Bioinformatics for Beginners (B4B) 2026

BTEP Team
BTEP/GAU/CCR/NCI/NIH - email ncibtep@mail.nih.gov
Bioinformatics Training and Education Program

Table of Contents

Course Overview

- Bioinformatics for Beginners: RNA-Seq 12
  - Course Description 12
    - Module 1: Unix and Biowulf 12
    - Module 2: Bulk RNA-Seq Analysis 12
    - Module 3: Differential Expression and Pathway Analysis 13
  - Course requirements: 13
    - Who can take this course? 13
    - How will we work through lesson content? 13

Module 1 - Unix and Biowulf

Lesson 1 15

- Lesson 1: Introduction to Biowulf & Unix Navigation 15
  - Learning Objectives 15
  - Part 1 — What is Biowulf? 15
    - Key components of the NIH HPC system 15
    - Why can't I just use my laptop? 15
    - Software 16
  - Part 2 — Connecting to Biowulf 16
    - Accessing Biowulf via SSH 16
    - Accessing Biowulf via HPC On Demand 16
      - Requirements 16

- How to connect 16
- What you can do from the dashboard 17
- Part 3 — Unix Navigation Basics 17
  - Orientation commands 17
  - Navigating directories 18
  - Creating and removing directories 18
- Part 4 — Working with Files 18
  - Creating a file 18
  - Viewing files 19
  - Copying files 19
  - Moving and renaming files 19
  - Deleting files 19
- Lesson 1 — End-of-Lesson Quiz 20

Lesson 2 21

- Lesson 2: Useful Unix & Running Jobs on Biowulf 21
  - Learning Objectives 21
  - Part 1 — Powerful Unix Tools 21
    - Combining commands with pipes | 21
    - Redirecting output 21
    - Searching files with grep 21
    - sort and uniq 22
  - Part 2 — The Module System 22
    - Why modules? 22
  - Part 3 — Running Jobs on Biowulf 23
    - Three ways to run work on Biowulf 23
    - Interactive sessions with sinteractive 23
    - Writing a batch script with sbatch 23
    - Monitoring your jobs 24

- Part 4 — Quick Practice 24
- Lesson 2 — End-of-Lesson Quiz 24

Lesson 3 26

- Lesson 3: Swarm Jobs, File Transfers & Downloading from the SRA 26
  - Learning Objectives 26
  - Part 1 — Swarm Jobs: Running Many Jobs in Parallel 26
    - Create a swarm file 26
    - Submit the swarm 26
    - Monitor swarm jobs 27
    - For loops vs. swarm — when to use which 27
  - Part 2 — Transferring Files 27
    - Copy from Biowulf to your local machine (run from your laptop terminal) 27
    - Copy a local file to Biowulf (run from your laptop terminal) 27
    - Copy an entire directory (use -r for recursive) 28
  - Part 3 — Introduction to the SRA 28
    - Key terminology 28
  - Part 4 — Downloading SRA Data with fasterq-dump 28
    - Copy the SRA accession list 29
    - Option A: Download interactively (single file, for testing) 29
    - Option B: Batch script (single file, unattended) 29
    - Option C: Swarm (multiple files in parallel — the real power) 29
    - Using prefetch + fasterq-dump (fastest method) 30
  - Part 5 — Checking Your Downloads 30
  - Lesson 3 — End-of-Lesson Quiz 30

Running RENEE 32

- Breaking Down the RENEE Command Line 32
  - 1. The program and subcommand 32

- 2. --input 32
- 3. --output /data/$USER/hcc1395_renee_b4b 33
- 4. --genome hg38_36 33
- 5. --mode slurm 33
- 6. --star-2-pass-basic 34
- 7. --sif-cache /data/CCBR_Pipeliner/SIFs 34
- 8. --partition student 34
- 9. --time 8:00:00 34
- Key Unix takeaways for students 34

Module 2 - Bulk RNA Sequencing

Detailed instructions for renee 37

- Running the RENEE Bulk RNA Sequencing Pipeline 37
  - Learning Objectives 37
  - Signing onto Biowulf's HPC OnDemand 37
  - Get an Interactive Terminal on HPC OnDemand 38
  - Change into Participant's Biowulf Data Directory 40
  - RENEE Pipeline General Usage and Documentation 41

Lesson 5 45

- Lesson 5: Quality Control and Cleaning of Next Generation Sequences 45
  - Lesson 5 Review 45
  - Learning Objectives 45
  - Sign onto Biowulf 45
  - Navigate the RENEE Output Directory 45
  - FASTQC 47
    - Components to a FastQC Report 48

- MultiQC 52
  - General Statistics Table 52
- Cutadapt Trimming 55
- Checking for Genomic Content from Other Species 57

Lesson 6 60

- Lesson 6: Alignment of Bulk RNA Sequencing Data 60
  - Lesson 5 Review 60
  - Learning objectives 60
  - Why Perform Alignment 60
  - SAM and BAM Files 61
  - Post Alignment Quality Metrics 63
    - Read Distribution 65
    - Coverage Distribution 65
    - Insert Size 66
  - Strandedness of Experiment 67
  - Visualizing Genomic Alingments 69

Lesson 7 71

- Lesson 7: Quantifying Gene Expression 71
  - Lesson 6 Review 71
  - Learning objectives 71
  - Goals for Gene Expression Quantification in Bulk RNA Sequencing 71
  - Annotation Files 73
  - Quantification Output 78
  - Quantification Metrics 80
  - Gene Coverage 81
  - Transcript Integrity 82
  - RNA Report 82

Module 3 - Differential Expression and Pathway Analysis

Lesson 8 87

- Lesson 1: What Is Differential Expression Analysis? 87
  - Learning Objectives 87
  - 1. What Can We Do with a Count Matrix? 87
  - 2. The Core Question 88
    - How We Measure Differential Expression 89
    - What Is Log2 Fold Change? 89
    - Why Use log2FC Instead of Raw Fold Change? 89
      - Quick Interpretation Guide 89
  - 3. Why Absolute Abundances Are Tricky 90
    - A Small Composition Example 91
  - 4. QC Before DEG Analysis 91
    - Pre-DEG QC Checklist 92
  - 5. Normalization Strategies 93
    - Low-Count Filtering. 93
    - CPM 94
    - RPKM, FPKM, and TPM 94
    - TMM 94
    - Median-of-Ratios / RLE 94
    - Transformations for Plots 95
      - Want more information? 95
  - 6. Meet the Tools 95
    - DESeq2 96
    - edgeR 96
    - limma-voom 96

- Summary Table 97
- 7. Going Further: R and Python 99
- 8. Introducing iDEP 100
  - What You'll Need 100
    - Getting the Data 101
- 9. Accessing iDEP via HPC Open OnDemand on Biowulf 101
- 10. Quick Knowledge Check 104
- Take-Home Messages 105
- Recommended Reviews and Tutorials by Section 106
- Resources Used 108

Lesson 9 111

- Lesson 9: Differential Expression Analysis with iDEP 111
  - Learning Objectives 111
  - Introduction to iDEP 111
  - Get Started with HPC OnDemand 112
  - Get the Files 113
    - Sample Names and Design File 113
  - iDEP Orientation 114
  - Upload the Data 114
  - Preprocess the Data 115
    - Filtering 115
    - Transformation 116
  - QC and Exploratory Analysis 116
    - Barplot of library sizes 117
    - Boxplot and Density plots of transformed expression distributions 118
    - Scatterplots of pairwise sample correlations 120
    - Mean vs standard deviation plot of transformed expression distributions 121
    - QC plot with the number of genes by type 122

- Gene plot comparing gene expression by condition or sample: Are the expected genes expressed? 123
- Clustering and Heatmap 124
  - Hierarchical clustering of samples and genes. 124
  - K-means clustering of genes 126
- PCA 127
- Batch Correction in RNA-Seq 128
- Run Differential Expression in DGE1 128
  - Experimental Design 129
    - Recommended Settings for a simple two-group comparison in iDEP 129
  - Other Options 130
- DEG Results Walkthrough 130
  - DEG Table 131
  - Interpreting the DEG Tab Plots 131
    - Heatmap 131
    - Volcano plot 134
    - MA plot 134
    - Scatter plot 135
- Genome Tab 136
- Reproducibility Checklist 137
- Resources Used 137

Lesson 10 138

- Introduction to Gene Ontology and Pathway Analysis 138
  - Objectives 138
  - Where have we been and where are we going? 138
  - What do we mean by gene ontology, functional enrichment, and pathway analysis? 139
  - Gene sets, GO terms, and pathways — what's the difference? 140
    - Quick Definitions. 140

- Why do gene set / pathway analysis? 141
- What is gene ontology? 141
  - What is GO-CAM? 143
  - GO by the numbers 143
  - What is GO Slim? Reducing semantic similarity 143
- Approaches to functional enrichment / pathway analysis 143
  - Over-representation analysis (ORA) 144
    - Background gene sets 145
  - Functional Class Scoring (FCS) 145
    - MSigDB and its relationship to GSEA 146
  - Pathway Topology (PT) 147
  - Single-Sample Gene Set Scoring 147
- What tools are available? 149
  - Other and related tools 149
  - BTEP Tutorials on Specific Tools 150
- Databases used by enrichment and pathway tools 150
  - Gene Ontology (GO) 150
  - Kyoto Encyclopedia of Genes and Genomes (KEGG) 150
  - Reactome 150
  - Pathway Commons 151
  - PANTHER 151
  - STRING 151
  - Enrichr 151
  - WikiPathways 152
  - NDEx 152
  - HumanCyc (See BioCyc) 152
  - Disease Ontology (DO) 152
  - DisGeNET 152

- Importance of Gene IDs 153
- Other considerations 154
- Functional enrichment and pathway analysis in iDEP 154
- References / Resources 155

Lesson 11 157

- Lesson 11: Pathway Analysis and Pre-ranked GSEA with iDEP 157
  - Learning Objectives 157
  - What Functional Enrichment / Pathway Analysis Is Available in iDEP? 157
  - Two Places to Start 159
    - Upload Fold Changes and Adjusted P-values 159
    - Upload Expression Matrix and Sample Design 160
  - DEG Functional Enrichment 160
    - Choosing Gene Sets and Databases 161
    - Results Interpretation 161
      - Pathways table 161
      - Tree and Network 162
      - Plot 162
      - Genes/Details 162
  - Pathway Analysis 163
    - Gene Set Enrichment Analysis (GSEA) / fgsea 163
      - Settings to control with iDEP: 164
      - Results 164
        - Significant Pathways Table 165
        - Example Interpretation of Results 165
  - Common Troubleshooting 166
  - Beyond iDEP 166
  - Resources Used 167

Resources

- Additional Resources 169
  - Module 1: Unix 169

Bioinformatics for Beginners: RNA-Seq

Course Description

This training series introduces bioinformatics concepts through a practical bulk RNA-Seq analysis workflow. Across three modules, participants build core skills in Unix and Biowulf navigation, file transfer and job submission, bulk RNA-Seq quality control and alignment, and differential expression and pathway analysis using iDEP. The course is designed for beginners and emphasizes hands-on analysis skills that can be applied to a wide range of bioinformatics projects.

Module 1: Unix and Biowulf

Learn the basic skills to work on the NIH HPC Biowulf, including login, basic Unix commands, working with files, and running various types of jobs.

- Bioinformatics for Beginners: Introduction to Biowulf and Unix Navigation — Wed, Jul 08, 2026, 2:00 pm - 3:00 pm
- Bioinformatics for Beginners: Useful Unix and Running Jobs on Biowulf — Thu, Jul 09, 2026, 2:00 pm - 3:00 pm
- Bioinformatics for Beginners: Swarm Jobs, File Transfers & Downloading from the SRA — Tue, Jul 14, 2026, 2:00 pm - 3:00 pm

Module 2: Bulk RNA-Seq Analysis

Learn bulk RNA-Seq experimental methods and data analysis, including quality metrics, genome alignment, and gene expression estimation.

- Introduction to Bulk RNA Sequencing *(https://bioinformatics.ccr.cancer.gov/btep/classes/introduction-to-bulk-rna-sequencing-1)* — Thu, Jul 16, 2026, 1:00 pm - 2:00 pm
- Bulk RNA Sequencing Quality Check and Trimming *(https://bioinformatics.ccr.cancer.gov/btep/classes/bulk-rna-sequencing-quality-check-and-trimming)* — Tue, Jul 21, 2026, 1:00 pm - 2:00 pm
- Alignment of Bulk RNA Sequencing Data *(https://bioinformatics.ccr.cancer.gov/btep/classes/alignment-of-bulk-rna-sequencing-data)* — Thu, Jul 23, 2026, 1:00 pm - 2:00 pm
- Estimating Gene Expression from Bulk RNA Sequencing Data *(https://bioinformatics.ccr.cancer.gov/btep/classes/estimating-gene-expression-from-bulk-rna-sequencing-data)* — Tue, Jul 28, 2026, 1:00 pm - 2:00 pm

Module 3: Differential Expression and Pathway Analysis

Learn the foundational concepts underlying differential expression and functional enrichment analysis while gaining hands-on experience using iDEP to perform these analyses.

- Introduction to Differential Gene Expression Analysis (RNA-Seq) — Tue, Aug 11, 2026, 2:00 pm - 3:00 pm
- Differential Expression Analysis with iDEP — Thu, Aug 13, 2026, 2:00 pm - 3:00 pm
- Introduction to Gene Ontology and Pathway Analysis — Tue, Aug 18, 2026, 2:00 pm - 3:00 pm
- Pathway Analysis and Pre-ranked GSEA with iDEP — Thu, Aug 20, 2026, 2:00 pm - 3:00 pm

Course requirements:

Who can take this course?

There are no prerequisites to take this course. This course is open to NIH researchers interested in learning bioinformatics skills, especially those relevant to analyzing bulk RNA sequencing data.

How will we work through lesson content?

For the hands-on sessions, participants will use NIH HPC Student Accounts. This requires access to the NIH VPN. ***Student accounts are only available to course registrants.***

Module 1 - Unix and Biowulf

Lesson 1: Introduction to Biowulf & Unix Navigation

Learning Objectives

- Understand what Biowulf is and why we use it.
- Connect to Biowulf.
- Navigate the file system using Unix commands.
- Create, move, copy, and delete files and directories.

Part 1 — What is Biowulf?

Biowulf is the NIH's high-performance computing (HPC) cluster, a massive Linux system with over 90,000 processors that tackles large computational tasks by dividing jobs across many computers (nodes) running simultaneously. Think of it as a supercomputer made of thousands of regular computers working together.

Key components of the NIH HPC system

Component	What it is	How you use it
Login node	Your entry point into Biowulf	SSH here first; do NOT run heavy jobs here
Compute nodes	Worker computers that run your jobs	Submit jobs via sbatch, sinteractive, or swarm
Helix	Dedicated file transfer node	For uploading/downloading data
/home/$USER	Personal home directory (16 GB)	Config files, scripts
/data/$USER	Primary working storage (100 GB)	Datasets, results

Why can't I just use my laptop?

Bioinformatics datasets (e.g., whole genome sequencing) can be hundreds of gigabytes. A single analysis may require 32+ CPU cores and 256 GB of RAM. Biowulf provides this on demand.

Software

Biowulf has 600+ pre-installed bioinformatics programs accessible via the module system (more on this in Lesson 2).

Part 2 — Connecting to Biowulf

Accessing Biowulf via SSH

Connect from your terminal (Mac/Linux) or an SSH client (Windows):

```
ssh username@biowulf.nih.gov
```

Note

You must be on the NIH campus network or NIH VPN to connect

Once logged in, you land on the **login node**. Your command prompt will show your username and the hostname.

Check-your-learning

After connecting, type `hostname`. What do you see? Why does this matter?

Accessing Biowulf via HPC On Demand

Not comfortable with the terminal yet? No problem. NIH HPC offers Open OnDemand, a browser-based interface that gives you access to Biowulf resources without any SSH client or command-line setup.

- URL: https://hpcondemand.nih.gov *(https://hpcondemand.nih.gov)*
- Preferred browser: Google Chrome

Requirements

- Connected to the NIH network (on campus or via VPN)
- NIH credentials + smart card (PIV) or multi-factor authentication (MFA)

How to connect

1. Open Chrome and navigate to https://hpcondemand.nih.gov *(https://hpcondemand.nih.gov)*
2. You will be redirected to the NIH central login page

3. Sign in with your smart card or MFA authenticator app
4. You will land on the HPC OnDemand Dashboard

What you can do from the dashboard

Feature	What it gives you
Files	Browse, upload, and download files in your /home and /data directories
Shell Access	A terminal in your browser — same as SSH, no client needed
Interactive Apps	Launch RStudio, JupyterLab, VS Code, and other GUI tools on compute nodes
Jobs	View and manage your Slurm job queue

Important

For this course, we will primarily use the Shell Access option in OnDemand, which gives you a terminal window in your browser. All of the commands you learn in this course work the same way whether you connect via SSH or OnDemand.

Check-your-learning

Log into HPC OnDemand and open a shell. Type `hostname` and `pwd`. How do the results compare to what you'd see with an SSH connection?

Part 3 — Unix Navigation Basics

Biowulf uses a Linux operating system. To use it, you need the command line interface (CLI) — you type commands rather than clicking icons.

The golden rule: You will make mistakes. That is okay and expected. It is how you learn. Biowulf has safeguards to prevent individual users from breaking the system.

Orientation commands

```
whoami          # print your username
pwd             # print working directory (where am I right now?)
ls              # list contents of current directory
ls -l           # list with details (permissions, size, date)
ls -lh          # same but with human-readable file sizes
```

Check-your-learning

Type `pwd`. What path do you see? What does the `$USER` variable represent?

Navigating directories

```
cd /data/$USER      # change to your data directory
pwd                 # confirm where you are
cd ~                # shortcut: go to home directory
cd ..               # go up one directory level
cd -                # go back to previous directory
```

Tip

Use the Tab key to auto-complete file and directory names. Use the ↑ arrow to recall previous commands.

Creating and removing directories

```
cd /data/$USER
mkdir Module_1                  # make a new directory
cd Module_1
mkdir directory_to_delete
ls
```

```
cd directory_to_delete
pwd
ls
```

To remove a directory, it must be empty:

```
cd ..
rmdir directory_to_delete       # only works if empty
```

Check-your-learning

Type `ls` to confirm the directory is gone. What happens if you try to `rmdir` a directory that contains files?

Part 4 — Working with Files

Creating a file

```
cd /data/$USER/Module_1
nano myseq.txt                  # open the nano text editor
```

Type a few lines of text, then press `Ctrl+X`, then `Y`, then `Enter` to save and exit.

Viewing files

```
cat myseq.txt                   # print entire file to screen
less myseq.txt                  # scroll through file (q to quit)
head myseq.txt                  # show first 10 lines
tail myseq.txt                  # show last 10 lines
head -20 myseq.txt              # show first 20 lines
wc -l myseq.txt                 # count lines in file
```

Copying files

```
# Copy a teaching file to your Module_1 directory
cp /data/classes/BTEP/B4B_2025/Module_1/sample.fast* .
ls
```

The `.` means "here" (current directory). The `*` is a wildcard that matches any characters.

Moving and renaming files

```
mv myseq.txt mysequence.txt     # rename a file
mv mysequence.txt /data/$USER/  # move to a different directory
```

Deleting files

```
rm mysequence.txt               # permanently deletes — no trash can
```

Warning

`rm` is permanent on Linux. There is no undo. Be careful.

Check-your-learning

What is the difference between `mv` and `cp`? What does the `-r` flag do with `rm`?

Lesson 1 — End-of-Lesson Quiz

1. What does HPC stand for, and why is it useful for bioinformatics?
2. What is the difference between the Biowulf login node and a compute node?
3. What command do you use to connect to Biowulf from your terminal?
4. What is Helix, and when should you use it instead of Biowulf?
5. What command prints your current directory location?
6. What are the two main storage spaces on Biowulf, and how much space does each provide by default?
7. You typed `cd /data/$USER/results` but got an error saying "No such file or directory." What likely went wrong, and how would you fix it?
8. What is the difference between `ls` and `ls -l`?
9. Why must you be careful before using `rm`?
10. What keyboard shortcut auto-completes file and directory names at the command line?

Lesson 2: Useful Unix & Running Jobs on Biowulf

Learning Objectives

- Use pipes, grep, and text utilities to process files.
- Load software modules.
- Run interactive sessions with `sinteractive`.
- Write and submit a batch job with `sbatch`.

Part 1 — Powerful Unix Tools

Combining commands with pipes |

The pipe (|) sends the output of one command as the input to the next. This lets you chain tools together into a mini-pipeline.

```
cat sample.fastq | head -8      # view just the first 8 lines of a FA
ls -l | wc -l                   # count the number of items in a dire
```

Redirecting output

```
ls -l > filelist.txt            # write output to a file (overwrite)
ls -l >> filelist.txt           # append output to an existing file
```

Check-your-learning

What is the difference between > and >>? What happens if you use > on a file that already exists?

Searching files with `grep`

`grep` searches a file for lines matching a pattern:

```
grep "ATCG" sample.fastq            # find lines containing "ATCG"
grep -c "ATCG" sample.fastq         # count matching lines
```

```
grep -i "atcg" sample.fastq          # case-insensitive search
grep -v "^@" sample.fastq            # show lines that do NOT start w
```

In FASTQ files, sequence identifier lines start with `@`. Using `grep` to find or exclude these is a common bioinformatics task.

Check-your-learning

In a FASTQ file, every 4th line starting from line 1 is a sequence header (`@`). Write a `grep` command to count only header lines.

`sort` and `uniq`

```
sort filelist.txt                # sort lines alphabetically
sort -k5 -n filelist.txt         # sort by column 5 numerically
sort filelist.txt | uniq         # remove duplicate lines
sort filelist.txt | uniq -c      # count occurrences of each unique l
```

Part 2 — The Module System

Biowulf has 600+ pre-installed programs. You load them with `module`:

```
module avail                     # list all available software (long!)
module avail sratoolkit          # search for a specific tool
module load sratoolkit           # load the tool into your environmen
module list                      # see what you have loaded
module unload sratoolkit         # unload a module
```

Why modules?

Different analyses require different software versions. The module system lets you load exactly what you need without conflicts.

Check-your-learning

Type `module avail samtools`. What versions are available? How would you load a specific version?

Part 3 — Running Jobs on Biowulf

On a shared HPC system, you cannot simply run heavy programs on the login node. You must submit jobs through the Slurm scheduler (Simple Linux Utility for Resource Management). Slurm manages the queue and allocates resources fairly.

Three ways to run work on Biowulf

Method	Command	Best for
Interactive session	`sinteractive`	Testing, debugging, short tasks
Batch job	`sbatch script.sh`	Long-running analyses
Job array (swarm)	`swarm -f commands.txt`	Many similar jobs in parallel

Interactive sessions with `sinteractive`

```
sinteractive                          # default: 1 core, 1.5 GB RAM, 8
sinteractive --cpus-per-task=4 --mem=8g    # request more resources
```

Once inside, your prompt changes — you are now on a compute node, not the login node. You can run computationally intensive commands here.

```
hostname                              # confirm you're on a compute noc
# do your work...
exit                                  # return to login node when done
```

Check-your-learning

Why is it important NOT to run CPU-intensive jobs on the login node?

Writing a batch script with `sbatch`

Batch jobs run in the background without you needing to stay connected. You write a script that tells Biowulf what resources you need and what to run.

```
nano myfirstjob.sh
```

Paste this into the file:

```
#!/bin/bash
#SBATCH --cpus-per-task=2
```

```
#SBATCH --mem=4g
#SBATCH --time=01:00:00
#SBATCH --partition=student
#SBATCH --job-name=my_first_job
#SBATCH --output=myfirstjob_%j.log
# Your commands go below this line
echo "Hello from Biowulf compute node!"
hostname
date
```

Save and exit (`Ctrl+X`, `Y`, `Enter`), then submit:

```
sbatch myfirstjob.sh
```

Monitoring your jobs

```
squeue -u $USER                    # list your running/pending jobs
scancel JOBID                      # cancel a job by its ID
cat myfirstjob_JOBID.log           # view job output after it finisl
```

Check-your-learning

In the script above, what does `#SBATCH --time=01:00:00` do? What happens to your job if it exceeds this limit?

Part 4 — Quick Practice

1. Start an `sinteractive` session on the student partition.
2. Load the `sratoolkit` module.
3. Run `fastq-dump --version` to confirm it loaded.
4. Exit the interactive session.

