Supplemental Code – Stemerdink et al., *Deciphering the largest disease-associated transcript isoforms in the human neural retina with advanced long-read sequencing approaches*

Analysis of long-read sequencing data with IsoQuant

```
# Dataset 1

# Analysis per sample
ccs subreads.bam ccs.bam  --min-rq 0.99 --num-threads 20 --maxLength=25000
--min-passes 3
lima ccs.bam primers.fasta fl.bam --isoseq -j 20 --peek-guess
isoseq3 refine fl.NEB_5p--NEB_Clontech_3p.bam primers.fasta flnc.bam -j 20
--require-polya
bam2fastq -o flnc.fastq flnc.bam
minimap2 -t 20 -ax splice -uf --secondary=no -C5 -O6,24 -B4 --MD hg38.fa
flnc.fastq > reads.sam
samtools sort -O sam -o aln.sorted.sam -@ 20 reads.sam
samtools view -b -o aln.sorted.bam aln.sorted.sam
samtools index -b -@ 20 aln.sorted.bam

# Analysis of all samples together (file names included in files.txt)
isoquant.py \
--reference hg38.fa  \
--genedb gencode.v39.primary_assembly.annotation.gtf \
--complete_genedb \
--bam_list files.txt \
--read_group file_name \
--data_type pacbio_ccs \
--fl_data \
--sqanti_output \
--count_exons \
--threads 20 \
--model_construction_strategy fl_pacbio \
-o dataset1_ccs3/isoquant \
--check_canonical \
--transcript_quantification unique_only \
--gene_quantification unique_only \
--splice_correction_strategy default_pacbio \
--force

# Dataset 2 - Same analysis as dataset 1

# Dataset 3 - Same analysis as dataset 1 except for CCS
ccs subreads.bamccs.bam  --min-rq 0.8 --num-threads 20 --maxLength=30000 --
min-passes 0
```

The following to uses output generated by the code on the previous page. With exception of Figure 2A, all code used to create the figures uses output generated by the IsoQuant algorithm.

Sample 1 (s1) -> human retina regular library prep #1
Sample 2 (s2) -> human retina regular library prep #2
Sample 3 (s3) -> human retina regular library prep #3
Sample 4 (s4) -> human retina long library prep, retina #2

Dataset 1 (CCS3) contains sample 1, 2 and 3 (sometimes referred to as s1-3)
Dataset 2 (CCS3) contains sample 1, 2, 3 and 4 (sometimes referred to as s1-4)
Dataset 3 (CCS0) contains sample 4 and is not analysed with code.

Figure 2A – read length distribution:

```
library(data.table)

#Sample 1 (human retina regular library prep #1)
#Read distribution in 500nt bins
s1_lenghts <- fread("ccs_length_sample1.txt")

group_bin.1 <- s1_lenghts %>%
  group_by(group = cut(V2, breaks = seq(0, max(V2), 500))) %>%
  count()

#calculate mean read length ± SD
mean(s1_lenghts$V2)
sd(s1_lenghts$V2)


#Sample 2 (human retina regular library prep #2)
#Read distribution in 500nt bins
s2_lenghts <- fread("ccs_length_sample2.txt")

group_bin.2 <- s2_lenghts %>%
  group_by(group = cut(V2, breaks = seq(0, max(V2), 500))) %>%
  count()

#calculate mean read length ± SD
mean(s2_lenghts$V2)
sd(s2_lenghts$V2)


#Sample 3 (human retina regular library prep #3)
#Read distribution in 500nt bins
s3_lenghts <- fread("ccs_length_sample3.txt")

group_bin.3 <- s3_lenghts %>%
  group_by(group = cut(V2, breaks = seq(0, max(V2), 500))) %>%
  count()

#calculate mean read length ± SD
mean(s3_lenghts$V2)
sd(s3_lenghts$V2)


#Sample 4 (human retina long library prep, retina #2)
#Read distribution in 500nt bins
s4_lenghts <- fread("ccs_length_sample2_long.txt")

group_bin.2long <- s4_lenghts %>%
  group_by(group = cut(V2, breaks = seq(0, max(V2), 500))) %>%
  count()

#calculate mean read length ± SD
mean(s4_lenghts$V2)
sd(s4_lenghts$V2)
```

Figure 2B – reads per USH gene

```
#Load packages
library(dplyr)
library(data.table)
library(splitstackshape)


#Read full data file containing read assignments from dataset 2.
s1_4_sorted_gene_assignments <- fread("00_aln.sorted.read_assignments.tsv")

# change col name to "run"
colnames(s1_4_sorted_gene_assignments)[1] ="run"

# filter for target genes (featureID)
USHs1_4_sorted_gene_assignments <- s1_4_sorted_gene_assignments %>%
dplyr::filter(gene_id %in% c("ENSG00000137474.23", "ENSG00000006611.17",
"ENSG00000107736.22", "ENSG00000150275.20", "ENSG00000182040.9",
"ENSG00000136425.14", "ENSG00000042781.14", "ENSG00000164199.18",
"ENSG00000095397.16", "ENSG00000163646.12", "ENSG00000141337.13"))

#save the file as backup (the original read assignment file is too large).
File is uploaded in Data folder
write.xlsx(USHs1_4_sorted_gene_assignments,
"USHs1_4_sorted_gene_assignments.xlsx")
USHs1_4_sorted_gene_assignments <-
read_xlsx("USHs1_4_sorted_gene_assignments.xlsx")


#rename ENSG codes to gene name (number by USH subtype)
USHs1_4_sorted_gene_assignments$gene_id[USHs1_4_sorted_gene_assignments$gen
e_id=="ENSG00000137474.23"]<-"01_MYO7A"
USHs1_4_sorted_gene_assignments$gene_id[USHs1_4_sorted_gene_assignments$gen
e_id=="ENSG00000006611.17"]<-"02_USH1C"
USHs1_4_sorted_gene_assignments$gene_id[USHs1_4_sorted_gene_assignments$gen
e_id=="ENSG00000107736.22"]<-"03_CDH23"
USHs1_4_sorted_gene_assignments$gene_id[USHs1_4_sorted_gene_assignments$gen
e_id=="ENSG00000150275.20"]<-"04_PCDH15"
USHs1_4_sorted_gene_assignments$gene_id[USHs1_4_sorted_gene_assignments$gen
e_id=="ENSG00000182040.9"]<-"05_SANS"
USHs1_4_sorted_gene_assignments$gene_id[USHs1_4_sorted_gene_assignments$gen
e_id=="ENSG00000136425.14"]<-"06_CIB2"
USHs1_4_sorted_gene_assignments$gene_id[USHs1_4_sorted_gene_assignments$gen
e_id=="ENSG00000042781.14"]<-"07_USH2A"
USHs1_4_sorted_gene_assignments$gene_id[USHs1_4_sorted_gene_assignments$gen
e_id=="ENSG00000164199.18"]<-"08_ADGRV1"
USHs1_4_sorted_gene_assignments$gene_id[USHs1_4_sorted_gene_assignments$gen
e_id=="ENSG00000095397.16"]<-"09_WHRN"
USHs1_4_sorted_gene_assignments$gene_id[USHs1_4_sorted_gene_assignments$gen
e_id=="ENSG00000163646.12"]<-"10_CLRN1"
USHs1_4_sorted_gene_assignments$gene_id[USHs1_4_sorted_gene_assignments$gen
e_id=="ENSG00000141337.13"]<-"11_ARSG"


#splice colum with to filter for structural categories and runs
s1_4.df.split2 <- cSplit(USHs1_4_sorted_gene_assignments, "structural_cat",
"=", direction = "wide")
s1_4.df.split3 <- cSplit(s1_4.df.split2, "run", "/", direction = "wide")


#rename structural categories to short name (for graph)
s1_4.df.split3$structural_cat_2[s1_4.df.split3$structural_cat_2=="full_spli
ce_match"]<-"05_FSM"
```

```
s1_4.df.split3$structural_cat_2[s1_4.df.split3$structural_cat_2=="incomplet
e_splice_match"]<-"04_ISM"
s1_4.df.split3$structural_cat_2[s1_4.df.split3$structural_cat_2=="novel_in_
catalog"]<-"03_NIC"
s1_4.df.split3$structural_cat_2[s1_4.df.split3$structural_cat_2=="novel_not
_in_catalog"]<-"02_NNIC"
s1_4.df.split3$structural_cat_2[s1_4.df.split3$structural_cat_2=="genic"]<-
"01_genic"

#filter for the different runs and create individual dfs containing
structural category counts per gene
sample1 <- s1_4.df.split3 %>% dplyr::filter(run_1 %in%
c("m64167e_210819_012015"))
sample1_s <- sample1 %>%
  group_by(gene_id, structural_cat_2) %>%
  summarise(n = n())

sample2 <- s1_4.df.split3 %>% dplyr::filter(run_1 %in%
c("m64167e_210902_121011"))
sample2_s <- sample2 %>%
  group_by(gene_id, structural_cat_2) %>%
  summarise(n = n())

sample3 <- s1_4.df.split3 %>% dplyr::filter(run_1 %in%
c("m64167e_210903_121024"))
sample3_s <- sample3 %>%
  group_by(gene_id, structural_cat_2) %>%
  summarise(n = n())

sample4 <- s1_4.df.split3 %>% dplyr::filter(run_1 %in%
c("m64037e_211216_160915"))
sample4_s <- sample4 %>%
  group_by(gene_id, structural_cat_2) %>%
  summarise(n = n())


#to obtain read transcript read counts for figure 2C, the sum of FSM, ISM,
NIC and NNIC was calculated. Reads assigned genic were excluded. Plot was
made with Graphpad Prism.
```

