ScisorWiz - Standard Workflow

The *ScisorWiz* package creates a plot utilizing single-cell long-read RNA sequencing data in order to display isoform expression differentiation across multiple cell types for a single gene.

Software required

- python version >= 3.7 with the following libraries:
  - pandas
- R version >= 3.5 with the following packages:
  - hash
  - Optional (for interactive option)
    - plotly (for R)
    - htmlwidgets
    - RCurl
- samtools (for MismatchFinder function – see Section 2 - Optional Preprocessing

NOTE: We recommend downloading ScisorWiz and running all functions of the package in R via a Mac or Linux terminal in an environment which has both R and python installed rather than in RStudio as the user may run into issues running the python code included in this package.

Quick cheat for viewing documentation

All the possible arguments, usage, and description for each function can be viewed interactively by putting a "?" in front of the function name in the R console like so:

```
?ScisorWiz_AllInfo
```

Section 1 - Main workflow - Organize data and create the plot

There are two options for running the pipeline to plot the data depending on the files available to the user:

- ScisorWiz_AllInfo
  - Used if the user has output from the scisorseqr package called the AllInfo file.
    - This method is recommended to provide more customizability and specificity to the output plots.
    - The AllInfo file is formatted with data in the following order by column:
      - Read ID
      - Gene ID
      - Cell type
      - Barcode

        - UMI
        - TSS peak + Intron Chain + PolyA peak
        - Exon Chain
        - Novelty Status
        - Number of Introns
    * For an idea of the formatting for the AllInfo file, please see the example AllInfo file provided in the package by inputting the command to open the AllInfo file below:

```
library(ScisorWiz)
allInfoExampleFile <- system.file("extdata", "userInput/AllInfo", package = "ScisorWiz")
system(paste("cat", allInfoExampleFile))
```

- ScisorWiz_2File
  - Used if the user has gff.gz file and genes.gz file filtered for detected cage and polyA peaks.
    * gff.gz file is a tab-separated file containing read info (1 read per line) and is formatted as follows:

chromosome	genome	readType	start	end	score	strand	frame
chr18	hg38	cDNA_match	263682	264092	.	-	.

read_id string	readID
read_id	"ExciteDG:CTCCTTTAGCCTGAAG:762e2e07-1cf8-4150-8f3a-60cc263a7445.path1";

    * genes.gz file is a tab-separated file containing readID to geneID mappings and is formatted as follows:

readID	geneID
RGL:GCAGCCAGTAAATGTG:m64013_190223_004143/73663463/ccs.path1	ENSMUSG00000041571.9
OPCs:TCAGGATAGTTCGCAT:m64013_190221_020520/30540150/ccs.path1	ENSMUSG00000038717.8

NOTE: The cell type from which the read came must be appended to the readID in the same format as above. This helps to sort each read into the appropiate cell type cluster.

Pipeline Option 1 - ScisorWiz_AllInfo

This option uses the AllInfo file output from the scisorseqr package. More information on preprocessing with scisorseqr can be found below in the section titled "Running scisorseqr".

Entering the command for the ScisorWiz_AllInfo pipeline option gives the user the ability to choose the clustering method to utilize for the data on the final plot. For that the user must specify:

- GENCODE annotation file for user data
- AllInfo file derived from scisorseqr
- Cell type file listing user-specified cell types of interest and the display color of each (example of document format below)*
- Gene of interest
- Clustering method**

- Optional: Confidence interval (CI) for alternative exon consideration. Default value for exon inclusion rate is .05 (5% < altExon inclusion < 95%)
- Optional: Mismatch Cutoff to eliminate sequencing errors. Default value is .05 (5% < mismatch inclusion < 95%)
- Output directory in which the user wants output files stored
- Optional: Mismatches file containing output from the MismatchFinder function (see Section 2). Default value is NULL
- Optional: Interactive plot (for exploratory purposes, see Section 3)

*Celltype file is **tab separated** with each cell type and display color on a new line, and cell type names must be written exactly as they appear in the GENCODE file. The file looks like the following:

ExcitNeuron	darkblue
InhibNeuron	darkblue
GranuleNB	darkblue
OPCs	antiquewhite4
Astro	darkred
Microglia	purple