Lesson 2 — End-of-Lesson Quiz

1. What does the pipe `|` do? Give an example using two commands.
2. What is the difference between > and >> when redirecting output?
3. Write a grep command to find all lines in `sample.fastq` that start with@.
4. What command lists all software modules currently loaded in your environment?
5. What is Slurm, and why does Biowulf use it?
6. What is the difference between `sinteractive` and `sbatch`?
7. In a batch script, what does the `#SBATCH` line do?

8. After submitting a job with `sbatch`, how do you check whether it is still running?
9. What does `--partition=student` specify in a Slurm job script?
10. You submitted a job but realize you made a mistake in the script. What command cancels a running job?

Lesson 3: Swarm Jobs, File Transfers & Downloading from the SRA

Learning Objectives

- Submit parallel jobs using `swarm`.
- Transfer files between Biowulf and your local computer.
- Understand the NCBI Sequence Read Archive (SRA).
- Download sequencing data using `fasterq-dump`.
- Scale downloads across multiple accessions.

Part 1 — Swarm Jobs: Running Many Jobs in Parallel

`swarm` is Biowulf's tool for running a batch of similar commands as independent parallel jobs — sometimes called an "embarrassingly parallel" workload.

Scenario: You have 10 samples to process with the same command. Instead of running them one by one (slow) or writing a complex loop, you write each command on its own line in a text file and let swarm handle the rest.

Create a swarm file

```
nano mycommands.swarm
```

Each line is one command (one subjob):

```
echo "Processing sample SRR001"
echo "Processing sample SRR002"
echo "Processing sample SRR003"
```

When finished, press `Ctrl+X`, then `Y`, then `Enter` to save and exit nano.

Submit the swarm

```
swarm -f mycommands.swarm --partition=student
```

Default allocation: Each subjob gets 1.5 GB RAM and 1 core. You can override:

```
swarm -f mycommands.swarm -g 4 -t 2 --partition=student
# -g = GB of memory per subjob, -t = threads (CPUs) per subjob
```

Monitor swarm jobs

```
squeue -u $USER                    # each subjob appears separately
```

Check-your-learning

What is the key advantage of using `swarm` over a `for` loop in a single batch script?

For loops vs. swarm — when to use which

Approach	How it works	Best when...
`for` loop in sbatch	Commands run one at a time, sequentially	Tasks are fast; few iterations
`swarm`	Each command runs as its own parallel job	Tasks are slow; many samples

Part 2 — Transferring Files

At some point you will need to move data between your local computer and Biowulf. Always use Helix (`helix.nih.gov`) for interactive file transfers — not the Biowulf login node.

```
ssh username@helix.nih.gov          # connect to Helix for transfers
```

Copy from Biowulf to your local machine (run from your laptop terminal)

```
scp username@helix.nih.gov:/data/username/Module_1/results.txt .
```

Copy a local file to Biowulf (run from your laptop terminal)

```
scp ./mydata.txt username@helix.nih.gov:/data/username/Module_1/
```

Copy an entire directory (use `-r` for recursive)

```
scp -r username@helix.nih.gov:/data/username/Module_1 .
```

Check-your-learning

Why should you connect to Helix rather than the Biowulf login node for file transfers?

Alternative: You can also mount HPC directories to your local machine using SSHFS or the HPC SMB share — see the NIH HPC documentation at https://hpc.nih.gov/docs/transfer.html *(https://hpc.nih.gov/docs/transfer.html)*.

Part 3 — Introduction to the SRA

The **Sequence Read Archive (SRA)** is NCBI's public repository for high-throughput sequencing data (RNA-seq, ChIP-seq, WGS, etc.). Most published sequencing datasets are deposited here.

Key terminology

Term	Meaning
BioProject	A study-level record, e.g., PRJNA578488
BioSample	A single biological sample within a project
SRR accession	A specific sequencing run, e.g., SRR10314042
FASTQ	The file format for raw sequencing reads

The **ENA (European Nucleotide Archive)** mirrors all SRA data — you can use `wget` or `curl` to pull gzipped FASTQ files directly from ENA without the SRA Toolkit, which can be faster for some use cases.

Check-your-learning

What is the difference between a BioProject and an SRR accession number?

Part 4 — Downloading SRA Data with fasterq-dump

`fasterq-dump` (part of the SRA Toolkit) downloads sequencing data from the SRA and converts it to FASTQ format. It uses multi-threading (default: 6 threads) and is significantly faster than the older `fastq-dump`.

Copy the SRA accession list

```
cd /data/$USER/Module_1
cp /data/classes/BTEP/B4B_2025/Module_1/sra_files_PRJNA578488.txt .
less sra_files_PRJNA578488.txt # view the list of SRR accessions
```

Option A: Download interactively (single file, for testing)

```
sinteractive --gres=lscratch:20 --cpus-per-task=6
module load sratoolkit
fasterq-dump -t /lscratch/$SLURM_JOBID SRR10314042 -O /data/$USER/Mo
exit
```

`-t` specifies a temp directory (local scratch space on the compute node — fast and automatically cleaned up). `-O` specifies the output directory.

Option B: Batch script (single file, unattended)

```
nano filedownload.sh
```

```
#!/bin/bash
#SBATCH --cpus-per-task=6
#SBATCH --gres=lscratch:10
#SBATCH --partition=student
#SBATCH --job-name=sra_download
#SBATCH --output=sra_download_%j.log
module load sratoolkit
fasterq-dump -t /lscratch/$SLURM_JOB_ID SRR10314042 -O /data/$USER/M
```

Press `Ctrl+X`, then `Y`, then `Enter` to save and exit nano.

```
sbatch filedownload.sh
squeue -u $USER
```

Option C: Swarm (multiple files in parallel — the real power)

First, build your swarm file:

```
nano download_all.swarm
fasterq-dump -t /lscratch/$SLURM_JOB_ID SRR10314042 -O /data/$USER/M
fasterq-dump -t /lscratch/$SLURM_JOB_ID SRR10314043 -O /data/$USER/M
fasterq-dump -t /lscratch/$SLURM_JOB_ID SRR10314044 -O /data/$USER/M
```

Press `Ctrl+X`, then `Y`, then `Enter` to save and exit nano.

```
swarm -f download_all.swarm -t 6 --gres=lscratch:10 --module sratoolk
```

Check-your-learning

Why do we use /lscratch/$SLURM_JOB_ID as the temp directory for fasterq-dump instead of /data/$USER?

Using `prefetch` + `fasterq-dump` (fastest method)

For large files, downloading with `prefetch` first is faster overall:

```
sinteractive --gres=lscratch:50 --cpus-per-task=6
module load sratoolkit
prefetch SRR10314042 -O /lscratch/$SLURM_JOBID/
fasterq-dump /lscratch/$SLURM_JOBID/SRR10314042/ -O /data/$USER/Modu
```

Part 5 — Checking Your Downloads

Use `seqkit` to get quick statistics about your FASTQ files:

```
module load seqkit
seqkit stats /data/$USER/Module_1/*.fastq
```

This reports: file name, format, type, number of sequences, sum of lengths, min/max/average read lengths.

Lesson 3 — End-of-Lesson Quiz

1. What is `swarm` used for, and how does it differ from a standard `sbatch` batch job?
2. How do you specify the amount of memory and number of CPUs for each subjob in a swarm?
3. Why should you use Helix (helix.nih.gov) for file transfers rather than the Biowulf login node?

4. Write an `scp` command to copy a file called `results.txt` from `/data/username/Module_1/` on Helix to your local current directory.
5. What is the SRA, and why is it important for bioinformatics research?
6. What is the difference between a BioProject accession (e.g., PRJNA578488) and an SRR run accession?
7. What does `fasterq-dump` do, and how is it different from `fastq-dump`?
8. Why do we specify `--gres=lscratch:20` when running `fasterq-dump`? What is `lscratch`?
9. What is the advantage of using `prefetch` before `fasterq-dump`?
10. You want to download 50 SRA accessions in parallel using swarm. Describe the steps you would take, from creating the swarm file to submitting and monitoring the jobs.

Breaking Down the RENEE Command Line

```
renee run \
  --input /data/classes/BTEP/hcc1395_renee_b4b/reads/*.R?.fastq.gz \
  --output /data/$USER/hcc1395_renee_b4b \
  --genome hg38_36 \
  --mode slurm \
  --star-2-pass-basic \
  --sif-cache /data/CCBR_Pipeliner/SIFs \
  --partition student \
  --time 8:00:00
```

This command runs **RENEE** (RNA sEquencing aNalysis pipElinE), an RNA-seq analysis pipeline. Even if you never touch this specific tool, the structure of this command is something you'll see over and over in Unix: `program subcommand --flag value --flag value ...`

---

1. The program and subcommand

```
renee run
```

- `renee` — the name of the program being executed. It's just a word that the shell looks up on your PATH to find the actual executable file.
- `run` — a **subcommand**. Many complex command-line tools (`git`, `docker`, conda`, and here`renee`) bundle multiple related actions into one program, and you pick which action with a subcommand. renee run starts a pipeline run, as opposed to other subcommands like renee build or renee unlock.

Everything after this is a series of **flags** (also called options or switches), each starting with `--` and usually followed by a value.

---

2. `--input`

```
/data/classes/BTEP/hcc1395_renee_b4b/reads/*.R?.fastq.gz
```

This tells RENEE which raw data files to process.

- `/data/...` — an **absolute path**, meaning it starts from the root directory (/) and describes the exact location of a file or folder, regardless of your current directory.
- `*` — the **wildcard** (or "glob") character meaning "match anything (zero or more characters)." The shell expands this *before* RENEE ever sees it — so if the folder contains `sample1.R1.fastq.gz`, `sample1.R2.fastq.gz`, `sample2.R1.fastq.gz`, etc., the shell turns `*.R?.fastq.gz` into a full list of all those filenames.
- `?` — a wildcard that matches exactly one character. So `R?` matches `R1` or `R2` (common notation for "read 1" and "read 2" in paired-end sequencing), but wouldn't match `R10`.
- `.fastq.gz` — the file extension, indicating these are FASTQ sequence files (the standard format for raw sequencing reads) that have been compressed with gzip.

> **Take Away**
>
> Wildcards let you specify many files with one short pattern instead of typing every filename.

---

3. `--output /data/$USER/hcc1395_renee_b4b`

Where RENEE should write its results.

- `$USER` — an **environment variable**. The `$` tells the shell "substitute the value of this variable here." `$USER` is set automatically by the system to your username. So if your username is `jsmith`, this path becomes `/data/jsmith/hcc1395_renee_b4b`.

Using `$USER` instead of hardcoding a username makes commands portable — the same command works for any user without editing it.

---

4. `--genome hg38_36`

Specifies which *reference genome* to align the sequencing reads against. `hg38` refers to a specific version (build) of the human reference genome; `_36` indicates a particular annotation release bundled with it. This isn't a general Unix concept — it's an option specific to RENEE.

---

5. `--mode slurm`

Tells RENEE *how* to execute its jobs. `slurm` refers to **Slurm**, a job scheduler used on many high-performance computing (HPC) clusters to allocate compute resources and queue jobs. In

`slurm` mode, RENEE submits its work as jobs to the cluster's scheduler rather than running everything directly on the machine you're typing on (which would be `--mode local`).

---

6. `--star-2-pass-basic`

A **boolean flag** (also called a "switch"). Notice it has no value after it — its mere presence turns a feature on. This one enables "2-pass" mode in STAR (the read-aligner RENEE uses internally), which improves detection of splice junctions by aligning reads twice: once to find junctions, and again using that extra information.

This is a good general lesson: not every flag takes an argument. Some just mean "yes, do this."

---

7. `--sif-cache /data/CCBR_Pipeliner/SIFs`

Points to a folder of pre-downloaded **Singularity Image Files (.sif)** — these are containers, self-contained bundles of software and dependencies, similar in spirit to Docker images but used on HPC systems. Pointing to a shared cache means RENEE reuses containers that already exist there instead of re-downloading them, saving time and storage.

---

8. `--partition student`

Another Slurm-related option: a **partition** is a named subset of a cluster's compute nodes (e.g., grouped by hardware type or by who's allowed to use them). `student` here refers to a partition reserved for training use.

---

9. `--time 8:00:00`

Sets the **walltime limit** for the Slurm job — the maximum time it's allowed to run, in HH:MM:SS format. Here, 8 hours. If the pipeline isn't finished by then, Slurm will terminate it. This is a standard part of requesting resources on shared HPC systems, since the scheduler needs to know how long to reserve resources for.

Key Unix takeaways for students

Concept	Example from this command	What it means
Subcommand	run`	

Concept	Example from this command	What it means
		Selects which action a multi-purpose tool performs
Absolute path	`/data/classes/...`	Full location starting from root /
Wildcard *	`*.R?.fastq.gz`	Matches any number of characters
Wildcard ?	`R?`	Matches exactly one character
Environment variable	`$USER`	Shell substitutes in a stored value
Flag with a value	`--genome hg38_36`	`--flag value` pairs configure behavior
Boolean flag	`--star-2-pass-basic`	Presence alone toggles a feature on
Line continuation	`\` at end of line	Lets one long command be split across lines for readability

Module 2 - Bulk RNA Sequencing

Running the RENEE Bulk RNA Sequencing Pipeline

Learning Objectives

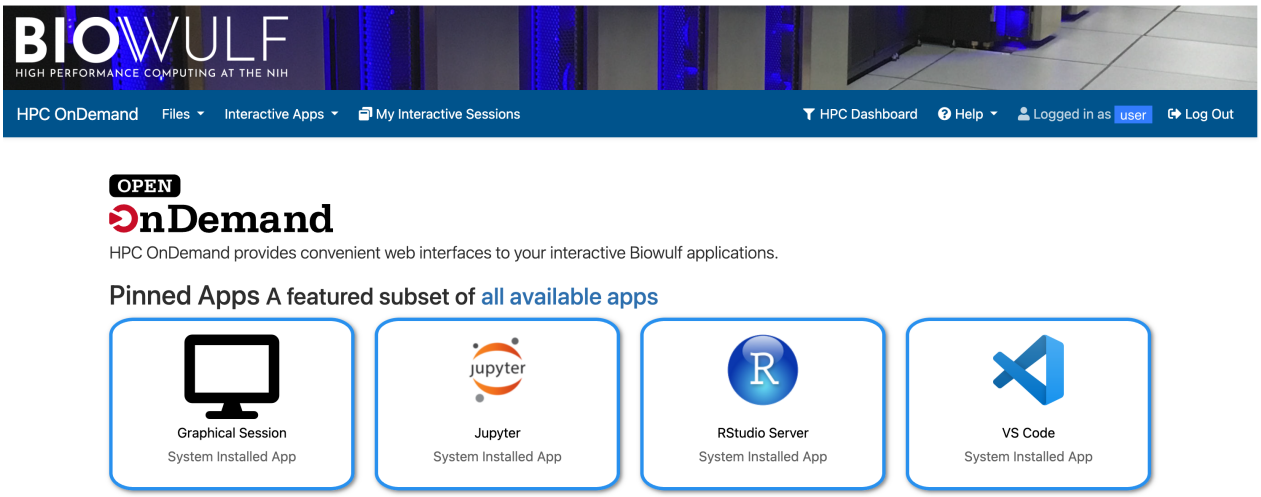
After this class, participants will be able to apply Unix skills to run a bulk RNA sequencing analysis pipeline on Biowulf.

Signing onto Biowulf's HPC OnDemand

Biowulf staff has provided this class 40 student accounts for accessing HPC OnDemand and each participant has been assigned to one.

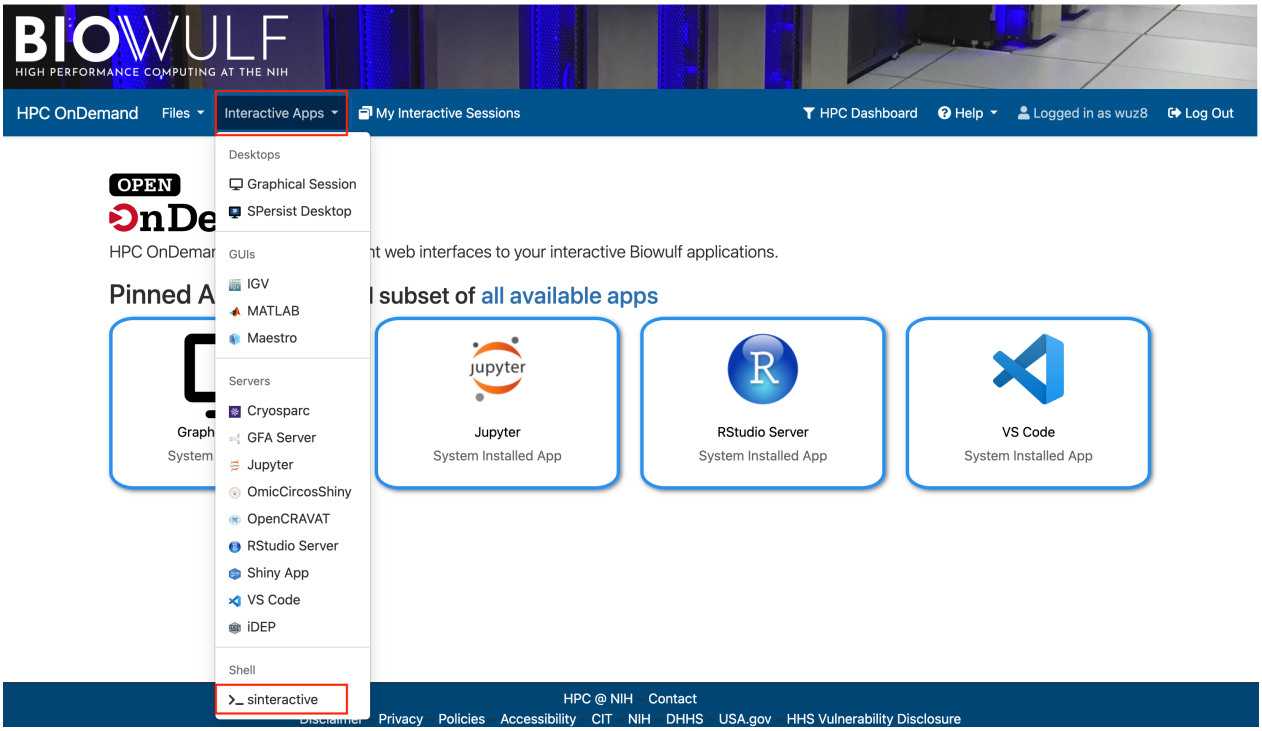
> **Note**
>
> For student accounts, use https://hpcclass.cit.nih.gov/ *(https://hpcclass.cit.nih.gov/)* to connect. For those connecting using personal Biowulf accounts use https://hpcondemand.nih.gov/ *(https://hpcondemand.nih.gov/)* and log in via PIV.

Upon logging in, users will see a sign that says "Logged in as" followed by whatever the username is on the top right. For student accounts, this will say "student#" where "#" is any number from 1 through 40. On the other hand, the participant's NIH username is displayed for personal Biowulf accounts. Several tabs presented in the homepage of HPC OnDemand enable launching some popular graphical applications such as R Studio and Jupyter Lab.

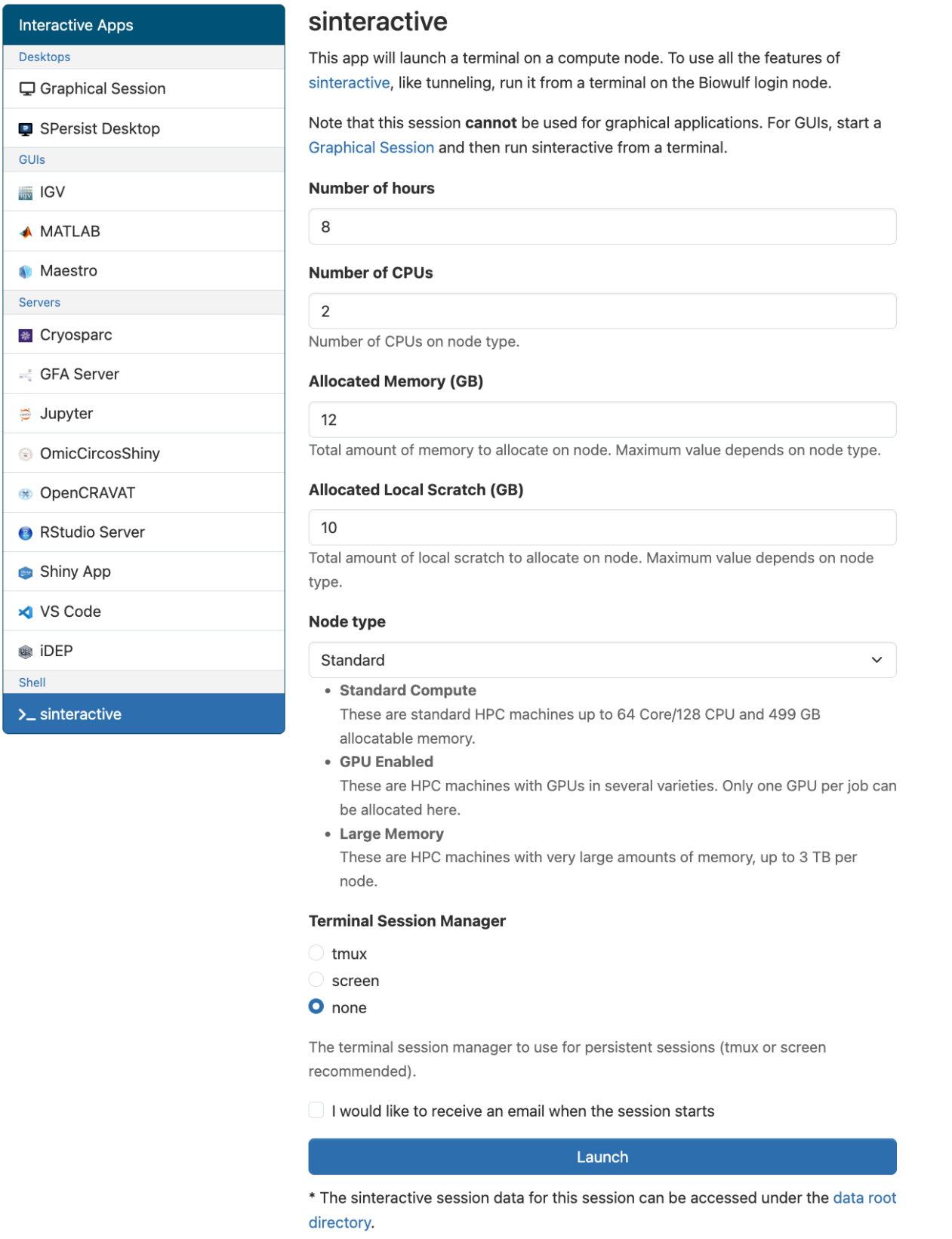


Get an Interactive Terminal on HPC OnDemand

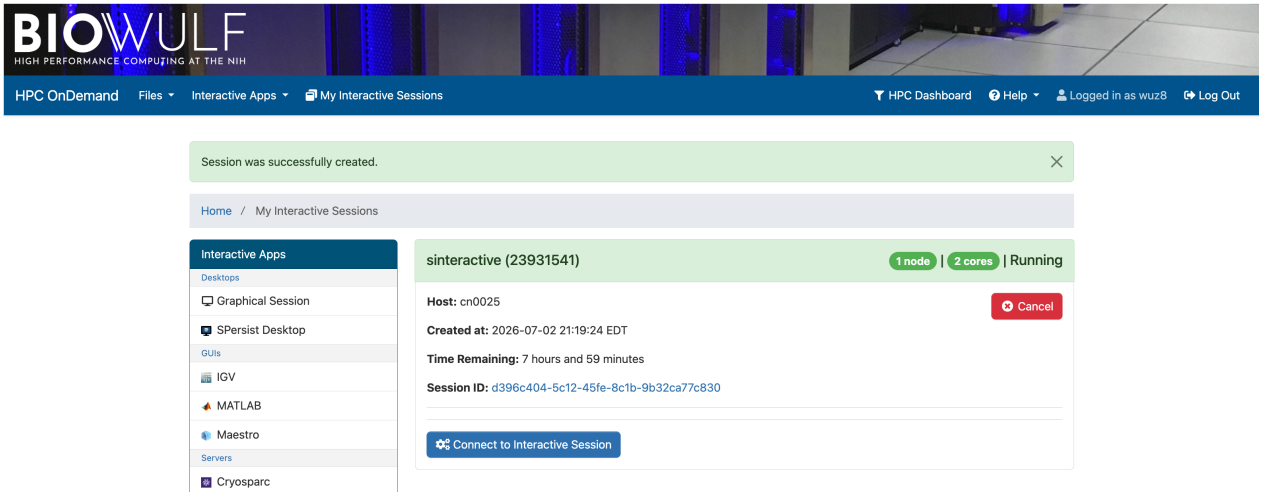
After signing onto HPC OnDemand, select the "Interactive Apps" drop down and choose to start a terminal session on one of Biowulf's compute nodes. This terminal session consumes one of the two allotted Biowulf interactive sessions.



