Retina LRS Analyses Overview

Sowmya Parthiban

2025-07-24

Table of contents

Code can be found here: https://github.com/sparthib/retina_lrs

1 FASTQ processing

1.0.1 Analysis 1: `code/01_fastq_processing/02_MinIONQC.sh`

- **Question/Aim**: Read statistics - read length, base quality, N50 distribution
- **Input**: Raw FASTQ
- **Method**: MinIONQC was ran.
- **Output**: summary yaml and individual plots on read length distribution, base quality distribution, yield over time

1.0.2 Analysis 2: `code/01_fastq_processing/03_fastq_qc.sh`

- **Question/Aim**: Remove low quality reads based on Phred score.
- **Input**: Raw FASTQ, counts matrix, etc.
- **Method**: Nanofilt removes reads in fastq based on ONT summary text file.
- **Output**: processed FASTQ

1.0.3 Analysis 3: `code/01_fastq_processing/minionQC_yaml.R`

- **Question/Aim**: Produce boxplots of mean read length, median q value, median N50 and total number of reads across all samples.
- **Input**: YAML summary file produced by MinIONQC.
- **Method**: ggplot2
- **Output**: boxplots

2 FASTQ to BAM

2.0.1 Analysis 4: `code/01b_fastq_to_bam/01_fastq_to_bam_gencode_splice.sh`

- **Question/Aim**: Alignment with genome
- **Input**: Nanofilt processed FASTQ
- **Method**: Minimap2 was used to align reads to the genome.
- **Output**: bam

2.0.2 Analysis 5: `code/01b_fastq_to_bam/01b_fastq_to_bam_transcriptome_gencode.sh`

- **Question/Aim**: Alignment with transcriptome
- **Input**: Nanofilt processed FASTQ
- **Method**: Minimap2 was used to align reads to the transcriptome.
- **Output**: bam

3 BAM QC Visualization

3.0.1 Analysis 6: `code/01b_fastq_to_bam/02_high_quality_bam_genome.sh`

- **Question/Aim**: Removes alignments with MAPQ < 30. Only keeps primary mapped alignments in chr 1-22, X, Y, M.
- **Input**: bam file from Analysis 4 and 5
- **Method**: samtools was used to filter the bam file, and create flagstat summary.
- **Output**: bam, summary stats on alignments.

3.0.2 Analysis 7: `code/02_bam_QC/01a_multi_exon_pcg_sample_specific.sh`

- **Question/Aim**: Exon-exon junction distribution
- **Input**: bam files produced in analysis 6
- **Method**: Multi-exon (PCG and all genes) junctions were quantified using python script `01_multi_exon_pcg.py` and `02_multi_exon_all_genes.py`. Visualization was done using R script `01_multi_exon_pcg.R` and `code/02_bam_QC/exon_exon_boxplots.R`
- **Output**: plots

3.0.3 Analysis 8: `code/02_bam_QC/read_type_percentages.R`

- **Question/Aim**: Percentage of alignments that are primary, supplementary, and unmapped.
- **Input**: `flagstat` file produced in analysis 4: `/retina_lrs/05_bams/genome/primary_assembly/logs/all.fl`
- **Method**: R script `read_type_percentages.R` was used to parse the flagstat file and create a bar plot of read types.
- **Output**: plots

3.0.4 Analysis 9:

- **Question/Aim**: Percentage of alignments that are primary, supplementary, and unmapped.
- **Input**: `flagstat` file produced in analysis 4: `/retina_lrs/05_bams/genome/primary_assembly/logs/all.fl`
- **Method**: R script `read_type_percentages.R` was used to parse the flagstat file and create a bar plot of read types.
- **Output**: plots

3.0.5 Quantification with `code/03_quantification/05_bambu`

3.0.6 Analysis 10:
`code/03_quantification/05_bambu/01_generate_sample_wise_read_class.sh`

- **Question/Aim**: Read class RDS files are generated for each sample individually.
- **Input**: bam files from analysis 6.
- **Method**: R script `read_type_percentages.R` was used to parse the flagstat file and create a bar plot of read types.
- **Output**: rds files

3.0.7 Analysis 11: `code/03_quantification/05_bambu/02_bambu_generate_rcs.sh`

- **Question/Aim**: Read classes are analyzed together for all samples to produce a common extended notation.
- **Input**: rds files from analysis 10.
- **Method**: R script `read_type_percentages.R` was used to parse the flagstat file and create a bar plot of read types.
- **Output**: final se object, counts matrix, extended annotation gtf.