Figure 2C – quantification of intron retention events

```
library(dplyr)
library(data.table)
library(splitstackshape)

#Read dataset 1 read assignment file.
s1_3_sorted_gene_assignments <- fread("00_aln.sorted.read_assignments.tsv")
colnames(s1_3_sorted_gene_assignments)[1] ="run"
s1_3_split <- cSplit(s1_3_sorted_gene_assignments, "run", "/", direction =
"wide")

#Read dataset 2 read assignment file.
s1_4_sorted_gene_assignments <- fread("00_aln.sorted.read_assignments.tsv")
colnames(s1_4_sorted_gene_assignments)[1] ="run"
s1_4_split <- cSplit(s1_4_sorted_gene_assignments, "run", "/", direction =
"wide")

#Produce and save individual datasets (because the complete files take long
to load)

s1_sorted_gene_assigments <- dplyr::filter(s1_3_sorted_gene_assignments,
grepl("m64167e_210819_012015", run))
s2_sorted_gene_assigments <- dplyr::filter(s1_3_sorted_gene_assignments,
grepl("m64167e_210902_121011", run))
s3_sorted_gene_assigments <- dplyr::filter(s1_3_sorted_gene_assignments,
grepl("m64167e_210903_121024", run))
s4_sorted_gene_assigments <- dplyr::filter(s1_4_sorted_gene_assignments,
grepl("m64037e_211216_160915", run))

write_csv(s1_sorted_gene_assigments, "s1_sorted_gene_assigments.csv")
write_csv(s2_sorted_gene_assigments, "s2_sorted_gene_assigments.csv")
write_csv(s3_sorted_gene_assigments, "s3_sorted_gene_assigments.csv")
write_csv(s4_sorted_gene_assigments, "s4_sorted_gene_assigments.csv")

#read individual datasets
s1_sorted_gene_assigments <- fread("s1_sorted_gene_assigments.csv")
s2_sorted_gene_assigments <- fread("s2_sorted_gene_assigments.csv")
s3_sorted_gene_assigments <- fread("s3_sorted_gene_assigments.csv")
s4_sorted_gene_assigments <- fread("s4_sorted_gene_assigments.csv")


#Grab, count intron retentions:

#sample 1
intron_ret_s1 <- dplyr::filter(s1_sorted_gene_assigments,
grepl("intron_retention", assignment_events))
sum_irs1 <- intron_ret_s1 %>%
  summarise(n = n())
sum_s1tot <- s1_sorted_gene_assigments %>%
  summarise(n = n())
sum_irs1 / sum_s1tot * 100  #to get percentage

#sample 2
intron_ret_s2 <- dplyr::filter(s2_sorted_gene_assigments,
grepl("intron_retention", assignment_events))
sum_irs2 <- intron_ret_s2 %>%
  summarise(n = n())
sum_s2tot <- s2_sorted_gene_assigments %>%
  summarise(n = n())
sum_irs2 / sum_s2tot * 100  #to get percentage

#sample 3
intron_ret_s3 <- dplyr::filter(s3_sorted_gene_assigments,
grepl("intron_retention", assignment_events))
```

```
sum_irs3 <- intron_ret_s3 %>%
  summarise(n = n())
sum_s3tot <- s3_sorted_gene_assigments %>%
  summarise(n = n())
sum_irs3 / sum_s3tot * 100  #to get percentage

#sample 4
intron_ret_s4 <- dplyr::filter(s4_sorted_gene_assigments,
grepl("intron_retention", assignment_events))
sum_irs4 <- intron_ret_s4 %>%
  summarise(n = n())
sum_s4tot <- s4_sorted_gene_assigments %>%
  summarise(n = n())
sum_irs4 / sum_s4tot * 100  #to get percentage
```

Figure 3A – GGtranscript *MYO7A* plot

```
# Packages needed:
# dplyr, ggplot2, ggtranscript, rtracklayer

# Load packages
library(dplyr)
library(ggplot2)
library(ggtranscript)
library(rtracklayer)

setwd("//")

#####  Visualize from data GTF  #################

# Import Gencode reference file into a temporary directory
gencode_path <- file("gencode.v45.annotation.gtf")
gencode <- rtracklayer::import(gencode_path)
gencode <- gencode %>% dplyr::as_tibble()

# Import dataset 1 isoQuant track into a temporary directory
s1_3_isoQ_path <- file.path("00_aln.sorted.transcript_models.gtf")
s1_3_isoQ <- rtracklayer::import(s1_3_isoQ_path)
s1_3_isoQ <- s1_3_isoQ %>% dplyr::as_tibble()

#Renome Gene_ID column to gene_name
colnames(s1_3_isoQ)[10] ="gene_name"


# filter GTF files for the gene of interest ("MYO7A" is used for the
gencode dataset, "ENSG00000137474.23" to filter in the isoquant data)
gene_of_interest <- "ENSG00000137474.23"

refseq_gencode <- gencode %>%   # creating a dataset to plot gencode
reference isoform
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest)

MYO7A_CCS3 <- s1_3_isoQ %>%    # creating a dataset to plot isoquant
isoforms
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest)

refseq_gilmore <- gencode %>%   # creating a dataset to plot the transcript
isoforms proposed by Gilmore et al (2023) - based on gencode data)
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest)

refseq_msHC <- s1_3_isoQ %>%   # creating a dataset to plot the orthologous
transcript isoforms of mouse auditory hair cells as published by Li et al
(2020) - based on isoquant data)
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest)

#Select MANE select/clinical as reference isoforms from Gencode dataset
#rename ENSTXXX transcripts in dataset to prevent redundant names.
refseq_gencode_clean <- refseq_gencode %>% dplyr::filter(transcript_id %in%
c("ENST00000409709.9"))
refseq_gencode_clean$origin <- "gencode"
refseq_gencode_clean$transcript_id[refseq_gencode_clean$transcript_id=="ENS
T00000409709.9"]<-"1_Gencode_ENST00000409709.9"
```

```
MYO7A_CCS3$origin <- "1-3 CCS3"
MYO7A_CCS3$transcript_id[MYO7A_CCS3$transcript_id=="ENST00000409709.9"]<-
"tAtlas_ENST00000409709.9"
MYO7A_CCS3$transcript_id[MYO7A_CCS3$transcript_id=="ENST00000458637.6"]<-
"tAtlas_ENST00000458637.6"

# selection of isoforms discussed in Gilmore et al from the Gencode
dataset. Gilmore 2023 refers to Weil 1996 for original identification of
isoforms.
# Gilmore mentions 2 variants of full lenght MYO7A -> have to assume that
IF1 and IF2 are annotated on ENST00000409709.9, implying that the short IF2
= ENST00000458637.6
# rename ENSTXXX transcripts in dataset to prevent redundant names.
refseq_gilmore <- refseq_gilmore %>% dplyr::filter(transcript_id %in%
c("ENST00000409709.9","ENST00000458637.6"))
refseq_gilmore$origin <- "gilmore"
refseq_gilmore$transcript_id[refseq_gilmore$transcript_id=="ENST00000409709
.9"]<-"Gilmore IF1"
refseq_gilmore$transcript_id[refseq_gilmore$transcript_id=="ENST00000458637
.6"]<-"Gilmore IF2"

# selection of mouse hair cell (msHC) isoforms discussed in Li et al 2020.
# orthologous transcripts are ENST00000409709.9 (refseq) and
transcript20052.chr11.nic take from the isoquant dataset
# We assume that the mouse short orthologue has the same 3' UTR as the long
orthologue. Hence, we reduce the UTR lenght of the
transcript20052.chr11.nic to produce the transcript for the figure.
# Rename ENSTXXX transcripts in dataset to prevent redundant names.
refseq_msHC <- refseq_msHC %>% dplyr::filter(transcript_id %in%
c("ENST00000409709.9","transcript20052.chr11.nic"))
refseq_msHC$origin <- "msHC"
refseq_msHC$transcript_id[refseq_msHC$transcript_id=="ENST00000409709.9"]<-
"Myo7a-C orthologue"
refseq_msHC$transcript_id[refseq_msHC$transcript_id=="transcript20052.chr11
.nic"]<-"Myo7a-S orthologue"
refseq_msHC[99,3] = 77215241
refseq_msHC[99,4] = 635


# combine df_columns for figure
MYO7A_fig_full <- bind_rows(refseq_gencode_clean, refseq_gilmore,
refseq_msHC, MYO7A_CCS3 )


# extract the required annotation columns
MYO7A_fig <- MYO7A_fig_full %>%
  dplyr::select(
    seqnames,
    start,
    end,
    strand,
    type,
    gene_name,
    transcript_type,
    transcript_id,
    tag,
    origin)
```

```
#####  Visualize transcript ################


# obtain cds
MYO7A_cds <- MYO7A_fig %>% dplyr::filter(type == "CDS")

# obtain exons
MYO7A_exons <- MYO7A_fig %>% dplyr::filter(type == "exon")

#  Reduce intron length

MYO7A_rescaled <- shorten_gaps(
  exons = MYO7A_exons,
  introns = to_intron(MYO7A_exons, "transcript_id"),
  group_var = "transcript_id", target_gap_width =50L)


# shorten_gaps() returns exons and introns all in one data.frame()
# let's split these for plotting
MYO7A_rescaled_exons <- MYO7A_rescaled %>% dplyr::filter(type == "exon")
MYO7A_rescaled_introns <- MYO7A_rescaled %>% dplyr::filter(type ==
"intron")

MYO7A_rescaled_exons %>%
  ggplot(aes(
    xstart = start,
    xend = end,
    y = transcript_id,
  ))  +
  geom_range(
    aes(fill = origin)
  ) + scale_fill_manual(values=c("#F8766D", "#7CAE00", "#00BFC4",
"#999999")) +
  geom_intron(
    data = MYO7A_rescaled_introns,
    aes(strand = strand),
    arrow.min.intron.length = 700
  ) +
  theme_classic(
  ) +
  scale_y_discrete(limits = rev)


ggsave("MYO7A_figure.pdf”)


# further processing of the plot was manually in adobe illustrator
```

Figure 4A – GGtranscript *WHRN* plot

```
# Packages needed:
# dplyr, ggplot2, ggtranscript, rtracklayer

# Load packages
library(dplyr)
library(ggplot2)
library(ggtranscript)
library(rtracklayer)

setwd("//")


#####  Visualize from data GTF  ################

# Import Gencode reference file into a temporary directory
gencode_path <- file("gencode.v45.annotation.gtf")
gencode <- rtracklayer::import(gencode_path)
gencode <- gencode %>% dplyr::as_tibble()

# Import dataset 1 isoQuant track into a temporary directory
s1_3_isoQ_path <- file.path("00_aln.sorted.transcript_models.gtf")
s1_3_isoQ <- rtracklayer::import(s1_3_isoQ_path)
s1_3_isoQ <- s1_3_isoQ %>% dplyr::as_tibble()

#Renome Gene_ID column to gene_name
colnames(s1_3_isoQ)[10] ="gene_name"


# filter GTF files for the gene of interest
gene_of_interest <- "WHRN"       #  "WHRN" in Gencode GTF &
"ENSG00000095397.16"   in IsoQuant GTF

refseq_gencode <- gencode %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest)

WHRN_CCS3 <- s1_3_isoQ %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest)


#Select MANE select/clinical REFSEQs, add origin column & rename ENST
(proir to making GIO_combi df, identical ENST in multiple datasets causes
an error)
refseq_gencode_clean <- refseq_gencode %>% dplyr::filter(transcript_id %in%
c("ENST00000362057.4"))
refseq_gencode_clean$origin <- "gencode"
refseq_gencode_clean$transcript_id[refseq_gencode_clean$transcript_id=="ENS
T00000362057.4"]<-"01_Gencode_ENST00000362057.4"

#Select validated retinal isoforms for Whirlin from the Gencode dataset
(selection based on van Wijk et al, 2006)
WHRN_Wijk2006 <- refseq_gencode %>% dplyr::filter(transcript_id %in%
c("ENST00000362057.4", "ENST00000674048.1"))
WHRN_Wijk2006$origin <- "WHRN_human retina Wijk2006"
WHRN_Wijk2006$transcript_id[WHRN_Wijk2006$transcript_id=="ENST00000362057.4
"]<-"02_WHRN-FL_ENST00000362057.4"
WHRN_Wijk2006$transcript_id[WHRN_Wijk2006$transcript_id=="ENST00000674048.1
"]<-"03_WHRN-C_(S2)_ENST00000674048.1"
```

```
#Select proposed orthologuous retinal isoforms for whirlin from the Gencode
dataset (based on published mouse data, Mburu et al., 2003; Belyantseva et
al., 2005; Ebrahim et al., 2016)
WHRN_proposed <- refseq_gencode %>% dplyr::filter(transcript_id %in%
c("ENST00000362057.4", "ENST00000674048.1"))
WHRN_proposed$origin <- "WHRN_mouse orthologs"
WHRN_proposed$transcript_id[WHRN_proposed$transcript_id=="ENST00000362057.4
"]<-"10_WHRN-FL_ENST00000362057.4"
WHRN_proposed$transcript_id[WHRN_proposed$transcript_id=="ENST00000674048.1
"]<-"11_WHRN-C_(S2)_ENST00000674048.1"

#IsoQ data s1-3 rename ENST and add origin column
WHRN_CCS3$origin <- "1-3 CCS3"
WHRN_CCS3$transcript_id[WHRN_CCS3$transcript_id=="ENST00000265134.10"]<-
"t3Atlas_ENST00000265134.10"
WHRN_CCS3$transcript_id[WHRN_CCS3$transcript_id=="ENST00000362057.4"]<-
"t1Atlas_ENST00000362057.4"
WHRN_CCS3$transcript_id[WHRN_CCS3$transcript_id=="ENST00000374057.3"]<-
"t2Atlas_ENST00000374057.3"


# combine df_columns for figure
WHRN_fig_full <- bind_rows(refseq_gencode_clean, WHRN_Wijk2006, WHRN_CCS3,
WHRN_proposed )


# extract the required annotation columns
WHRN_fig <- WHRN_fig_full %>%
  dplyr::select(
    seqnames,
    start,
    end,
    strand,
    type,
    gene_name,
    transcript_type,
    transcript_id,
    tag,
    origin)


#####  Visualize transcripts  #################

# obtain exons
GOI_exons <- WHRN_fig %>% dplyr::filter(type == "exon")


# Reduce intron length
GOI_rescaled <- shorten_gaps(
  exons = GOI_exons,
  introns = to_intron(GOI_exons, "transcript_id"),
  group_var = "transcript_id", target_gap_width =50L)


# shorten_gaps() returns exons and introns all in one data.frame()
# let's split these for plotting
GOI_rescaled_exons <- GOI_rescaled %>% dplyr::filter(type == "exon")
GOI_rescaled_introns <- GOI_rescaled %>% dplyr::filter(type == "intron")

GOI_rescaled_exons %>%
  ggplot(aes(
    xstart = start,
    xend = end,
    y = transcript_id,
  ))  +
```

```
  scale_y_discrete(limits = rev)  +         #reverts y axis
  coord_cartesian(xlim = c(max(GOI_rescaled_exons$end),
(min(GOI_rescaled_exons$start))))+       #invert orientation

  geom_range(
    aes(fill = origin)
  ) + scale_fill_manual(values=c("#00BFC4", "#7CAE00","#F8766D",
"#999999")) +
  geom_intron(
    data = GOI_rescaled_introns,
    aes(strand = strand),
    arrow.min.intron.length = 700
  ) +
  theme_classic(
  )

ggsave("WHRN_figure.pdf")


# additional annotations were made manually in Adobe Illustrator.
#The mouse ortholog of FL whirlin with exon 7B was copied manually from
transcript.nnic13724.chr9.
#The mouse ortholog of whirlin N was produced by copying FL transcript and
removing intron 4 and beyond
```

