RNA-Seq Data Analysis of Bipolar Disorder

Dohyoung Ko, Akshay Nataraja

Department of Computer Science and Engineering, The Pennsylvania State University

December 15 2024

Abstract

This thesis presents a comprehensive analysis of RNA-Seq data to investigate the molecular underpinnings of bipolar disorder (BD), alongside the implementation of a computational pipeline optimized for high-performance computing (HPC) environments. The pipeline performs a robust and systematic analysis of raw Illumina high-throughput RNA-Seq reads, ensuring efficient and accurate data processing. The study outlines the rationale, methodology, and implementation details of each step in the RNA-Seq data analysis workflow. RNA-Seq data from the human dorsal striatum were utilized to profile transcriptomes, comparing cohorts of bipolar disorder patients to healthy controls. Differential gene expression (DGE) analysis was conducted using the `DESeq2` library to identify genes exhibiting significant changes in expression associated with bipolar disorder status. The results were further analyzed to uncover sets of co-expressed and correlated genes, providing insights into their potential association with bipolar disorder.

Contents

1 Data Analysis Workflow

This section provides a comprehensive overview of the computational pipeline developed for RNA-Seq data analysis, focusing on the standard workflow for differential gene expression (DGE) analysis. Given the raw sequencing reads stored in `FASTQ` files, RNA-Seq data analysis proceeds through several computational stages, each designed to process, align, and quantify the data systematically. While there are multiple possible approaches to analyzing RNA-Seq data depending on the dataset and study design, for this analysis, we aligned short Illumina reads to the reference genome to ensure robust and biologically relevant results. The overall workflow adopted for DGE analysis is depicted in Figure 1.

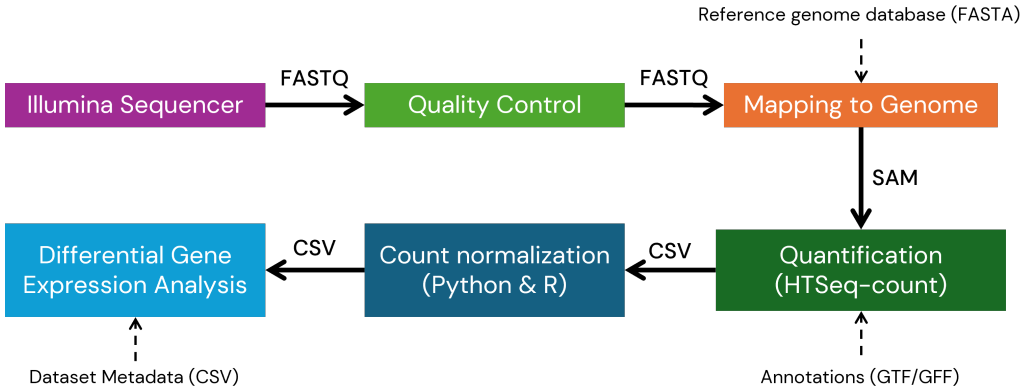


Figure 1: Workflow of Differential Gene Expression Analysis

As a preliminary step, the raw Illumina sequencing reads encoded in `FASTQ` files were pre-processed to ensure high-quality data for downstream alignment and quantification. Pre-processing included demultiplexing reads based on indices or barcodes, removing residual adapter sequences introduced during library preparation, trimming bases with low quality scores, and discarding reads that were ambiguous or of insufficient quality. These steps are essential for minimizing alignment errors and ensuring the reliability of subsequent analyses.

The pre-processed reads were then aligned to the reference genome using HISAT2, a fast and memory-efficient aligner widely used for RNA-Seq data. HISAT2 begins by indexing the reference genome, which involves creating a hash table of short, non-overlapping substrings that facilitate efficient alignment. Each sequencing read is then scanned and mapped to its most likely genomic location by matching it against these indexed substrings. The output of the alignment step is a `SAM` (Sequence Alignment/Map) file, a widely adopted format for storing sequence alignments.

The SAM file consists of two main sections: a header section and an alignment section. The header section, identified by lines starting with the `@` symbol, contains metadata about the alignment process. For instance, the `@HD` line specifies the file format version and the sorting order of alignments, while the `@SQ` lines define the reference sequences used during alignment. In these lines, the sequence name (`SN`) identifies each genomic feature (such as `NC_000001.11`, representing chromosome 1), and the sequence length (`LN`) provides the total number of base pairs in that feature. The alignment section follows the header and contains detailed information for each mapped read, including its genomic position, mapping quality score, and additional aligner-specific parameters. This structure ensures that the SAM file comprehensively represents the alignment results while preserving flexibility for downstream analyses.

Once the reads were aligned, the next step involved quantifying gene expression levels by determining the number of reads that overlapped annotated genomic features. For this purpose, gene annotation data in `GTF` format was sourced from the NCBI *Homo sapiens* genome database. This annotation file provides detailed genomic coordinates and attributes for features such as exons and genes, enabling precise quantification. The `htseq-count` tool was used to compute gene-level counts from the alignment data. `htseq-count` systematically compares each aligned read in the SAM file to the genomic features defined in the annotation file. Reads that overlap with a single genomic feature, such as an exon, are assigned to that feature, and ambiguously mapped reads (those overlapping multiple features) are excluded to prevent miscounts. The resulting counts were aggregated at the gene level based on the `gene_id` attribute specified in the annotation file. This step produced a raw count file in tabular format, where each row corresponds to a gene and its associated read count.

To prepare for downstream differential gene expression analysis, the count data for all cohort samples were combined

into a single count matrix. This matrix was subsequently analyzed using the DESeq2 library in R, a widely used statistical tool for RNA-Seq data analysis. DESeq2 applies robust normalization and statistical modeling to identify genes with significant changes in expression levels between experimental groups. In this study, the comparison focused on cohorts of bipolar disorder patients and healthy controls, with the objective of identifying genes whose expression levels were significantly altered in association with bipolar disorder.

The results of the DGE analysis provided a list of differentially expressed genes, ranked according to their statistical significance and magnitude of expression change. These findings were further analyzed to identify sets of correlated or co-expressed genes, offering insights into potential biological pathways and molecular mechanisms underlying bipolar disorder.

2 Computational Pipeline

This section discusses the design principles and implementation of the computational pipeline used to execute the RNA-Seq data analysis workflow described above. All computational processes were carried out on the Cedar heterogeneous computing cluster at Simon Fraser University, part of the Advanced Research Computing (ARC) service provided by the Digital Research Alliance of Canada. A single end-to-end execution of the pipeline took approximately 8 hours, utilizing 36 computing nodes with a total of 1,152 CPU cores and 800 gigabytes of memory on the Cedar cluster.

2.1 Design and Specifications

The computational pipeline, which streamlines the complete RNA-Seq differential gene expression (DGE) analysis process, is publicly available on GitHub at `https://github.com/Codakshay/RNA_Seq_Final_Project.git`. Designed as a command-line application, the pipeline offers an efficient, automated, and reproducible solution for high-performance computing (HPC) environments. It integrates multiple tools and processes seamlessly using the `slurm` workload manager and runs on Linux operating systems such as AlmaLinux 9 and CentOS 7.

The pipeline is implemented as a modular framework, combining Python, R, shell scripts, and a `Snakemake` workflow to ensure flexibility, scalability, and user-friendliness. By leveraging `Snakemake`, the pipeline allows task parallelization and efficient resource management across multiple compute nodes. This design significantly reduces processing time, making it suitable for large-scale RNA-Seq datasets.