3.0.8 Analysis 12: `code/03_quantification/05_bambu/05_sqanti.sh`

- **Question/Aim**: What type of novel isoforms were discovered?
- **Input**: `extended annotation` from analysis 11 and GENCODE references. polyA motif and CAGE peak experiments available from SQANTI example data.
- **Output**: `classification.txt`, CDS `gtf`, `corrected gtf` and `corrected fasta` files.

3.0.9 Analysis 13:
`code/03_quantification/05_bambu/07_gene_names_for_novel_isoforms.R`

- **Question/Aim**: Get gene names for isoforms that are common between `bambu` and `isoquant`.
- **Input**: `"/dcs04/hicks/data/sparthib/retina_lrs/06_quantification/bambu"`, `"bambu_isoquant_refmap.txt"` from analysis 15.
- **Method**: R script for getting the gene names of common novel isoforms between bambu and isoquant.
- **Output**: tsv

TODO: Archive `03_bambu_quantification.R`, `04_switch_plots.R`, `06_number_of_isoforms_per_gene.R`

4 Isoquant

5 Analysis 14: `code/03_quantification/01_IsoQuant/isoquant_all_samples.sh`

- **Question/Aim**: Isoquant quantification of all samples.
- **Input**: `bam` files from analysis 6.
- **Method**: Isoquant was used to quantify the reads.
- **Output**: `isoquant` output files including counts matrix, extended annotation, and `SQANTI3` like output of quality of isoforms.

6 Compare gtfs

6.0.1 Analysis 15: `code/03_quantification/10_compare_gtfs/compare_gtfs.sh`

- **Question/Aim**: Compare the GTFs produced by bambu and isoquant.
- **Input**: GTF files from bambu and isoquant.
- **Method**: `gffcompare` was used to compare the GTF files and produce a summary of the differences.
- **Output**: `txt` file of common isoforms between bambu and isoquant.

7 Cleaning up counts matrix

7.0.1 Analysis 16:
`code/03_quantification/11_cleaning_up_counts_matrix/cleaning_up_counts_matrix.R`

- **Question/Aim**: Clean up column and row names.
- **Input**: Counts matrix from bambu.
- **Output**: FT vs RGC and ROs specific gene and isoforms counts matrices.

7.0.2 Analysis 17: `code/04_dtu_dge_dte/01b_filter_matrix_by_common_isoforms.R`

- **Question/Aim**: Filter the counts matrix by common isoforms between bambu and isoquant.
- **Input**: Isoform counts matrices from Analysis 16, and output from Analysis 15.
- **Method**: R script `filter_matrix_by_common_isoforms.R` was used to filter the counts matrix by common isoforms between bambu and isoquant and other known isoforms.

7.0.3 Analysis 18: `code/04_dtu_dge_dte/01c_filter_by_gene_biotypes.R`

- **Question/Aim**: Filter the counts matrix by gene biotypes.
- **Input**: Counts matrices from Analysis 17.
- **Method**: R script `filter_by_gene_biotypes.R` was used to filter the counts matrix to only keep protein coding genes. `edgeR::filterByExpr` was used to filter the counts matrix by expression levels, for gene counts and isoform counts separately, and converted to cpm.
- **Output**: PCG gene and isoform counts and cpm matrices.

8 DTU DGE DTE Analysis

8.0.1 Analysis 19:

`code/04_dtu_dge_dte/bambu/FT_vs_RGC/bambu_FT_vs_RGC_DTE_DGE.R`
`code/04_dtu_dge_dte/bambu/ROs/bambu_ROs_DGE_DTE.R code/04_dtu_dge_dte/bambu/RO_vs_RGC/RO_vs_RGC`

- **Question/Aim**: Differential transcript expression (DTE) and differential gene expression (DGE) analysis.
- **Input**: counts matrix from bambu.
- **Method**: R script `____DTE_DGE.R` was used to perform DTE and DGE analysis for 1. between FT and RGC, 2. between RO stages, 3. among RO stages and RGCs.
- **Output**: `tsv` files of DGE and DTE results.