Figure 5A – GGtranscript *USH2A* IsoQuant plot

```
# Packages needed:
# dplyr, ggplot2, ggtranscript, rtracklayer

# Load packages
library(dplyr)
library(ggplot2)
library(ggtranscript)
library(rtracklayer)

setwd("//")



#####  Visualize from data GTF  #################

# Import Gencode reference file into a temporary directory
gencode_path <- file("gencode.v45.annotation.gtf")
gencode <- rtracklayer::import(gencode_path)
gencode <- gencode %>% dplyr::as_tibble()

# Import dataset 1 isoQuant track into a temporary directory
s1_3_isoQ_path <- file.path("00_aln.sorted.transcript_models.gtf")
s1_3_isoQ <- rtracklayer::import(s1_3_isoQ_path)
s1_3_isoQ <- s1_3_isoQ %>% dplyr::as_tibble()

#Renome Gene_ID column to gene_name
colnames(s1_3_isoQ)[10] ="gene_name"


# filter GTF files for the gene of interest
gene_of_interest <- "USH2A"       #  "USH2A" in Gencode GTF &
"ENSG00000042781.14" in Isoquant GTF

refseq_gencode <- gencode %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest)

USH2A_CCS3 <- s1_3_isoQ %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest)

human_retina <- gencode %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest)



# change df_columns for figure

#Select MANE select, add origin column & rename ENST (proir to making
GIO_combi, identical ENST in multiple datasets causes an error)
refseq_gencode_clean <- refseq_gencode %>% dplyr::filter(transcript_id %in%
c("ENST00000307340.8"))
refseq_gencode_clean$origin <- "gencode"
refseq_gencode_clean$transcript_id[refseq_gencode_clean$transcript_id=="ENS
T00000307340.8"]<-"1_Gencode_ENST00000307340.8"
```

```
#IsoQ data s1-3 rename ENST and add origin column
USH2A_CCS3$origin <- "1-3 CCS3"
USH2A_CCS3$transcript_id[USH2A_CCS3$transcript_id=="ENST00000307340.8"]<-
"tAtlas_ENST00000307340.8"


# Human_retina isoform based on van Wijk et al 2003
# to produce these transcript in the figure, they are taken from the
Gencode GTF and add origin column & rename ENST (proir to making GIO_combi,
identical ENST in multiple datasets causes an error)
human_retina <- human_retina %>% dplyr::filter(transcript_id %in%
c("ENST00000307340.8","ENST00000366942.3"))
human_retina$origin <- "retina"
human_retina$transcript_id[human_retina$transcript_id=="ENST00000366942.3"]
<-"2_USH2A_isoA"
human_retina$transcript_id[human_retina$transcript_id=="ENST00000307340.8"]
<-"3_USH2A_isoB"

# combine df_columns for figure
USH2A_fig_full <- bind_rows(refseq_gencode_clean, USH2A_CCS3, human_retina)


# extract the required annotation columns
USH2A_fig <- USH2A_fig_full %>%
  dplyr::select(
    seqnames,
    start,
    end,
    strand,
    type,
    gene_name,
    transcript_type,
    transcript_id,
    tag,
    origin)



#####  Visualize transcripts #################


# obtain cds
GOI_cds <- USH2A_fig %>% dplyr::filter(type == "CDS")

# obtain exons
GOI_exons <- USH2A_fig %>% dplyr::filter(type == "exon")


#  Reduce intron length
GOI_rescaled <- shorten_gaps(
  exons = GOI_exons,
  introns = to_intron(GOI_exons, "transcript_id"),
  group_var = "transcript_id", target_gap_width =100L)


# shorten_gaps() returns exons and introns all in one data.frame()
# let's split these for plotting
GOI_rescaled_exons <- GOI_rescaled %>% dplyr::filter(type == "exon")
GOI_rescaled_introns <- GOI_rescaled %>% dplyr::filter(type == "intron")

GOI_rescaled_exons %>%
  ggplot(aes(
    xstart = start,
    xend = end,
    y = transcript_id,
```

```
  ))  +
  scale_y_discrete(limits = rev)  +       #reverts y axis
  coord_cartesian(xlim = c(max(GOI_rescaled_exons$end),
(min(GOI_rescaled_exons$start))))+       #invert orientation

  geom_range(
    aes(fill = origin)
  ) +scale_fill_manual(values=c("#F8766D", "#7CAE00", "#00BFC4")) +

  geom_intron(
    data = GOI_rescaled_introns,
    aes(strand = strand),
    arrow.min.intron.length = 700
  ) +
  theme_classic(
  )

ggsave("USH2A_figure part1.pdf", width = 40, height = 28, units = "cm")

# additional annotations were made manually
```

Figure 5B – GGtranscript *USH2A* manual curation and Samplix plot

```
# Packages needed:
# dplyr, ggplot2, ggtranscript, rtracklayer

# Load packages
library(dplyr)
library(ggplot2)
library(ggtranscript)
library(rtracklayer)

setwd("//")


#####  Visualize from data GTF  ################

# Import Gencode reference file into a temporary directory
gencode_path <- file("gencode.v45.annotation.gtf")
gencode <- rtracklayer::import(gencode_path)
gencode <- gencode %>% dplyr::as_tibble()

# Import dataset 1 isoQuant track into a temporary directory
s1_3_isoQ_path <- file.path("00_aln.sorted.transcript_models.gtf")
s1_3_isoQ <- rtracklayer::import(s1_3_isoQ_path)
s1_3_isoQ <- s1_3_isoQ %>% dplyr::as_tibble()

#Renome Gene_ID column to gene_name
colnames(s1_3_isoQ)[10] ="gene_name"


# filter GTF files for the gene of interest
gene_of_interest <- "USH2A"       #  USH2A  ENSG00000042781.14

refseq_gencode <- gencode %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest
  )

USH2A_CCS3 <- s1_3_isoQ %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest
  )

human_retina <- gencode %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest)


# change df_columns for figure

#Select MANE select, add origin column & rename ENST (proir to making
GIO_combi, identical ENST in multiple datasets causes an error)
refseq_gencode_clean <- refseq_gencode %>% dplyr::filter(transcript_id %in%
c("ENST00000307340.8"))
refseq_gencode_clean$origin <- "gencode"
refseq_gencode_clean$transcript_id[refseq_gencode_clean$transcript_id=="ENS
T00000307340.8"]<-"1_Gencode_ENST00000307340.8"


#Isoform predictions based on manual curation of BAM files (cryptic exon
track)
```

```
USH2A_BAMp <- read_xlsx("USH genes/USH2A/USH2A_i20 events.xlsx",
"USH2A_i20")
USH2A_BAMp$origin <- "BAM"

#Annotated REFSEQs based on BAM. selected transcripts taken from gencode.
USH2A_annotated <- refseq_gencode %>% dplyr::filter(transcript_id %in%
c("ENST00000307340.8","ENST00000366942.3"))
USH2A_annotated$transcript_id[USH2A_annotated$transcript_id=="ENST000003073
40.8"]<-"2_ENST00000307340.8"
USH2A_annotated$transcript_id[USH2A_annotated$transcript_id=="ENST000003669
42.3"]<-"3_ENST00000366942.3"
USH2A_annotated$origin <- "BAM"

#Isoform predictions based on manual curation of BAM files (5' variation)
USH2A_BAM5 <- read_xlsx("USH genes/USH2A/USH2A_5-variation.xlsx", "Sheet1")
USH2A_BAM5$origin <- "BAM"


#Visualize Samplix FL isoform, taken representative isoform from gencode.
samplix <- refseq_gencode %>% dplyr::filter(transcript_id %in%
c("ENST00000307340.8"))
samplix$origin <- "samplix"
samplix$transcript_id[samplix$transcript_id=="ENST00000307340.8"]<-
"ZZ_samplix_ENST00000307340.8"


# combine df_columns for figure
USH2A_fig_full <- bind_rows(refseq_gencode_clean, USH2A_annotated,
USH2A_BAM5, USH2A_BAMp, samplix)


# extract the required annotation columns
USH2A_fig <- USH2A_fig_full %>%
  dplyr::select(
    seqnames,
    start,
    end,
    strand,
    type,
    gene_name,
    transcript_type,
    transcript_id,
    tag,
    origin)



#####  Visualize transcript type  #################


# obtain cds
GOI_cds <- USH2A_fig %>% dplyr::filter(type == "CDS")

# obtain exons
GOI_exons <- USH2A_fig %>% dplyr::filter(type == "exon")

#  Reduce intron length
GOI_rescaled <- shorten_gaps(
  exons = GOI_exons,
  introns = to_intron(GOI_exons, "transcript_id"),
  group_var = "transcript_id", target_gap_width =150L)


# shorten_gaps() returns exons and introns all in one data.frame()
# let's split these for plotting
```

```
GOI_rescaled_exons <- GOI_rescaled %>% dplyr::filter(type == "exon")
GOI_rescaled_introns <- GOI_rescaled %>% dplyr::filter(type == "intron")

GOI_rescaled_exons %>%
  ggplot(aes(
    xstart = start,
    xend = end,
    y = transcript_id,
  ))  +
  scale_y_discrete(limits = rev) +                    #reverts y axis
  coord_cartesian(xlim = c(max(GOI_rescaled_exons$end),
(min(GOI_rescaled_exons$start))))+       #invert orientation

  geom_range(
    aes(fill = origin)
  ) +scale_fill_manual(values=c("#CC0033", "#7CAE00", "#E68613")) +

  geom_intron(
    data = GOI_rescaled_introns,
    aes(strand = strand),
    arrow.min.intron.length = 700
  ) +
  theme_classic(
  )

ggsave("USH2A_figure part2.pdf", width = 40, height = 12, units = "cm")

# save on fixed width of 40 with height of 2 per transcript isoform
# xls files were made in which refseq data was manually adjusted to match
the information obtained with manual curation and Samplix. Files are
deposited on github:
https://github.com/erikdevrieze/USH_retina/tree/main/Figures/Data
# additional annotations were made manually in Adobe Illustrator.
```