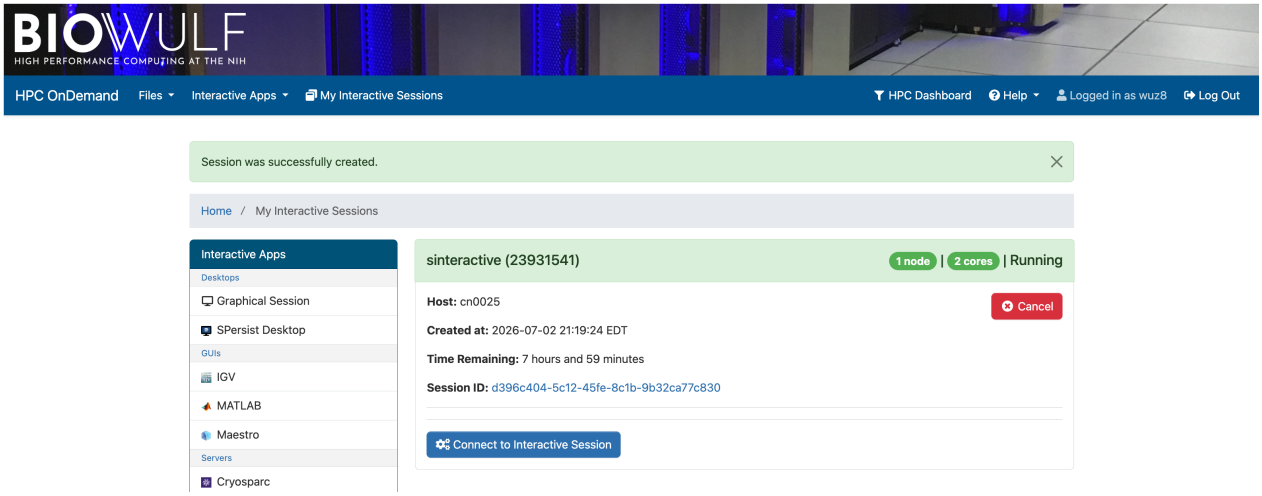
In the next page enter the compute resources needed. Here, the default will do. Scroll to the bottom and click launch when ready.



After resources have been granted, a button labeled "Connect to Interactive Session" will appear. Click on this.



Users will then be taken to a terminal.



Change into Participant's Biowulf Data Directory

Change into the participant's Biowulf data directory using the command below. The variable `$USER` points to the participant's Biowulf user ID.

```
cd /data/$USER
```

Then make a folder called `hcc1395_renee_b4b`. List the content of the `data` folder to confirm that the output directory has been created. The output from the example bulk RNA sequencing analysis will be written here.

```
mkdir hcc1395_renee_b4b
```

RENEE Pipeline General Usage and Documentation

The GitHub repository for RENEE can be found at https://github.com/CCBR/RENEE *(https://github.com/CCBR/RENEE)*. A detail set of instructions for running this pipeline can be found at https://ccbr.github.io/RENEE/latest/ *(https://ccbr.github.io/RENEE/latest/)*.

> **Tip**
>
> If working from the terminal rather than HPC OnDemand, use `sinteractive` to get a session on Biowulf's compute node to run the RENEE.

The following messages will be printed to the terminal once the interactive compute session is granted.

```
salloc: Pending job allocation 25537775
salloc: job 25537775 queued and waiting for resources
salloc: job 25537775 has been allocated resources
salloc: Granted job allocation 25537775
salloc: Nodes cn0011 are ready for job
```

Next, the `ccbrpipeliner` module will need to be loaded. This module enables scientists to access all CCBR analysis pipelines.

```
module load ccbrpipeliner
```

After the `ccbrpipeliner` is module loaded, users will see the message below listing all of the available pipelines for analyzing different sequencing data types.

```
    "ccbrpipeliner" is a suite of end-to-end pipelines and tools
    Visit https://github.com/CCBR for more details.
    Pipelines are available on BIOWULF and FRCE.
    Tools are available on BIOWULF, HELIX and FRCE.

    The following pipelines/tools will be loaded in this module:

    PIPELINES:
    ASPEN       v1.1    ATAC-seq        https://ccbr.github.io/ASPEN/
    CARLISLE    v2.7    CUT&RUN         https://ccbr.github.io/CARLIS
    CHAMPAGNE   v0.5    ChIP-seq        https://ccbr.github.io/CHAMPA
    CHARLIE     v0.12   circRNAs        https://ccbr.github.io/CHARLI
    CRISPIN     v1.2    CRISPR          https://ccbr.github.io/CRISPI
    ESCAPE      v1.2    EV-seq          https://ccbr.github.io/ESCAPE
    LOGAN       v0.3    whole genome seq    https://ccbr.github.io/LO
```

```
    RENEE       v2.7    bulk RNA-seq        https://ccbr.github.io/RI
    SINCLAIR    v0.3    scRNA-seq       https://ccbr.github.io/SINCL/
    XAVIER      v3.2    whole exome-seq     https://ccbr.github.io/X/

    TOOLS:
    spacesavers2    v0.14           https://ccbr.github.io/spacesaver
    permfix     v0.6            https://github.com/ccbr/permfix/
    ccbr_tools  v0.4            https://ccbr.github.io/Tools/

################################################################################
            Thank you for using CCBR Pipeliner
            Comments/Questions/Requests:
            CCBR_Pipeliner@mail.nih.gov
################################################################################
```

Use `renee --help` to see what scientists can do with this pipeline.

```
renee --help
```

```
[+] Loading singularity  4.3.7  on cn4310
[+] Loading snakemake  7.32.4
Python version: 3.12.11
usage: renee [-h] [--version] {run,gui,build,unlock,cache,debug} ...

a highly-reproducible RNA-seq pipeline

positional arguments:
  {run,gui,build,unlock,cache,debug}
                        List of available sub-commands
    run                 Run the RENEE pipeline with your FastQ files
    gui                 Launch the RENEE pipeline with a Graphical Us
    build               Builds the reference files for the RENEE pipe
    unlock              Unlocks a previous runs output directory.
    cache               Cache software containers locally.
    debug               Debug the RENEE pipeline base directory.

options:
  -h, --help            show this help message and exit
  --version             show program's version number and exit
```

The `run` subcommand in RENEE enables users to run the pipeline. Further, `build` allows users to build a custom genome for their analysis. However, this not necessary as CCBR has already built genomes for human and mouse (see https://ccbr.github.io/RENEE/latest/RNA-seq/Resources/#1-reference-genomes *(https://ccbr.github.io/RENEE/latest/RNA-seq/Resources/#1-*

reference-genomes)). For this class, the RNA sequencing data will be aligned to hg38_36 *(https://www.gencodegenes.org/human/release_36.html)*.

Next, the command construct below will be used to get the bulk RNA analysis on the example dataset going. Remember the `\` at the end of each line denotes continuation so the terminal runs all lines as one command. This is included to enhance document readability. Explanation for the command construct is below.

- `renee run`:
  - Each RENEE task starts with `renee` followed by the sub-command (`run` here to run an analysis).
- `--input`:
  - Prompts for path to `FASTQ` files. These are stored in `/data/classes/BTEP/hcc1395_renee_b4b/reads`.
  - The use of globbing enables RENEE to grab all FASTQ files at the same time.
    - `*` matches any sequence of characters, including no characters.
    - `?` matches exactly one character and can be used to distinguish read 1 from read 2 in paired-end sequencing files.
- `--output`:
  - Prompts for the output directory. Here is `hcc1395_renee_b4b`.
- `--genome`:
  - Here, users will enter reference genome (ie. hg38_36).
  - For custom genomes, supply the path for the reference genome `json` file.
- `--mode`:
  - `slurm` is used for this argument to allow for submitting of this analysis to the Biowulf cluster and leave it running.
- `--star-2-pass-basic`:
  - Enables better splice junction detection.
- `--sif-cache`:
  - Here, users will enter the path for cache files that act as share resources for the RENEES pipeline. The path used in this examples is `/data/CCBR_Pipeliner/SIFs`.
- `--partition student`:
  - Ensures that the analysis is sent to the `student` partition on Biowulf as student accounts cannot access the `norm` partition.
- `--time`:
  - 8 hours is set here since this the maximum time allowed on a job submitted to the Biowulf student partition.

```
renee run \
  --input /data/classes/BTEP/hcc1395_renee_b4b/reads/*.R?.fastq.gz \
  --output /data/$USER/hcc1395_renee_b4b \
  --genome hg38_36 \
```

```
--mode slurm \
--star-2-pass-basic \
--sif-cache /data/CCBR_Pipeliner/SIFs \
--partition student \
--time 8:00:00
```

Lesson 5: Quality Control and Cleaning of Next Generation Sequences

Lesson 5 Review

Lesson 5 introduced the basics bulk of RNA sequencing, including experimental considerations and basic ideas behind data analysis. In this class, participants will initiate a bulk RNA sequencing analysis using a pipeline and be introduced to quality control metrics for FASTQ files derived from Next Generation Sequencing and data cleanup steps required for downstream analysis.

Learning Objectives

After this class, participants will be able to:

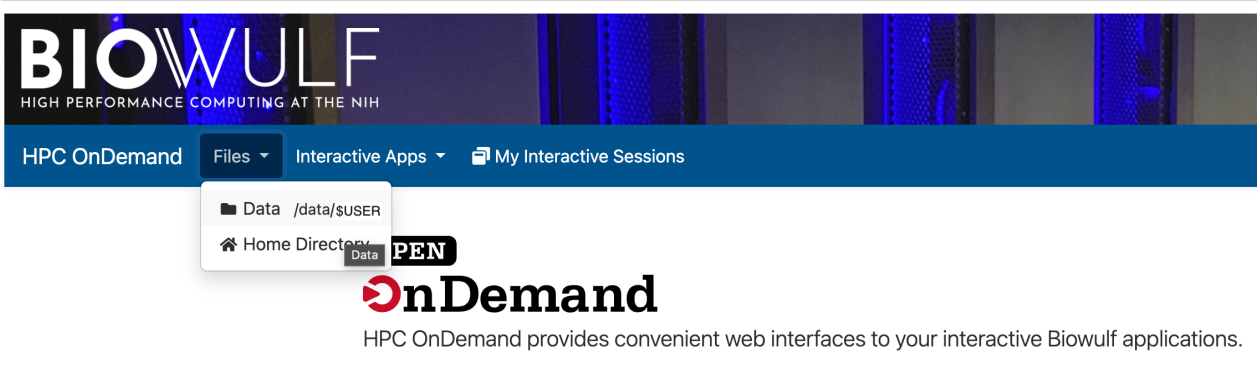
- Apply Unix skills to run a bulk RNA sequencing analysis pipeline.
- Interpret quality assessment metrics for next generation sequences.
- List tools and describe data cleanup procedures for next generation sequencing data.

Sign onto Biowulf

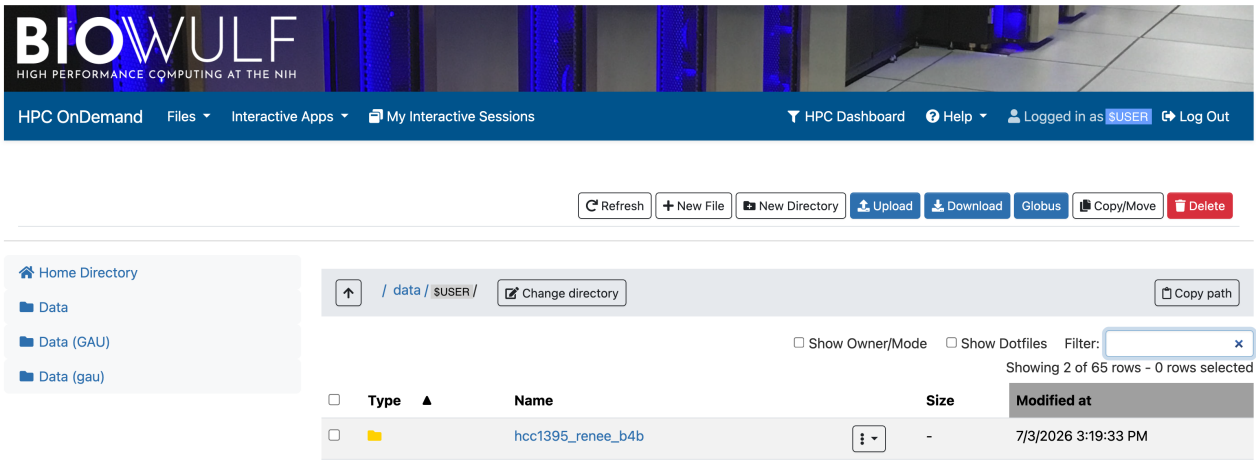
Before getting started, sign onto Biowulf's HPC OnDemand either through student account at https://hpcclass.cit.nih.gov/ *(https://hpcclass.cit.nih.gov/)* or using personal account at https://hpcondemand.nih.gov/ *(https://hpcondemand.nih.gov/)*.

Navigate the RENEE Output Directory

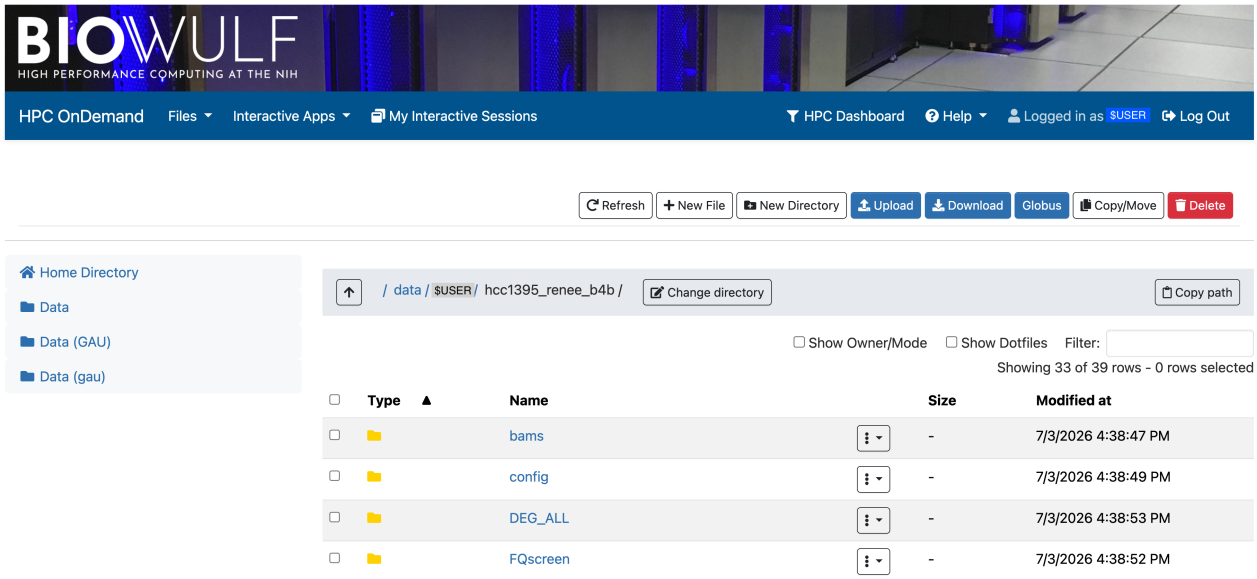
After signing on, navigate to the participant's `/data/$USER` folder by clicking on "Files". Recall that `$USER` is a variable that points to participant's Biowulf user ID.



Next, navigate to the `hcc1395_renee_b4b` folder.



This folder contains alignment results stored in the subfolder `BAM`, `gene expression tables stored in`DEG_ALL`, and other output such as QC metrics.



List the contents of the `hcc1395_renee_b4b` folder in the terminal.

```
ls -al
```

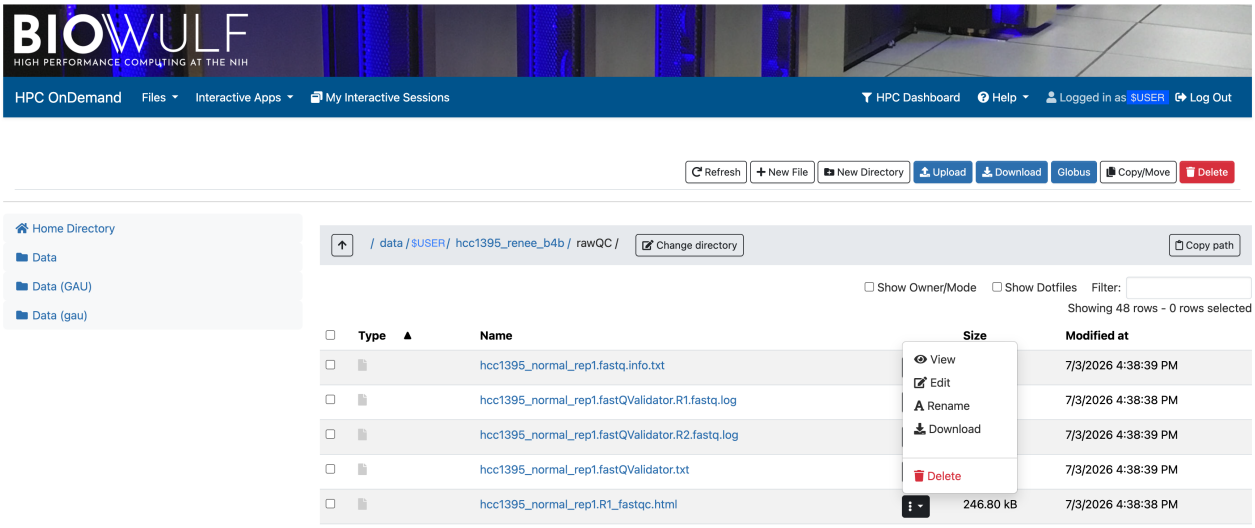
```
total 11
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:39 .
drwxrwx--- 2 $USER $USER 4096 Jul  3 16:38 ..
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 bams
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 .cache
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 .config
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 config
-rw-r----- 1 $USER $USER 8963 Jul  3 16:38 config.json
```

```
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 DEG_ALL
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 FQscreen
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 FQscreen2
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 fusions
lrwxrwxrwx 1 $USER $USER   66 Jul  3 16:38 hcc1395_normal_rep1.R1.fas
lrwxrwxrwx 1 $USER $USER   66 Jul  3 16:38 hcc1395_normal_rep1.R2.fas
lrwxrwxrwx 1 $USER $USER   66 Jul  3 16:38 hcc1395_normal_rep2.R1.fas
lrwxrwxrwx 1 $USER $USER   66 Jul  3 16:38 hcc1395_normal_rep2.R2.fas
lrwxrwxrwx 1 $USER $USER   66 Jul  3 16:38 hcc1395_normal_rep3.R1.fas
lrwxrwxrwx 1 $USER $USER   66 Jul  3 16:38 hcc1395_normal_rep3.R2.fas
lrwxrwxrwx 1 $USER $USER   65 Jul  3 16:38 hcc1395_tumor_rep1.R1.fast
lrwxrwxrwx 1 $USER $USER   65 Jul  3 16:38 hcc1395_tumor_rep1.R2.fast
lrwxrwxrwx 1 $USER $USER   65 Jul  3 16:38 hcc1395_tumor_rep2.R1.fast
lrwxrwxrwx 1 $USER $USER   65 Jul  3 16:38 hcc1395_tumor_rep2.R2.fast
lrwxrwxrwx 1 $USER $USER   65 Jul  3 16:38 hcc1395_tumor_rep3.R1.fast
lrwxrwxrwx 1 $USER $USER   65 Jul  3 16:38 hcc1395_tumor_rep3.R2.fast
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 .java
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 kraken
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 logfiles
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 .parallel
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 preseq
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 QC
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 QualiMap
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 rawQC
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 Reports
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 resources
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 RSeQC
-rw-r----- 1 $USER $USER  139 Jul  3 16:38 run_log.R
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 .singularity
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 .snakemake
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:39 STAR_files
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 trim
-rw-r----- 1 $USER $USER  583 Jul  3 16:39 websockify.log
drwxr-x--- 2 $USER $USER 4096 Jul  3 16:38 workflow
```

FASTQC

FastQC *(https://www.bioinformatics.babraham.ac.uk/projects/fastqc/)* is a tool used to assess quality of next generation sequencing data derived from Illumina instruments.

The FastQC reports for each FASTQ file are stored in the `rawQC` folder. Users will typically inspect the `html` report files.

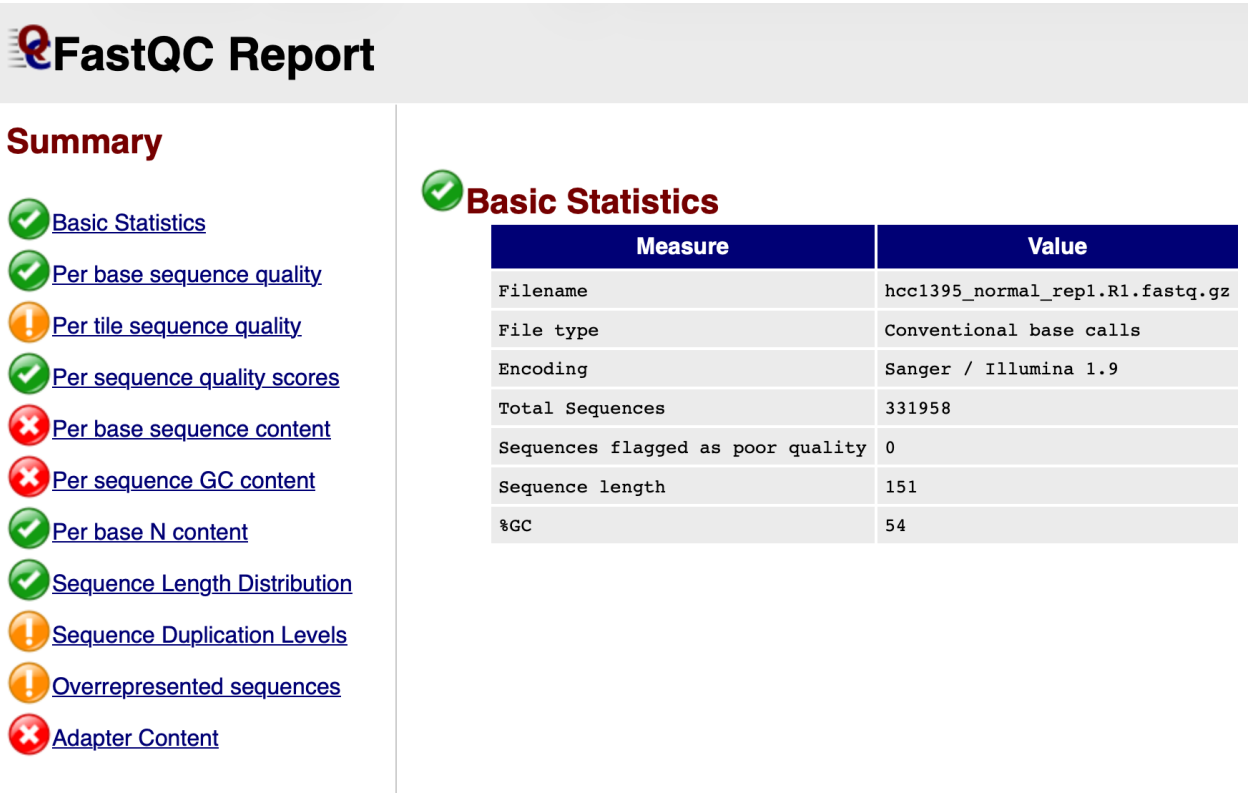


Click the drop down next to `hcc1395_normal_rep1.R1_fastqc.html` to download it on to local computer to learn what is inside a FastQC report.

Components to a FastQC Report

Upon opening a FastQC report, users will see a navigator menu that links to different sections of the report. In addition, there is a basic statistics table. This table contains the information below.

- Name of the FASTQ file in which the report was generated for (hcc1395_normal_rep1.R1 in this case)
- The number of sequences in the file.
- Number of sequences flagged with poor quality.
- The sequence length (ie. how many bases are in each sequencing read of the FASTQ file).