8.0.2 Analysis 20:

`code/04_dtu_dge_dte/bambu/FT_vs_RGC/bambu_FT_vs_RGC_DTU.R` `code/04_dtu_dge_dte/bambu/ROs/bambu_R` `code/04_dtu_dge_dte/bambu/RO_vs_RGC/RO_vs_RGC_DTU.R` - **Question/Aim**: Differential transcript usage (DTU) analysis using `IsoformSwitchAnalyzeR`.

- **Input**: counts and cpm matrix from Analysis 20, `extended annotation` from bambu, CDS annotation from SQANTI3.
- **Method**: R script `____DTU.R` was used to perform DTU analysis for 1. between FT and RGC, 2. between RO stages, 3. among RO stages and RGCs.
- **Output**: `tsv` files of DTU results, other files from `IsoformSwitchAnalyzeR` such as on splicing, switch consequences, switchplots.

8.0.3 Analysis 21: `code/04_dtu_dge_dte/pfam/external_protein_analysis.sh`

- **Question/Aim**: `Pfam domain analysis` , `SignalP` and `CPC2`.
- **Input**: `SwitchAnalysisPart1` input from Analysis 20.
- **Method**: `CPC2` was used to predict coding potential, `SignalP` was used to predict signal peptides, and `Pfam` was used to predict protein domains.
- **Output**: `tsv` files of coding potential, signal peptides, and protein domains incorporated into the switch plots in Analysis 20.

8.0.4 Analysis 22: `code/04_dtu_dge_dte/02_create_DGE_DTE_DTU.R`

- **Question/Aim**: Create a summary of DGE, DTE, and DTU results.
- **Input**: DGE, DTE, and DTU results from Analysis 19 and 21.
- **Method**: R script `create_DGE_DTE_DTU.R` was used to create a summary of DGE, DTE, and DTU results.
- **Output**: merged `tsv` files of DGE, DTE, and DTU results.

9 DGE_DTE_DTU Visualization

9.0.1 Analysis 23: `code/05_visualization/01_PCA.R`

- **Question/Aim**: PCA of gene and isoform expression for all comparisons.
- **Input**: CPM matrix from Analysis 18.
- **Method**: R script `PCA.R` was used to perform PCA analysis on the counts matrix.
- **Output**: PCA plots for gene and isoform expression.

9.0.2 Analysis 24: `code/05_visualization/02_heatmaps.R`

- **Question/Aim**: Heatmaps of gene and isoform expression for DGE genes and DTE/DTU isoforms for all comparisons.
- **Input**: CPM matrix from Analysis 18.
- **Method**: `ComplexHeatmap` was used to create heatmaps of gene and isoform expression.
- **Output**: Heatmap pdfs for gene and isoform expression.

9.0.3 Analysis 24: `code/05_visualization/03_volcano_plots.R`

- **Question/Aim**: Volcano plots of DGE, DTE, and DTU results.
- **Input**: DGE, DTE, and DTU results from Analysis 22.
- **Method**: R script `volcano_plots.R` was used to create volcano plots of DGE, DTE, and DTU results.
- **Output**: Volcano plots for DGE, DTE, and DTU results.

9.0.4 Analysis 25: `code/05_visualization/04_retnet_dtu_genes.R`

- **Question/Aim**: Which IRD genes have strong DTU or DTE events in our comparisons?
- **Input**: DGE, DTE, and DTU results from Analysis 22, gene list from RetNet database.
- **Method**: R script was used to create heatmaps for 30 or less genes with top DTU or DTE events.
- **Output**: Heatmaps of IRD genes with DTE and DTU events in all comparisons.

9.0.5 Analysis 26: `code/05_visualization/04b_retnet_dtu_switchplots.R`

- **Question/Aim**: Switchplots of DTU and DTE genes from Analysis 25.
- **Input**: SwitchAnalyzeR results from Analysis 22, gene list from Analysis 25.
- **Method**: R script was used to create switchplots for 30 or less genes with top DTU or DTE events.
- **Output**: Switchplots of IRD genes with DTE and DTU events in all comparisons.

9.0.6 Analysis 27: `code/05_visualization/05_splicing_factor_analysis.R`

- **Question/Aim**: Which splicing factor genes have strong DTE or DTU events in our comparisons?
- **Input**: DGE, DTE, and DTU results from Analysis 22, gene list from gene cards.
- **Method**: R script was used to create heatmaps similar to Analysis 25 for splicing factors.
- **Output**: Heatmaps of splicing factor genes with DTE and DTU events in all comparisons.