Figure 6 – GGtranscript *ADGRV1* manual curation and Samplix plot

```
# Packages needed:
# dplyr, ggplot2, ggtranscript, rtracklayer

# Load packages
library(dplyr)
library(ggplot2)
library(ggtranscript)
library(rtracklayer)

setwd("//")


#####  Visualize from data GTF  ################

# Import Gencode reference file into a temporary directory
gencode_path <- file("gencode.v45.annotation.gtf")
gencode <- rtracklayer::import(gencode_path)
gencode <- gencode %>% dplyr::as_tibble()


# filter GTF files for the gene of interest
gene_of_interest <- "ADGRV1"

refseq_gencode <- gencode %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest )


# change df_columns for figure

#Select MANE select refseq, add origin column & rename ENST (proir to
making GIO_combi, identical ENST in multiple datasets causes an error)
refseq_gencode_clean <- refseq_gencode   %>% dplyr::filter(transcript_id
%in% c("ENST00000405460.9"))
refseq_gencode_clean$origin <- "gencode"
refseq_gencode_clean$transcript_id[refseq_gencode_clean$transcript_id=="ENS
T00000405460.9"]<-"01_Gencode_ENST00000405460.9"

#produce proposed Samplix FL transcript
ADGRV1_samplix <- refseq_gencode   %>% dplyr::filter(transcript_id %in%
c("ENST00000405460.9"))
ADGRV1_samplix$origin <- "samplix"
ADGRV1_samplix$transcript_id[ADGRV1_samplix$transcript_id=="ENST00000405460
.9"]<-"Xsamplix_ENST00000405460.9"

#produce proposed Samplix 1e transcript. We used to manually produced xls
file containg the information on the Vlgr1e transcript isoform.
#input file is uploaded to the data folder on github
ADGRV1_samplix_e <- read.xlsx("USH
Genes/ADGRV1/ADGRV1_Samplix_Vlgr1e.xlsx", "ADGRV1_Samplix_Vlgr1e")
ADGRV1_samplix_e$origin <- "Xx_ADGRV1_samplix_e"
ADGRV1_samplix_e$transcript_id[ADGRV1_samplix_e$transcript_id=="X_ADGRV1_Sa
mplix_Vlgr1e"]<-"XX_ADGRV1_Samplix_Vlgr1e"

#produce proposed Pacbio manual curation isoforms that are not refseq (read
from xls). Similar to the Vlgr1e trnascript, we have taken the FL ADGRV1
isoform and manually annotated exon 39A in an xls file
#input file is uploaded to the data folder
Pacbio_manual1 <- read.xlsx("USH Genes/ADGRV1/ADGRV1_Full length incl
39A.xlsx", "ADGRV1_Exon39A")
Pacbio_manual1$origin <- "Pacbio_manual1"
```

```
#produce proposed Pacbio manual curation isoforms that are not refseq (read
from xls) Similar to the Vlgr1e trnascript, we have taken the FL ADGRV1
isoform and manually annotated the U2_VLGR1-c found in the BAM file.
#input file is uploaded to the data folder
Pacbio_manual2 <- read.xlsx("USH Genes/ADGRV1/ADGRV1_Pacbio_isoC.xlsx",
"U2_VLGR1-c")
Pacbio_manual2$origin <- "Pacbio_manual2"


# combine df_columns for figure
ADGRV1_fig_full <- bind_rows(refseq_gencode_clean, Pacbio_manual1,
Pacbio_manual2, ADGRV1_samplix, ADGRV1_samplix_e)


# extract the required annotation columns
ADGRV1_fig <- ADGRV1_fig_full %>%
  dplyr::select(
    seqnames,
    start,
    end,
    strand,
    type,
    gene_name,
    transcript_type,
    transcript_id,
    tag,
    origin )



#####  Visualize transcript type  #################


# obtain exons
GOI_exons <- ADGRV1_fig %>% dplyr::filter(type == "exon")

# Reduce intron length
GOI_rescaled <- shorten_gaps(
  exons = GOI_exons,
  introns = to_intron(GOI_exons, "transcript_id"),
  group_var = "transcript_id", target_gap_width =50L)


# shorten_gaps() returns exons and introns all in one data.frame()
# let's split these for plotting
GOI_rescaled_exons <- GOI_rescaled %>% dplyr::filter(type == "exon")
GOI_rescaled_introns <- GOI_rescaled %>% dplyr::filter(type == "intron")

GOI_rescaled_exons %>%
  ggplot(aes(
    xstart = start,
    xend = end,
    y = transcript_id,
  ))  +
  scale_y_discrete(limits = rev)  +           #reverts y axis


  geom_range(
    aes(fill = origin)
  ) + scale_fill_manual(values=c("#7CAE00", "#CC0033", "#CC0033",
"#E68613","#E68613")) +

  geom_intron(
    data = GOI_rescaled_introns,
```

```
    aes(strand = strand),
    arrow.min.intron.length = 700
  ) +
  theme_classic()


ggsave("ADGRV1_figure part2.pdf")

# save on fixed width of 40 with height of 2 per transcript isoform
# xls files were made in which refseq data was manually adjusted to match
the information obtained with manual curation and Samplix. Files are
deposited on github:
https://github.com/erikdevrieze/USH_retina/tree/main/Figures/Data
# additional annotations were made manually in Adobe Illustrator.
```

Figure S1 – GGtranscript *USH1C* plot

```
# Packages needed:
# dplyr, ggplot2, ggtranscript, rtracklayer

# Load packages
library(dplyr)
library(ggplot2)
library(ggtranscript)
library(rtracklayer)

setwd("//")


#####  Visualize from data GTF  #################

# Import Gencode reference file into a temporary directory
gencode_path <- file("gencode.v45.annotation.gtf")
gencode <- rtracklayer::import(gencode_path)
gencode <- gencode %>% dplyr::as_tibble()

# Import dataset 1 isoQuant track into a temporary directory
s1_3_isoQ_path <- file.path("00_aln.sorted.transcript_models.gtf")
s1_3_isoQ <- rtracklayer::import(s1_3_isoQ_path)
s1_3_isoQ <- s1_3_isoQ %>% dplyr::as_tibble()

#Renome Gene_ID column to gene_name
colnames(s1_3_isoQ)[10] ="gene_name"


# filter GTF files for the gene of interest
gene_of_interest <- "USH1C"       #  USH1C  ENSG00000006611.17

refseq_gencode <- gencode %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest)

USH1C_CCS3 <- s1_3_isoQ %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest)


# change df_columns for figure

#Select MANE select/clinical refseqs, add origin column & rename ENST
(proir to making GIO_combi, identical ENST in multiple datasets causes an
error)
refseq_gencode_clean <- refseq_gencode %>% dplyr::filter(transcript_id %in%
c("ENST00000005226.12", "ENST00000318024.9"))
refseq_gencode_clean$origin <- "gencode"
refseq_gencode_clean$transcript_id[refseq_gencode_clean$transcript_id=="ENS
T00000005226.12"]<-"01_Gencode_ENST00000005226.12"
refseq_gencode_clean$transcript_id[refseq_gencode_clean$transcript_id=="ENS
T00000318024.9"]<-"02_Gencode_ENST00000318024.9"


#Select published isoforms (nagel-Wolfrum 2023 and refs therein). Isoforms
selected from ENS transcripts that encode the proposed protein isoforms.
#Additional alternaitve splicing events are shown in literature, but appear
to be "splicing in progress". Determining events are inclusion/exclusion
middle exons, and exon 15 skipping -> FS
```

```
#add origin column & rename ENST (proir to making GIO_combi, identical ENST
in multiple datasets causes an error)
refseq_gencode_clean <- refseq_gencode %>% dplyr::filter(transcript_id %in%
c("ENST00000527720.5", "ENST00000527020.5", "ENST00000005226.12",
"ENST00000318024.9", "ENST00000526313.5"))
refseq_gencode_clean$origin <- "published"
refseq_gencode_clean$transcript_id[refseq_gencode_clean$transcript_id=="ENS
T00000527720.5"]<-"05_Harmonin_A2_ENST00000527720.5"
refseq_gencode_clean$transcript_id[refseq_gencode_clean$transcript_id=="ENS
T00000527020.5"]<-"04_Harmonin_A1_ENST00000527020.5"
refseq_gencode_clean$transcript_id[refseq_gencode_clean$transcript_id=="ENS
T00000005226.12"]<-"07_Harmonin_B_ENST00000005226.12"
refseq_gencode_clean$transcript_id[refseq_gencode_clean$transcript_id=="ENS
T00000318024.9"]<-"06_Harmonin_A3_ENST00000318024.9"
refseq_gencode_clean$transcript_id[refseq_gencode_clean$transcript_id=="ENS
T00000526313.5"]<-"08_Harmonin_C_ENST00000526313.5"

#IsoQ data s1-3 rename ENST and add origin column
USH1C_CCS3$origin <- "1-3 CCS3"
USH1C_CCS3$transcript_id[USH1C_CCS3$transcript_id=="ENST00000640109.1"]<-
"tAtlas_ENST00000640109.1"
USH1C_CCS3$transcript_id[USH1C_CCS3$transcript_id=="ENST00000640281.1"]<-
"tAtlas_ENST00000640281.1"
USH1C_CCS3$transcript_id[USH1C_CCS3$transcript_id=="ENST00000638316.1"]<-
"tAtlas_ENST00000638316.1"
USH1C_CCS3$transcript_id[USH1C_CCS3$transcript_id=="ENST00000639884.1"]<-
"tAtlas_ENST00000639884.1"


# combine df_columns for figure
USH1C_fig_full <- bind_rows(refseq_gencode_clean, USH1C_CCS3)


# extract the required annotation columns
USH1C_fig <- USH1C_fig_full %>%
  dplyr::select(
    seqnames,
    start,
    end,
    strand,
    type,
    gene_name,
    transcript_type,
    transcript_id,
    tag,
    origin )



#####  Visualize transcript type  #################

# obtain exons
GOI_exons <- USH1C_fig %>% dplyr::filter(type == "exon")

#  Reduce intron length
GOI_rescaled <- shorten_gaps(
  exons = GOI_exons,
  introns = to_intron(GOI_exons, "transcript_id"),
  group_var = "transcript_id", target_gap_width =50L)

# shorten_gaps() returns exons and introns all in one data.frame()
# let's split these for plotting
GOI_rescaled_exons <- GOI_rescaled %>% dplyr::filter(type == "exon")
GOI_rescaled_introns <- GOI_rescaled %>% dplyr::filter(type == "intron")
```

```
GOI_rescaled_exons %>%
  ggplot(aes(
    xstart = start,
    xend = end,
    y = transcript_id,
  ))  +
  scale_y_discrete(limits = rev)  +                    #reverts y axis
  coord_cartesian(xlim = c(max(GOI_rescaled_exons$end),
(min(GOI_rescaled_exons$start))))+       #invert orientation


  geom_range(
    aes(fill = origin)
  ) + scale_fill_manual(values=c("#F8766D", "#7CAE00", "#00BFC4",
"#00BFC4","#E68613", "#999999")) +
  geom_intron(
    data = GOI_rescaled_introns,
    aes(strand = strand),
    arrow.min.intron.length = 700
  ) +
  theme_classic(
  )


ggsave("USH1C_figure.pdf")

# save on fixed width of 40 with height of 2 per transcript isoform
# additional annotations were made manually in Adobe Illustrator.
```