FastQC Report

Summary

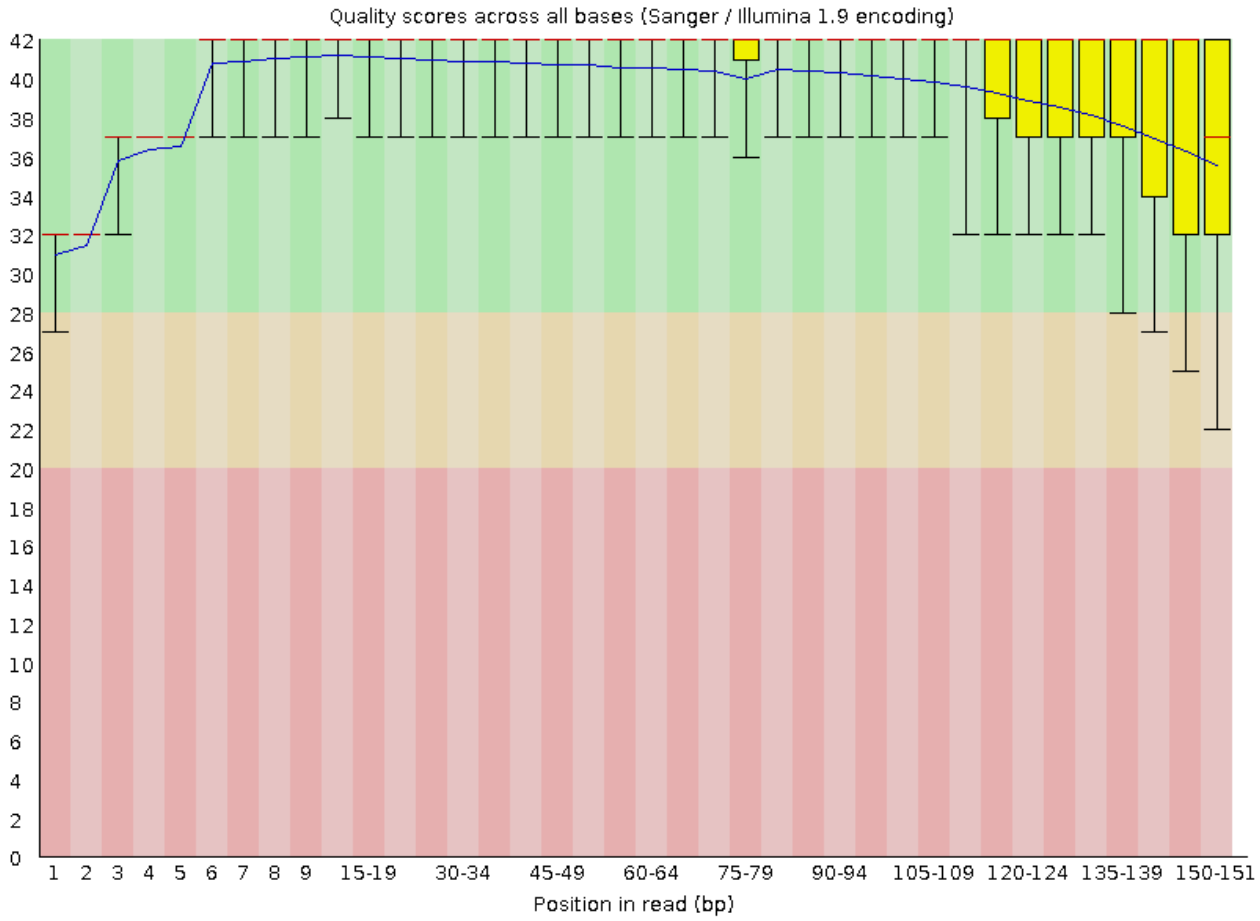
- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content

Basic Statistics

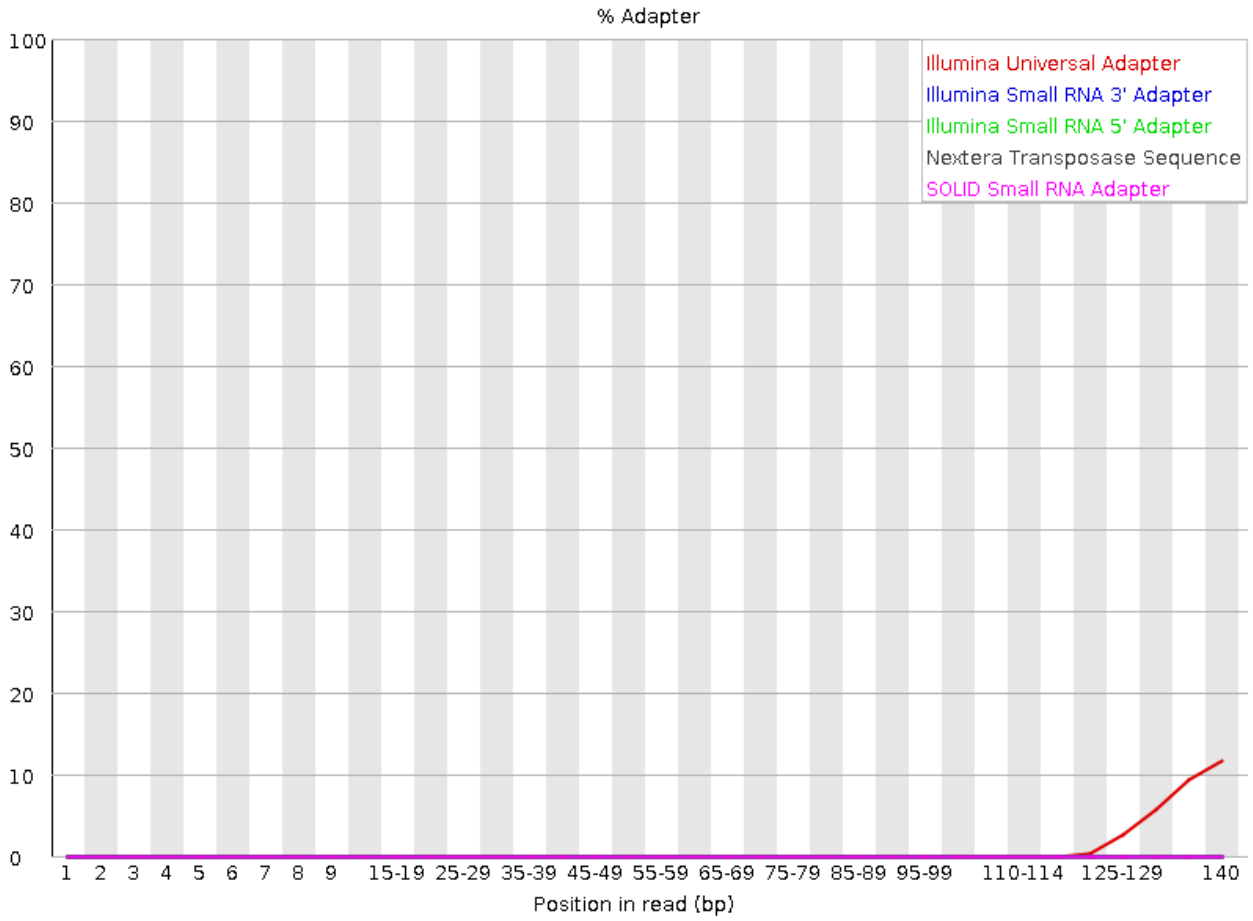
Measure	Value
Filename	hcc1395_normal_rep1.R1.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	331958
Sequences flagged as poor quality	0
Sequence length	151
%GC	54

To not be exhaustive, this class will point out two important quality metrics in the FastQC report. First, there is the "Per base sequence quality" plot, which plots the error likelihood at each base position averaged over all sequences in a FASTQ file.

- On the vertical axis are the quality scores that are in row 4 of the sequencing reads in a FASTQ file. These quality scores represent error probabilities, where:
    - 10 corresponds to 10% error (1/10),
    - 20 corresponds to 1% error (1/100),
    - 30 corresponds to 0.1% error (1/1000) and
    - 40 corresponds to one error every 10,000 measurements (1/10,000) that is an error rate of 0.01%
- The three colored bands (green, yellow, red) illustrate the typical labels assigned to these measure:
    - reliable (28-40, green)
    - less reliable (20-28, yellow)
    - error prone (1-20, red)
- The yellow boxes contain 50% of the data, the whiskers indicate the 75% outliers.
- The red line inside the yellow boxes is the median quality score for that base.
- The blue line is the average quality score at a particular base.



Then, if there is adapter contamination in the sequences. This example shows that the raw FASTQ file contains adapter contamination.



The quality of and whether there are adapter contamination in the sequences are important QC metrics because:

1. Scientists would want high quality sequences (ie. those that have low likelihood for error).
2. Sequencing adapters interfere with identifying where in the genome each read came from.

There are other metrics in a FastQC report. Please see https://www.bioinformatics.babraham.ac.uk/projects/fastqc/ *(https://www.bioinformatics.babraham.ac.uk/projects/fastqc/)* to learn more.

The presence of over represented sequences may indicate adapter contamination, low complexity sequencing library, or other artifacts. FASTQC will trigger a warning when the content of over represented sequences are more than 0.1%. Failure occurs when the content of overrepresented sequences is more than 1%.

Overrepresented sequences

Sequence	Count	Percentage	Possible Source
GTGGAAAATAACTTTTATTGAGACCCCACCAACTGCAAAATCTGTTCCTG	643	0.19369920291121165	No Hit
GCCAATTCTTATAGAAACGCCTCTTGCATTCATCACTGATGTGCTCAGCA	463	0.13947547581320527	No Hit
GGTCACTTCACCATAGTGGACAAAGCCACCCAGAGGGTTGATGCTCTTGT	378	0.11386982690581339	No Hit
GTTGTCAATGAGGATATTTATTGGGGTTTCATGAGTGCAGGGAGAAGGGC	354	0.1066399966260792	No Hit
CTGGAGTCTTGGAAGCTTGACTACCCTACGTTCTCCTACAAATGGACCTT	335	0.10091638098795631	No Hit

MultiQC

Because FastQC generates quality metrics for each FASTQ file in a study, it becomes inefficient to interrogate QC metrics. MultiQC *(https://github.com/MultiQC/MultiQC)* is a tool that can merge individual FastQC reports as well as enable the inclusion of things such as alignment metrics and other quality measures that are important in Next Generation Sequencing. MultiQC generate interactive `html` reports. Similar to the FASTQC report, users have quick access to different portions of the MultiQC report via a navigation panel on the left.

General Statistics Table

The first item in the MultiQC report is a table that summarizes sample level QC metrics. The information here varies by analysis. The information in this table include the following.

- Whether there is genetic contamination from species that are not of interest. This is a human study, so it is expected that the sequencing reads for the most part will map to the human genome even though some reads may map to different species due to conservation. Here, over 99% of reads in each sample map to human while an additional small percentage map to other species. This provides one level of confidence that the data is sound.
- Information about the sequences:
    - %Dup shows the percentage of duplicate sequences found in each sample. Here, the duplication rates for the samples are approximately 40%, which will raise a flag in FASTQC (see FASTQC section on duplication) *(https://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help/3%20Analysis%20Modules/8%20Duplicate%20Sequences.html)*. When it comes to RNA sequencing, duplicate reads can be biological meaningful as there could be many copies of a transcript. However, duplication could also indicate artifact generated during PCR amplification or low complexity library.
    - %GC contains the GC content for each sample. Ideally, this should be same as the GC content for the specie's genome.
- The subsequent columns summarize downstream analysis results, including the STAR alignment rate, coverage-related metrics, and RSEM mapping or assignment rates. These

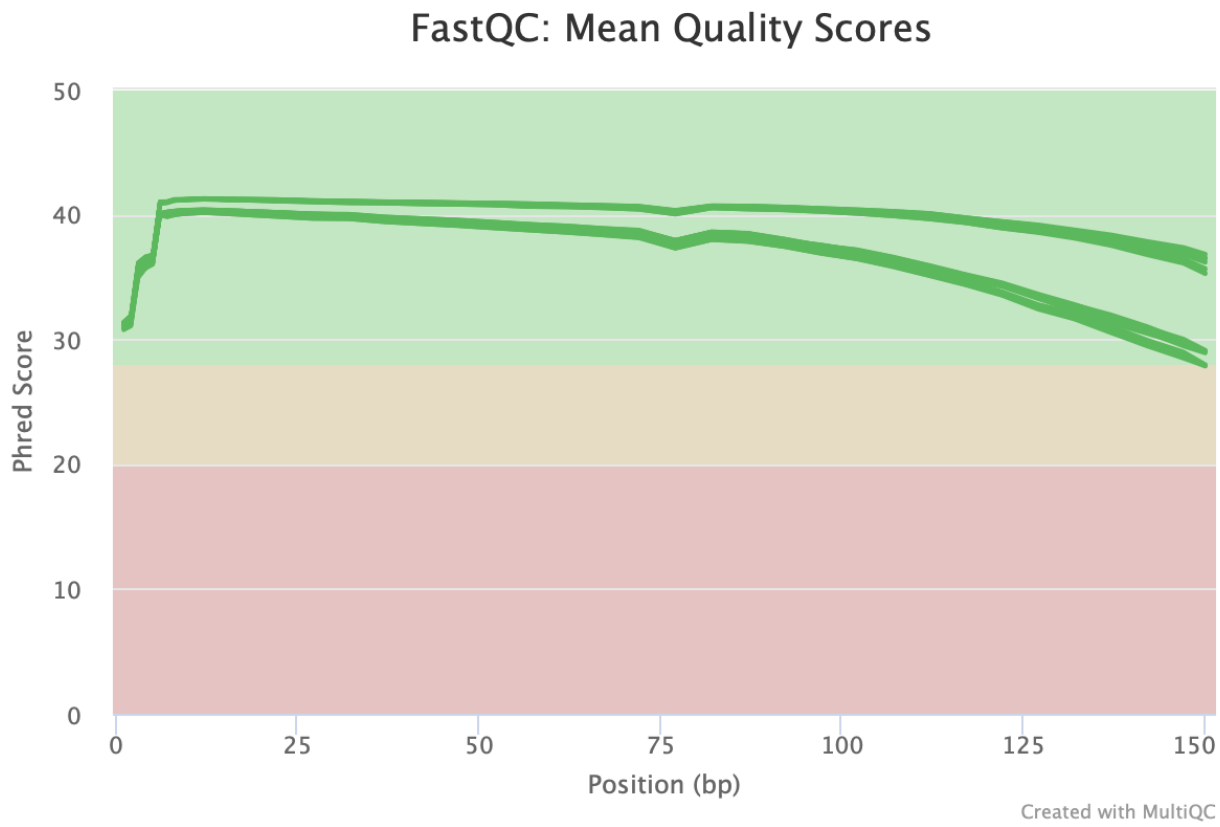
metrics help assess how well the reads align to the reference genome and how effectively they are used for transcript-level quantification.

General Statistics

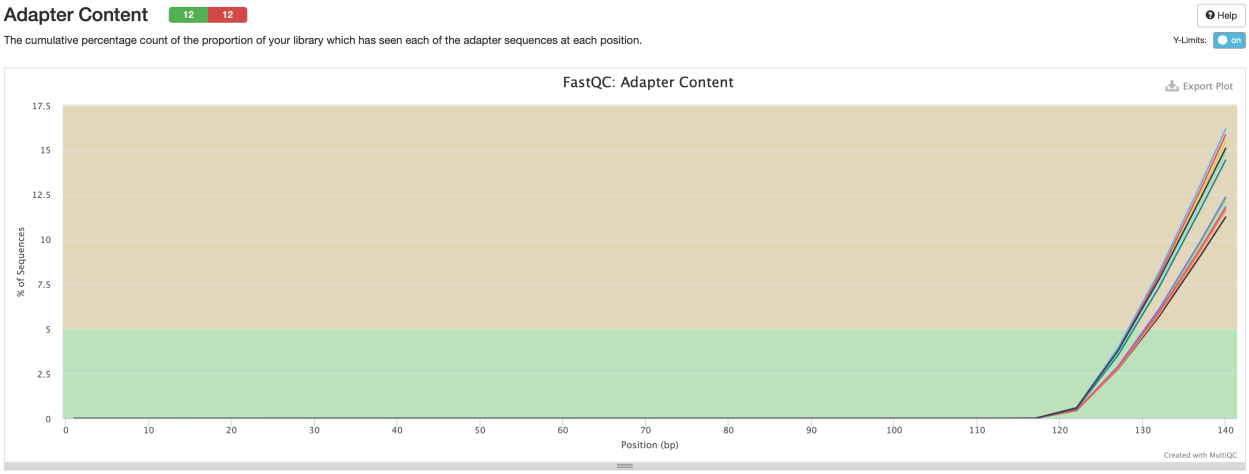
Copy table | Configure Columns | Plot | Showing 36/36 rows and 19/34 columns.

Sample Name	% Homo sapiens	% Top 5 Species	% Unclassified	% Dups	% GC	M Seqs	% BP Trimmed	% Aligned	M Aligned	% Alignable	Ins. size	Median cov	Mean cov	% Aligned	TIN	% rRNA	% mRNA	% Dups	M Reads Mapped
hcc1395_normal_rep1								93.2%	0.3	94.2%	224	0.0X	0.7X	95.4%	71.2	0.0%	86.2%	13.6%	0.7
hcc1395_normal_rep1.R1				43.3%	54%	0.3													
hcc1395_normal_rep1.R1.trim				43.7%	54%	0.3													
hcc1395_normal_rep1.R2				38.9%	54%	0.3	2.6%												
hcc1395_normal_rep1.R2.trim				39.4%	54%	0.3													
hcc1395_normal_rep1.trim.kraken_bacteria.taxa	99.2%	99.7%	0.1%																
hcc1395_normal_rep2								93.4%	0.3	94.5%	222	0.0X	0.7X	95.7%	71.9	0.0%	86.2%	13.9%	0.7
hcc1395_normal_rep2.R1				43.0%	54%	0.3													
hcc1395_normal_rep2.R1.trim				43.4%	54%	0.3													
hcc1395_normal_rep2.R2				40.1%	54%	0.3	2.7%												
hcc1395_normal_rep2.R2.trim				40.7%	54%	0.3													
hcc1395_normal_rep2.trim.kraken_bacteria.taxa	99.2%	99.7%	0.1%																
hcc1395_normal_rep3								93.5%	0.3	94.7%	221	0.0X	0.7X	95.8%	70.6	0.0%	86.1%	14.1%	0.7
hcc1395_normal_rep3.R1				44.1%	54%	0.3													
hcc1395_normal_rep3.R1.trim				44.5%	54%	0.3													

MultiQC aggregates the sequencing quality histogram generated by FASTQC for each sample. The per base qualities for untrimmed and trimmed FASTQ files are available on the same plot.



The adapter content for both the untrimmed and trimmed sequences are aggregated in the MultiQC report as well. Note that prior to trimming, this module failed (ie. 12 FASTQ files contain sequences with adapter read through).



Looking at all samples in a MultiQC plot can be inundating. The filter toolbox can be used to select and highlight those samples that the researcher wants to view.

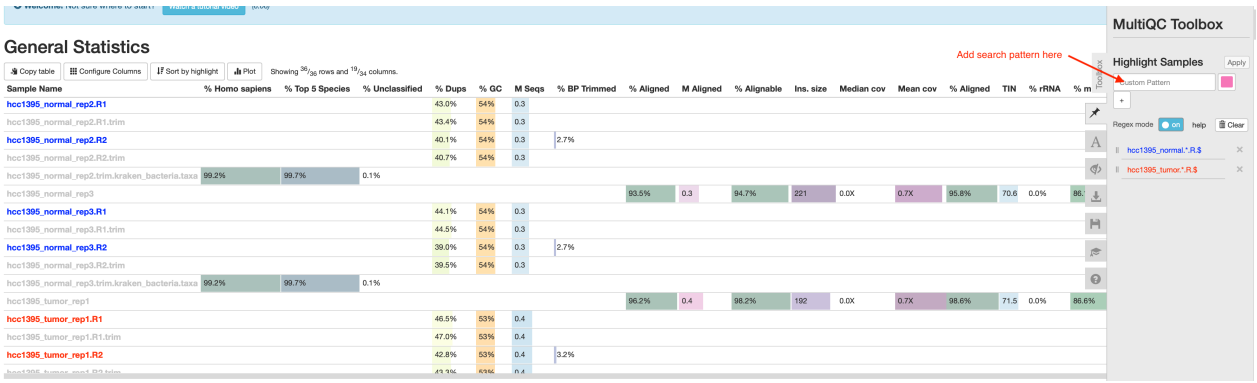
To filter only those samples that the user wants to view, click on the "Toolbox". Enter the search patterns, click "+". When finished adding search patterns, click apply. MultiQC enables pattern search in regular expression or regex mode *(https://www.geeksforgeeks.org/python/python-regex/)*. See https://learn.microsoft.com/en-us/dotnet/standard/base-types/regular-expression-language-quick-reference *(https://learn.microsoft.com/en-us/dotnet/standard/base-types/regular-expression-language-quick-reference)* for a list of regex patterns.

The following patterns will filter and highlight the QC metrics for the untrimmed normal and tumor FASTQ files. The normal samples will be highlighted in blue while tumor samples will be highlighted in orange. Users can select the color in the color drop down box. In the pattern below:

- The first part of the pattern is either `hcc1395_normal` or `hcc1395_tumor`.
- The following `.` denotes matches anyone character and the `*` after says any number of occurences are allowed.
- `R` is then added to signify read 1 (R1) or read 2 (R2), although there is no need to be explicit since the proceeding `.` tells MultiQC to grab any one character at the end of search pattern (denoted by `$`).

```
hcc1395_normal.*.R.$
```

```
hcc1395_tumor.*.R.$
```



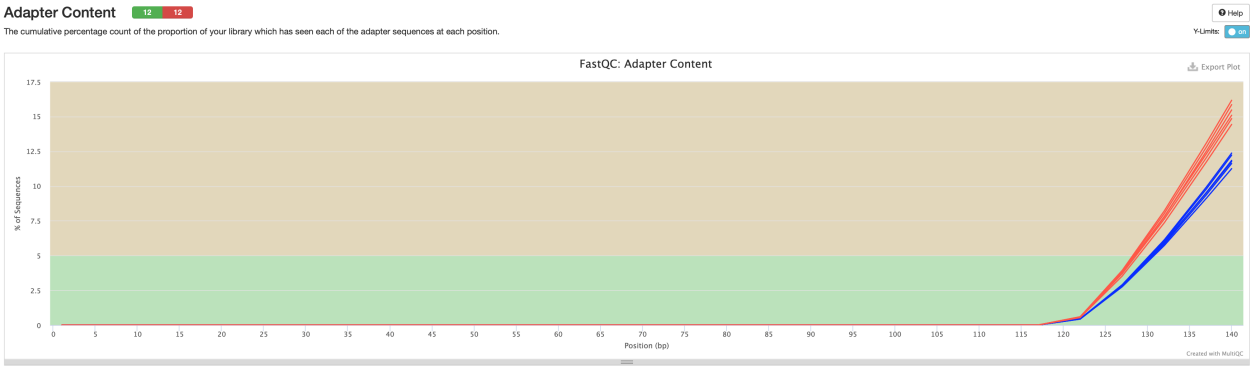
Use the following regex patterns to filter out the normal and tumor untrimmed FASTQ adapter content plots. The `\s` in the search pattern denote spaces (`\` serves as an escape for `s` so that the computer does interprets as a space rather than the letter s).

```
hcc1395_normal.*R.\s-\sillumina
```

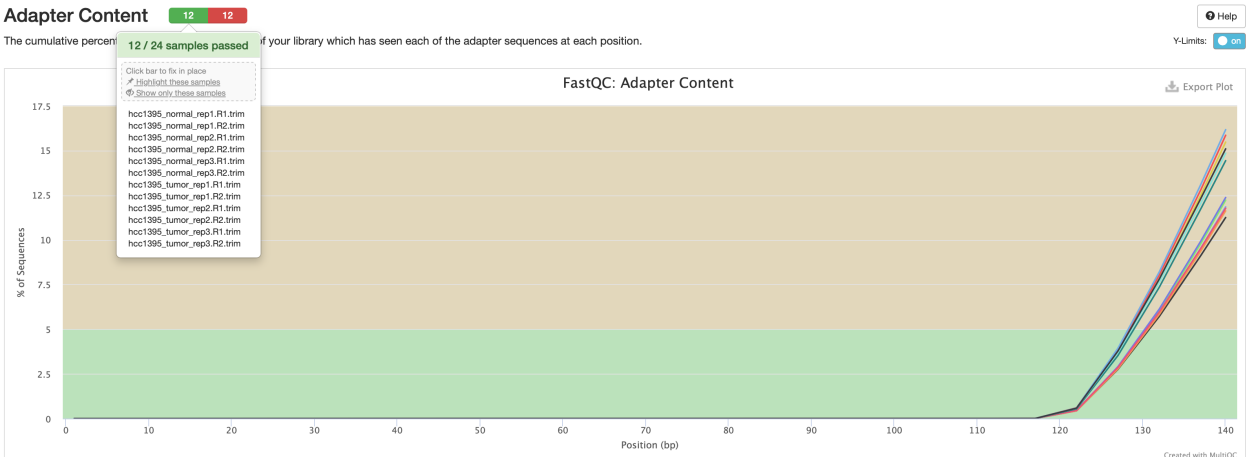
```
hcc1395_tumor.*R.\s-\sillumina
```

Cutadapt Trimming

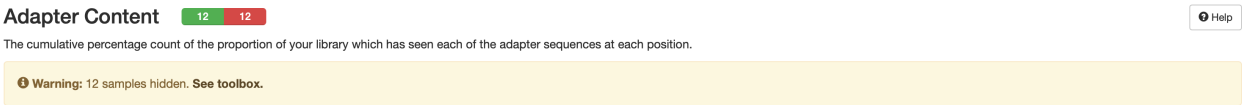
RENEE uses Cutadapt *(https://cutadapt.readthedocs.io/en/stable/)* to remove adapter contamination from sequencing reads. The plot below from MultiQC shows the distribution of the number of reads trimmed from each FASTQ file. Most FASTQ files have 30 or less reads trimmed from them.