9.0.7 Analysis 28: `code/05_visualization/05b_splicing_factor_volcano.R`

- **Question/Aim**: Volcano plots of DGE, DTE, and DTU results for splicing factor genes.
- **Input**: DGE, DTE, and DTU results from Analysis 22, gene list from gene cards.
- **Method**: R script was used to create volcano plots for splicing factor genes.

9.0.8 Analysis 29: `code/05_visualization/06_splicing_factor_switchplots.R`

- **Question/Aim**: Switchplots of DTU and DTE splicing factor genes from Analysis 28
- **Input**: SwitchAnalyzeR results from Analysis 20, gene list from Analysis 27.
- **Method**: R script was used to create switchplots for 30 or less splicing factor genes with top DTU or DTE events.

9.0.9 Analysis 30: `code/05_visualization/06_go_analysis.R`

- **Question/Aim**: What are the main biological processes that are associated at the gene and isoform level with DGE, DTE, and DTU for each comparison?
- **Input**: DGE, DTE, and DTU results from Analysis 22.
- **Method**: R script was used to perform GO analysis using `clusterProfiler`.
- **Output**: `enrichGO` dotplots for DGE, DTE, and DTU genes and isoforms.

9.0.10 Analysis 31: `code/05_visualization/07_upset.R`

- **Question/Aim**: What are the common genes and isoforms between DGE, DTE, and DTU for each comparison?
- **Input**: DGE, DTE, and DTU results from Analysis 22.
- **Method**: UpsetR was used to create upset plots for common genes and isoforms between DGE, DTE, and DTU for each comparison.

9.0.11 Analysis 32: `code/05_visualization/07b_upset_only_DTUs.R`

- **Question/Aim**: What are the genes that had DTU events common between multiple RO stages? Which ones were unique to a certain pairwise comparisons?
- **Input**: DTU results from Analysis 22.
- **Method**: UpsetR was used to create upset plots for common DTU genes between multiple RO stages.
- **Output**: Upset plot of DTU genes across the multiple RO stages.

9.0.12 Analysis 33: `code/05_visualization/07c_DTU_only_GO_analysis.R`

- **Question/Aim**: What are the main biological processes that are associated with the 166 genes that were DTU across all stage comparisons?
- **Input**: DTU results from Analysis 22.
- **Method**: R script was used to perform GO analysis using `clusterProfiler`.
- **Output**: `enrichGO` dotplot for DTU genes across all stage comparisons.

9.0.13 Analysis 34: `code/05_visualization/08_short_read_comparison.R`

- **Question/Aim**: How does the long read gene expression in our RO samples compare to the short read gene expression in a previous study?
- **Input**: RO gene CPM matrix from Analysis 18.
- **Method**: R script was used to create a geom tile heatmap of the sample-wise spearman correlations.
- **Output**: Heatmap of sample-wise spearman correlations between long read and short read gene expression across the different RO stages.

9.0.14 Analysis 35: `code/05_visualization/09_isoforms_per_gene.R`

- **Question/Aim**: How many isoforms are there per gene in our RO samples?
- **Input**: `extended annotation` from bambu filtered by isoforms present in Analysis 18 of all RO samples.
- **Method**: R script was used to create the distribution of isoform counts per gene number across samples.
- **Output**: Barplot of isoforms per gene across samples.

TODO: remove `code/06_rbp_analysis` dir TODO: archive `code/09_RBFOX_motif_genes` dir

10 Allele Specific Analysis

10.0.1 Analysis 36: `code/08_ASE/short_reads/01_bowtie.sh`

- **Question/Aim**: Align publicly available H9 WGS short-reads.
- **Input**: FASTQ files for samples `SRR1091088`, `SRR1091091` and `SRR1091092`.
- **Method**: `bowtie2` was used to align the reads to the genome.
- **Output**: bam files for each sample.

10.0.2 Analysis 37: `code/08_ASE/short_reads/02_filter_bams.sh`

- **Question/Aim**: Filter the bam files to only keep high quality primary mapped alignments in chr 1-22, X, Y, M.
- **Input**: bam files merged from Analysis 36.
- **Method**: `gatk` samtools' were used for QC.
- **Output**: filtered bam files.