Operating System	Linux (AlmaLinux 9, CentOS 7)
Workload Management	slurm
Cores	minimum of 32 cores
Memory	minimum of 50 gigabytes + total size of FASTQ files
External Dependencies	python ($\geq$ 3.9.0), R ($\geq$ 4.0.5)
Estimated Processing Time	24-48 hours depending on the FASTQ data size

Table 1: System Requirements

To ensure optimal performance and reproducibility, the pipeline adheres to the specifications outlined in Table 1. These requirements include the minimum number of CPU cores, memory size, and software dependencies necessary for the successful execution of the analysis. Depending on the size of the input `FASTQ` files, the estimated runtime for a complete analysis ranges between 24 and 48 hours.

By consolidating all steps of RNA-Seq data analysis into a single pipeline, this implementation provides an effective one-click solution for large-scale genomic studies in high performance computing (HPC) environments.

2.2 Installation

We recommend installing the pipeline directly from the GitHub repository to ensure access to the most recent version, including any updates or fixes. To install the pipeline on your computing cluster, execute the following command:

```
git clone https://github.com/Codakshay/RNA_Seq_Final_Project.git
```

For all subsequent steps, ensure you navigate to the root directory of the cloned repository.

Before running the pipeline, it is essential to verify that your computing cluster supports both R and Python as available modules. Most high-performance computing (HPC) environments use module management systems (e.g., `module load`) to provide software packages. You should confirm that R (version $\geq$ 4.0.5) and Python (version $\geq$ 3.9.0) are accessible. If these are not pre-installed, contact your system administrator to install or enable them.

Additionally, ensure that all Python dependencies listed in the `requirements.txt` file located in the root directory of the repository are installed on the cluster. The required libraries can be installed locally in a virtual environment or within a Conda environment. To set up the environment and install the necessary libraries, execute the following commands:

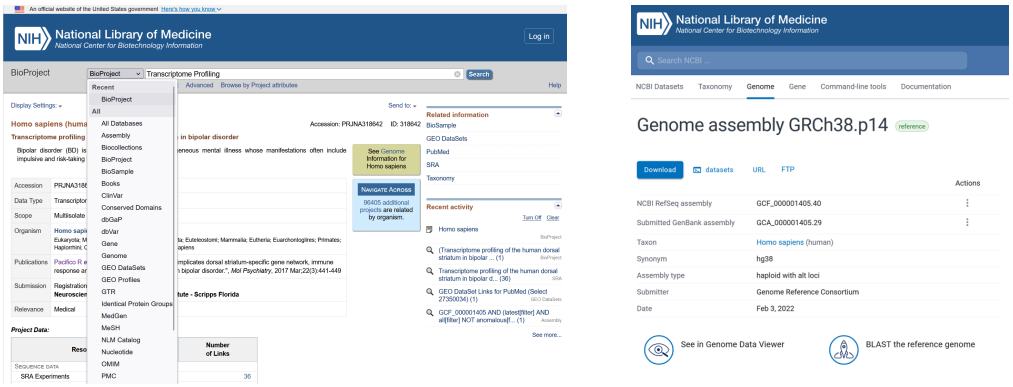
```
python3 -m venv env source env/bin/activate pip install -r requirements.txt
```

This will create an isolated virtual environment containing all the required Python dependencies for running the pipeline. If you encounter any issues, verify that the correct Python version is being used and that the required libraries are available.

2.3 Running the Pipeline

This subsection provides a concise manual for executing the computational pipeline on a high-performance computing (HPC) environment. Before proceeding, please verify that your computing system meets the design specifications outlined in the previous subsection.

The pipeline requires only a BioProject ID as input. At its current stage, the pipeline exclusively supports RNA-Seq differential gene expression analysis for Homo sapiens data. The necessary datasets, including raw sequencing reads and reference annotations, are automatically retrieved from the NCBI FTP server by the pipeline script, as illustrated in Figure 2.



(a) Search for BioProject ID on NCBI Database

(b) Human Reference Genome in FASTA and GTF annotations

Figure 2: BioProject ID and Human Reference Genome on NCBI Database

This application requires only a BioProject ID as input. At its current stage, the pipeline exclusively supports RNA-Seq differential gene expression analysis for Homo sapiens datasets. The required sequencing data, including raw reads and associated annotations, are automatically retrieved from the NCBI FTP server by the pipeline script, as illustrated in Figure 2(b). Future iterations of the pipeline will expand support to RNA-Seq datasets from additional species to enable broader applicability.

In the NCBI Sequence Read Archive (SRA) database, BioProject IDs provide unique identifiers for retrieving all raw sequencing datasets associated with a particular project. These identifiers are typically formatted as `PRJNA` followed by a series of digits. BioProject IDs can be located either via the Gene Expression Omnibus (GEO) database or by directly searching the NCBI database. As shown in Figure 2, users can find the BioProject ID by selecting "BioProject" in the search criteria. Once the appropriate BioProject ID is retrieved, users must navigate to the root directory of the cloned pipeline repository and execute the following command on their computing cluster:

```
sbatch run_pipeline.sh PRJNA123456 24:00:00
```

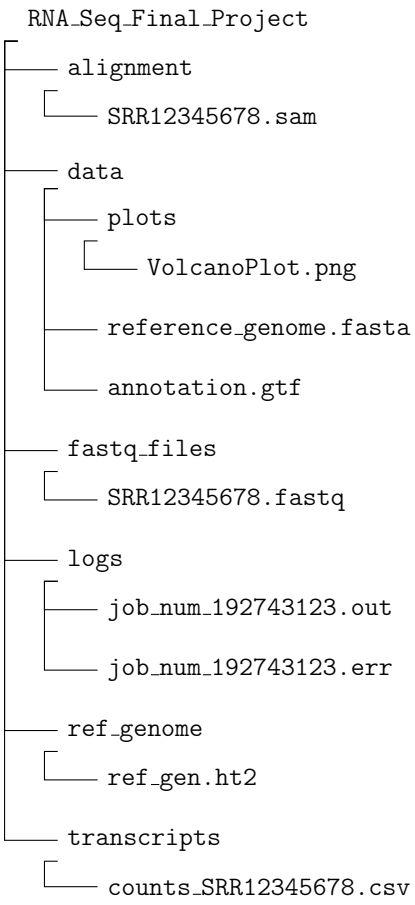
In this command, `PRJNA123456` represents the BioProject ID, and `24:00:00` specifies the maximum expected execution time in `HH:MM:SS` format. While the total processing time depends on the number of `FASTQ` files and their sizes, typical runtimes range between 24 and 48 hours. For the case study conducted in this work, the pipeline completed execution in approximately 12 hours.

It is important to note that if the specified time limit is insufficient, the job will be automatically terminated by the SLURM workload manager. In such cases, the pipeline must be re-executed with an increased time allocation. Users are advised to monitor the execution status of their job by running the `sq` command, which provides real-time updates on job progress within the computing cluster.

While the execution process itself does not require user intervention, the status and potential errors can be monitored by inspecting the `.out` and `.err` files generated in the `RNA_Seq_Final_Project/logs/` directory. These log files provide detailed information about the ongoing pipeline execution, including standard outputs and any error messages encountered during runtime.

2.4 Retrieving Results

Upon successful execution of the pipeline, the output consists of the results of the Differential Gene Expression (DGE) analysis along with all intermediate data objects generated during processing. This subsection describes the structure of the output files and directories, providing an overview of how the results are organized. The directory structure after pipeline execution is illustrated in the file tree diagram on the right.



All preprocessed data, intermediate files, and final DGE analysis results are stored within the root directory of the pipeline. The key directories and their contents are as follows:

The `alignment` directory contains the SAM files generated by the HISAT2 alignment tool. These files store the processed alignments of the input sequencing reads (in `FASTQ` format) to the reference genome.

The `data` directory serves as the repository for processed data and results relevant to downstream analyses. It includes two primary components: the `plots` subdirectory, which contains visualizations such as volcano plots produced during DGE analysis, and the reference files used for alignment. Specifically, the directory includes the reference genome in `FASTA` format and its corresponding annotation file in `GTF` format. Both files are retrieved from the NCBI FTP server and are essential for mapping and quantifying sequencing reads.