Click on the green box and select "Show only these samples" to see the adapter contamination information for the trimmed FASTQ files.



The adapter content plot dissappeared since the trimmed FASTQ files do not contain adapter contamination.



Other tools for trimming include

- Trimmomatic *(http://www.usadellab.org/cms/?page=trimmomatic)* performs adapter and quality trimming for Illumina data.
- BBduk *(https://bbmap.org/tools/bbduk)* uses a kmer based algorithm for adapter and quality trimming.

Rationale for choosing Cutadapt in RENEE.

- Ability to better control adapter trimming.
- Higher reliability.
- Enables the fine tuning of error rates.
- Capability to handle many adapters at once.
- Actively maintained.

Note that post trimming, the quality of the sequences are still good (30 or above).

Enter the search pattern below to view the quality score histogram for the trimmed FASTQ files.

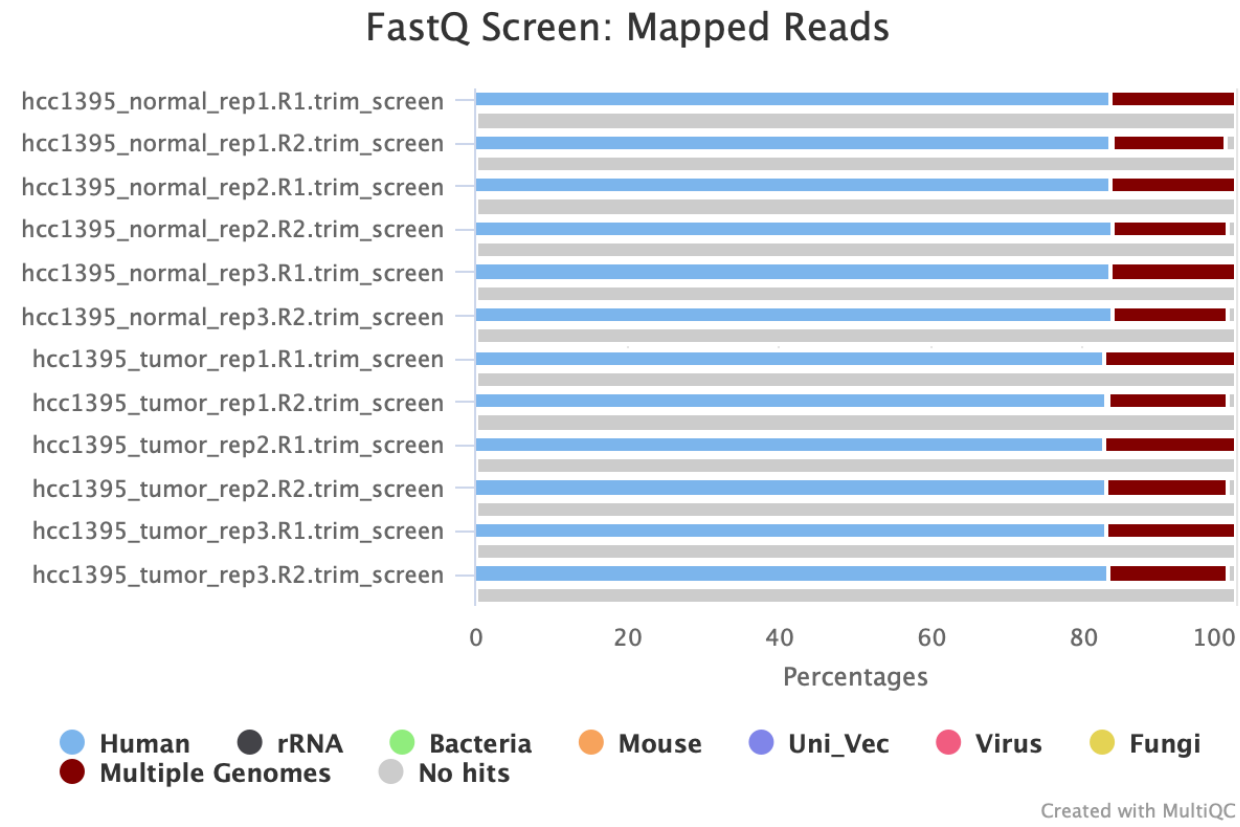
```
hcc1395_normal.*trim$
```

```
hcc1395_tumor.*trim$
```

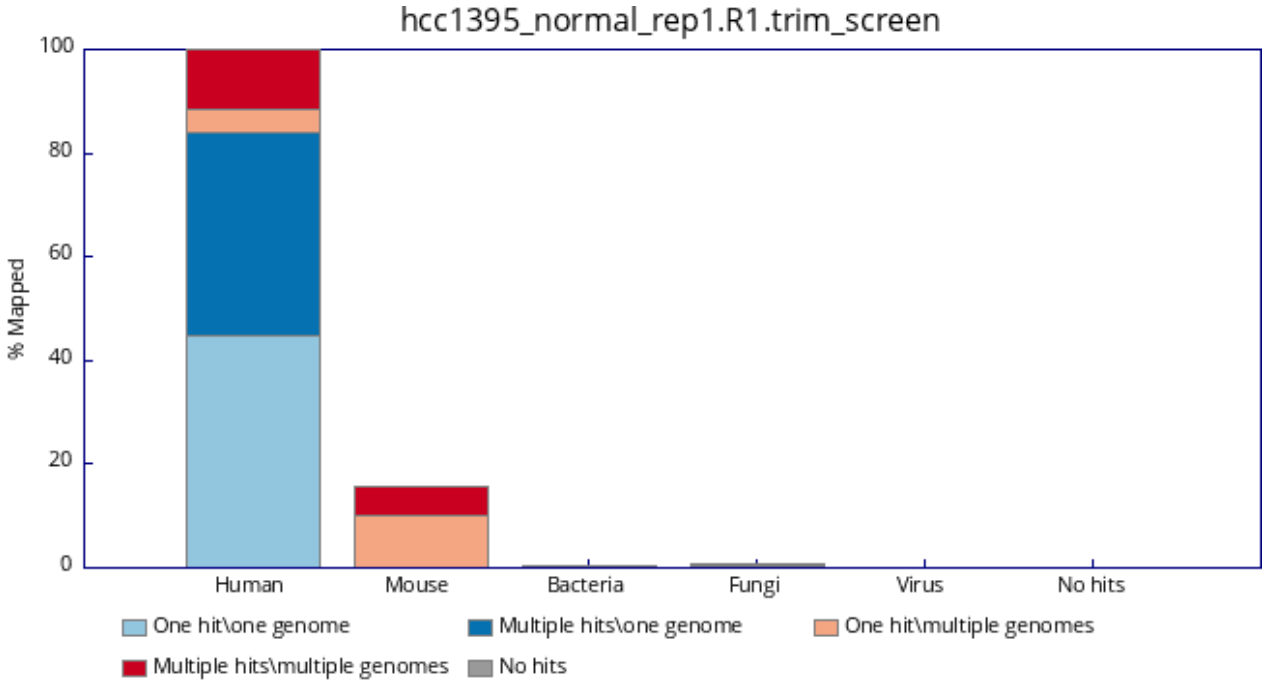
Checking for Genomic Content from Other Species

RENEE uses two software to check whether there are potential genomic contamination from species that are not of interest in the study. For instance, a human bulk RNA sequencing study should generate sequences that map to the human genome. These tools are Kraken *(https://github.com/DerrickWood/kraken)* and Fastqscreen *(https://www.bioinformatics.babraham.ac.uk/projects/fastq_screen/)*(target=_blank). Fastqscreen allows scientists to compare their next generation sequencing data to a database (think of BLAST). On the other hand, Kraken is a taxonomic classification system. As shown in the general statistics table, it is unlikely that there is contamination from non-human genomic content.

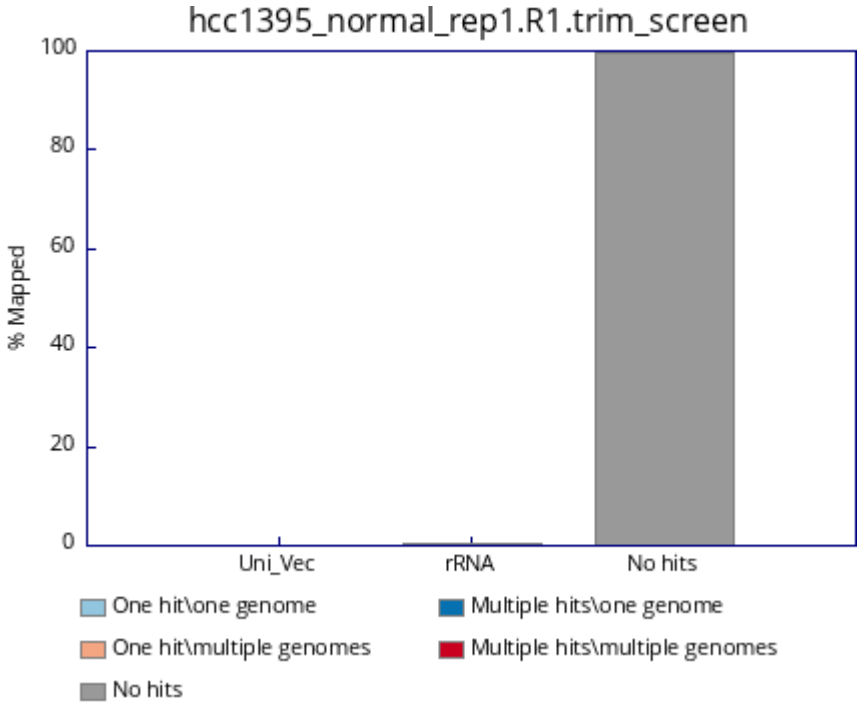
Note that there are two Fastqscreen plots per trimmed FASTQ file. The genomic hits from other species are on the top graph. The bottom reports matches to Uni_vect *(https://www.ncbi.nlm.nih.gov/tools/vecscreen/univec/)* and ribosomal RNA (rRNA) hits. Uni_Vec is a database that houses sequences for vectors, adapters, linkers, and primers. There traces of hits to rRNA sequences but none for Uni_vec.



Navigate to the FQscreen and FQscreen2 folders to take a look at the single sample reports. In terms of genomic content from other species, the plot below shows that 100% of the sequences in this study match humans. Due to things like conservation, less than 20% match that for mice.



The same sample shows no contamination from Uni_vec but some from rRNA while almost 100% of the sequences returned to hits when compared to those databases.


hcc1395_normal_rep1.R1.trim_screen
% Mapped
100
80
60
40
20
0
Uni_Vec
rRNA
No hits
One hit\one genome
Multiple hits\one genome
One hit\multiple genomes
Multiple hits\multiple genomes
No hits

Lesson 6: Alignment of Bulk RNA Sequencing Data

Lesson 5 Review

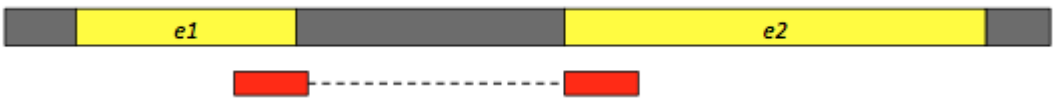
Lesson 5 introduced participants to quality control and cleanup (ie. removal of adapters and/or error prone reads) from bulk RNA data. After cleanup, it is time to determine where in the genome or transcriptome each sequence the samples came from in what is known as the alignment step.

Learning objectives

At the end of this session, participants will be able to describe tools for aligning bulk RNA sequencing data and interpret post alignment quality metrics.

Why Perform Alignment

Bulk RNA sequencing generates millions of short reads per sample. Alignment identifies where in the genome or transcriptome these reads came from. In short, this process takes each of the sequencing reads and identifies the most likely match along a genome or transcriptome. Think of BLAST *(https://blast.ncbi.nlm.nih.gov/Blast.cgi)*. The challenge with bulk RNA sequencing is that sometimes reads map across exons (see figure below), which can be overcome by using splice aware aligners.



A sequencing read (red fragments) aligning two exons (e1 and e2). Modified from: https://training.galaxyproject.org/training-material/topics/transcriptomics/tutorials/rb-rnaseq/tutorial.html *(https://training.galaxyproject.org/training-material/topics/transcriptomics/tutorials/rb-rnaseq/tutorial.html)*.

HISAT2 *(https://daehwankimlab.github.io/hisat2/)* and Spliced Transcripts Alignment to a Reference (STAR) *(https://pmc.ncbi.nlm.nih.gov/articles/PMC3530905/#bts635-F1)* are two aligners for bulk RNA sequencing. RENEE uses the aligner STAR since some advantages include:

- Splice aware
- Ultra fast

- Accurate
- Able to detect unannotated splice junctions (ie. those that are not in the GTF), which preserves reads that were not mapped to annotated splice sites

There also pseudo-aligners that maps reads to transcript and performs quanitification at the same time. These include Salmon *(https://salmon.readthedocs.io/en/latest/salmon.html)* and Kallisto *(https://pachterlab.github.io/kallisto/about)*.

Common to all aligners are reference genome or transcriptome indices. The indices make mapping easier. Think of a library having different sections divided according to topic of the books. When searching for chemistry, there is no need to search starting from A, just start from C. Each aligner requires its own index.

SAM and BAM Files

Next Generation Sequencing alignment results are stored in SAM or BAM *(https://www.htslib.org/doc/samtools-view.html)* files. SAM stands for **sequence** alignment mapped and BAM files are just the binary and compressed version of these. RENEE keeps the BAM files from STAR alignment in the `bams` folder of the output directory, which in this class it is `hcc1359_renee_b4b` in the participants Biowulf `data` directory.

SAM Tools *(https://www.htslib.org/doc/samtools-view.html)* can be used to view and manipulate alignment results such as sorting, indexing, and subsetting to a specific chromosome. Grab an interactive session on Biowulf using `sinteractive` or launch a interactive terminal session through HPC OnDemand.

```
sinteractive
```

Then, `module load samtools`. There is a `bam` file for each sample.

Use the construct below to print the first line of alignment result in `hcc1395_normal_rep1.star_rg_added.sorted.dmark.bam`.

- `samtools`: initiate SAM Tools tasks with this.
- `view`: a subcommand for viewing SAM or BAM output and is followed by the file that the user wants to view (ie. `hcc1395_normal_rep1.star_rg_added.sorted.dmark.bam`)
- The result of `samtools view` is then sent via pipe (`|`) to `head` where the `-1` option prints out the first line.

```
samtools view hcc1395_normal_rep1.star_rg_added.sorted.dmark.bam | he
```

```

```

```
K00193:38:H3MYFBBXX:4:1121:15352:28970  355 chr1    177500  0   148M
```

Including `chr22` after the file name pulls only the alignments for chromosome 22 since this dataset was subsetted to human chromosome 22.

```
samtools view hcc1395_normal_rep1.star_rg_added.sorted.dmark.bam chr
```

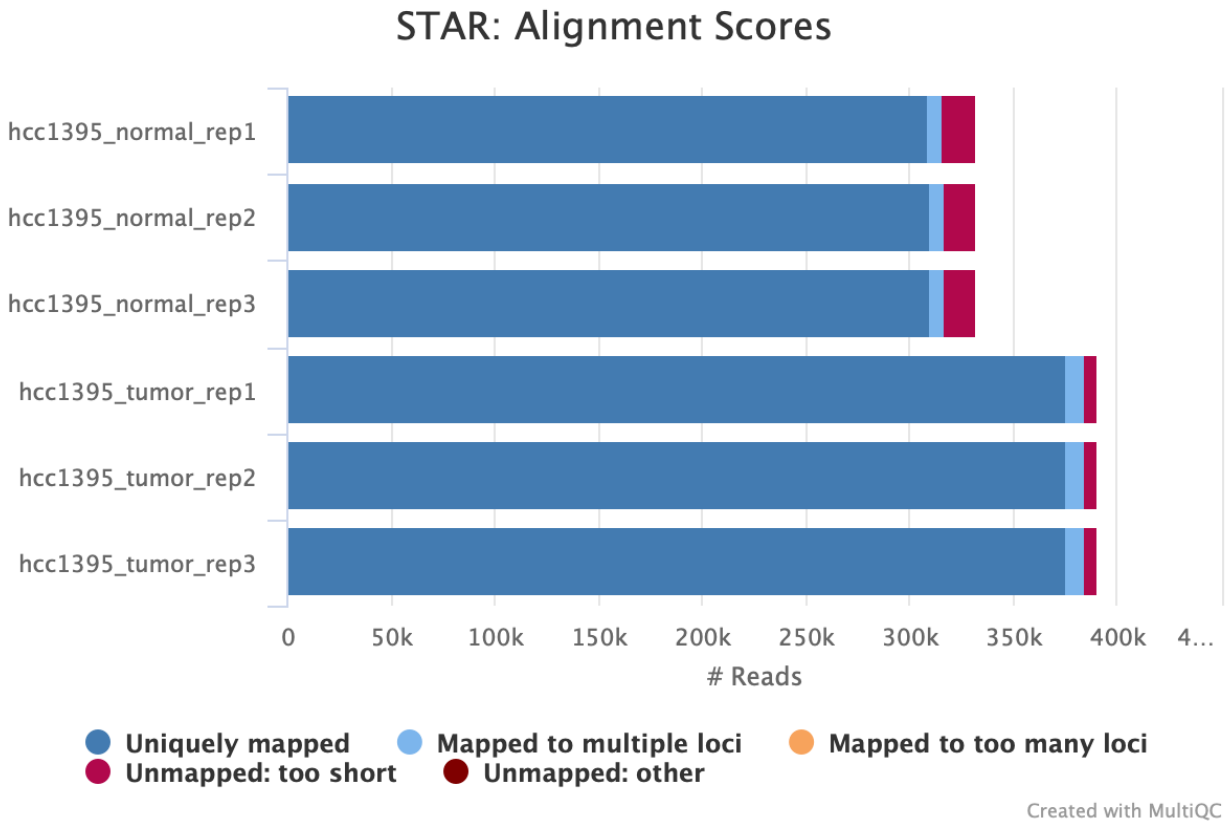
The fields in a SAM/BAM file are explained below.

1. QNAME or query template name, which is essentially the sequencing read that was mapped onto a reference genome.
2. The second column is a FLAG that provides detail on mapping. See https://broadinstitute.github.io/picard/explain-flags.html *(https://broadinstitute.github.io/picard/explain-flags.html)* to learn about these FLAGS. For instance, the first alignment in the hcc1395_normal_rep1 SAM file has a FLAG of 99, which indicates that:
    1. The read is paired, indicating paired end sequencing.
    2. Read is mapped in proper pair.
    3. The mate is in the reverse strand.
    4. The read is the first in a pair.
3. Column three contains the name of our reference genome.
4. Column four provides the left most genomic coordinate where the sequencing read maps.
5. The mapping quality (MAPQ) is provided in the fifth column (the higher the number, the less likely that the mapping is due to error); a value of 255 in this column means that the mapping quality is not available.
6. Column six presents the CIGAR string, which informs about the match/mismatch, insertion/deletion, etc. found in the alignment.
7. Column seven is the reference sequence name of the primary alignment of the NEXT read in the template. A "=" will appear if this name is the same as the current (which is expected for paired reads).
8. The alignment position of the next read in the template is provided in column eight. When this is not available, the value is set to 0.
9. Column nine provides the template length (TLEN), which should reflect the DNA fragment that was size selected for during library preparation.
10. The tenth column is just the sequencing read (some are written as the reverse complement so be cautious. The FLAGS in column two will tell whether the sequence is reverse complemented).
11. The eleventh column is the Phred quality scores of the sequencing read.
12. For definition of optional fields, see https://samtools.github.io/hts-specs/ *(https://samtools.github.io/hts-specs/)*.

Note that in the alignment folder, there are correspond indexed `bam` files with extension `bai` for each sample.

Post Alignment Quality Metrics

The alignment rate or percent of reads that map to the genome in bulk RNA sequencing is a good metric for determining whether the experiment is successful. The plot below shows that for this dataset, all samples achieved greater 90% unique mapping (ie. where sequencing reads map to one genomic location).



Researchers can interrogate the sample level statistics stored in the `Log.final.out` files in the `STAR_files` of the output directory.

Assuming participants are in the output directory `hcc1395_renee_b4b`, use the `cd` command to navigate to the `STAR_file` folder.

```
cd STAR_files
```

Use `cat` to look at the alignment metrics for the `hcc1395_normal_rep1` sample.

```
cat hcc1395_normal_rep1.Log.final.out
```

Detailed alignmnet statistcs are shown and these include:

- Unique mapping percentage.

- Number and type of splice junctions found.
- Deletion or insertion rates.
- Number of reads that map to multiple places in the genome (multimapping) and umapped reads.