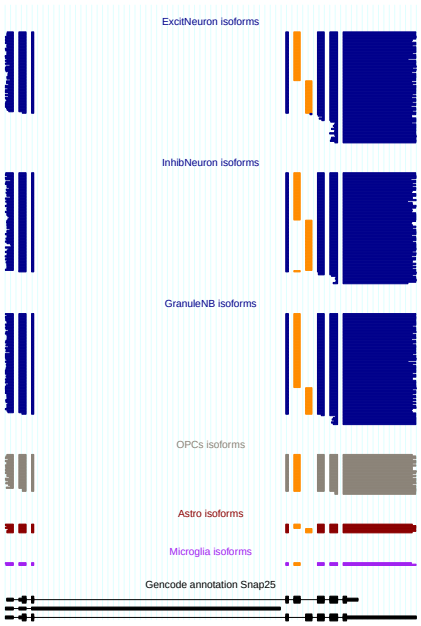
**The user can choose from one of the clustering methods by inputting the number that is next to the method as shown below:

1. Intron chain
2. TSS site
3. PolyA site
4. Intron chain, TSS site, and PolyA site

```
library(ScisorWiz)

gencodeAnnoFile <- system.file("extdata", "gencode.vM21.annotation.gtf.gz",
                           package = "ScisorWiz")
allInfoFile <- system.file("extdata", "userInput/AllInfo.gz", package = "ScisorWiz")
cTypeFile <- system.file("extdata", "userInput/cellTypeFile_Snap25.tab",
                         package = "ScisorWiz")

## Run command without plotting mismatches
ScisorWiz_AllInfo(gencodeAnno = gencodeAnnoFile, AllInfoInput = allInfoFile,
                  cellTypeFile = cTypeFile, gene = "Snap25", cluster = 1,
                  ci = .05, outputDir = "outputDir/")
```

Example of the output at the top of next page:

The output after running the data through the ScisorWiz_AllInfo function without mismatches is as follows:



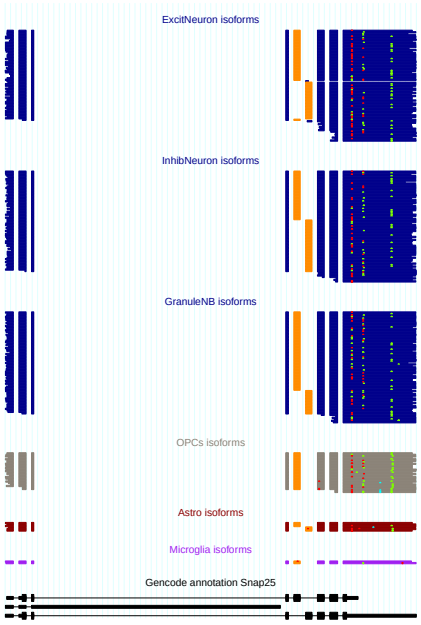
NOTE 1: The example data included in this package should yield a plot very similar to the one above.

NOTE 2: You may notice that Step 14 was skipped in this test run. This is because no mismatches file was input as an argument, and as a result mismatches will not be present on the output plot. To run this with mismatches, see the below chunk of code:

```
##_____________________________________OR_____________________________________##

library(ScisorWiz)

# In order to retain a smaller overall size for the ScisorWizpackage, the
# example data mismatches file is located in extData/, rather than within the
# outputDir subdirectory where normal output from the first step would go.

gencodeAnnoFile <- system.file("extdata", "gencode.vM21.annotation.gtf.gz",
                           package = "ScisorWiz")
allInfoFile <- system.file("extdata", "userInput/AllInfo.gz", package = "ScisorWiz")
cTypeFile <- system.file("extdata", "userInput/cellTypeFile_Snap25.tab",
                         package = "ScisorWiz")
mismatchesFile <- system.file("extdata", "Snap25.mismatches.txt.gz",
                           package = "ScisorWiz")

## Run command with plotting mismatches
ScisorWiz_AllInfo(gencodeAnno = gencodeAnnoFile, AllInfoInput = allInfoFile,
                  cellTypeFile = cTypeFile, gene = "Snap25", cluster = 1,
                  ci = .05, mismatchCutoff = .05,
                  outputDir = "outputDir/", mismatchFile = mismatchesFile)
```

Example of the output at the top of next page:

The output after running the data through the MismatchFinder function and then the ScisorWiz_AllInfo function is as follows:



NOTE: The example data included in this package should yield a plot very similar to the one above.

Running scisorseqr

The scisorseqr package was built with the goal of performing differential isoform expression tests between samples either at the cell-type level using single cell barcodes, or at the spatial level using spatial 'spot' barcodes. The package performs extensive QC and filtering of the data to ensure that only barcoded, full-length, spliced reads are taken into consideration. Due to the rigorous filtering that the package performs as well as the the data formats it generates, we recommend users run that pipeline to ensure the highest quality reads make it into the ScisorWiz plot.