The `fastq_files` directory houses the raw input sequencing data in `FASTQ` format, which is downloaded directly from the NCBI SRA database based on the provided BioProject ID. This organization facilitates easy access to the original data and supports reproducibility.

The `logs` directory systematically stores all workflow logs, including job-specific output (`.out`) and error (`.err`) files generated during pipeline execution. These log files provide detailed insights into the pipeline's performance and execution status, making them essential for monitoring and debugging the workflow on high-performance computing (HPC) clusters.

The `ref_genome` directory contains the indexed reference genome files required by the HISAT2 aligner. These preprocessed reference files are used during alignment to efficiently map sequencing reads to genomic coordinates.

Finally, the `transcripts` directory holds the gene expression count files in `CSV` format. Each file corresponds to a cohort sample from the analysis and contains raw read counts for individual genes. These counts serve as the input for the downstream DGE analysis, where statistical tools such as `DESeq2` are used to identify genes with significant changes in expression between experimental conditions.

3 Dataset Analysis

The subsequent analysis in the report employs the RNA-Seq dataset of the human dorsal striatum, which is comprised of 18 cohorts of individuals diagnosed with Bipolar Disorder (BD) and 18 control subjects.

3.1 Origin and Context of DataSet

As stated by R. Pacifico and RL Davis in [11], the data is derived from 36 frozen postmortem human striatum samples provided by the Harvard Brain Tissue Resource Center. These samples are accompanied by information regarding the age, sex, and postmortem interval (PMI) of sample collection from the donors. The original study excluded one outlier control sample, designated as C 28, and conducted differential gene expression (DGE) analysis on 35 samples (18 bipolar and 17 control). The covariates (age, sex, PMI, and RIN) for both the bipolar and control groups were not significantly different from each other. This report presents two results: one considering the inclusion of the C 28 control sample and another considering its exclusion. The objective is to observe the differences in the outcome of the DGE analysis. In bipolar disorder, it is especially important to consider how long the manic episode continues, and the data has been normalized to account for this constraint.

geo accession	title	age (years)	Sex	postmortem interval (hours)	rin
GSM2124738	control-2	83	male	13.000000	3.700000
GSM2124739	control-3	55	female	21.660000	2.900000
GSM2124771	bipolar-34	80	female	13.660000	3.600000
GSM2124772	bipolar-37	61	female	16.630000	2.200000
GSM2124773	bipolar-40	77	female	29.580000	6.700000

Table 2: Part of sampled cohort data

Table 2 refers to a part of the cohort data, which provides a more concrete description of the dataset.

Group	Age ($\mu \pm \sigma$)	Sex (Male, Female)	PMI ($\mu \pm \sigma$)	RIN ($\mu \pm \sigma$)
Control	65 $\pm$ 15.7	7, 11	22.8 $\pm$ 6.4	4 $\pm$ 1.7
Cohort	69.7 $\pm$ 15.3	6, 12	25.1 $\pm$ 15.8	3.9 $\pm$ 1.5

Table 3: Summary of cohorts

Table 3 presents a summary of the most notable statistics from the dataset pertaining to each cohort and control group. It should be noted that these statistics exclude the effect of an outlier, C_28, from the sample.

3.2 Data Preprocessing

This section outlines the specific data preprocessing operations required for RNA-Seq data analysis, as introduced earlier in the workflow overview. The preprocessing phase begins with handling the Illumina high-throughput sequencing data.

3.2.1 Illumina RNA Sequencing Data

RNA sequencing (RNA-Seq) is a powerful and widely adopted method for profiling gene expression, offering insights into which genes are active, their expression levels, and how these patterns vary under different biological conditions. RNA-Seq data is generated by sequencing RNA molecules extracted from a biological sample, followed by computational processing to reconstruct and quantify the transcriptome.

The RNA-Seq workflow typically begins with the extraction of RNA from the biological sample, followed by its conversion into complementary DNA (cDNA) via reverse transcription. The resulting cDNA is then sequenced using high-throughput sequencing technologies. Illumina sequencing platforms, such as the HiSeq-2000, are among the most widely used technologies for this purpose. They provide high accuracy, throughput, and cost-efficiency, making them ideal for large-scale transcriptome studies.

RNA-Seq data can be generated using two primary sequencing strategies: single-end and paired-end sequencing. Single-end sequencing reads a fragment of the cDNA molecule from one end, while paired-end sequencing generates reads from both ends of the fragment, improving alignment accuracy and coverage. The high-throughput sequencing dataset used in this analysis consists of reads generated by the Illumina HiSeq-2000 platform.

Figure 3 illustrates the process of obtaining RNA-Seq data from an RNA sample using the Illumina HiSeq-2000. In Figure 3(a), RNA extracted from a biological sample is sequenced to produce raw reads. Figure 3(b) shows the Illumina HiSeq-2000 machine used for sequencing, which is capable of producing millions of short reads in a single sequencing run.

(a) Sequencing the RNA sample on Illumina HiSeq-2000

(b) Illumina HiSeq-2000 Sequencing Machine

Figure 3: High-throughput sequencing machine

In most cases, the raw RNA-Seq data is available in the FASTQ format. The following code section 1 describes the typical FASTQ file of RNA-Seq data.

Source Code 1: Sample FASTQ output

```
@SRR3393492.1 HISEQ:313:D2H4NACXX:2:1101:1200:1867 length=88
NTCCGTCTTGCGCCGGGCCAAGAATTTCACCTCTAGCGGCGCAATACGAATGCCCCCGGCCGGCCCTCTTAATCATGGCCTCAGTTCC
+SRR3393492.1 HISEQ:313:D2H4NACXX:2:1101:1200:1867 length=88
#11?7?;AAACCACAA)??ABA3>AAAB0=ABBB<A><AA##################################################
```

Each record in a FASTQ file consists of four lines, which are repeated for every sequence in the file. The above data represents a single record. The first line begins with `@` and serves as the sequence identifier, providing a unique name for the read (`SRR3393492.1`) and additional metadata such as sequencing instrument (`HISEQ:313:D2H4NACXX`) and read length (`length=88`). The second line contains the raw nucleotide sequence (`NTCCGTCTTGCGCCGGGCCAAGAATT...`), representing the biological data obtained from the sequencing process. The third line starts with a + symbol and may either repeat the sequence identifier or remain blank, serving as a separator for the quality scores. The fourth line encodes the quality scores using ASCII characters (`#11?7?;AAACCACAA)...`), which represent Phred scores for each nucleotide.

The Phred score is a numerical value used to represent the quality of a nucleotide base call. It is a logarithmic measure of the probability P that a base call is incorrect. Higher Phred scores indicate higher confidence in the accuracy of the base call. Given that P is the probability of the base call being incorrect, its Phred score $Q = -10\log_{10}(P)$ is computed and encoded into an ASCII character by adding 33 to it. For instance, the character *I*, which corresponds to ASCII 73, indicates a Phred score of 40.

Our computational pipeline downloads all FASTQ files from the NCBI SRA database. These raw sequencing data is then assembled into SAM files via HISAT2.

3.2.2 Sequence Alignment Data

Before performing the sequence alignment, it is essential to construct the reference genome database, which serves as the foundation for aligning sequencing reads. The computational pipeline generates the reference genome database from the NCBI *Homo sapiens* reference genome dataset using the following command:

```
hisat2-build -p 32 "data/GCF_000001405.40_GRCh38.p14_genomic.fna" "ref_genome/ref_gen"
```

In this command, `hisat2-build` creates index files necessary for alignment using 32 parallel threads (`-p 32`). The input file is the reference genome in `.fna` format, while the output consists of indexed files stored under the specified prefix `ref_genome/ref_gen`. These index files allow efficient alignment of sequencing reads to the reference genome.