Figure S2 – GGtranscript *CDH23* plot

```
# Packages needed:
# dplyr, ggplot2, ggtranscript, rtracklayer

# Load packages
library(dplyr)
library(ggplot2)
library(ggtranscript)
library(rtracklayer)

setwd("//")


#####  Visualize from data GTF  ################

# Import gencode reference file into a temporary directory
gencode_path <- file("gencode.v45.annotation.gtf")
gencode <- rtracklayer::import(gencode_path)
gencode <- gencode %>% dplyr::as_tibble()

# Import dataset 1 isoQuant track into a temporary directory
s1_3_isoQ_path <- file.path("00_aln.sorted.transcript_models.gtf")
s1_3_isoQ <- rtracklayer::import(s1_3_isoQ_path)
s1_3_isoQ <- s1_3_isoQ %>% dplyr::as_tibble()

#Renome Gene_ID column to gene_name
colnames(s1_3_isoQ)[10] ="gene_name"


# filter GTF files for the gene of interest
gene_of_interest <- "CDH23"       #  CDH23  ENSG00000107736.22

refseq_gencode <- gencode %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest )

CDH23_CCS3 <- s1_3_isoQ %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest )


# change df_columns for figure

#Select MANE select refseqs, add origin column & rename ENST (proir to
making GIO_combi, identical ENST in multiple datasets causes an error)
refseq_gencode_clean <- refseq_gencode %>% dplyr::filter(transcript_id %in%
c("ENST00000224721.12"))
refseq_gencode_clean$origin <- "gencode"
refseq_gencode_clean$transcript_id[refseq_gencode_clean$transcript_id=="ENS
T00000224721.12"]<-"01_Gencode_ENST00000224721.12"

#IsoQ data s1-3 rename ENST and add origin column
CDH23_CCS3$origin <- "1-3 CCS3"
CDH23_CCS3$transcript_id[CDH23_CCS3$transcript_id=="ENST00000461841.7"]<-
"tAtlas_ENST00000461841.7"
CDH23_CCS3$transcript_id[CDH23_CCS3$transcript_id=="transcript11235.chr10.n
nic"]<-"tAtlas_transcript11235.chr10.nnic"
CDH23_CCS3$transcript_id[CDH23_CCS3$transcript_id=="ENST00000461841.7"]<-
"tAtlas_ENST00000461841.7"
```

```
# combine df_columns for figure
CDH23_fig_full <- bind_rows(refseq_gencode_clean, CDH23_CCS3)


# extract the required annotation columns
CDH23_fig <- CDH23_fig_full %>%
  dplyr::select(
    seqnames,
    start,
    end,
    strand,
    type,
    gene_name,
    transcript_type,
    transcript_id,
    tag,
    origin  )


#####  Visualize transcript type  ################


# obtain exons
GOI_exons <- CDH23_fig %>% dplyr::filter(type == "exon")


#  Reduce intron length
GOI_rescaled <- shorten_gaps(
  exons = GOI_exons,
  introns = to_intron(GOI_exons, "transcript_id"),
  group_var = "transcript_id", target_gap_width =50L)


# shorten_gaps() returns exons and introns all in one data.frame()
# let's split these for plotting
GOI_rescaled_exons <- GOI_rescaled %>% dplyr::filter(type == "exon")
GOI_rescaled_introns <- GOI_rescaled %>% dplyr::filter(type == "intron")

GOI_rescaled_exons %>%
  ggplot(aes(
    xstart = start,
    xend = end,
    y = transcript_id,
  ))  +
  scale_y_discrete(limits = rev)  +            #reverts y axis


  geom_range(
    aes(fill = origin)
  ) + scale_fill_manual(values=c("#F8766D", "#7CAE00", "#00BFC4",
"#999999","#E68613", "#999999")) +
  geom_intron(
    data = GOI_rescaled_introns,
    aes(strand = strand),
    arrow.min.intron.length = 700
  ) +
  theme_classic(
  )

ggsave("Figure_ CDH23.pdf)

# save on fixed width of 40 with height of 2 per transcript isoform
# additional annotations were made manually in Adobe Illustrator.
```

Figure S3 – GGtranscript *PCDH15* plot

```
# Packages needed:
# dplyr, ggplot2, ggtranscript, rtracklayer

# Load packages
library(dplyr)
library(ggplot2)
library(ggtranscript)
library(rtracklayer)

setwd("//")


#####  Visualize from data GTF  ################

# Import Gencode reference file into a temporary directory
gencode_path <- file("gencode.v45.annotation.gtf")
gencode <- rtracklayer::import(gencode_path)
gencode <- gencode %>% dplyr::as_tibble()

# Import dataset 1 isoQuant track into a temporary directory
s1_3_isoQ_path <- file.path("00_aln.sorted.transcript_models.gtf")
s1_3_isoQ <- rtracklayer::import(s1_3_isoQ_path)
s1_3_isoQ <- s1_3_isoQ %>% dplyr::as_tibble()


#Rename Gene_ID column to gene_name
colnames(s1_3_isoQ)[10] ="gene_name"


# filter GTF files for the gene of interest
gene_of_interest <- "ENSG00000150275.20"     #  PCDH15  ENSG00000150275.20

refseq_gencode <- gencode %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest
  )

PCDH15_CCS3 <- s1_3_isoQ %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest
  )


# change df_columns for figure

#Select MANE select/clinical refseqs, add origin column & rename ENST
(proir to making GIO_combi, identical ENST in multiple datasets causes an
error)
refseq_gencode_clean <- refseq_gencode %>% dplyr::filter(transcript_id %in%
c("ENST00000320301.11", "ENST00000644397.2"))
refseq_gencode_clean$origin <- "gencode"
refseq_gencode_clean$transcript_id[refseq_gencode_clean$transcript_id=="ENS
T00000644397.2"]<-"01_Gencode_ENST00000644397.2"   # MANE select
refseq_gencode_clean$transcript_id[refseq_gencode_clean$transcript_id=="ENS
T00000320301.11"]<-"02_Gencode_ENST00000320301.11"  # MANE clinical


#Select published isoforms (Ahmed et al 2008). Isoforms selected from ENS
transcripts that contain the exons shows in the paper.
```

```
#add origin column & rename ENST (proir to making GIO_combi, identical ENST
in multiple datasets causes an error)
PCDH15_pub <- refseq_gencode %>% dplyr::filter(transcript_id %in%
c("ENST00000320301.11", "ENST00000395445.6", "ENST00000616114.4",
"ENST00000463095.2", "ENST00000373955.5"))
PCDH15_pub$origin <- "Ahmed_2008"
PCDH15_pub$transcript_id[PCDH15_pub$transcript_id=="ENST00000320301.11"]<-
"03_PCDH15_CD1_ENST00000320301.11"
PCDH15_pub$transcript_id[PCDH15_pub$transcript_id=="ENST00000395445.6"]<-
"04_PCDH15_CD2 ENST00000395445.6"
PCDH15_pub$transcript_id[PCDH15_pub$transcript_id=="ENST00000616114.4"]<-
"05_PCDH15_CD3_ENST00000616114.4"
PCDH15_pub$transcript_id[PCDH15_pub$transcript_id=="ENST00000463095.2"]<-
"06_PCDH15_Cterm_ENST00000463095.2"
PCDH15_pub$transcript_id[PCDH15_pub$transcript_id=="ENST00000373955.5"]<-
"07_PCDH15_SI_ENST00000373955.5"

#IsoQ data s1-3 rename ENST and add origin column
PCDH15_CCS3$origin <- "1-3 CCS3"
PCDH15_CCS3$transcript_id[PCDH15_CCS3$transcript_id=="ENST00000621708.4"]<-
"08_tAtlas_ENST00000621708.4"
PCDH15_CCS3$transcript_id[PCDH15_CCS3$transcript_id=="ENST00000373957.7"]<-
"08_tAtlas_ENST00000373957.7"
PCDH15_CCS3$transcript_id[PCDH15_CCS3$transcript_id=="ENST00000638316.1"]<-
"08_tAtlas_ENST00000638316.1"
PCDH15_CCS3$transcript_id[PCDH15_CCS3$transcript_id=="ENST00000639884.1"]<-
"08_tAtlas_ENST00000639884.1"

# combine df_columns for figure
PCDH15_fig_full <- bind_rows(refseq_gencode_clean, PCDH15_pub, PCDH15_CCS3)


# extract the required annotation columns
PCDH15_fig <- PCDH15_fig_full %>%
  dplyr::select(
    seqnames,
    start,
    end,
    strand,
    type,
    gene_name,
    transcript_type,
    transcript_id,
    tag,
    origin )


#####  Visualize transcript type  #################

# obtain exons
GOI_exons <- PCDH15_fig %>% dplyr::filter(type == "exon")


#  Reduce intron length
GOI_rescaled <- shorten_gaps(
  exons = GOI_exons,
  introns = to_intron(GOI_exons, "transcript_id"),
  group_var = "transcript_id", target_gap_width =50L)


# shorten_gaps() returns exons and introns all in one data.frame()
# let's split these for plotting
GOI_rescaled_exons <- GOI_rescaled %>% dplyr::filter(type == "exon")
GOI_rescaled_introns <- GOI_rescaled %>% dplyr::filter(type == "intron")
```

```
GOI_rescaled_exons %>%
  ggplot(aes(
    xstart = start,
    xend = end,
    y = transcript_id,
  ))  +
  scale_y_discrete(limits = rev)  +                      #reverts y axis
  coord_cartesian(xlim = c(max(GOI_rescaled_exons$end),
(min(GOI_rescaled_exons$start))))+       #invert orientation


  geom_range(
    aes(fill = origin)
  ) + scale_fill_manual(values=c("#F8766D", "#E68613", "#00BFC4",
"#7CAE00","#CC66CC", "#999999")) +
  geom_intron(
    data = GOI_rescaled_introns,
    aes(strand = strand),
    arrow.min.intron.length = 700
  ) +
  theme_classic(
  ) +

  ggsave("PCDH15_figure.pdf")


# save on fixed width of 40 with height of 2 per transcript isoform
# xls files were made in which refseq data was manually adjusted to match
the information obtained with manual curation and Samplix. Files are
deposited on github:
https://github.com/erikdevrieze/USH_retina/tree/main/Figures/Data
# additional annotations were made manually
```

