significantly the number of expressed genes, pointing out that probably RNA sequencing alone is not sufficient to provide an in-depth analysis of the NMD transcriptome.

Even though the 1676 transcripts that are identified by DIU are treated as NMD-sensitive transcripts, this is not a correct assumption. Since NMD not only targets aberrant mRNA (primary targets), but is also involved in post-transcriptional gene regulation by degrading mRNAs that would code for e.g. splicing factors (secondary targets), such differentially expressed isoforms could also arise due to the increase in expression of otherwise regulated secondary targets. To assess those isoforms, however, in addition to the dKD and Scr experiments a rescue experiment would have to be performed.

By comparing the three methods of annotating transcripts as NMD-sensitive, 93 of them were observed in both the list of highly confident NMD-sensitive isoforms of Karousis et al, the Gencode annotation and our DIU analysis, reflecting the need to reconsider the usage of long RNA sequencing alone to perform an in-depth NMD analysis.

References

[1] E. D. Karousis, F. Gypas, M. Zavolan, and O. Mühlemann, "Nanopore sequencing reveals endogenous NMD-targeted isoforms in human cells," *Genome biology*, vol. 22, no. 1, pp. 1–23, 2021.

[2] E. D. Karousis and O. Mühlemann, "The broader sense of nonsense," *Trends in biochemical sciences*, 2022.

[3] D. Aird *et al.*, "Analyzing and minimizing PCR amplification bias in illumina sequencing libraries," *Genome biology*, vol. 12, no. 2, pp. 1–14, 2011.

[4] T. Steijger *et al.*, "Assessment of transcript reconstruction methods for RNA-seq," *Nature methods*, vol. 10, no. 12, pp. 1177–1184, 2013.

[5] J. Gleeson *et al.*, "Accurate expression quantification from nanopore direct RNA sequencing with NanoCount," *Nucleic Acids Research*, vol. 50, no. 4, pp. e19–e19, 2022.

[6] I. A. Roundtree, M. E. Evans, T. Pan, and C. He, "Dynamic RNA modifications in gene expression regulation," *Cell*, vol. 169, no. 7, pp. 1187–1200, 2017.

[7] A. Kahles, C. S. Ong, Y. Zhong, and G. Rätsch, "SplAdder: Identification, quantification and testing of alternative splicing events from RNA-seq data," *Bioinformatics*, vol. 32, no. 12, pp. 1840–1847, 2016.

[8] A. D. Tang *et al.*, "Full-length transcript characterization of SF3B1 mutation in chronic lymphocytic leukemia reveals downregulation of retained introns," *Nature Communications*, vol. 11, no. 1, p. 1438, 2020, doi: 10.1038/s41467-020-15171-6.

[9] G. Pertea and M. Pertea, "GFF utilities: GffRead and GffCompare," *F1000Research*, vol. 9, 2020.