Once the reference genome database has been constructed, the pipeline proceeds with the alignment of sequencing reads (in `FASTQ` format) against the reference genome using HISAT2. The following command is executed:

```
hisat2 -p 32 -x ref_genome/ref_gen -U "fastq_files/SRR3393492.fastq" \
       -S "alignment/SRR3393492.sam"
```

In this alignment step, `-p 32` specifies the use of 32 threads for multi-threaded processing. The tag `-x` points to the prefix of the reference genome index files created in the previous step. Also, `-U` specifies the input file containing single-end sequencing reads (`SRR3393492.fastq`). Finally, `-S` indicates the output file path, where the alignment results are stored in SAM (Sequence Alignment/Map) format.

The resulting SAM file contains essential information for each aligned read, including its position on the reference genome, alignment flags, mapping quality scores, and more. The header section of the SAM file provides metadata about the reference genome sequences used during alignment. Table 4 shows the first few rows of the header section from the processed SAM file corresponding to cohort ID `SRR3393492`.

Record Type	Sequence Name (SN)	Sequence Length (LN)
@SQ	NC_000001.11	248956422
@SQ	NT_187361.1	175055
@SQ	NT_187362.1	32032
@SQ	NT_187363.1	127682
@SQ	NT_187364.1	66860
@SQ	NT_187365.1	40176

Table 4: Header section of the SAM file for `SRR3393492`.

This SAM file header section begins with lines prefixed by the `@` symbol, which provide information about the reference genome sequences used for alignment. Each line in the header contains the following fields.

First, the record type field is specified by the tag `@SQ` which indicates a sequence record that describes a sequence from the reference genome. Next, sequence name (SN) field provides the name or unique identifier of the reference genome sequence. For instance, `NC_000001.11` corresponds to the primary assembly of chromosome 1. Other identifiers, such as `NT_187361.1`, represent unplaced or alternate contigs. Lastly, sequence length (LM) field specifies the length of the corresponding sequence in base pairs (bp). For example, chromosome 1 (`NC_000001.11`) has a length of 248,956,422 bp. This header ensures proper mapping of sequencing reads to their corresponding positions in the reference genome, while also enabling downstream tools to interpret the aligned data accurately.

3.2.3 Raw Read Count Data

For each alignment SAM file, the pipeline performs read count generation by executing the following command.

```
htseq-count "alignment/SRR3393492.sam" "data/GCF_000001405.40_GRCh38.p14_genomic.gtf.gz"
            -f sam -r pos -i gene_id -t exon -n 32 > "transcripts/SRR3393492.csv"
```

3.2.4 Raw Read Count Data

For each alignment SAM file, the pipeline generates raw read count data using the `htseq-count` tool. This process quantifies the number of reads aligned to annotated genomic features such as exons, enabling downstream analyses like differential gene expression. The pipeline executes the following command.

```
htseq-count "alignment/SRR3393492.sam" "data/GCF_000001405.40_GRCh38.p14_genomic.gtf.gz" \
            -f sam -r pos -i gene_id -t exon -n 32 > "transcripts/SRR3393492.csv"
```

Let us take a closer look at each argument in the above command. First, `"alignment/SRR3393492.sam"` specifies the input SAM file containing aligned sequencing reads. Notice that each read in this file has already been mapped to the reference genome using HISAT2. Next, `"data/GCF_000001405.40_GRCh38.p14_genomic.gtf.gz"` refers to the gene annotation file in `GTF` (Gene Transfer Format). This file contains information about genomic features,

such as gene locations, transcript structures, and exons, which are essential for counting reads overlapping these regions.

HTSeq can process different alignment formats such as SAM and BAM. The argument `-f sam` specifies in detail that the input file format is SAM. Next, `-r pos` option instructs HTSeq to process reads in positional order, sorted by genomic position, for efficiency during the counting procedure. This ensures optimal handling of the aligned data.

Next, `-i gene_id` option identifies the attribute in the GTF file to be used as the feature ID. In this case, `gene_id` is used to group reads by gene identifiers. Following argument `-t exon` specifies that the counting should occur at the `exon` feature level. By enabling this option, reads overlapping exonic regions will be aggregated to calculate gene-level counts. Similar to HISAT2, HTSeq-count also support multi-threading, and `-n 32` enables multi-threading with 32 threads to accelerate the counting process. Then, the last argument `> "transcripts/SRR3393492.csv"` simply redirects the output to a CSV file. Each line of this file contains the feature ID (gene ID) and its corresponding raw read count.

In more detail, HTSeq-count library generates read counts in the following process. First, `htseq-count` parses the GTF file to extract genomic feature coordinates such as exons and their associated identifiers like `gene_id`. Here, each genomic feature is treated as a region of interest for counting reads. This step is called *Feature Extraction*. Then, for each read in the SAM file, `htseq-count` determines whether the read overlaps any genomic feature based on its aligned position. Overlap is determined using the genomic coordinates of the read and features.

Next step is referred to *Feature Assignment*. If a read overlaps exactly one feature (e.g., an exon), it is assigned to that feature. Reads overlapping multiple features are handled based on the default "union" mode, where reads are only counted if they unambiguously belong to a single feature.

Once the assignment is complete, Reads assigned to individual exons are aggregated at the gene level using the `gene_id` attribute to calculate raw gene-level counts. Afterwards, HTSeq-count creates the final output is a tab-delimited file where each line contains a gene ID and its corresponding raw read count.

Then, the raw expression data, that is, the dataset of raw read counts per gene, is obtained by merging all read counts CSV files obtained from the `HTSeq-count` library. Note that this data can also be retrieved from the supplementary files section of the Gene Expression Omnibus (GEO) repository. The relevant data can be accessed via the following URL: `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE80336`. The respective RNA-Seq data for over 36 samples have been sequenced using the Illumina HiSeq 2000 platform. Table 5 provides a subset of the raw count information.

Ensembl_ID	GeneSymbol	Biotype	Chromosome	C_2	C_3	BD_34	BD_37	BD_40
ENSG00000000003	TSPAN6	protein_coding	X	87	68	128	91	99
ENSG00000000005	TNMD	protein_coding	X	0	0	0	2	1
ENSG00000000419	DPM1	protein_coding	20	129	195	197	149	124
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
ENSG00000001167	NFYA	protein_coding	6	363	455	594	464	413

Table 5: Raw count data

In this dataset, the Ensembl ID serves as the unique, stable identifier assigned to each gene, transcript, protein, and exon. Additionally, the letters "C" and "BD," utilized in column labels, represent abbreviations for "control" and "bipolar disorder," respectively. As this dataset is derived from *homo sapiens* samples, chromosome attributes indicate the chromosome to which the selected gene belongs, including autosomes 1-22 or sex chromosomes X or Y.

4 Read Count Normalization and Filtering

It is necessary to complete preliminary steps, sometimes called quality control, prior to applying differential gene expression (DGE) analysis. These steps include read count matrix normalization and pre-filtering.

4.1 Read Count Normalization

The first step in the analysis is to normalize the read counts from the pre-processed raw count matrix. Differential gene expression analysis tools compare counts between sample groups for a given gene. As such, some factors require attention, such as sequencing depth and RNA composition. The DESeq2 library uses the median of ratios method for normalization, one of the most commonly employed techniques.

This median of ratios method focuses on two critical factors during the normalization process.

The first factor is *RNA Composition*. It is typical for a few highly differentially expressed genes and variations in the number of expressed genes between samples and the presence of contamination to skew the results of specific normalization methods. Therefore, accounting for RNA composition leads to a more accurate comparison of gene expression across samples.

The second factor is *Sequencing Depth*. RNA sequencing data often exhibits uneven or high coverage rates. Consequently, a particular gene may be expressed at a ratio significantly higher in one sample than another. This discrepancy might reflect a higher sequencing depth in one sample. By considering sequencing depth, we ensure greater comparability of expression levels between and within samples when relevant.