Figure S4 – GGtranscript *SANS/USH1G* plot

```
# Packages needed:
# dplyr, ggplot2, ggtranscript, rtracklayer

# Load packages
library(dplyr)
library(ggplot2)
library(ggtranscript)
library(rtracklayer)

setwd("//")

#####  Visualize from data GTF  ################

# Import Gencode reference file into a temporary directory
gencode_path <- file("gencode.v45.annotation.gtf")
gencode <- rtracklayer::import(gencode_path)
gencode <- gencode %>% dplyr::as_tibble()

# Import dataset 1 isoQuant track into a temporary directory
s1_3_isoQ_path <- file.path("00_aln.sorted.transcript_models.gtf")
s1_3_isoQ <- rtracklayer::import(s1_3_isoQ_path)
s1_3_isoQ <- s1_3_isoQ %>% dplyr::as_tibble()

#Renome Gene_ID column to gene_name
colnames(s1_3_isoQ)[10] ="gene_name"


# filter GTF files for the gene of interest
gene_of_interest <- "USH1G"       #  USH1G  ENSG00000182040.9

refseq_gencode <- gencode %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest)

SANS_CCS3 <- s1_3_isoQ %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest)


# change df_columns for figure

#Select MANE select refseqs, add origin column & rename ENST (proir to
making GIO_combi, identical ENST in multiple datasets causes an error)
refseq_gencode_clean <- refseq_gencode #%>% dplyr::filter(transcript_id
%in% c("ENST00000614341.5"))
refseq_gencode_clean$origin <- "gencode"
refseq_gencode_clean$transcript_id[refseq_gencode_clean$transcript_id=="ENS
T00000614341.5"]<-"01_Gencode_ENST00000614341.5"

#IsoQ data s1-3 rename ENST and add origin column
SANS_CCS3$origin <- "1-3 CCS3"
SANS_CCS3$transcript_id[SANS_CCS3$transcript_id=="ENST00000614341.5"]<-
"tAtlas_ENST00000614341.5"
SANS_CCS3$transcript_id[SANS_CCS3$transcript_id=="ENST00000475158.1"]<-
"tAtlas_ENST00000475158.1"
SANS_CCS3$transcript_id[SANS_CCS3$transcript_id=="transcript11235.chr10.nni
c"]<-"tAtlas_transcript11235.chr10.nnic"
```

```
# combine df_columns for figure
SANS_fig_full <- bind_rows(refseq_gencode_clean, SANS_CCS3)


# extract the required annotation columns
SANS_fig <- SANS_fig_full %>%
  dplyr::select(
    seqnames,
    start,
    end,
    strand,
    type,
    gene_name,
    transcript_type,
    transcript_id,
    tag,
    origin)



#####  Visualize transcript type  #################


# obtain exons
GOI_exons <- SANS_fig %>% dplyr::filter(type == "exon")


#  Reduce intron length
GOI_rescaled <- shorten_gaps(
  exons = GOI_exons,
  introns = to_intron(GOI_exons, "transcript_id"),
  group_var = "transcript_id", target_gap_width =50L)


# shorten_gaps() returns exons and introns all in one data.frame()
# let's split these for plotting
GOI_rescaled_exons <- GOI_rescaled %>% dplyr::filter(type == "exon")
GOI_rescaled_introns <- GOI_rescaled %>% dplyr::filter(type == "intron")

GOI_rescaled_exons %>%
  ggplot(aes(
    xstart = start,
    xend = end,
    y = transcript_id,
  ))  +
  scale_y_discrete(limits = rev)  +                  #reverts y axis


  geom_range(
    aes(fill = origin)
  ) + scale_fill_manual(values=c("#F8766D", "#7CAE00", "#00BFC4",
"#E68613","#E68613", "#999999")) +
  geom_intron(
    data = GOI_rescaled_introns,
    aes(strand = strand),
    arrow.min.intron.length = 700
  ) +
  theme_classic()

  ggsave("SANS_figure.pdf")



# additional annotations were made manually in illustrator
```

Figure S5 – GGtranscript *CIB2* plot

```
# Packages needed:
# dplyr, ggplot2, ggtranscript, rtracklayer

# Load packages
library(dplyr)
library(ggplot2)
library(ggtranscript)
library(rtracklayer)

setwd("//")

#####  Visualize from data GTF  #################

# Import Gencode reference file into a temporary directory
gencode_path <- file("gencode.v45.annotation.gtf")
gencode <- rtracklayer::import(gencode_path)
gencode <- gencode %>% dplyr::as_tibble()

# Import dataset 1 isoQuant track into a temporary directory
s1_3_isoQ_path <- file.path("00_aln.sorted.transcript_models.gtf")
s1_3_isoQ <- rtracklayer::import(s1_3_isoQ_path)
s1_3_isoQ <- s1_3_isoQ %>% dplyr::as_tibble()

#Renome Gene_ID column to gene_name
colnames(s1_3_isoQ)[10] ="gene_name"


# filter GTF files for the gene of interest
gene_of_interest <- "ENSG00000136425.14"       #  CIB2  ENSG00000136425.14

refseq_gencode <- gencode %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest
  )

CIB2_CCS3 <- s1_3_isoQ %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest
  )

# change df_columns for figure

#Select MANE select refseqs, add origin column & rename ENST (proir to
making GIO_combi, identical ENST in multiple datasets causes an error)
refseq_gencode_clean <- refseq_gencode %>% dplyr::filter(transcript_id %in%
c("ENST00000258930.8"))
refseq_gencode_clean$origin <- "gencode"
refseq_gencode_clean$transcript_id[refseq_gencode_clean$transcript_id=="ENS
T00000258930.8"]<-"01_Gencode_ENST00000258930.8"


#IsoQ data s1-3 rename ENST and add origin column  (dataset is empty, no
isoforms predicted by IsoQuant)
CIB2_CCS3$origin <- "1-3 CCS3"
```

```
# combine df_columns for figure
CIB2_fig_full <- bind_rows(refseq_gencode_clean, CIB2_CCS3)


# extract the required annotation columns
CIB2_fig <- CIB2_fig_full %>%
  dplyr::select(
    seqnames,
    start,
    end,
    strand,
    type,
    gene_name,
    transcript_type,
    transcript_id,
    tag,
    origin)


#####  Visualize transcript type  #################


# obtain exons
GOI_exons <- CIB2_fig %>% dplyr::filter(type == "exon")


#  Reduce intron length
GOI_rescaled <- shorten_gaps(
  exons = GOI_exons,
  introns = to_intron(GOI_exons, "transcript_id"),
  group_var = "transcript_id", target_gap_width =50L)

# shorten_gaps() returns exons and introns all in one data.frame()
# let's split these for plotting
GOI_rescaled_exons <- GOI_rescaled %>% dplyr::filter(type == "exon")
GOI_rescaled_introns <- GOI_rescaled %>% dplyr::filter(type == "intron")

GOI_rescaled_exons %>%
  ggplot(aes(
    xstart = start,
    xend = end,
    y = transcript_id,
  ))  +
  scale_y_discrete(limits = rev)  +               #reverts y axis
  coord_cartesian(xlim = c(max(GOI_rescaled_exons$end),
(min(GOI_rescaled_exons$start))))+       #invert orientation

  geom_range(
    aes(fill = origin)
  ) + scale_fill_manual(values=c("#F8766D", "#7CAE00", "#00BFC4",
"#E68613","#E68613", "#999999")) +
  geom_intron(
    data = GOI_rescaled_introns,
    aes(strand = strand),
    arrow.min.intron.length = 700
  ) +
  theme_classic()

ggsave("CIB2_figure.pdf")
#save on fixed width of 10 with height of 2 per transcript isoform


# additional annotations were made manually
```