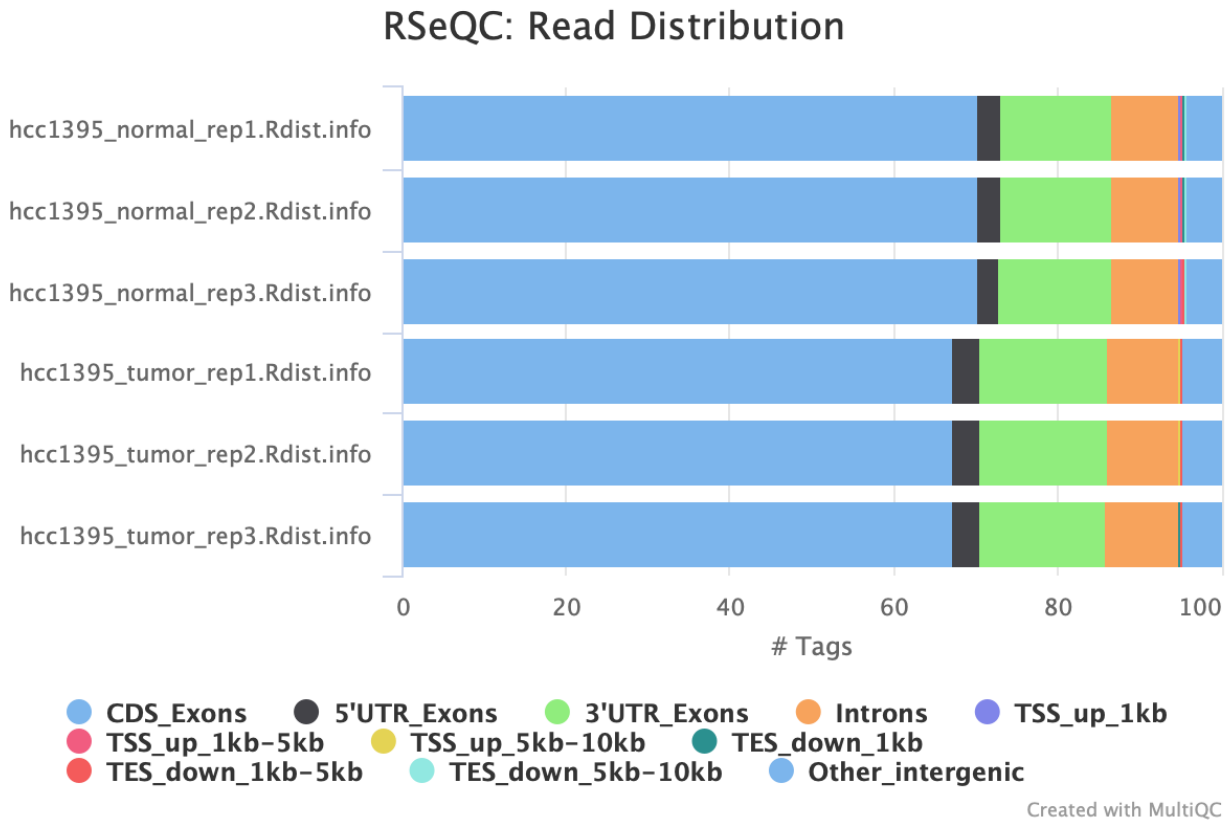
```
Started job on |    Feb 02 17:02:36
                            Started mapping on |   Feb 02 17:02:56
                                   Finished on |   Feb 02 17:03:51
       Mapping speed, Million of reads per hour |   21.70

                               Number of input reads |   331576
                         Average input read length |   294
                                       UNIQUE READS:
                      Uniquely mapped reads number |   311032
                           Uniquely mapped reads % |   93.80%
                             Average mapped length |   292.51
                          Number of splices: Total |   344806
               Number of splices: Annotated (sjdb) |   340737
                          Number of splices: GT/AG |   344131
                          Number of splices: GC/AG |   604
                          Number of splices: AT/AC |   63
                     Number of splices: Non-canonical |   8
                            Mismatch rate per base, % |   0.51%
                               Deletion rate per base |   0.00%
                             Deletion average length |   1.27
                             Insertion rate per base |   0.00%
                           Insertion average length |   1.47
                                 MULTI-MAPPING READS:
        Number of reads mapped to multiple loci |   3911
             % of reads mapped to multiple loci |   1.18%
        Number of reads mapped to too many loci |   0
             % of reads mapped to too many loci |   0.00%
                                   UNMAPPED READS:
  Number of reads unmapped: too many mismatches |   0
       % of reads unmapped: too many mismatches |   0.00%
            Number of reads unmapped: too short |   11318
                 % of reads unmapped: too short |   3.41%
                Number of reads unmapped: other |   5315
                     % of reads unmapped: other |   1.60%
                                   CHIMERIC READS:
                         Number of chimeric reads |   0
                              % of chimeric reads |   0.00%
```

Read Distribution

Another important QC metric after alignment is the type of genomic features are the reads mapping to. The `read_distribution` module in `RSeQC` helps determine this (https://rseqc.sourceforge.net/#read-distribution-py *(https://rseqc.sourceforge.net/#read-distribution-py)*). For this experiment, most reads map to coding sequencing (CDS), 5'-UTR, or 3'-UTR exons. Some reads do map to other genomics regions such as introns which could be caused by:

- Genomic DNA contamination
- Biology such as retained introns
- Nascent transcript
- Unspliced or partially spliced pre-mRNA
- Incomplete annotation

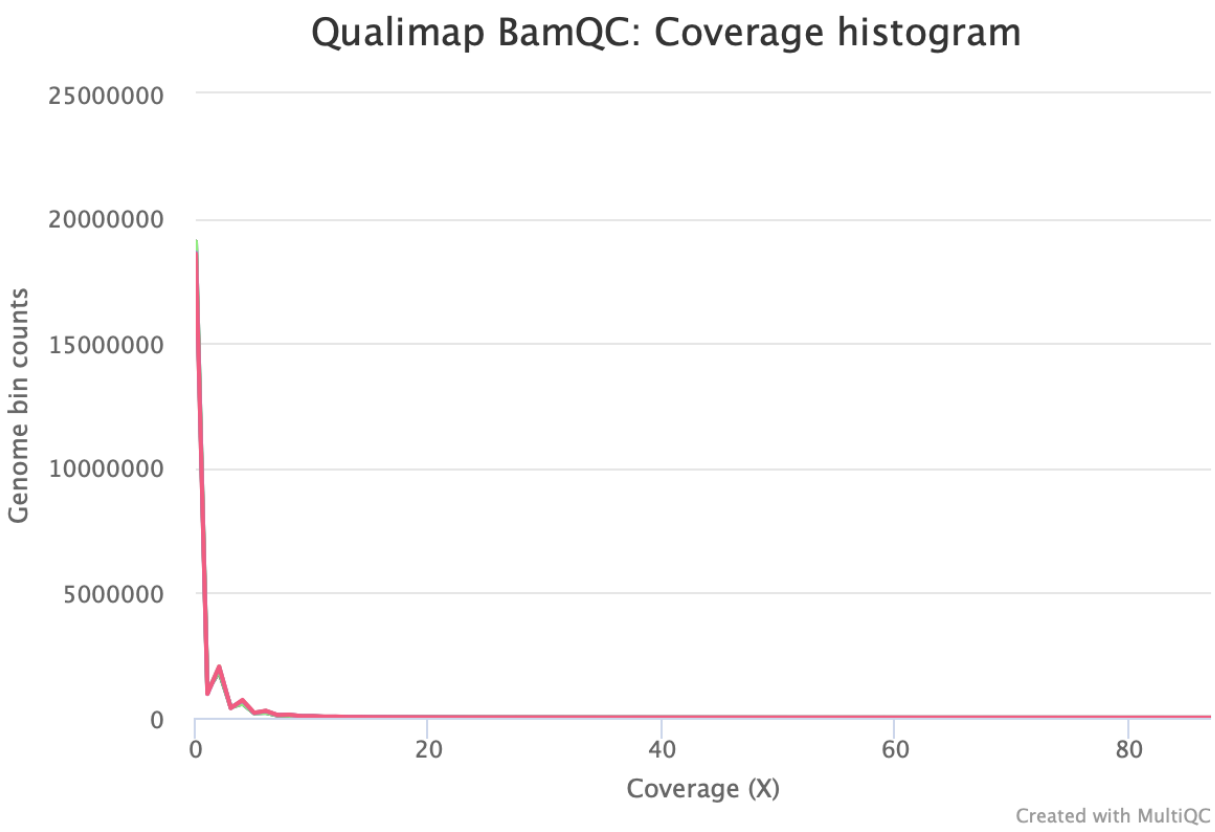


Read distribution results can be reviewed in the `Rdist.info` files of the `RSeQC` directory inside the RENEE output folder (in this class it is `hcc1395_renee_b4b`).

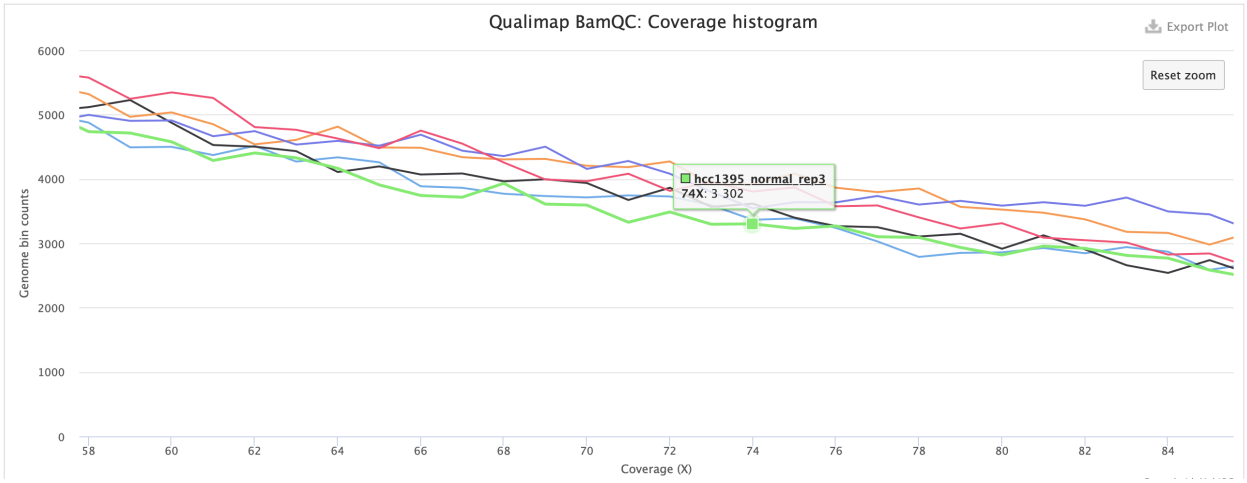
Coverage Distribution

Qualimap *(http://qualimap.conesalab.org)* can be used to look at the sequencing coverage. The plot below shows the number of genomic locations with a certain amount of reads mapped to them. As expected with RNA sequencing, the majority of genomic locations have no reads

map to them as the whole genome is not sequenced. Further, this dataset was subsetted to chromosome 22 and it was aligned to the entire human genome.

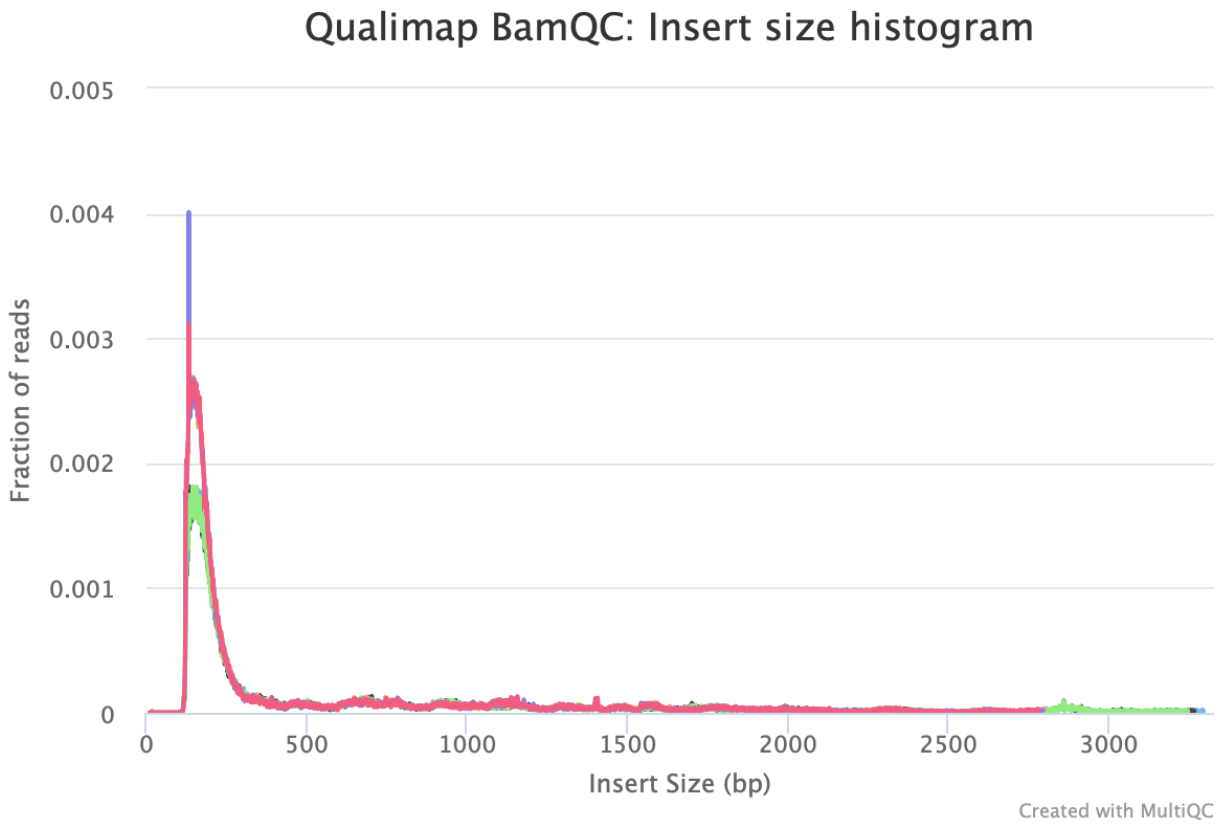


But zooming in on the plot, it is evident there are thousands of locations in the genome with over 60x coverage.



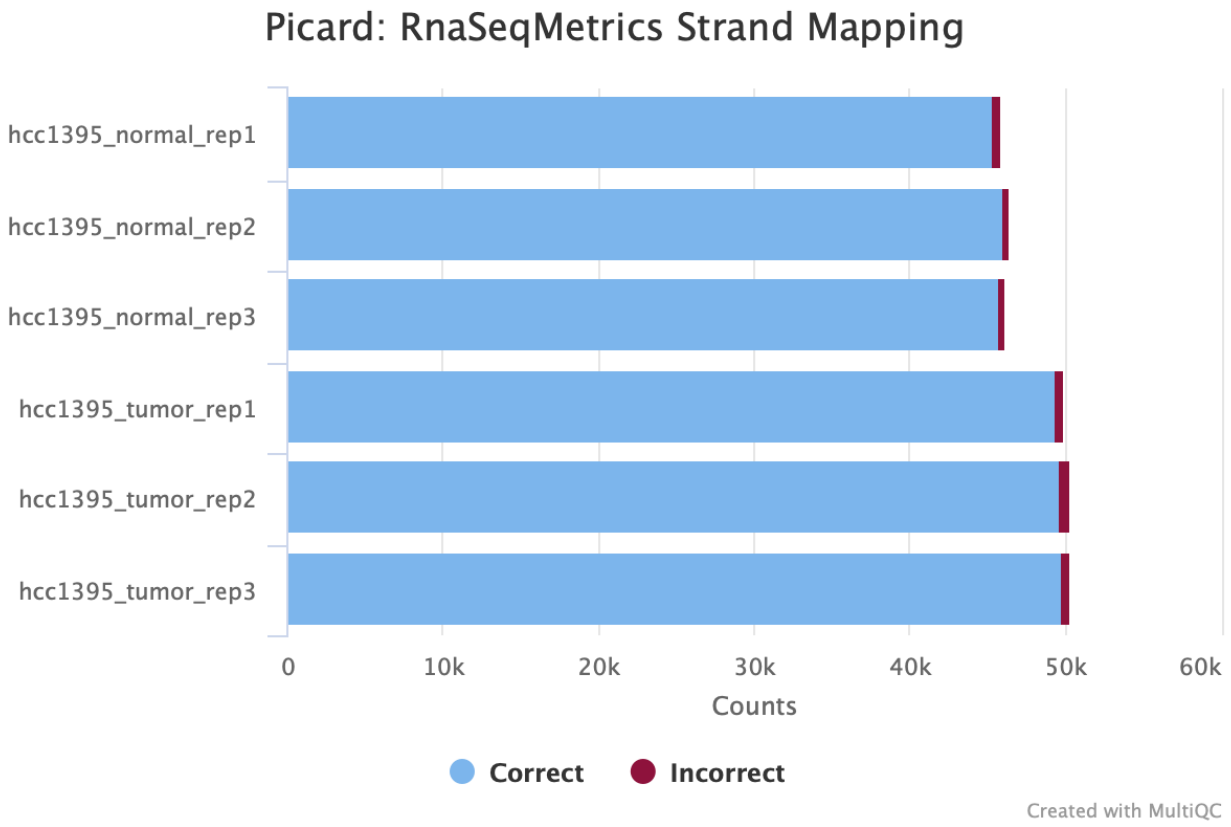
Insert Size

Qualimap *(http://qualimap.conesalab.org/doc_html/analysis.html)* calculates and plots the distribution of insert size from alignment results. Insert size distributions should be checked to ensure that the observed library fragment sizes are consistent with the size-selection criteria used during fragmentation and library preparation.

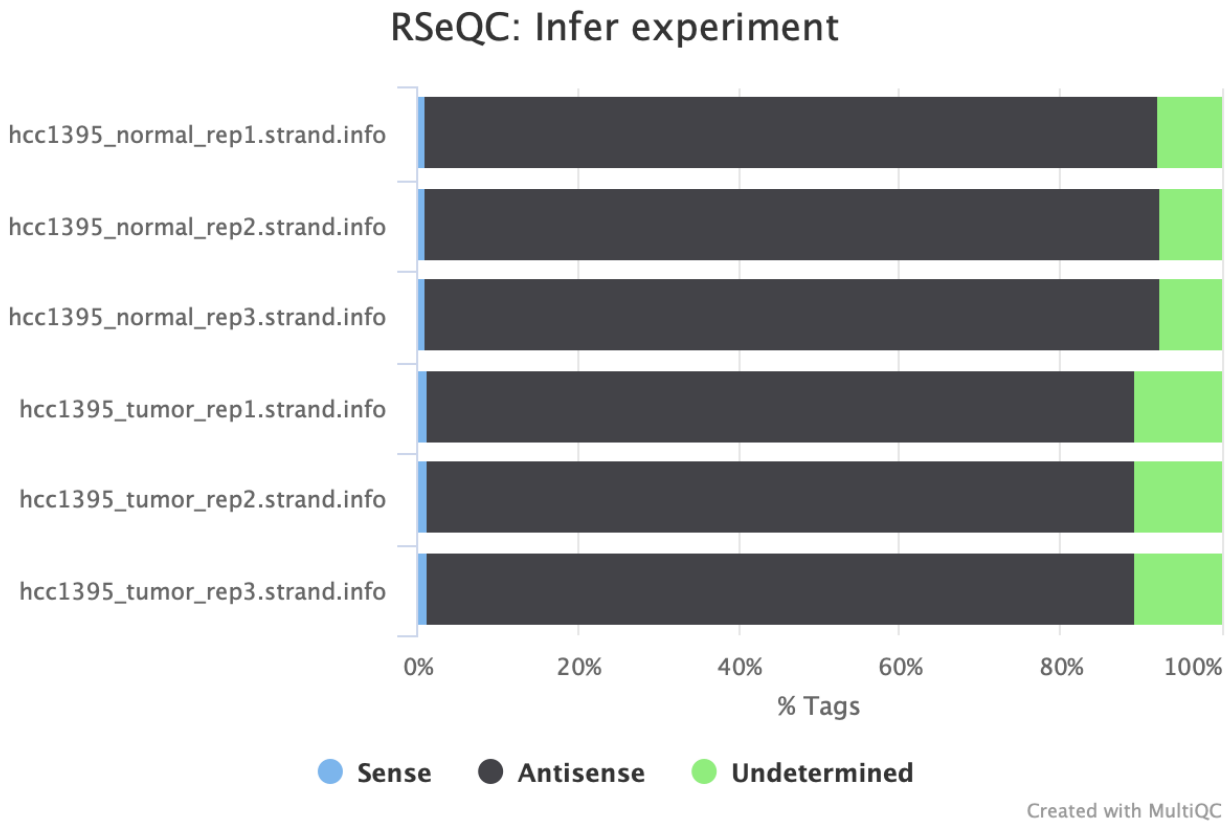


Strandedness of Experiment

Another important post alignment QC metric is whether the sequencing reads mapped to the correct strand. This information can be obtained using the tool Picard *(https://broadinstitute.github.io/picard/)*. The plot below shows that an overwhelming majority of reads in this experiment mapped to the correct strand of the DNA double helix. This is critical as mapping to the wrong strand will lead assigning of reads to the wrong gene or transcript during the expression quantification stage.



The tool rseqc *(https://rseqc.sourceforge.net)* can be used infer whether the bulk RNA sequencing protocol utilized a stranded protocol. It counts the number of reads or read pairs that matches the strand in which their overlapping transcript is on. The result below indicates that the experiment utilized a 1st strand protocol.



Visualizing Genomic Alingments

The Integrative Genomics Viewr *(https://igv.org)* is a popular tool for visualizing genomic alignment data. This enables scientists to visually inspect alignment for oddities, capture alignment results for regions of interest and/or get an idea of whether certain genes are expressed differently between conditions based on coverage. IGV can be run on local computer or launched via Biowulf HPC OnDemand.

An IGV snapshot for the HCC1395 bulk RNA data is shown below. The gray bar plots represent read coverage or how many reads map to a genomic location. The pink arches designate splice junctions where reads map across multiple exons, which is unique to RNA sequencing. The track with the colored boxes contain read alignments. Finally, the track on the bottom shows the reference genome.


Coverage
Splice junction
Alignment track
Refseq transcripts

Lesson 7: Quantifying Gene Expression

Lesson 6 Review

Lesson 6 introduced alignment of bulk RNA sequences in order to find where in the genome each read came from. Post alignment QC metrics were also covered. The next step, covered in this class will deal with quantifying gene expression based on the number of reads that map to each gene.

Learning objectives

After this lesson, participants will be able to:

- Describe the goal of gene expression quantification in bulk RNA sequencing and name tools that can be used for this task.
- Understand the content of GTF annotation file and why it is needed in quantification.
- Interpret post quantification QC metrics.
- Become familiar with the quantification table.

Goals for Gene Expression Quantification in Bulk RNA Sequencing

When the origin in the genome of each sequencing read in bulk RNA has been identified, the next step is quantifying gene expression levels. Here, the number of reads that map to a particular gene or transcript is used as an estimate for its expression level. This should be a simple counting exercise but the process is more complicated.

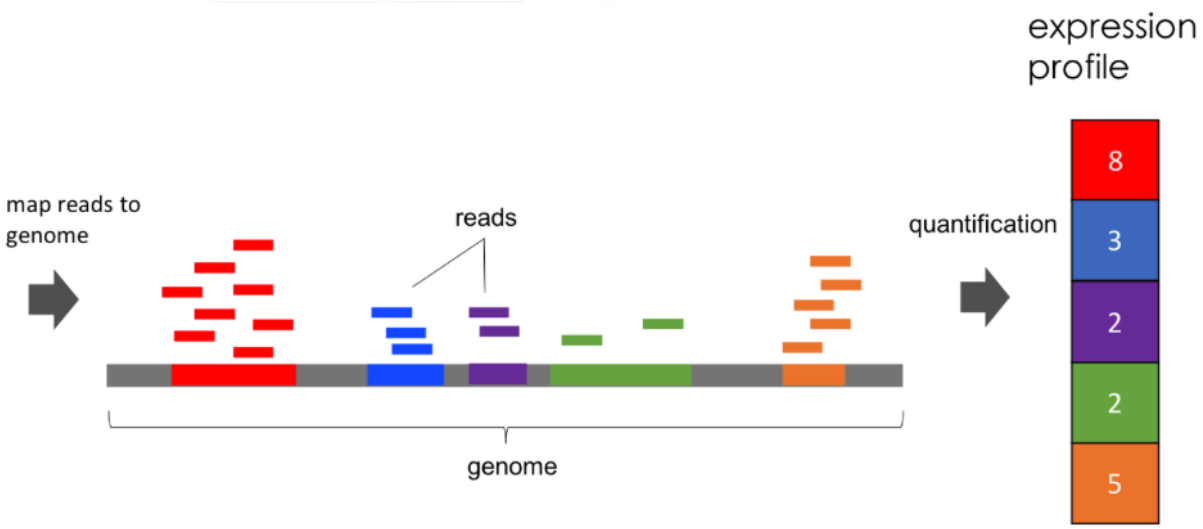


Image was modified from https://mbernste.github.io/posts/rna_seq_basics/ *(https://mbernste.github.io/posts/rna_seq_basics/)*

As it turns out, there are various scenarios in the gene expression quantification process. For instance:

- Reads that map to multiple genes.
- Reads that partially overlap a gene.