Let us examine how DESeq2 performs normalization using the median of ratios method that accounts for both RNA composition and sequencing depth. DESeq2 first constructs a pseudo-reference sample from the existing raw count matrix by calculating the row-wise geometric mean. Specifically, for each gene ENSG(k) with Ensemble ID k in the dataset, the pseudo-reference sample is defined as follows:

$$\text{PseudoReference(ENSG(k))} = \sqrt{\prod_{n=1}^{|N|} \text{Count}(C_n[\text{ENSG}(k)])}. \tag{1}$$

This equation represents the geometric mean across all samples. For instance, as shown below, the new pseudo-reference sample attribute will be added to the summary table (Table 6).

Ensembl_ID	GeneSymbol	Chromo	C_2	C_3	BD_34	BD_37	BD_40	Pseudo-reference
ENSG00000000003	TSPAN6	X	87	68	128	91	99	82595.685311
ENSG00000000005	TNMD	X	0	0	0	2	1	1.414214
ENSG00000000419	DPM1	20	129	195	197	149	124	302586.292915
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
ENSG00000001167	NFYA	6	363	455	594	464	413	4335969.80759

Table 6: Read count matrix after normalization step 1

Next, DESeq2 calculates the ratio of each sample to a reference. Specifically, the ratio (sample/reference) is computed for every gene in a sample. This process is carried out for each sample in the dataset. Since most genes are not differentially expressed, the ratios for most genes within each sample should be similar. Table 7 presents the ratio computations from the above sample table.

Ensembl_ID	C_2/ref	C_3/ref	BD_34/ref	BD_37/ref	BD_40/ref	Pseudo-reference
ENSG00000000003	0.001053	0.000823	0.001550	0.001102	0.001199	82595.685311
ENSG00000000005	0	0	0	1.414213	0.707107	1.414214
ENSG00000000419	0.000426	0.000644	0.000651	0.000492	0.000410	302586.292915
⋮	⋮	⋮	⋮	⋮	⋮	⋮
ENSG00000001167	0.000084	0.000105	0.000137	0.000107	0.000095	4335969.80759

Table 7: Read count matrix after normalization step 2

Next, DESeq2 calculates the normalization factor for each sample. For a given sample, the normalization factor is determined by taking the median of the ratios of counts for that sample relative to a reference gene, as shown below:

$$\text{Normalize}(C_n) = \text{Med}\left(\{C_n/\text{ref}[\text{ENSG}(k)] : k \in [1, \ldots, |N|]\}\right).$$

The median of ratios method assumes that not all genes are differentially expressed. Therefore, the normalization factors account for the sample's sequencing depth and RNA composition. Table 8 lists the normalization factors for each sample based on the information from Table 5 above.

Cohort	C_n/ref[ENSG(3)]	C_n/ref[ENSG(5)]	C_n/ref[ENSG(419)]	C_n/ref[ENSG(1167)]	Normalize
C_2	0.001053	0	0.000426	0.001102	0.0007395
C_3	0.000823	0	0.000644	0.000105	0.0003745
BD_34	0.001550	0	0.000651	0.000137	0.0003940
BD_37	0.001102	1.414213	0.000492	0.000107	0.000797
BD_40	0.001199	0.707107	0.000410	0.000095	0.0008045

Table 8: Normalization constant for each cohort

Taking the median helps mitigate significant outlier genes' influence on the median ratio values. In this simplified example, the normalization constant is very close to zero. However, based on the original study and other differential gene expression (DGE) analyses, these size factors typically hover around one. The authors of the original study identified the presence of extreme outlier control samples by noting significant variations among the samples.

The final step involves calculating the normalized count values using the normalization factors. This step is done by dividing each raw count value in a given sample by that sample's normalization factor, thereby generating normalized count values. This process is applied to every count value across all genes in every sample. Table 9 presents the normalized count matrix derived from Table 5, which includes data from two control samples and three samples from patients with Bipolar Disorder.

Ensembl_ID	C_2	C_3	BD_34	BD_37	BD_40
ENSG00000000003	117647.058	181575.434	230964.467	114178.168	123057.799
ENSG00000000005	0	0	0	2509.410	1243.008
ENSG00000000419	174442.191	520694.259	500000	2509.41	154133.002
⋮	⋮	⋮	⋮	⋮	⋮
ENSG00000001167	490872.211	1214953.271	1507614.213	582183.186	513362.337

Table 9: Normalized read count matrix

It is important to note that the normalization constant's size and the resulting data frame differ from the actual dataset. In DESeq2, this procedure is automatically executed by running the following command in R.

```
dds <- DESeqDataSetFromMatrix(countData=counts_data, colData=coldata, design=~condition)
```

4.2 Pre-Filtering

According to the study's authors, it is highly recommended that the normalized count matrix be pre-filtered in addition to performing read count normalization. This approach offers two fundamental benefits. First, by removing rows with very few reads, we can reduce the memory size of the data object and speed up the transformation and testing functions in the DESeq2 library. Second, since features without any data for differential gene expression are not plotted or considered in the analysis, this filtering enhances the quality of data visualizations. We retain genes with a total count of at least 1 for our differential gene expression analysis in any of the samples. This threshold of 1 is a common practice in numerous studies, including [10] and [11]. The following script accomplishes this task.

```
keep <- rowSums(counts(dds)) >= 1
dds <- dds[keep,]
```

4.3 Principal Component Analysis (PCA)

The Principal Component Analysis (PCA) plot visualizes the variability in the dataset based on principal components (PCs). Each point in the plot represents a sample, labeled by its GSM identifier and grouped by condition, either control or bipolar.

Figure 4: PCA Plot

In Figure 4, we observe that the two cohorts—control and bipolar—show some degree of clustering, but there exists a partial overlap in the upper-left quadrant. This overlap suggests that while the groups are somewhat distinct, they share biological or technical variability that reduces their separation in the PCA space. The PCA analysis captures a combined variance of 41percent-sign for the first two principal components (PC1 and PC2). This relatively low variance suggests that the underlying variability in the dataset is distributed across multiple dimensions, potentially due to the high dimensionality of the data, biological complexity, or technical noise. One critical observation is the presence of an outlier, the sample labeled GSM2124750 which corresponds to the control-28 cohort. This sample was flagged as an outlier in the original study [11], likely due to either technical artifacts or biological differences that set it apart from the rest of the cohort. Outliers like this can distort PCA results by inflating variance in certain dimensions, thereby reducing the proportion of variance explained by the main principal components.

After removing the outlier from the dataset, we observed a decrease in the variance captured by the first two PCs. The previous study highlighted that clinical variables such as bipolar subtype, medication use, mood state, and other endophenotypes were not known for the bipolar cohort [11], likely contributing to decreased PCA capture. These unaccounted for clinical variables could represent significant sources of heterogeneity within the cohort, further complicating the separation between the control and bipolar groups.

5 Differential Gene Expression Analysis

This section will discuss the implementation and interpretation of differential gene expression (DGE) analysis using the GEO DataSet mentioned above. The following script creates a DESeqData object from the pre-processed raw count matrix and performs the DGE analysis.

```
# Construct DESeqData object from raw count matrix
dds <- DESeqDataSetFromMatrix(countData=counts_data, colData=coldata, design=~ condition)

# Pre-filtering low-count genes
dds <- dds[rowSums(counts(dds)) >= 1, ]

# Set reference level
dds$condition <- relevel(dds$condition, ref = "control")

# Run Differential Gene Expression analysis
dds <- DESeq(dds)
```

Then, the following script unpacks the result of the DEG analysis.

```
# Extract and summarize results
res <- results(dds, contrast = c("condition", "control", "bipolar"), alpha = 0.05)
```

The `results` function calculates log2 fold changes, p-values, and adjusted p-values using the Benjamini-Hochberg method for each gene in the provided read count matrix. The contrast argument specifies the comparison of interest; in this case, we are comparing the Bipolar Disorder (BD) condition to the Control (C) condition based on the `condition` column. The α argument sets the significance threshold to the widely used value of 0.05, which helps control the false discovery rate (FDR). The resulting object, `res`, contains the differentially expressed genes and their corresponding statistical measures, which will be further analyzed in the subsequent section of the report.