Figure S6 – GGtranscript *ADGRV1* Isoquant output plot

```
# Packages needed:
# dplyr, ggplot2, ggtranscript, rtracklayer

# Load packages
library(dplyr)
library(ggplot2)
library(ggtranscript)
library(rtracklayer)

setwd("//")

#####  Visualize from data GTF  #################

# Import Gencode reference file into a temporary directory
gencode_path <- file("gencode.v45.annotation.gtf")
gencode <- rtracklayer::import(gencode_path)
gencode <- gencode %>% dplyr::as_tibble()

# Import dataset 1 isoquant track into a temporary directory
s1_3_isoQ_path <- file.path("00_aln.sorted.transcript_models.gtf")
s1_3_isoQ <- rtracklayer::import(s1_3_isoQ_path)
s1_3_isoQ <- s1_3_isoQ %>% dplyr::as_tibble()

#Renome Gene_ID column to gene_name
colnames(s1_3_isoQ)[10] ="gene_name"

# filter GTF files for the gene of interest ("ADGRV1" is used for the
gencode dataset, "ENSG00000164199.18" to filter in the isoquant data)
gene_of_interest <- "ADGRV1"

refseq_gencode <- gencode %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest
  )

ADGRV1_CCS3 <- s1_3_isoQ %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest
  )



# Select MANE select/clinical as reference isoforms from Gencode dataset
# rename ENSTXXX transcripts in dataset to prevent redundant names.
refseq_gencode_clean <- refseq_gencode   %>% dplyr::filter(transcript_id
%in% c("ENST00000405460.9"))
refseq_gencode_clean$origin <- "gencode"
refseq_gencode_clean$transcript_id[refseq_gencode_clean$transcript_id=="ENS
T00000405460.9"]<-"01_Gencode_ENST00000405460.9"

# Select human transcript isoforms published in McMillan et al (2002) -
VLGR1a and VLGR1b
# rename ENSTXXX transcripts in dataset to prevent redundant names.
ADGRV1_McMillan_ab <- refseq_gencode   %>% dplyr::filter(transcript_id %in%
c("ENST00000405460.9", "ENST00000638510.1"))
ADGRV1_McMillan_ab$origin <- "McMillan_a"
ADGRV1_McMillan_ab$transcript_id[ADGRV1_McMillan_ab$transcript_id=="ENST000
00405460.9"]<-"03_VLGR1-b_ENST00000405460.9"
ADGRV1_McMillan_ab$transcript_id[ADGRV1_McMillan_ab$transcript_id=="ENST000
00638510.1"]<-"04_VLGR1-a_ENST00000638510.1"
```

```
#Select and create orthologs of mouse transcripts isoform published in
McMillan (2002) - Vlgr1a, Vlgr1b, Vlgr1c
Mouse_ab <- refseq_gencode   %>% dplyr::filter(transcript_id %in%
c("ENST00000405460.9"))
Mouse_ab$origin <- "Mouse_ab"
Mouse_ab$transcript_id[Mouse_ab$transcript_id=="ENST00000405460.9"]<-
"06_1VLGR1-b_ENST00000405460.9"
Mouse_ab$transcript_id[Mouse_ab$transcript_id=="ENST00000638510.1"]<-
"06_2VLGR1-a_ENST00000638510.1"

# One isoform published in McMillan did not have an immediate human
orthologue in the gencode dataset. And xlsx file containing the transcript
isoform was created manually.
ADGRV1_McMillan_c <- read.xlsx("USH Genes/ADGRV1/ADGRV1_McMillan_c.xlsx",
"McMillan_c")
ADGRV1_McMillan_c$origin <- "McMillan_c"


# create orthologues of mouse transcript isoforms published in Yagi et al
(2005) - Vlgr1d and Vlgr1e
# both are created manually and uploaded as xlsx file
ADGRV1_Yagi_d <- read.xlsx("USH Genes/ADGRV1/Adgrv1_Yagi_d.xlsx", "Yagi_d")
ADGRV1_Yagi_d$origin <- "Yagi_d"
ADGRV1_Yagi_e <- read.xlsx("USH Genes/ADGRV1/Adgrv1_Yagi_e.xlsx", "Yagi_e")
ADGRV1_Yagi_e$origin <- "Yagi_e"

# create orthologues of mouse transcript isoforms published in Skradski
2001 orthologous - Mass1.1, Mass1.2 and Mass1.3
# End of all Mass1 isoforms annoated based on 3'extension of exon 39 as
seen in mouse
# Start site of Mass1.1 annotated based on murine 5'extension of exon 5
# Start site of Mass1.2 annotated based on EST DA391827
# Start site of Mass1.3 annotated based on murine 5'extension of exon 26
# all are created manually and uploaded as a single xlsx file
Adgrv1_Skradski <- read.xlsx("USH Genes/ADGRV1/ADGRV1_Mass1.xlsx", "Mass
combined")
Adgrv1_Skradski$origin <- "Skradski"

# isoquant data
# rename ENSTXXX transcripts in dataset to prevent redundant names.
ADGRV1_CCS3$origin <- "1-3 CCS3"
ADGRV1_CCS3$transcript_id[ADGRV1_CCS3$transcript_id=="ENST00000640109.1"]<-
"tAtlas_ENST00000640109.1"
ADGRV1_CCS3$transcript_id[ADGRV1_CCS3$transcript_id=="ENST00000640281.1"]<-
"tAtlas_ENST00000640281.1"
ADGRV1_CCS3$transcript_id[ADGRV1_CCS3$transcript_id=="ENST00000638316.1"]<-
"tAtlas_ENST00000638316.1"
ADGRV1_CCS3$transcript_id[ADGRV1_CCS3$transcript_id=="ENST00000639884.1"]<-
"tAtlas_ENST00000639884.1"


# combine df_columns for figure
ADGRV1_fig_full <- bind_rows(refseq_gencode_clean, ADGRV1_McMillan_ab,
ADGRV1_McMillan_c, mouse_C, Mouse_ab, ADGRV1_Yagi_d, ADGRV1_Yagi_e,
Adgrv1_Skradski, ADGRV1_CCS3)


# extract the required annotation columns
ADGRV1_fig <- ADGRV1_fig_full %>%
  dplyr::select(
    seqnames,
    start,
    end,
    strand,
    type,
```

```
    gene_name,
    transcript_type,
    transcript_id,
    tag,
    origin
  )



#####  Visualize transcript type  #################


# obtain exons
GOI_exons <- ADGRV1_fig %>% dplyr::filter(type == "exon")


# Reduce intron length
GOI_rescaled <- shorten_gaps(
  exons = GOI_exons,
  introns = to_intron(GOI_exons, "transcript_id"),
  group_var = "transcript_id", target_gap_width =100L
)


# shorten_gaps() returns exons and introns all in one data.frame()
# let's split these for plotting
GOI_rescaled_exons <- GOI_rescaled %>% dplyr::filter(type == "exon")
GOI_rescaled_introns <- GOI_rescaled %>% dplyr::filter(type == "intron")

GOI_rescaled_exons %>%
  ggplot(aes(
    xstart = start,
    xend = end,
    y = transcript_id,
  ))  +
  scale_y_discrete(limits = rev)  +                    #reverts y axis


  geom_range(
    aes(fill = origin)
  ) + scale_fill_manual(values=c("#F8766D", "#7CAE00", "#00BFC4",
"#00BFC4","#999999", "#999999","#D3D3D3","#999999", "#999999")) +

  geom_intron(
    data = GOI_rescaled_introns,
    aes(strand = strand),
    arrow.min.intron.length = 700
  ) +
  theme_classic(
  )

Ggsave("ADGRV1_figure.pdf")

# save on fixed width of 40 with height of 2 per transcript isoform
# xls files were made in which refseq data was manually adjusted to match
the information obtained with manual curation and Samplix. Files are
deposited on github:
https://github.com/erikdevrieze/USH_retina/tree/main/Figures/Data
# additional annotations were made manually
```

Figure S7 – GGtranscript *CLRN1* plot

```
# Packages needed:
# dplyr, ggplot2, ggtranscript, rtracklayer

# Load packages
library(dplyr)
library(ggplot2)
library(ggtranscript)
library(rtracklayer)

setwd("//")

#####  Visualize from data GTF  #################

# Import Gencode reference file into a temporary directory
gencode_path <- file("gencode.v45.annotation.gtf")
gencode <- rtracklayer::import(gencode_path)
gencode <- gencode %>% dplyr::as_tibble()

# Import dataset 1 isoQuant  track into a temporary directory
s1_3_isoQ_path <- file.path("00_aln.sorted.transcript_models.gtf")
s1_3_isoQ <- rtracklayer::import(s1_3_isoQ_path)
s1_3_isoQ <- s1_3_isoQ %>% dplyr::as_tibble()

#Renome Gene_ID column to gene_name
colnames(s1_3_isoQ)[10] ="gene_name"


# filter GTF files for the gene of interest
gene_of_interest <- "ENSG00000163646.12"       #  CLRN1  ENSG00000163646.12

refseq_gencode <- gencode %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest)

CLRN1_CCS3 <- s1_3_isoQ %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest)


# change df_columns for figure

#Select MANE select and MANE clinical as reference for annotation. add
origin column & rename ENST (proir to making GIO_combi, identical ENST in
multiple datasets causes an error)
refseq_gencode_clean <- refseq_gencode %>% dplyr::filter(transcript_id %in%
c("ENST00000327047.6"))
refseq_gencode_clean$origin <- "gencode"
refseq_gencode_clean$transcript_id[refseq_gencode_clean$transcript_id=="ENS
T00000327047.6"]<-"01_Gencode_ENST00000327047.6"

#IsoQ data s1-3 rename ENST and add origin column
CLRN1_CCS3$origin <- "1-3 CCS3"
CLRN1_CCS3$transcript_id[CLRN1_CCS3$transcript_id=="ENST00000472224.1"]<-
"07_tAtlas_ENST00000472224.1"
CLRN1_CCS3$transcript_id[CLRN1_CCS3$transcript_id=="ENST00000327047.6"]<-
"08_tAtlas_ENST00000327047.6"
CLRN1_CCS3$transcript_id[CLRN1_CCS3$transcript_id=="transcript28193.chr3.ni
c"]<-"09_tAtlas_transcript28193.chr3.nic"
```

```
# combine df_columns for figure
CLRN1_fig_full <- bind_rows(refseq_gencode_clean, CLRN1_CCS3)

# extract the required annotation columns
CLRN1_fig <- CLRN1_fig_full %>%
  dplyr::select(
    seqnames,
    start,
    end,
    strand,
    type,
    gene_name,
    transcript_type,
    transcript_id,
    tag,
    origin )


#####  Visualize transcript type  ################

# obtain exons
GOI_exons <- CLRN1_fig %>% dplyr::filter(type == "exon")


#  Reduce intron length
GOI_rescaled <- shorten_gaps(
  exons = GOI_exons,
  introns = to_intron(GOI_exons, "transcript_id"),
  group_var = "transcript_id", target_gap_width =50L)


# shorten_gaps() returns exons and introns all in one data.frame()
# let's split these for plotting
GOI_rescaled_exons <- GOI_rescaled %>% dplyr::filter(type == "exon")
GOI_rescaled_introns <- GOI_rescaled %>% dplyr::filter(type == "intron")

GOI_rescaled_exons %>%
  ggplot(aes(
    xstart = start,
    xend = end,
    y = transcript_id,
  ))  +
  scale_y_discrete(limits = rev)  +                #reverts y axis
  coord_cartesian(xlim = c(max(GOI_rescaled_exons$end),
(min(GOI_rescaled_exons$start))))+       #invert orientation


  geom_range(
    aes(fill = origin)
  ) + scale_fill_manual(values=c("#F8766D", "#00BFC4", "#7CAE00",
"#E68613","#E68613", "#999999")) +
  geom_intron(
    data = GOI_rescaled_introns,
    aes(strand = strand),
    arrow.min.intron.length = 700
  ) +
  theme_classic( )

ggsave("Fig Sx_CLRN1.pdf)

#save on fixed width of 10 with height of 2 per transcript isoform
# additional annotations were made manually
```

Figure S8 – GGtranscript *ARSG* plot

```
# Packages needed:
# dplyr, ggplot2, ggtranscript, rtracklayer

# Load packages
library(dplyr)
library(ggplot2)
library(ggtranscript)
library(rtracklayer)

setwd("//")

#####  Visualize from data GTF  #################

# Import Gencode reference file into a temporary directory
gencode_path <- file("gencode.v45.annotation.gtf")
gencode <- rtracklayer::import(gencode_path)
gencode <- gencode %>% dplyr::as_tibble()

# Import dataset 1 isoQuant track into a temporary directory
s1_3_isoQ_path <- file.path("00_aln.sorted.transcript_models.gtf")
s1_3_isoQ <- rtracklayer::import(s1_3_isoQ_path)
s1_3_isoQ <- s1_3_isoQ %>% dplyr::as_tibble()

#Renome Gene_ID column to gene_name
colnames(s1_3_isoQ)[10] ="gene_name"

# filter GTF files for the gene of interest
gene_of_interest <- "ARSG"       #  ARSG  ENSG00000141337.13

refseq_gencode <- gencode %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest)

ARSG_CCS3 <- s1_3_isoQ %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest)


# change df_columns for figure

#Select MANE select and MANE clinical as reference for annotation. add
origin column & rename ENST (proir to making GIO_combi, identical ENST in
multiple datasets causes an error)
refseq_gencode_clean <- refseq_gencode %>% dplyr::filter(transcript_id %in%
c("ENST00000621439.5"))
refseq_gencode_clean$origin <- "gencode"
refseq_gencode_clean$transcript_id[refseq_gencode_clean$transcript_id=="ENS
T00000621439.5"]<-"01_Gencode_ENST00000621439.5"

#IsoQ data s1-3 rename ENST and add origin column
ARSG_CCS3$origin <- "1-3 CCS3"
ARSG_CCS3$transcript_id[ARSG_CCS3$transcript_id=="ENST00000472224.1"]<-
"02_tAtlas_ENST00000472224.1"
ARSG_CCS3$transcript_id[ARSG_CCS3$transcript_id=="ENST00000327047.6"]<-
"03_tAtlas_ENST00000327047.6"
ARSG_CCS3$transcript_id[ARSG_CCS3$transcript_id=="transcript28193.chr3.nic"
]<-"04_tAtlas_transcript28193.chr3.nic"
```

```
# combine df_columns for figure
ARSG_fig_full <- bind_rows(refseq_gencode_clean, ARSG_CCS3)

# extract the required annotation columns
ARSG_fig <- ARSG_fig_full %>%
  dplyr::select(
    seqnames,
    start,
    end,
    strand,
    type,
    gene_name,
    transcript_type,
    transcript_id,
    tag,
    origin)


#####  Visualize transcript type  #################

# obtain exons
GOI_exons <- ARSG_fig %>% dplyr::filter(type == "exon")

#  Reduce intron length
GOI_rescaled <- shorten_gaps(
  exons = GOI_exons,
  introns = to_intron(GOI_exons, "transcript_id"),
  group_var = "transcript_id", target_gap_width =50L)


# shorten_gaps() returns exons and introns all in one data.frame()
# let's split these for plotting
GOI_rescaled_exons <- GOI_rescaled %>% dplyr::filter(type == "exon")
GOI_rescaled_introns <- GOI_rescaled %>% dplyr::filter(type == "intron")

GOI_rescaled_exons %>%
  ggplot(aes(
    xstart = start,
    xend = end,
    y = transcript_id,
  ))  +
  scale_y_discrete(limits = rev)  +                #reverts y axis

  geom_range(
    aes(fill = origin)
  ) + scale_fill_manual(values=c("#F8766D", "#7CAE00", "#7CAE00",
"#E68613","#E68613", "#999999")) +
  geom_intron(
    data = GOI_rescaled_introns,
    aes(strand = strand),
    arrow.min.intron.length = 700
  ) +
  theme_classic()

ggsave("ARSG_figure.pdf")

#save on fixed width of 40 with height of 2 per transcript isoform
# additional annotations were made manually
```

Figure S9 – Isoform structural category reads per USH gene

Data from generated with the code for Figure 2B was plotted for the individual structural categories.