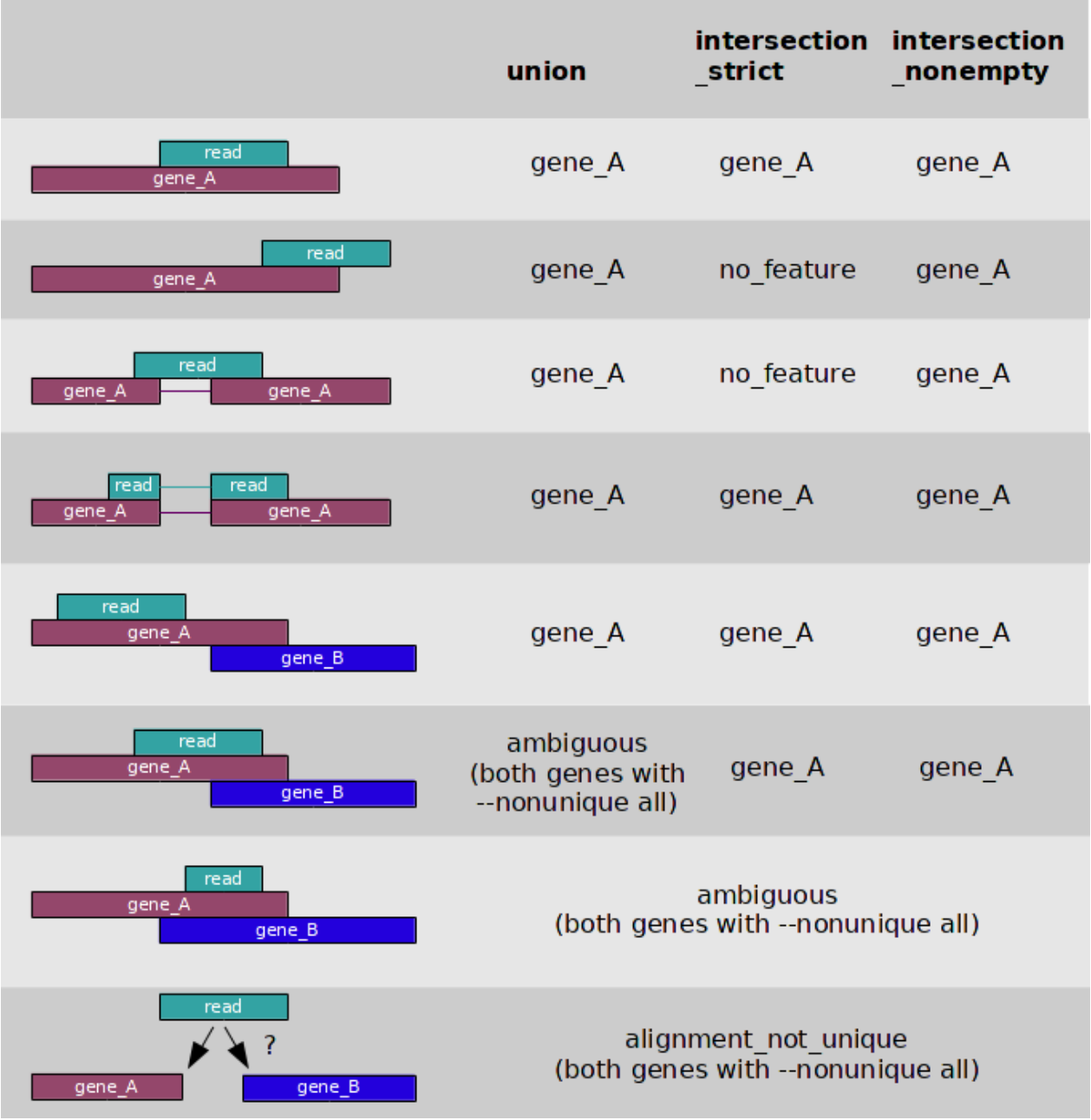


Image source: https://htseq.readthedocs.io/en/latest/htseqcount.html *(https://htseq.readthedocs.io/en/latest/htseqcount.html)*

Tools used for bulk RNA sequencing expression quantification include HTSeq *(https://htseq.readthedocs.io/en/latest/index.html)*, featureCounts *(https://subread.sourceforge.net/featureCounts.html)*, and RSEM *(https://github.com/deweylab/RSEM)*. Below is a comparison of

these three tools. RSEM is more sophisticated and can better quantify reads that map to multiple spots in the genome so it is used in RENEE.

HTSeq	featureCounts	RSEM
Overlap based	Overlap based	Expectation-Maximization
Assign fraction to multi-mappers	Assign fraction to multi-mappers	More accurately quantify expression for multi-mappers
No long reads	Long reads	Long reads

Pseudo aligners such as Salmon *(https://salmon.readthedocs.io/en/latest/salmon.html)* can map and quantify to transcriptome at the same time.

Annotation Files

During the alignment step, a reference genome or transcriptome is needed. This is to guide the aligner as it determines where is the best match for each read. When it comes to quantifying expression, a file that contains information on the location of genes, transcripts, exons, coding sequences, introns is needed. A common format is the General Transfer Format (GTF), which is tab separated table that holds genomic annotations. Below is a glimpse of the contents of a GTF file.

Chromosome	Annotation source	Feature	Start	End	Score	Strand	Frame	Attributes
chr22	ENSEMBL	gene	10736171	10736283	.	-	.	**"gene_id ""ENSG00000277248.1"" gene_type ""snRNA""; gene_status ""NOVEL""; gene_name ""U2""; level 3;"**
chr22	ENSEMBL	transcript	10736171	10736283	.	-	.	**"gene_id ""ENSG00000277248.1"" transcript_id ""ENST00000615943.1"" gene_type ""snRNA""; gene_status ""NOVEL""; gene_name ""U2""; transcript_type ""snRNA""; transcript_status ""NOVEL""; transcript_name ""U2.14-201""; level 3;

Chromosome	Annotation source	Feature	Start	End	Score	Strand	Frame	Attributes
								tag ""basic""; transcript_support_leve ""NA"";"**
chr22	ENSEMBL	exon	10736171	10736283	.	-	.	**"gene_id ""ENSG00000277248.1" transcript_id ""ENST00000615943.1"" gene_type ""snRNA""; gene_status ""NOVEL""; gene_name ""U2""; transcript_type ""snRNA""; transcript_status ""NOVEL""; transcript_name ""U2.14-201""; exon_number 1; exon_i ""ENSE00003736336.1"" level 3; tag ""basic""; transcript_support_leve ""NA"";"**
chr22	ENSEMBL	gene	10936023	10936161	.	-	.	**"gene_id ""ENSG00000274237.1" gene_type ""miRNA""; gene_status ""NOVEL""; gene_name ""CU459211.1""; level 3;"**
chr22	ENSEMBL	transcript	10936023	10936161	.	-	.	**"gene_id ""ENSG00000274237.1" transcript_id ""ENST00000618365.1"" gene_type ""miRNA""; gene_status ""NOVEL""; gene_name ""CU459211.1""; transcript_type ""miRNA""; transcript_status ""NOVEL";

Chromosome	Annotation source	Feature	Start	End	Score	Strand	Frame	Attributes
								transcript_name ""CU459211.1-201""; level 3; tag ""basic""; transcript_support_leve ""NA"";"**
chr22	ENSEMBL	exon	10936023	10936161	.	-	.	**"gene_id ""ENSG00000274237.1" transcript_id ""ENST00000618365.1"" gene_type ""miRNA""; gene_status ""NOVEL""; gene_name ""CU459211.1""; transcript_type ""miRNA""; transcript_status ""NOVEL""; transcript_name ""CU459211.1-201""; exon_number 1; exon_i ""ENSE00003712615.1"" level 3; tag ""basic""; transcript_support_leve ""NA"";"**
chr22	ENSEMBL	gene	11065974	11067346	.	-	.	**"gene_id ""ENSG00000280363.1" gene_type ""protein_coding""; gene_status ""KNOWN"" gene_name ""CU104787.1""; level 3;"**
chr22	ENSEMBL	transcript	11065974	11067346	.	-	.	**"gene_id ""ENSG00000280363.1" transcript_id ""ENST00000623473.1"" gene_type ""protein_coding""; gene_status ""KNOWN"" gene_name

Chromosome	Annotation source	Feature	Start	End	Score	Strand	Frame	Attributes
								""CU104787.1""; transcript_type ""protein_coding""; transcript_status ""KNOWN""; transcript_name ""CU104787.1-201""; level 3; protein_id ""ENSP00000485388.1"" tag ""basic""; transcript_support_leve ""5""; tag ""appris_principal_1"";"**
chr22	ENSEMBL	exon	11065974	11066015	.	-	.	**"gene_id ""ENSG00000280363.1" transcript_id ""ENST00000623473.1"" gene_type ""protein_coding""; gene_status ""KNOWN"" gene_name ""CU104787.1""; transcript_type ""protein_coding""; transcript_status ""KNOWN""; transcript_name ""CU104787.1-201""; exon_number 2; exon_i ""ENSE00003755067.1"" level 3; protein_id ""ENSP00000485388.1"" tag ""basic""; transcript_support_leve ""5""; tag ""appris_principal_1"";"**
chr22	ENSEMBL	CDS	11065974	11066015	.	-	0	**"gene_id ""ENSG00000280363.1" transcript_id ""ENST00000623473.1"" gene_type

Chromosome	Annotation source	Feature	Start	End	Score	Strand	Frame	Attributes
								""protein_coding""; gene_status ""KNOWN"" gene_name ""CU104787.1""; transcript_type ""protein_coding""; transcript_status ""KNOWN""; transcript_name ""CU104787.1-201""; exon_number 2; exon_i ""ENSE00003755067.1"" level 3; protein_id ""ENSP00000485388.1"" tag ""basic""; transcript_support_leve ""5""; tag ""appris_principal_1"";"**
chr22	ENSEMBL	gene	11066501	11068089	.	+	.	**"gene_id ""ENSG00000279973.1" gene_type ""protein_coding""; gene_status ""KNOWN"" gene_name ""BAGE5""; level 3;"**
chr22	ENSEMBL	transcript	11066501	11068089	.	+	.	**"gene_id ""ENSG00000279973.1" transcript_id ""ENST00000624155.1"" gene_type ""protein_coding""; gene_status ""KNOWN"" gene_name ""BAGE5""; transcript_type ""protein_coding""; transcript_status ""KNOWN""; transcript_name ""BAGE5-201""; level 3; protein_id ""ENSP00000485185.1"

Chromosome	Annotation source	Feature	Start	End	Score	Strand	Frame	Attributes
								tag ""basic""; transcript_support_leve ""1""; tag ""appris_principal_1"";"**
chr22	ENSEMBL	exon	11066501	11066515	.	+	.	**""gene_id ""ENSG00000279973.1" transcript_id ""ENST00000624155.1"" gene_type ""protein_coding""; gene_status ""KNOWN"" gene_name ""BAGE5""; transcript_type ""protein_coding""; transcript_status ""KNOWN""; transcript_name ""BAGE5-201""; exon_number 1; exon_i ""ENSE00003758096.1"" level 3; protein_id ""ENSP00000485185.1"" tag ""basic""; transcript_support_leve ""1""; tag ""appris_principal_1"";"**

Quantification Output

RENEE saves the expression results in the `DEG_ALL` folder of the output directory. Change into it and list the contents one item per line.

```
cd DEG_ALL
```

```
ls -1
```

RSEM provides quantification estimates on the gene and isoform level. Per sample basis (`sample.RSEM.genes.results` and `sample.RSEM.isoforms.results`) as well as combined quantifications for all samples (see

`RSEM.genes.expected_count.all_samples.txt` and `RSEM.isoforms.expected_count.all_samples.txt`) are provided. The files labeled `FPKM` and `TPM` are normalized expression estimates, however differential expression tools such as DESeq2 and edgeR do not use these as they take unormalized expression data as input. For the purposes of the class, look at `RSEM.genes.expected_count.all_samples.txt`.

```
-rw-r----- 1 $USER $USER  49351 Jul  3 16:38 combined_TIN.tsv
-rw-r----- 1 $USER $USER 274485 Jul  3 16:38 hcc1395_normal_rep1.RSEM
-rw-r----- 1 $USER $USER 809683 Jul  3 16:38 hcc1395_normal_rep1.RSEM
drwxr-x--- 2 $USER $USER   4096 Jul  3 16:38 hcc1395_normal_rep1.RSEM
-rw-r----- 1 $USER $USER     98 Jul  3 16:38 hcc1395_normal_rep1.RSEM
-rw-r----- 1 $USER $USER 274483 Jul  3 16:38 hcc1395_normal_rep2.RSEM
-rw-r----- 1 $USER $USER 809538 Jul  3 16:38 hcc1395_normal_rep2.RSEM
drwxr-x--- 2 $USER $USER   4096 Jul  3 16:38 hcc1395_normal_rep2.RSEM
-rw-r----- 1 $USER $USER     98 Jul  3 16:38 hcc1395_normal_rep2.RSEM
-rw-r----- 1 $USER $USER 274467 Jul  3 16:38 hcc1395_normal_rep3.RSEM
-rw-r----- 1 $USER $USER 809708 Jul  3 16:38 hcc1395_normal_rep3.RSEM
drwxr-x--- 2 $USER $USER   4096 Jul  3 16:38 hcc1395_normal_rep3.RSEM
-rw-r----- 1 $USER $USER     98 Jul  3 16:38 hcc1395_normal_rep3.RSEM
-rw-r----- 1 $USER $USER 274554 Jul  3 16:38 hcc1395_tumor_rep1.RSEM
-rw-r----- 1 $USER $USER 809469 Jul  3 16:38 hcc1395_tumor_rep1.RSEM
drwxr-x--- 2 $USER $USER   4096 Jul  3 16:38 hcc1395_tumor_rep1.RSEM
-rw-r----- 1 $USER $USER     98 Jul  3 16:38 hcc1395_tumor_rep1.RSEM
-rw-r----- 1 $USER $USER 274594 Jul  3 16:38 hcc1395_tumor_rep2.RSEM
-rw-r----- 1 $USER $USER 809729 Jul  3 16:38 hcc1395_tumor_rep2.RSEM
drwxr-x--- 2 $USER $USER   4096 Jul  3 16:38 hcc1395_tumor_rep2.RSEM
-rw-r----- 1 $USER $USER     98 Jul  3 16:38 hcc1395_tumor_rep2.RSEM
-rw-r----- 1 $USER $USER 274547 Jul  3 16:38 hcc1395_tumor_rep3.RSEM
-rw-r----- 1 $USER $USER 809614 Jul  3 16:38 hcc1395_tumor_rep3.RSEM
drwxr-x--- 2 $USER $USER   4096 Jul  3 16:38 hcc1395_tumor_rep3.RSEM
-rw-r----- 1 $USER $USER     98 Jul  3 16:38 hcc1395_tumor_rep3.RSEM
-rw-r----- 1 $USER $USER  81819 Jul  3 16:38 RSEM.genes.expected_cou
-rw-r----- 1 $USER $USER  77987 Jul  3 16:38 RSEM.genes.expected_cou
-rw-r----- 1 $USER $USER  81809 Jul  3 16:38 RSEM.genes.expected_cou
-rw-r----- 1 $USER $USER  85394 Jul  3 16:38 RSEM.genes.FPKM.all_sam
-rw-r----- 1 $USER $USER  86000 Jul  3 16:38 RSEM.genes.TPM.all_samp
-rw-r----- 1 $USER $USER 386203 Jul  3 16:38 RSEM.isoforms.expected_
-rw-r----- 1 $USER $USER 277331 Jul  3 16:38 RSEM.isoforms.expected_
-rw-r----- 1 $USER $USER 392811 Jul  3 16:38 RSEM.isoforms.FPKM.all_
-rw-r----- 1 $USER $USER 394765 Jul  3 16:38 RSEM.isoforms.TPM.all_sa
```

Use the `column` command construct below to get a glimpse of the gene level expression estimates for all samples saved in `RSEM.genes.expected_count.all_samples.txt`.

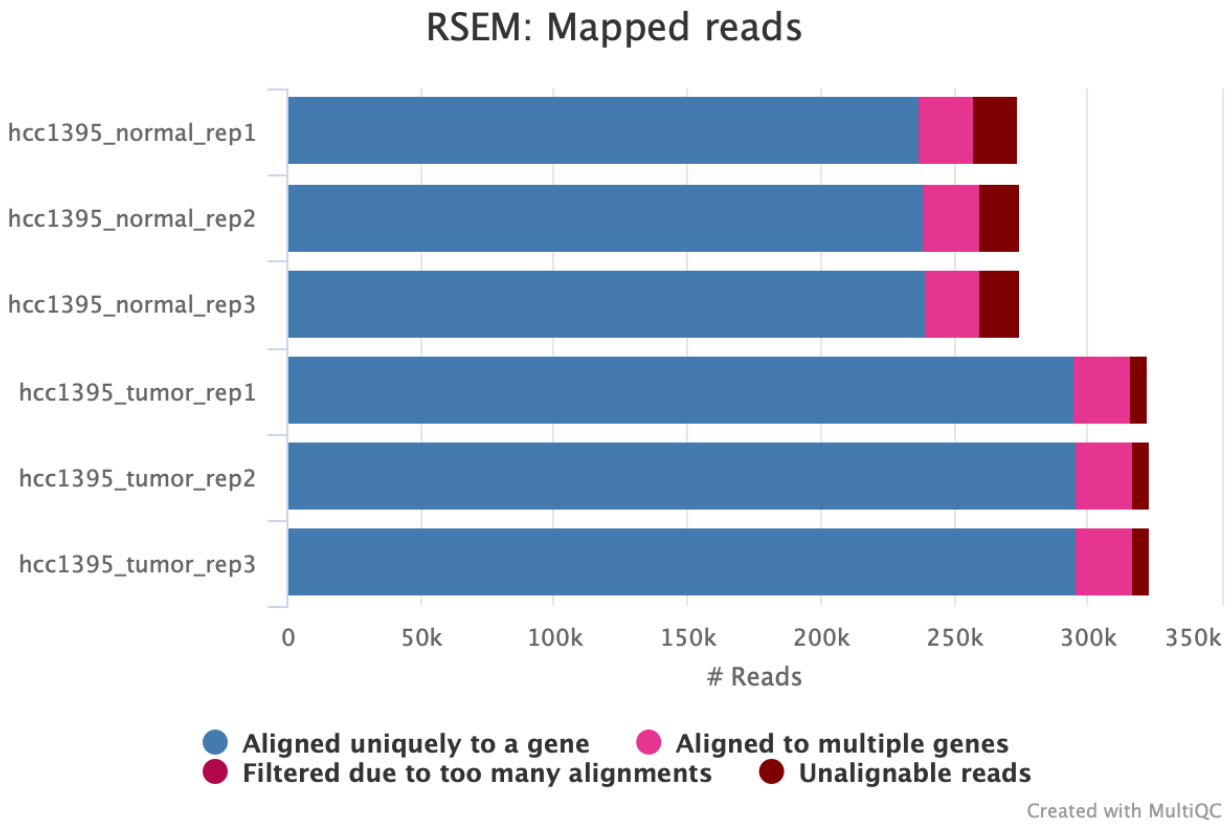
- `column` is a command that prints tabular data to the terminal with columns nicely aligned.
  - `-t` is an option to tell `column` to create a table.
  - The argument is the file that the user wants to display (ie. `RSEM.genes.expected_count.all_samples.txt`).
- The output of `column` is sent to `less` using pipe (`|`). The `-S` option in `less` enables horizontal scrolling on the terminal.

```
column -t RSEM.genes.expected_count.all_samples.txt | less -S
```

```
gene_id             GeneName           hcc1395_normal_rep1  hcc1395_r
ENSG00000276871.1   5_8S_rRNA          0.0                  0.0
ENSG00000128274.17  A4GALT             187.0                212.0
ENSG00000225293.1   ABCD1P4            0.0                  0.0
ENSG00000229107.2   ABHD17AP4          5.0                  14.0
ENSG00000263366.2   ABHD17AP5          0.0                  0.0
ENSG00000100412.17  ACO2               3538.0               3517.89
```

Quantification Metrics

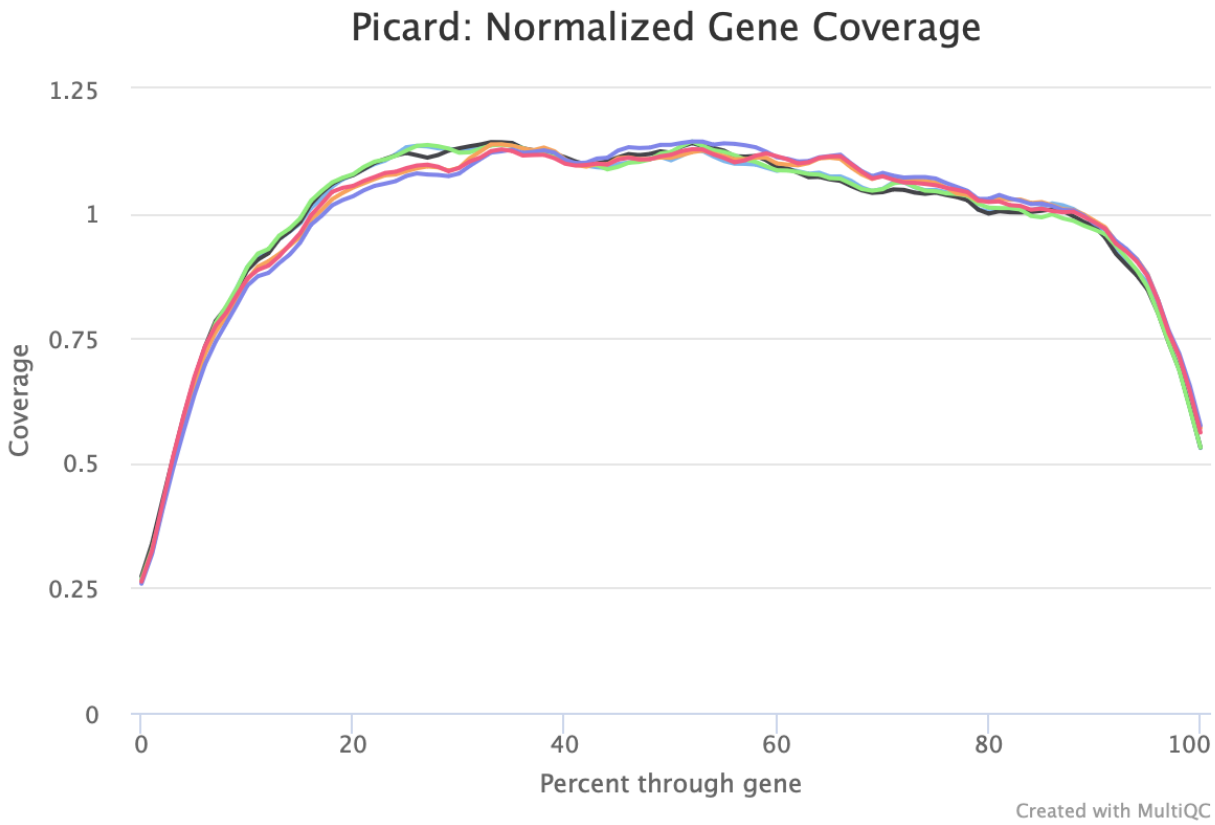
An important metric for estimation of gene expression is how many reads was the quantifier able to assign to a gene or transcript. Below is a plot that shows this for RSEM. For normal samples, approximately 87% of aligned reads were assigned to a unique gene, while tumor samples had approximately a 91% assignment rate.



Gene Coverage

Picard's normalized gene-body coverage plot shows how sequencing coverage is distributed from the annotated 5′ end to the 3′ end of transcripts. It is mainly used to detect technical coverage bias, such as RNA degradation, incomplete reverse transcription, or library-preparation bias.

- The samples have very similar profiles, which suggests good consistency.
- Coverage is relatively flat across most of the gene body.
- The lower coverage at both ends is common and can result from transcript-end annotation, read-length constraints, fragmentation, and alignment effects. There is no strong 3′ or 5′ bias visible.



Transcript Integrity

RSeQC calculates a transcript integrity number (TIN) that informs of the intactness of RNA at the transcript level and acts as metric for determining whether there is uniform read coverage across the transcript. This can be reviewed in the general statistics table. Just select "Configure Columns" and deselect everything except "RSeQC TIN stdev" and "RSeQC TIN". The fact that all TIN in this study are greater than 70% means that there is no degradation and uniform read coverage across the transcript. The high standard deviation could be due to low expression genes no have uniform coverage or certain transcripts just produce non-uniform coverage.

General Statistics

Copy table | Configure Columns | Plot | Showing 6/36 rows and 2/34 columns.

Sample Name	TIN stdev	TIN
hcc1395_normal_rep1	24.5	71.6
hcc1395_normal_rep2	24.7	71.9
hcc1395_normal_rep3	24.4	71.6
hcc1395_tumor_rep1	25.6	72.5
hcc1395_tumor_rep2	25.7	72.3
hcc1395_tumor_rep3	24.4	73.0

RNA Report

The file `RNA_Report.html` in the `Reports` directory of the output folder provides additional QC metrics for the RNA sequencing study. This report provides a metadata table provides

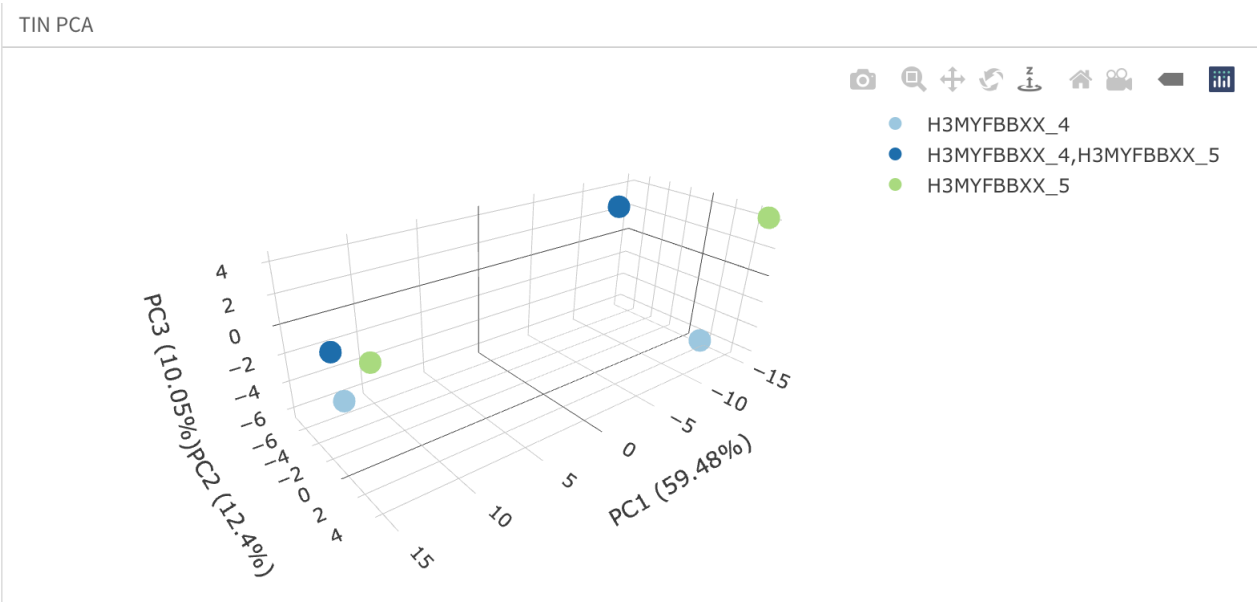
information including total number of read pairs in a sample, GC percentage, average mapping quality, coverage, median TIN, and flow cell in which the sample was sequenced on.