10.0.3 Analysis 38: `code/08_ASE/short_reads/03_bam2vcf.sh`

- **Question/Aim**: Convert the bam files to VCF format.
- **Input**: filtered bam files from Analysis 37.
- **Method**: `gatk` was used to convert the bam files to VCF format and QC.
- **Output**: joint VCF file with only high quality SNPs and INDELs.

10.0.4 Analysis 39: `code/08_ASE/short_reads/03b_vcf_stats.sh`

- **Question/Aim**: Get statistics on the VCF file.
- **Input**: VCF file from Analysis 38.
- **Method**: `vcfstats` was used to get statistics on the VCF file.
- **Output**: VCF stats plots on VCF file phased using H9 and EP1 long reads by `whatshap` in Analysis 40.

10.0.5 Analysis 40: `code/08_ASE/short_reads/04_whatshap_phase.sh`

- **Question/Aim**: Phase the VCF file using long reads.
- **Input**: VCF file from Analysis 38, long read bam files from Analysis 6.
- **Method**: `whatshap` was used to phase the VCF file using long reads.
- **Output**: phased VCF file.

10.0.6 Analysis 41: `code/08_ASE/short_reads/05_whatshap_haplotag.sh`

- **Question/Aim**: Haplotag the long read bam files with the phased VCF file.
- **Input**: phased VCF file from Analysis 40, long read bam files from Analysis 6.
- **Method**: `whatshap` was used to haplotag the long read bam files with the phased VCF file.
- **Output**: haplotagged long read bam files split into `HP1` and `HP2`.

10.0.7 Analysis 42: `code/08_ASE/short_reads/06_ase_read_counts.sh`

- **Question/Aim**: Get allele specific read counts from the haplotagged long read bam files.
- **Input**: haplotagged long read bam files from Analysis 41, phased VCF file from Analysis 40.
- **Method**: `featureCounts` was used to get allele specific read counts from the haplotagged long read bam files.
- **Output**: allele specific read counts matrix for all H9 and EP1 samples.

10.0.8 Analysis 43:

`code/08_ASE/short_reads/07_allele_spec_expression_modeling.R code/08_ASE/short_reads/07b_FT_vs_` `code/08_ASE/short_reads/07c_all_ROs_DGE.R` - **Question/Aim**: 1. What are the DEGs in the H9 samples between haplotypes? 2. What are the DEGs between the two haplotypes accounting for cell type FT, RGC, or ROs? 3. What are the DEGs between celltypes FT, RGC, or ROs accounting for haplotype? - **Input**: DEGs from Analysis 42. - **Method**: R script `allele_spec_expression_modeling.R` was used to model allele specific expression using `edgeR`. - **Output**: `tsv` files of DEGs between the two haplotypes accounting for cell type FT, RGC, or ROs, and DEGs between celltypes FT, RGC, or ROs accounting for haplotype. Volcano plot of DEGs. GO plot of DEGs.

10.0.9 Analysis 44: `code/08_ASE/short_reads/08_chr_viz.R`

- **Question/Aim**: Find the hromosome Distribution of the DEGs.
- **Input**: DEGs from Analysis 42.
- **Method**: R script `chr_viz.R` was used to visualize the chromosome distribution of the DEGs.
- **Output**: Barplot of chromosome distribution of the DEGs.

10.0.10 Analysis 45: `code/08_ASE/short_reads/08b_num_variants_per_read.R`

- **Question/Aim**: How many variants are there per read in the untagged long read BAM files from Analysis 6?
- **Input**: Long read bam files from Analysis 6, phased VCF file from Analysis 40.
- **Method**: `findOverlaps` function was used to count the number of variants per read in the untagged long read BAM files.
- **Output**: `tsv` and histogram of number of variants per read in the untagged long read BAM files.

10.0.11 Analysis 46: `code/08_ASE/short_reads/08c_variant_percent_all_samples.R`

- **Question/Aim**: What is the percentage of reads that have a specific number of variants in the untagged long read BAM files from Analysis 6?
- **Input**: `tsv` file from Analysis 45.
- **Method**: R script `variant_percent_all_samples.R` was used to calculate the percentage of reads that have a specific number of variants in the untagged long read BAM files.
- **Output**: `tsv` file combined across samples from Analysis 45 and histogram of percent of reads that have a specific number of variants faceted by sample.