Figure S10 – unique isoforms per structural category (dataset 1 vs dataset 2)

```
#Load packages
library(dplyr)
library(readr)
library(data.table)
library(splitstackshape)

setwd()



######## Dataset 1   ####################################################

#read dataset 1 00_aln.sorted.transcript_model_grouped_counts
AT_transcript_model_counts <-
fread("00_aln.sorted.transcript_model_grouped_counts.tsv")
colnames(AT_transcript_model_counts)[1] ="transcript_id"

s1_AT_transcript_model_counts <- AT_transcript_model_counts %>%
dplyr::filter(AT_transcript_model_counts$s1 >= 1)
s2_AT_transcript_model_counts <- AT_transcript_model_counts %>%
dplyr::filter(AT_transcript_model_counts$s2 >= 1)
s3_AT_transcript_model_counts <- AT_transcript_model_counts %>%
dplyr::filter(AT_transcript_model_counts$s3 >= 1)



#Grab, count and plot ENST matching reads:
ENST_s1 <- dplyr::filter(s1_AT_transcript_model_counts, grepl("ENST",
transcript_id))
ENST_s2 <- dplyr::filter(s2_AT_transcript_model_counts, grepl("ENST",
transcript_id))
ENST_s3 <- dplyr::filter(s3_AT_transcript_model_counts, grepl("ENST",
transcript_id))


#Grab, count and plot NNIC matching reads:
NNIC_s1 <- dplyr::filter(s1_AT_transcript_model_counts, grepl(".nnic",
transcript_id))
NNIC_s2 <- dplyr::filter(s2_AT_transcript_model_counts, grepl(".nnic",
transcript_id))
NNIC_s3 <- dplyr::filter(s3_AT_transcript_model_counts, grepl(".nnic",
transcript_id))

#Grab, count and plot NIC matching reads:
NIC_s1 <- dplyr::filter(s1_AT_transcript_model_counts, grepl(".nic",
transcript_id)) # captures both NIC and NNIC (calculated NIC manually (NIC
= total .nic and .nnic combined - NNIC))
NIC_s2 <- dplyr::filter(s2_AT_transcript_model_counts, grepl(".nic",
transcript_id))
NIC_s3 <- dplyr::filter(s3_AT_transcript_model_counts, grepl(".nic",
transcript_id))
```

```
######## Dataset 2   ##################################################

#read dataset 2 00_aln.sorted.transcript_model_grouped_counts
s1_4_transcript_model_counts <-
fread("00_aln.sorted.transcript_model_grouped_counts.tsv")
colnames(s1_4_transcript_model_counts)[1] ="transcript_id"

s1_14_AT_transcript_model_counts <- s1_4_transcript_model_counts %>%
dplyr::filter(s1_4_transcript_model_counts$s1 >= 1)
s2_14_AT_transcript_model_counts <- s1_4_transcript_model_counts %>%
dplyr::filter(s1_4_transcript_model_counts$s2 >= 1)
s3_14_AT_transcript_model_counts <- s1_4_transcript_model_counts %>%
dplyr::filter(s1_4_transcript_model_counts$s3 >= 1)
s4_14_AT_transcript_model_counts <- s1_4_transcript_model_counts %>%
dplyr::filter(s1_4_transcript_model_counts$s2_long >= 1)

#Grab, count and plot ENST matching reads:
ENST_s1_14 <- dplyr::filter(s1_14_AT_transcript_model_counts, grepl("ENST",
transcript_id))
ENST_s2_14 <- dplyr::filter(s2_14_AT_transcript_model_counts, grepl("ENST",
transcript_id))
ENST_s3_14 <- dplyr::filter(s3_14_AT_transcript_model_counts, grepl("ENST",
transcript_id))
ENST_s4_14 <- dplyr::filter(s4_14_AT_transcript_model_counts, grepl("ENST",
transcript_id))

#Grab, count and plot NNIC matching reads:
NNIC_s1_14 <- dplyr::filter(s1_14_AT_transcript_model_counts,
grepl(".nnic", transcript_id))
NNIC_s2_14 <- dplyr::filter(s2_14_AT_transcript_model_counts,
grepl(".nnic", transcript_id))
NNIC_s3_14 <- dplyr::filter(s3_14_AT_transcript_model_counts,
grepl(".nnic", transcript_id))
NNIC_s4_14 <- dplyr::filter(s4_14_AT_transcript_model_counts,
grepl(".nnic", transcript_id))


#Grab, count and plot NNIC and NIC matching reads:
NIC_s1_14 <- dplyr::filter(s1_14_AT_transcript_model_counts, grepl(".nic",
transcript_id)) # captures both NIC and NNIC (calculated NIC manually (NIC
= total .nic and .nnic combined - NNIC))
NIC_s2_14 <- dplyr::filter(s2_14_AT_transcript_model_counts, grepl(".nic",
transcript_id))
NIC_s3_14 <- dplyr::filter(s3_14_AT_transcript_model_counts, grepl(".nic",
transcript_id))
NIC_s4_14 <- dplyr::filter(s4_14_AT_transcript_model_counts, grepl(".nic",
transcript_id))


# obtained counts were plotted with Graphpad Prism
```

Figure S11 – GGtranscript *MYO7A* IsoQuant dataset 1 vs dataset 2 plot

```
# Packages needed:
# dplyr, ggplot2, ggtranscript, rtracklayer

# Load packages
library(dplyr)
library(ggplot2)
library(ggtranscript)
library(rtracklayer)

setwd("//")

#####  Visualize from data GTF  #################

# Import Gencode reference file into a temporary directory
gencode_path <- file("gencode.v45.annotation.gtf")
gencode <- rtracklayer::import(gencode_path)
gencode <- gencode %>% dplyr::as_tibble()

# Import dataset 1 isoQuant GTF file into a temporary directory
s1_3_isoQ_path <- file.path("dataset
1/00_aln.sorted.transcript_models.gtf")
s1_3_isoQ <- rtracklayer::import(s1_3_isoQ_path)
s1_3_isoQ <- s1_3_isoQ %>% dplyr::as_tibble()

# Import dataset 2 isoQuant GTF file into a temporary directory (dataset
including the 4th long enriched library)
s1_4_isoQ_path <- file.path("dataset
2/Isoquant/00_aln.sorted.transcript_models.gtf")
s1_4_isoQ <- rtracklayer::import(s1_4_isoQ_path)
s1_4_isoQ <- s1_4_isoQ %>% dplyr::as_tibble()

#Renome Gene_ID column to gene_name
colnames(s1_3_isoQ)[10] ="gene_name"
colnames(s1_4_isoQ)[10] ="gene_name"

# filter GTF files for the gene of interest
gene_of_interest <- "ENSG00000137474.23"       #  MYO7A  ENSG00000137474.23

refseq_gencode <- gencode %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest
  )

MYO7A_CCS3 <- s1_3_isoQ %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest
  )

MYO7A_s14 <- s1_4_isoQ %>%
  dplyr::filter(
    !is.na(gene_name),
    gene_name == gene_of_interest
  )


# change df_columns for figure

#Select MANE select/clinical as reference isoforms from Gencode dataset
#rename ENSTXXX transcripts in dataset to prevent redundant names.
```

```
refseq_gencode_clean <- refseq_gencode %>% dplyr::filter(transcript_id %in%
c("ENST00000409709.9"))
refseq_gencode_clean$origin <- "gencode"
refseq_gencode_clean$transcript_id[refseq_gencode_clean$transcript_id=="ENS
T00000409709.9"]<-"1_Gencode_ENST00000409709.9"


#rename ENSTXXX transcripts in CCS3 dataset to prevent redundant names.
MYO7A_CCS3$origin <- "1-3 CCS3"
MYO7A_CCS3$transcript_id[MYO7A_CCS3$transcript_id=="ENST00000409709.9"]<-
"tAtlas_ENST00000409709.9"
MYO7A_CCS3$transcript_id[MYO7A_CCS3$transcript_id=="ENST00000458637.6"]<-
"tAtlas_ENST00000458637.6"

#rename ENSTXXX transcripts in sample 1-4 dataset to prevent redundant
names.
MYO7A_s14$origin <- "1-4 CCS3"
MYO7A_s14$transcript_id[MYO7A_s14$transcript_id=="ENST00000409709.9"]<-
"tAtlas_ENST00000409709.9"
MYO7A_s14$transcript_id[MYO7A_s14$transcript_id=="ENST00000458637.6"]<-
"tAtlas_ENST00000458637.6"


# combine df_columns for figure
MYO7A_fig_full <- bind_rows(refseq_gencode_clean, MYO7A_s14, MYO7A_CCS3 )

# extract the required annotation columns
MYO7A_fig <- MYO7A_fig_full %>%
  dplyr::select(
    seqnames,
    start,
    end,
    strand,
    type,
    gene_name,
    transcript_type,
    transcript_id,
    tag,
    origin
  )


#####  Visualize transcript type  ################


# obtain cds
MYO7A_cds <- MYO7A_fig %>% dplyr::filter(type == "CDS")

# obtain exons
MYO7A_exons <- MYO7A_fig %>% dplyr::filter(type == "exon")


#  Reduce intron length
MYO7A_rescaled <- shorten_gaps(
  exons = MYO7A_exons,
  introns = to_intron(MYO7A_exons, "transcript_id"),
  group_var = "transcript_id", target_gap_width =50L
)


# shorten_gaps() returns exons and introns all in one data.frame()
# let's split these for plotting
MYO7A_rescaled_exons <- MYO7A_rescaled %>% dplyr::filter(type == "exon")
MYO7A_rescaled_introns <- MYO7A_rescaled %>% dplyr::filter(type ==
"intron")
```

```
MYO7A_rescaled_exons %>%
  ggplot(aes(
    xstart = start,
    xend = end,
    y = transcript_id,
  ))  +
  geom_range(
    aes(fill = origin)
  ) + scale_fill_manual(values=c("#F8766D", "#00BFC4","#7CAE00",
"#999999")) +
  geom_intron(
    data = MYO7A_rescaled_introns,
    aes(strand = strand),
    arrow.min.intron.length = 700
  ) +
  theme_classic(
  ) +
  scale_y_discrete(limits = rev)

  ggsave("MYO7A_suppl_figure.pdf")


# further processing of the plot was manually in adobe illustrator
```