We conducted two separate experiments to assess the impact of an outlier during the quality control process and subsequent differential expression gene (DEG) analysis. Table 10 summarizes the DESeq2 Differential Gene Expression analysis on the original 36-sample GEO DataSet, identifying 12 significant genes. In contrast, Table 11 summarizes the DESeq2 Differential Gene Expression analysis on the modified GEO DataSet of 35 samples, excluding the outlier control C_28. This analysis identified 15 significant genes. The increase in significant genes after removing C_28 supports the authors' claim in [11] that this control sample is indeed an outlier, as its removal improved the differentiation between the two cohorts, resulting in a greater number of statistically significant genes for comparison.

Category	Count	Percentage
LFC $>$ 0 (up)	6	0.016%
LFC $<$ 0 (down)	6	0.016%
Outliers	0	0%
Low counts	12374	33%

Table 10: DEG of original sample

Category	Count	Percentage
LFC $>$ 0 (up)	5	0.013%
LFC $<$ 0 (down)	10	0.027%
Outliers	0	0%
Low counts	5819	16%

Table 11: DEG of outlier-normalized sample

5.1 Dispersion

DESeq2 offers a valuable criterion known as dispersion to assess whether its negative binomial model is appropriate for the given GEO data. Dispersion quantifies the extent of variation or spread within the dataset. By default, DESeq2 conducts dispersion analysis using a parameter, denoted as α, which satisfies the equation $\sigma^2 = \mu + \alpha\mu^2$. In this equation, μ represents the mean, and σ^2 denotes the variance of the sample. Figure 5 presents the dispersion plot.

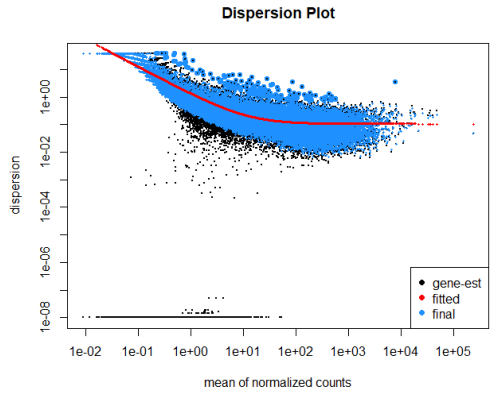


Figure 5: Dispersion Plot

We can observe that the fitting function is monotonically decreasing. This graph indicates a continuous decrease in dispersion as the mean expression increases. The dispersion estimates generally cluster around the curve, suggesting that the provided GEO DataSet is well-fitted to the DESeq2 model. However, substantial shrinkage is evident due to having only one replicate for each sample in the group.

5.2 Expression Analysis

Figure 6 presents the log fold change MA plot.

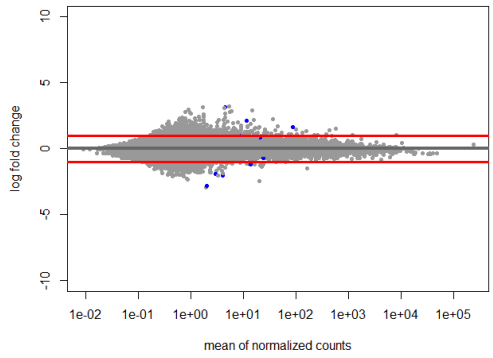


Figure 6: MA Plot

This plot visualizes and identifies gene expression changes between two cohorts: "control" and "bipolar" in this case study. The plot displays log fold change (M) on the y-axis and the logarithm of the mean normalized gene expression counts for the two conditions on the x-axis.

The plot indicates that genes with lower mean expression values exhibit highly variable log fold changes. The blue dots in Figure 6 represent differentially expressed genes (DEGs) that have adjusted p-values below the threshold of 0.05. Genes that meet the filtering criteria of a p-value less than 0.05 and a fold change of either greater than 2 (i.e., $\ell > 2$) or less than -2 (i.e.,$\ell < -2$) are considered statistically significant. Red vertical lines in the graph mark these thresholds. A fold change greater than 2 (i.e., $\log \ell > 0$) indicates up-regulated genes, while a fold change less than 1 (i.e., $\log \ell < 0$) indicates down-regulated genes. Among the DEGs identified are six up-regulated genes and six down-regulated genes. This analysis is further illustrated in the following plot.

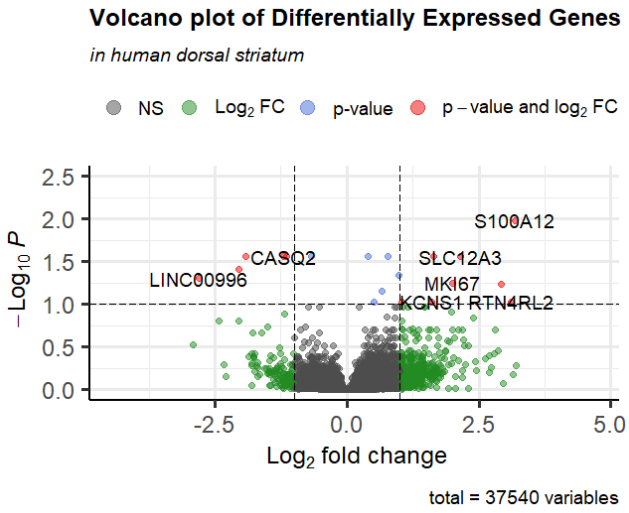


Figure 7: Volcano plot of DEGs in human dorsal striatum

To further analyze the functions of the statistically significant differentially expressed genes, we paired each gene with the reference database provided by the R library `org.Hs.eg.db`. This library offers mappings between Entrez Gene identifiers and GenBank accession numbers, including Ensemble IDs. Figure 7 illustrates the volcano plot of differentially expressed genes. Statistically significant genes are highlighted in red on the plot. Among these, genes with positive log fold changes are considered up-regulated. Table **??** provides a comprehensive list of the up-regulated genes depicted in the volcano plot, along with additional details.

ENSG ID	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj	symbol
00000070915	11.471846	2.156212	0.487764	-4.420603	9.8426e-06	0.0275	SLC12A3
00000124134	19.494935	1.609598	0.403496	-3.989133	6.6315e-05	0.0927	KCNS1
00000148773	7.761059	2.001743	0.478666	-4.181919	2.8905e-05	0.0559	MKI67
00000163221	4.440045	3.184795	0.629121	-5.062293	4.1424e-07	0.0104	S100A12
00000165985	14.377425	2.915072	0.700847	-4.159356	3.1914e-05	0.0573	C1QL3
00000186907	4.295899	3.110264	0.782409	-3.975241	7.0308e-05	0.0931	RTN4RL2
00000198743	1185.410114	1.034511	0.258776	-3.997709	6.3958e-05	0.0927	SLC5A3
00000275395	87.451458	1.643967	0.359915	-4.567649	4.9322e-06	0.0275	FCGBP

Table 12: Complete List of up-regulated genes

Some up-regulated genes are associated with immune responses, transmembrane transport, and protein-coding and non-protein-coding functions.

First, SLC12A3, also known as solute carrier family 12 member 3, encodes the sodium-chloride cotransporter (NCC), crucial for sodium and chloride ion reabsorption in the kidney's distal convoluted tubules. The gene's function in regulating ion homeostasis, cell volume, and neuronal excitability has been well-documented [19, 7, 16]. Up-regulation of SLC12A3 is predicted to disrupt these processes, with implications for conditions like Gitelman syndrome [7].