Instead of multiple files containing various pieces of information about a read, scisorseqr combines all the necessary information needed to produce the isoform plot. Particularly, it combines the read name along with a cell type annotation, barcode, and UMI information in columns of the file, and additionally collates the coordinates of exons that map to a read within a single column of this file as opposed to having multiple lines e.g. in a GTF format

We have included the output from scisorseqr i.e. the AllInfo.gz file, which is used as input for the ScisorWiz isoform plot. However, if the user wishes to run the whole pipeline including the steps of scisorseqr, we have provided the necessary input for two example genes: *Snap25* and *Pkm*

The user will have to download bed files for annotated CAGE peaks and PolyA sites. If the user wishes to proceed without them, simply set the parameter filterFullLength=FALSE in Step3 in the code chunk below:

NOTE: You must download the scisorseqr package to run the following code.

```
library(scisorseqr)

genomeFa <- args[2]
```

```
annoGZ <- args[3]
cageBedFile <- args[4]
polyaBedFile <- args[5]
seqDir <- args[6]

print("++++++++ Step 1: Getting barcodes and filtering those reads out")

fqFolderPath <- system.file("extdata/", "userInput/fqFolder/", package = "ScisorWiz")
ctAssignments <- system.file("extdata/", "userInput/barcode_cellType_assignments",
                             package = "ScisorWiz")

GetBarcodes(fqFolder=fqFolderPath,
        BCClustAssignFile=ctAssignments,
        chemistry='v2', filterReads=TRUE, numProcesses=12,
        concatenate=FALSE)

print("++++++++ Step 2: Aligning with minimap2")
mmProgPath <- '~/minimap2/minimap2' ## Please provide path to minimap2 aligner
genomeFa <- '~/genomes/M.musculus/mm10.fa' ## Please provide path to a mouse genome

MMalign(fqFolder=fqFolderPath,mmProgPath,
        refGenome=genomeFa,
        numThreads=12)

print("++++++++ Step 3: Map and filter function")

gencodeAnnoFile <- system.file("extdata/", "gencode.vM21.annotation.gtf.gz", package = "ScisorWiz")
chr_fa_dir <- '~/genomes/M.musculus/mm10/chromFa/' ## Please provide a path to a directory containing o

# Below two arguments are optional. Can instead set filterFullLength=FALSE in the function
polyABed_path <- 'atlas.clusters_chr.mm10.2-0.bed.gz'
cageBed_path <- 'mm10_fair+new_CAGE_peaks_phase1and2.bed.gz'

MapAndFilter(numThreads=12, filterFullLength=TRUE,
        polyABed=polyABed_path,
        cageBed=cageBed_path,
        annoGZ=gencodeAnnoFile,
        seqDir=chr_fa_dir, genomeVersion='mm10')

print("++++++++ Step 4: Getting All-Info files")
## If default parameters for Step 1 are used, the the barcodeOutput file will be autogenenerated in the
## "OutputFiltered" folder

InfoPerLongRead(barcodeOutputFile='OutputFiltered/FilteredDeconvBC_P7HIPP_subset.csv',
        mapAndFilterOut='LRProcessingOutput/', minTimesIsoObserve=1, rmTmpFolder=FALSE)

print("+++++++ All done!")
```

Pipeline Option 2 - ScisorWiz_2File

Without the AllInfo file from the scisorseqr pipeline, the user will not be able to choose a clustering method for the data, so using this option will cluster the data automatically by the intron chain. However, the user

can still run the ScisorWiz_2File function on their gff.gz and genes.gz files which are filtered for detected cage and PolyA peaks. For that the user must specify:

- GENCODE annotation file for user data
- gff.gz file containing read-specific information
- genes.gz file
- Cell type file listing user-specified cell types of interest and the display color of each
- Gene of interest
- Optional: Confidence interval (CI) for alternative exon consideration. Default value for exon inclusion rate is .05 (5% < altExon inclusion < 95%)
- Optional: Mismatch Cutoff to eliminate sequencing errors. Default value is .05 (5% < mismatch inclusion < 95%)
- Output directory in which the user wants output files stored
- Optional: Mismatches file containing output from the MismatchFinder function (see Section 2). Default value is NULL
- Optional: Interactive plot (for exploratory purposes, see Section 3)

```
## Run command without plotting mismatches
ScisorWiz_2File(gencodeAnno = "gencodeAnnoFile.gz", gffInput = "CagePolyA.gff.gz",
                genesInput = "reads2genes.gz",cellTypeFile = "cellTypeFile_Snap25.tab",
                gene = "Snap25", ci = .05, outputDir = "outputDir/")

##____________________________________OR____________________________________##

## Run command with plotting mismatches
ScisorWiz_2File(gencodeAnno = "gencodeAnnoFile.gz",
                gffInput = "CagePolyA.gff.gz", genesInput = "reads2genes.gz",
                cellTypeFile = "cellTypeFile_Snap25.tab",gene = "Snap25",
                ci = .05, mismatchCutoff = .05, outputDir = "outputDir/",
                mismatchFile = "Snap25.mismatches.txt.gz")
```

NOTE: The data included in the package is not for this function due to raw file size, to test ScisorWiz using included data, please refer to Pipeline Option 1 - ScisorWiz_AllInfo section example.

Section 2 - Optional Pre-processing: Getting SNVs with MismatchFinder Function

Prior to using ScisorWiz main pipeline, the user has the option to run the MismatchFinder function. MismatchFinder will specify any SNVs, insertions, or deletions in the data as compared to the reference genome. For that the user must specify:

- Sorted .bam file
- Reference .fasta file
- GENCODE annotation for the data
- Gene of interest
- Output directory in which the user wants the mismatch file stored

```
## Run command
library(ScisorWiz)
MismatchFinder(BAM = "sorted.bestperRead.mapping.bam", fasta = "mm10.fa",
               gencodeAnno = "gencode.vM21.annotation.gtf.gz", gene = "Snap25",
               outputDir = "outputDir/")
```

User will see the output directory with a subdirectory specifically named for the gene of interest:

- outputDir
  - Snap25
    - Snap25.info.tab (Provides chromosome, start, and end for MismatchFinder script)
    - Snap25.mismatches.txt.gz (One line per readID with following structure:)

chrom	readID	SNV
chr2	m64013_190219_195127/88146585/ccs	136713563_G|A;136764203_C|A

insertion	deletion
136781282_136781283_A;136781282_136781283_C	136781219_C;136781421_C

NOTE: In order to run the MismatchFinder function, the user must install samtools. Installation for samtools can be found via the following link: samtools

Section 3 - Option to Create Interactive Plot

When calling the ScisorWiz_AllInfo or ScisorWiz_2File functions, users have an option to create an interactive plot in which dynamic, windowed zooming and panning functionality will allow the user to explore the plot printout closely. For this option, the user must specify:

```
library(ScisorWiz)

gencodeAnnoFile <- system.file("extdata", "gencode.vM21.annotation.gtf.gz",
                               package = "ScisorWiz")
allInfoFile <- system.file("extdata", "userInput/AllInfo.gz", package = "ScisorWiz")
cTypeFile <- system.file("extdata", "userInput/cellTypeFile_Snap25.tab",
                         package = "ScisorWiz")

## Run command without plotting mismatches
ScisorWiz_AllInfo(gencodeAnno = gencodeAnnoFile, AllInfoInput = allInfoFile,
                  cellTypeFile = cTypeFile, gene = "Snap25", cluster = 1,
                  ci = .05, outputDir = "outputDir/", interactive = "y")
```

When this option is chosen, the output plots will be contained in two files. The first is a .jpg file which will be visually different than the default pdf plot. The second output file is an interactive html file which is automatically made using the .jpg file. Opening the html file will allow the user to interact with the plot, zooming and panning throughout it for a deeper look at any data of interest.

NOTE: The .jpg and html files are intended to be used for exploratory purposes only. If you would like to use a ScisorWiz plot as a figure in a paper or presentation, please keep the interactive option set to "n" and use the pdf version of the plot. Thank you!

Section 4 - UCSC Genome Browser Reference File

As an additional way to explore the data for the user's desired gene and cell types, ScisorWiz produces a file entitled (gene_name).ucscReference.gtf.gz. This file can be uploaded directly by clicking the link here: UCSC Genome Browser and clicking on the "Custom Tracks" option on the "My Data" dropdown tab. Once you have accessed this page, click the "add custom tracks" button and you will be redirected to a page with a series of empty text boxes. Above the first box will be an option to upload a file. Upload the (gene_name).ucscReference.gtf.gz file, and you can then access your data on the UCSC Genome Browser.

Done!