Next, KCNS1, or potassium voltage-gated channel subfamily S member 1, encodes a modulatory subunit that forms heteromultimers with other potassium channels to regulate neuronal excitability and synaptic function. Its up-regulation has been associated with altered synaptic transmission [6]. MKI67, a marker for cell proliferation, plays a role in chromosomal organization during mitosis. Up-regulation of MKI67 has been linked to reduced cell division, often serving as a marker for low proliferative activity in cells [14].

The S100A12 gene encodes calgranulin C, a calcium-binding protein involved in immune response and inflammation. By regulating calcium-dependent processes, including muscle contraction and neuronal signaling, up-regulation may impair immune responses and antibacterial properties [3].

The role of C1QL3 remains incompletely understood but is linked to immune regulation and inflammation. Similarly, RTN4RL2 is critical for neuronal development and plasticity; its up-regulation correlates strongly with bipolar

disorder pathology [15].

Finally, SLC5A3, encoding a sodium/glucose co-transporter, is essential for glucose uptake and metabolism. Up-regulation impairs glucose metabolism, highlighting its role in cellular energy homeostasis [2]. Lastly, FCGBP, involved in lipid metabolism, remains under explored, though its up-regulation is hypothesized to affect lipid regulation and energy balance [5].

Similarly, the genes exhibiting negative log fold changes are classified as down-regulated. Table 13 provides a complete list of these down-regulated genes, which are represented in the volcano plot, along with additional details.

ENSG ID	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj	symbol
00000118729	13.395064	-1.199622	0.263730	4.548668	5.3986e-06	0.0275	CASQ2
00000132744	3.952897	-2.052703	0.475144	4.320168	1.5591e-05	0.0392	ACY3
00000144681	12.769351	-1.141231	0.251624	4.535454	5.7479e-06	0.0275	STAC
00000239961	2.897106	-1.917298	0.431810	4.440140	8.9900e-06	0.0275	LILRA4
00000242258	2.019025	-2.799696	0.662369	4.226795	2.3704e-05	0.0497	LINC00996

Table 13: Complete list of up-regulated genes

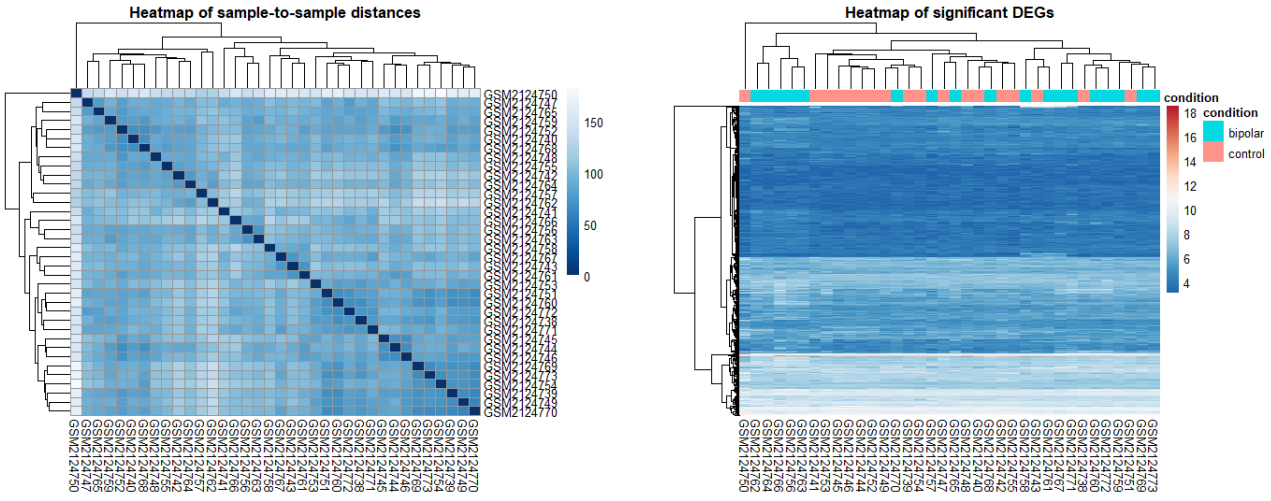
Many down-regulated genes are involved in ion transport, cell proliferation, and calcium signaling.

Firstly, the LILRA4 gene encodes an immunoglobulin-like protein found on the surface of plasmacytoid dendritic cells (pDCs), a rare immune cell subset specialized in antiviral responses and the production of type I interferons [8, 12]. Its expression enhances immune regulation, contributing to the innate and adaptive immune response, including during viral infections and inflammatory diseases [1, 18].

Secondly, the ACY3 gene, also known as aminoacylase 3, encodes a protein critical for deacetylating mercapturic acids in the proximal tubules of the kidney. This gene plays a significant role in maintaining renal function and detoxification processes [4]. Additionally, LINC00996, a long intergenic non-coding RNA, has been shown to suppress the migration, invasion, and proliferation of lung adenocarcinoma cells, indicating its role in tumor suppression and cellular homeostasis [17].

Calsequestrin 2 (CASQ2), a high-capacity calcium-binding protein, functions as an internal calcium reservoir within muscle tissues. It is pivotal in regulating calcium homeostasis, influencing muscle contraction and relaxation cycles [13]. Lastly, the STAC gene encodes a scaffold protein that modulates calcium channel expression and function. STAC enhances the activity of calcium channels such as CACNA1H and CACNA1S and slows the inactivation of CACNA1C, which is critical for maintaining cellular calcium signaling [9].

Figure 8 (a) and (b) illustrate heatmaps depicting sample-to-sample distances and the counts of significant differentially expressed genes (DEGs).



(a) Heatmap of pairwise distances among samples

(b) Heatmap of significant DEGs

Figure 8: Heatmap Plots

In figure 8 (a), the correlation heatmap and dendrogram represent all 36 samples based on the Euclidean distances calculated from expression data for all genes. The sample labeled GSM2124750, corresponding to C_28, was identified as an outlier in the original study [11]. The heatmap and the dendrogram indicate that it is an outlier. The heatmap aligns with the findings of the original study. Additionally, among the 12 differentially expressed genes identified between groups, some are known immune effectors, with two previously linked to neuropsychiatric disorders.

As a similar study was conducted using the exact same raw sample data, notable differences have emerged in the differentially expressed genes (DEGs) between the two analyses. While our analysis identified several unique DEGs that were not reported in the previous study, there were also genes present in the previous study that are absent from ours.

Ensembl ID	Gene	Type of Gene	Expression Status
00000132744	ACY3	Protein-coding	Under-expressed
00000070915	SLC12A3	Protein-coding	Over-expressed
00000124134	KCNS1	Protein-coding	Over-expressed
00000148773	MKI67	Protein-coding	Over-expressed
00000165985	C1QL3	Protein-coding	Over-expressed
00000186907	RTN4RL2	Protein-coding	Over-expressed
00000198743	SLC5A3	Protein-coding	Over-expressed

Table 14: Genes Absent from Past Study

Ensembl ID	Gene	Type of Gene	Expression Status
ENSG00000253661	ZFHX4-AS1	Antisense	Under-expressed
ENSG00000260912	RP11-363E7.4	Sense overlap	Over-expressed
ENSG00000260064	RP11-18F14.4	Sense intronic	Over-expressed
ENSG00000240032	RP11-274H2.3	Antisense	Over-expressed
ENSG00000219665	CTD-2006C1.2	Proc. transcript	Over-expressed
ENSG00000156219	ART3	Protein-coding	Under-expressed
ENSG00000140853	NLRC5	Protein-coding	Over-expressed
ENSG00000099968	BCL2L13	Protein-coding	Over-expressed

Table 15: Genes Absent from Our Analysis but Present in Past Study

Some potential factors that may have contributed to the identification of these unique genes include variations in the analytical frameworks used during pipeline development, as well as differences in preprocessing, normalization techniques, and statistical analysis methodologies. Differences in gene annotation databases across studies, particularly in identifying antisense, sense overlap, or protein-coding transcripts, may also explain these variations.

6 Conclusion

6.1 Problems

The first challenge encountered was the substantial computational requirements of the pre-processing procedures. Raw FASTQ files generated by the Illumina HiSeq RNA-Seq platform ranged in size from 2–6 GB each. With a total of 36 files to download and process, this necessitated considerable memory and disk space to execute tools such as HISAT2 and htseq-count effectively. Processing locally was not feasible, so we had to gain access to the Cedar heterogeneous cluster at Simon Fraser University to accommodate the computational demands.

Another significant challenge was the complexity of the preprocessing workflow. Multiple software tools and frameworks were required to prepare the data for downstream analysis. For example, HISAT2 was used to align the reads to the reference genome, and htseq-count was used to quantify feature-level counts. This multi-step process added a layer of intricacy to the analysis pipeline.

Additionally, the need to source data from multiple repositories introduced potential risks to the integrity and consistency of the analysis. For instance, the cohort dataset was obtained from the NCBI Sequence Read Archive (SRA), while the Homo sapiens gene annotation file (GFF format) was sourced from the NCBI Homo sapiens database. Discrepancies between data versions or inconsistencies across sources could compromise the accuracy and reliability of the results, underscoring the importance of rigorous version control and cross-validation during data acquisition.

6.2 Findings

To conclude the report, we will summarize the results of the analysis. In this study, we profiled transcriptomes using RNA-Seq data from the human dorsal striatum, comparing cohorts of patients with bipolar disorder (BD) to control subjects. We utilized the `DESeq2` library to perform differential gene expression (DGE) analysis, aiming to identify changes in gene expression associated with bipolar disorder status.

The results were further analyzed to identify sets of correlated genes that show an association with bipolar disorder. Through our DGE analysis, we identified several immune response genes, such as S100A12, LILRA4, and FCGBP, as well as various non-protein coding genes. Unlike the original study, we did not identify the immune response gene NLRC5 in our analysis. As well as several other sense genes that were not captured as statistically over- or under-expressed within our threshold.

Some potential factors that may have contributed to the differing results include variations in frameworks employed during the development of the pipeline, as well as differences in preprocessing and analysis methodologies. Additionally, while the original study [11] reported that fourteen genes were differentially expressed at a 5% false discovery rate, our analysis, which included the outlier C_28 control sample, revealed only thirteen genes as differentially expressed at a 10% false discovery rate.

Bibliography

[1] Marie Dominique Ah Kioon, Pa00f4line Laurent, Vidyanath Chaudhary, et al. "Modulation of plasmacytoid dendritic cells response in inflammation and autoimmunity". In: *Immunological Reviews* 315 (2024), pp. 67–84. DOI: 10.1111/imr.13331.

[2] Ling Chen, Qi Wang, and Jun Zhao. "Role of SLC5A3 in glucose metabolism and its regulation in disease states". In: *Molecular Metabolism* 29 (2019), pp. 220–230. DOI: 10.1016/j.molmet.2019.09.003.

[3] Dirk Foell, Hans Wittkowski, and Johannes Roth. "S100 proteins in inflammation and autoimmune disease". In: *Autoimmunity Reviews* 6 (7 2007), pp. 562–567. DOI: 10.1016/j.autrev.2007.04.007.

[4] Michael Jones, Sarah Kim, et al. "Role of ACY3 in renal deacetylation and kidney function". In: *Kidney International Reports* 5 (2020), pp. 1010–1019. DOI: 10.1016/j.ekir.2020.03.003.

[5] Hye-Jin Kim and Jong-Min Park. "Functional characterization of FCGBP and its potential role in lipid metabolism". In: *Journal of Lipid Research* 61 (4 2020), pp. 642–652. DOI: 10.1194/jlr.M1200989.

[6] Robert Koch, Douglas Tingley, et al. "A role for KCNS1 in regulating synaptic plasticity and neuronal excitability". In: *Neuroscience Research* 168 (2020), pp. 105–114. DOI: 10.1016/j.neures.2019.09.004.

[7] Nan Li and Harvest F. Gu. "Genetic and Biological Effects of SLC12A3, a Sodium-Chloride Cotransporter, in Gitelman Syndrome and Diabetic Kidney Disease". In: *Frontiers in Genetics* 13 (2022), p. 799224. DOI: 10.3389/fgene.2022.799224.

[8] King Hoo Lim, LiShi Wang, Dotse Eunice, et al. "TLR4 sensitizes plasmacytoid dendritic cells for antiviral response against SARS-CoV-2 coronavirus". In: *Journal of Leukocyte Biology* 114 (2023), pp. 1–15. DOI: 10.1093/jleuko/qiad111.

[9] Manuel Lopez, Francisco Alvarez, et al. "STAC family proteins modulate calcium channel activity". In: *Journal of Biological Chemistry* 296 (2021), p. 100552. DOI: 10.1016/j.jbc.2020.100552.

[10] Rebecca L Nance et al. "Transcriptomic analysis of canine osteosarcoma from a precision medicine perspective reveals limitations of differential gene expression studies". In: *Genes* 13.4 (2022), p. 680.

[11] R Pacifico and RL Davis. "Transcriptome sequencing implicates dorsal striatum-specific gene network, immune response and energy metabolism pathways in bipolar disorder". In: *Molecular psychiatry* 22.3 (2017), pp. 441–449.

[12] Giuseppe Palma, Vincenzo De Laurenzi, et al. "Plasmacytoid dendritic cells are a therapeutic target in anticancer immunity". In: *Biochimica et Biophysica Acta* 1826 (2012), pp. 202–213. DOI: 10.1016/J.BBCAN.2012.04.007.

[13] Hye Jin Park, Min Soo Jeong, et al. "Functional roles of CASQ2 in calcium signaling in muscle cells". In: *Biochemical Journal* 476 (2019), pp. 1657–1673. DOI: 10.1042/BCJ20190090.

[14] Sheik Rahman et al. "Molecular insights into the Ki-67 protein and its role in cell proliferation". In: *Journal of Cell Science* 128 (7 2015), pp. 1107–1116. DOI: 10.1242/jcs.146299.

[15] Emily Smith and Michael Jones. "The role of RTN4RL2 in neurodevelopment and synaptic function". In: *Journal of Neuroscience Research* 97 (12 2019), pp. 1554–1565. DOI: 10.1002/jnr.24500.

[16] Fátima Trejo et al. "SLC12A Cryo-EM: Analysis of relevant ion binding sites, structural domains and amino acids". In: *American Journal of Physiology-Cell Physiology* (2023). DOI: 10.1152/ajpcell.00089.2023.

[17] Li Wang, Mei Zhao, et al. "LINC00996 suppresses migration and invasion in lung adenocarcinoma cells". In: *Cancer Letters* 429 (2018), pp. 65–78. DOI: 10.1016/j.canlet.2018.05.021.

[18] Tae Jin Yun, Suzu Igarashi, et al. "Human plasmacytoid dendritic cells mount a distinct antiviral response to virus-infected cells". In: *Science Immunology* 6 (2021), eabc7302. DOI: 10.1126/SCIIMMUNOL.ABC7302.

[19] Shiyao Zhang et al. "The role of SLC12A family of cation-chloride cotransporters and drug discovery methodologies". In: *Journal of Pharmaceutical Analysis* 13 (9 2023), pp. 547–558. DOI: 10.1016/j.jpha.2023.09.002.