Metadata

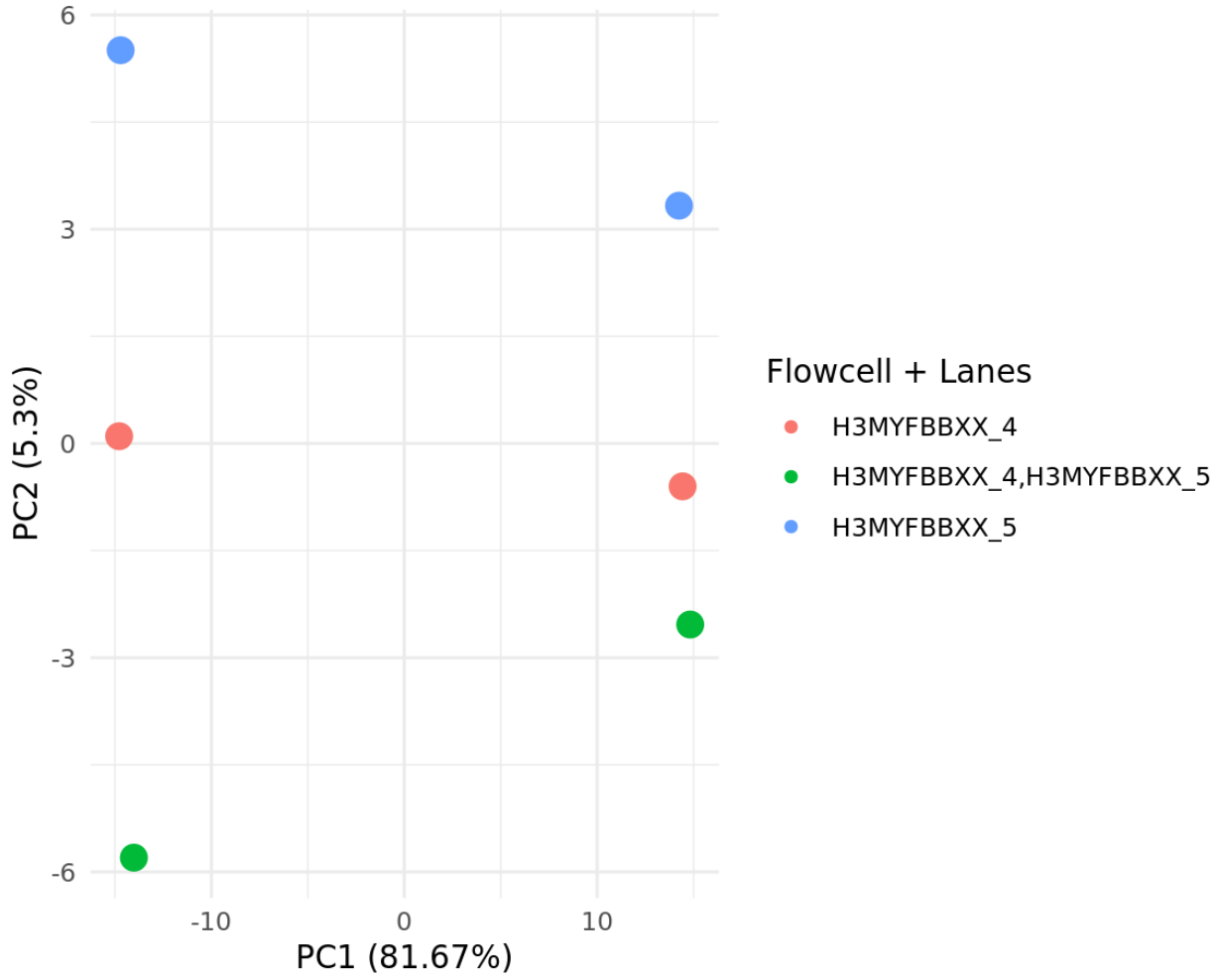
Column visibility CSV Excel

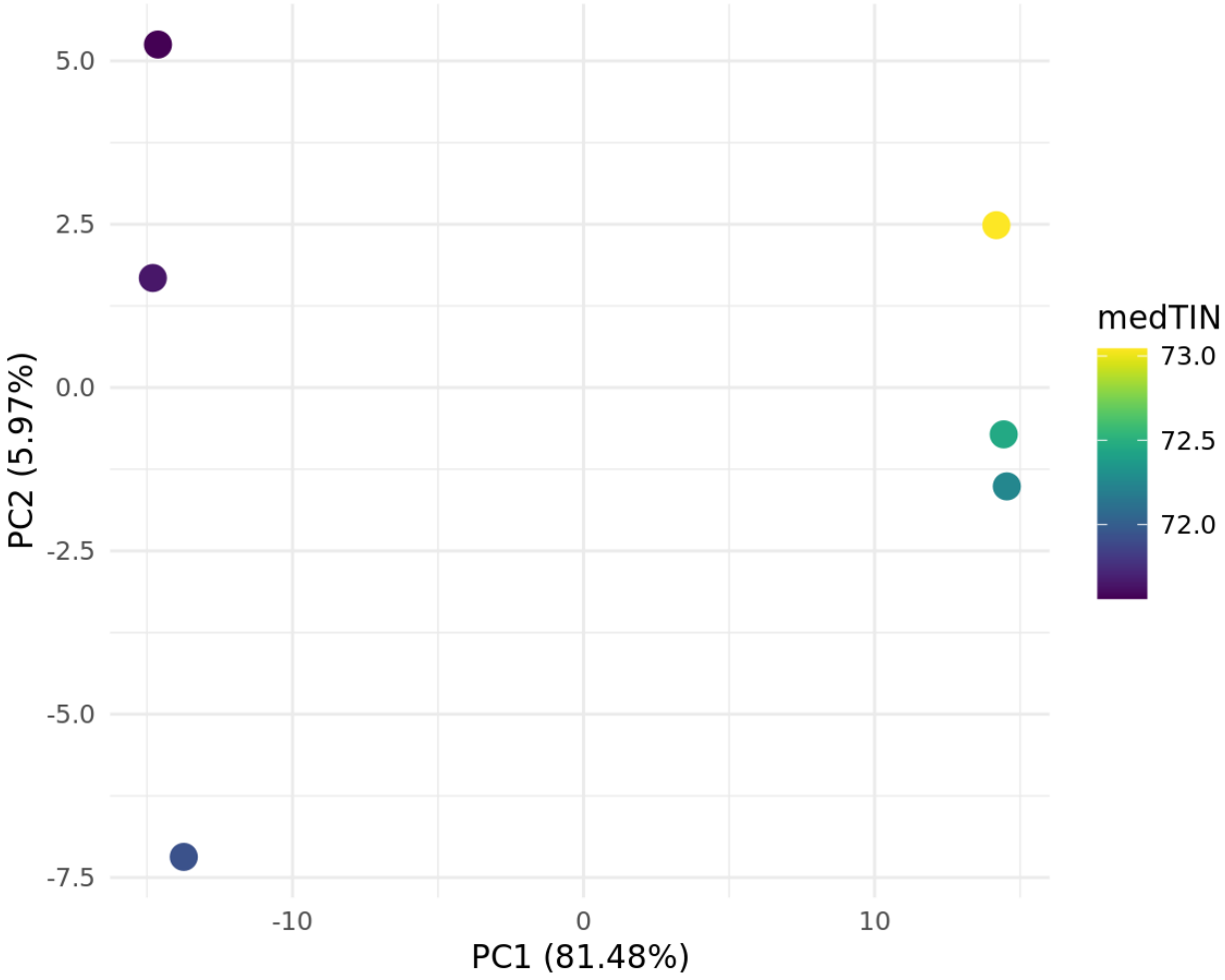
Sample ID	Total Reads	Trimmed Reads	Avg Seq Length	Seq Range	GC	% Dup	% Aligned	Inner Distance Maxima	Insert Size	Avg MapQ	Coverage	% Coding	% Intronic	CV Coverage	% rRNA	% UniVec	% Anti-sense	medTIN	Flowcell Lanes
hcc1395_normal_rep1	331958	331576	147.262	35-151	54	13.7	93.8	-140	284	248.7	48.716	59.405	11.727	0.511	0.39	0.04	91.1	71.648	H3MYFBBXX_4
hcc1395_normal_rep2	331958	331561	147.181	35-151	54	14	94.19	-145	281	248.692	48.873	59.229	11.694	0.512	0.4	0.04	91.2	71.933	H3MYFBBXX_4,H3MY
hcc1395_normal_rep3	331956	331574	147.15	35-151	54	14.1	94.29	-145	277	248.738	48.207	59.218	11.786	0.516	0.41	0.04	91.2	71.56	H3MYFBBXX_5
hcc1395_tumor_rep1	390607	390162	146.45	35-151	53	18.4	97.44	-145	223	249.626	51.515	56.008	11.746	0.518	0.27	0.02	88	72.453	H3MYFBBXX_4

Below this table are two Principle Components (PCA) plots. One of which shows whether flow cell lane (legend) influences transcript integrity or TIN. In this case, TIN is not influenced by flow cell lane as we can distinguished clusters of samples from each flow cell clustering together and segregated along PC1 from the other trio. Mousing over this interactive plot, users will learn that in terms of TIN, tumor samples are clustered together and have similar TIN as do the normal samples. If flow cell affected TIN, then it would be expected that dots of the same color will cluster together more closely.



Moving to the subplots tab users will see 2 dimensional PCA plots indicating how differing QC metrics influence expression. As PC1 accounts for greater than 80% of the variance in these PCA plots, it can be concluded that it is expression that is driving apart the samples in the two clusters while QC metrics such as median TIN (medTIN) and flow cell lane play a minor role in differentiating between samples.


6
3
0
-3
-6
PC2 (5.3%)
-10
0
10
PC1 (81.67%)
Flowcell + Lanes
H3MYFBBXX_4
H3MYFBBXX_4,H3MYFBBXX_5
H3MYFBBXX_5


5.0
2.5
0.0
-2.5
-5.0
-7.5
PC2 (5.97%)
-10
0
10
PC1 (81.48%)
medTIN
73.0
72.5
72.0

Module 3 - Differential Expression and Pathway Analysis

Lesson 1: What Is Differential Expression Analysis?

Learning Objectives

By the end of this lesson, learners should be able to:

- identify the RNA-Seq count matrix and understand the data within;
- explain what differential expression analysis tests in an RNA-Seq experiment;
- describe why raw read counts and absolute abundances are not directly comparable across samples;
- distinguish common normalization strategies and when they are appropriate;
- identify common warning signs from QC that should be addressed before DEG testing;
- name the most widely used DE tools and know where to find them.
- know where to look when a specialized experimental design calls for a method beyond DESeq2, edgeR, or limma;
- know where to go next, iDEP for a GUI-driven analysis, or R/Python for full flexibility.

---

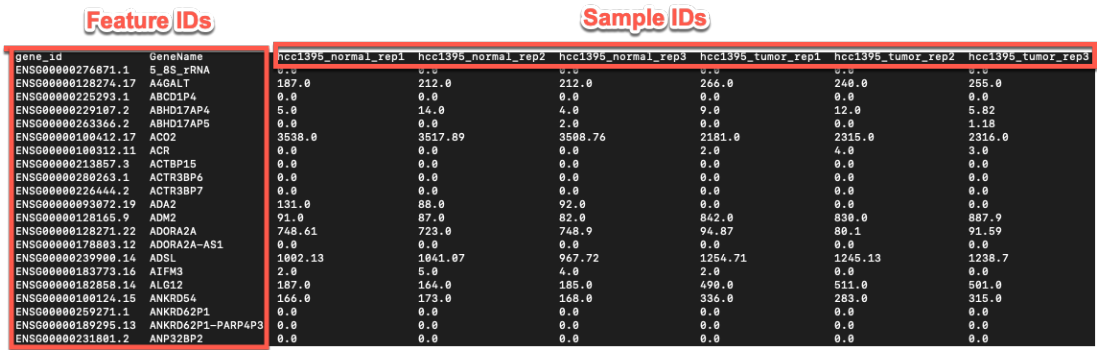
1. What Can We Do with a Count Matrix?

We are at the point in our workflow where we have a count matrix and sample metadata. The count matrix is the hub of our analysis. From it, we can do quality control (QC), filtering, differential expression (DE) testing, and pathway interpretation.

> **Where is my count matrix?**
>
> The RENEE workflow outputs multiple count matrices to `DEG_ALL`. This directory contains multiple count matrices by sample, across all samples, and by gene or isoform. See the RENEE documentation *(https://ccbr.github.io/RENEE/latest/RNA-seq/output/#3-deg_all)* for details.

For this lesson, we will focus on the gene level counts across all samples. These are found in `RSEM.genes.expected_count.all_samples.txt`. If you do not have this file, you can copy or download it from `/data/classes/BTEP/hcc1395_renee_out/DEG_ALL`. We will discuss how to access this file for iDEP at the end of this lesson.

The count matrix is usually a tab-delimited text file with rows as genes and columns as samples. The first column is usually gene identifiers, and the first row is usually sample identifiers.



gene_id	GeneName	hcc1395_normal_rep1	hcc1395_normal_rep2	hcc1395_normal_rep3	hcc1395_tumor_rep1	hcc1395_tumor_rep2	hcc1395_tumor_rep3
ENSG00000276871.1	5_8S_rRNA	0.0	0.0	0.0	0.0	0.0	0.0
ENSG00000128274.17	A4GALT	187.0	212.0	212.0	266.0	240.0	255.0
ENSG00000225293.1	ABCD1P4	0.0	0.0	0.0	0.0	0.0	0.0
ENSG00000229107.2	ABHD17AP4	5.0	14.0	4.0	9.0	12.0	5.82
ENSG00000263366.2	ABHD17AP5	0.0	0.0	2.0	0.0	0.0	1.18
ENSG00000100412.17	ACO2	3538.0	3517.89	3508.76	2181.0	2315.0	2316.0
ENSG00000100312.11	ACR	0.0	0.0	0.0	2.0	4.0	3.0
ENSG00000213857.3	ACTBP15	0.0	0.0	0.0	0.0	0.0	0.0
ENSG00000280263.1	ACTR3BP6	0.0	0.0	0.0	0.0	0.0	0.0
ENSG00000226444.2	ACTR3BP7	0.0	0.0	0.0	0.0	0.0	0.0
ENSG00000093072.19	ADA2	131.0	88.0	92.0	0.0	0.0	0.0
ENSG00000128165.9	ADM2	91.0	87.0	82.0	842.0	830.0	887.9
ENSG00000128271.22	ADORA2A	748.61	723.0	748.9	94.87	80.1	91.59
ENSG00000178803.12	ADORA2A-AS1	0.0	0.0	0.0	0.0	0.0	0.0
ENSG00000239900.14	ADSL	1002.13	1041.07	967.72	1254.71	1245.13	1238.7
ENSG00000183773.16	AIFM3	2.0	5.0	4.0	2.0	0.0	0.0
ENSG00000182858.14	ALG12	187.0	164.0	185.0	490.0	511.0	501.0
ENSG00000100124.15	ANKRD54	166.0	173.0	168.0	336.0	283.0	315.0
ENSG00000259271.1	ANKRD62P1	0.0	0.0	0.0	0.0	0.0	0.0
ENSG00000189295.13	ANKRD62P1-PARP4P3	0.0	0.0	0.0	0.0	0.0	0.0
ENSG00000231801.2	ANP32BP2	0.0	0.0	0.0	0.0	0.0	0.0

Figure 1: Gene Level Count Matrix from RENEE (ALL Samples)

2. The Core Question

Differential expression (DE) analysis asks whether the expected expression of a gene differs between experimental conditions after accounting for technical and biological variation.

For each gene, we want:

- an effect estimate, commonly a log2 fold change;
- uncertainty around that effect;
- a p-value from a statistical test;
- an adjusted p-value or false discovery rate (FDR) across thousands of genes.

This is not the same question as "Which genes have the largest counts?" A gene can have high expression in all samples and no meaningful condition effect. Another gene can have moderate expression but a consistent and statistically supported difference between groups.

Gene-level DE, isoform-level DE, and DTU are different questions

These three analyses sound similar but answer different biological questions:

- **Gene-level DE:** Is total expression of this gene different between conditions?
- **Isoform-level DE (DTE):** Is this specific isoform's expression different between conditions?
- **Differential transcript usage (DTU):** Did the relative proportions of isoforms within a gene shift between conditions regardless of whether total gene expression changed? DTU is inherently compositional, so it's better handled by purpose-built tools (DRIMSeq, DEXSeq, satuRn, IsoformSwitchAnalyzeR) rather than by treating isoforms as independent features in DESeq2/edgeR.

Event-level differential splicing: Did the inclusion of a specific splicing event (e.g., a skipped exon, alternative splice site, retained intron) change between conditions? Tools like **rMATS** *(https://bioinformatics.ccr.cancer.gov/docs/btep-coding-club/CC2023/rmats/)* (and similarly LeafCutter, MAJIQ) test this directly from exon-exon junction reads, without needing to reconstruct or quantify whole isoforms.

Isoform-level tests generally have less statistical power per feature than gene-level tests, as there are more features to test, counts per feature are often lower (reads are split across isoforms of the same gene), and EM-based (Expectation-Maximization) assignment of ambiguous reads adds estimation uncertainty. Together, these push toward wider confidence intervals and weaker power unless the true effect is large.

How We Measure Differential Expression

In practice, we summarize each gene with two key outputs:

1. **Effect size** (usually log2 fold change): how much expression changed between groups.
2. **Statistical confidence** (p-value and adjusted p-value/FDR): how likely that change is to be real rather than noise.

Adjusted p-value (FDR)

- An adjusted p-value corrects for testing many genes (running multiple tests), thereby reducing false positives from chance alone. The purpose is to control the false positive rate or false discovery rate that will naturally be inflated by chance when testing thousands of genes.
- At an FDR cutoff of 0.05, among the genes you call significant, the expected proportion of false positives is about 5% (on average, across repeated studies)

Learn more at:
- R p.adjust docs: https://stat.ethz.ch/R-manual/R-devel/library/stats/html/p.adjust.html *(https://stat.ethz.ch/R-manual/R-devel/library/stats/html/p.adjust.html)*
- Bioconductor discussion: https://support.bioconductor.org/p/49864/ *(https://support.bioconductor.org/p/49864/)*

What Is Log2 Fold Change?

Fold change compares expression between two groups:

Fold change = (expression in group A) / (expression in group B)

Most RNA-seq tools report this on a log base 2 scale:

$\log_2$FC = $\log_2$((group A) / (group B))

Why Use log2FC Instead of Raw Fold Change?

- The scale is symmetric around zero (up and down are easier to compare).
- Large ratios are compressed to a more interpretable range.
- It pairs naturally with MA plots and volcano plots.

Quick Interpretation Guide

- **log2FC = 0**: no change.
- **log2FC = +1**: about 2x higher in group A.
- **log2FC = -1**: about 2x lower in group A (or 2x higher in group B).
- **log2FC = +2**: about 4x higher in group A.
- **log2FC = -2**: about 4x lower in group A.

Interpret effect size and FDR together

- Large log2FC with weak FDR can be unstable.

- Tiny log2FC with very strong FDR may be statistically real but biologically small.

3. Why Absolute Abundances Are Tricky

Raw counts are affected by several factors.

Factor	What happens	Why it matters
Sequencing depth	A sample with twice as many mapped reads tends to have twice as many counts	Raw counts inflate with library size
Gene length	Longer genes generate more fragments than shorter genes at the same expression level	Counts can compare poorly across genes
Library composition	A few highly expressed genes consume a large fraction of reads	Other genes can appear lower even if unchanged
Mapping and annotation	Multi-mapping, gene model differences, and unannotated transcripts affect assignment	Gene-level counts depend on the reference
Batch and quality	Library prep, sequencing lane, RIN, and contamination can shift distributions	Technical variation can mimic biology

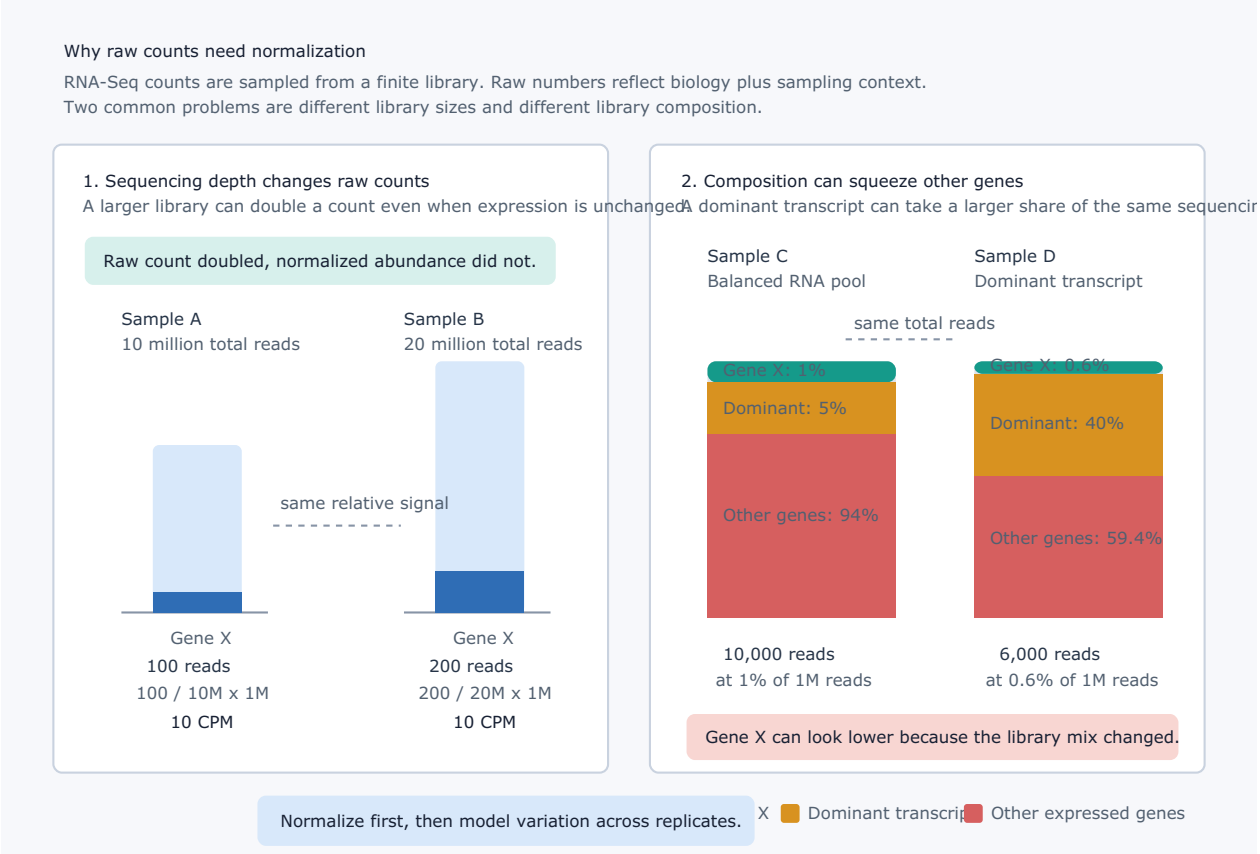
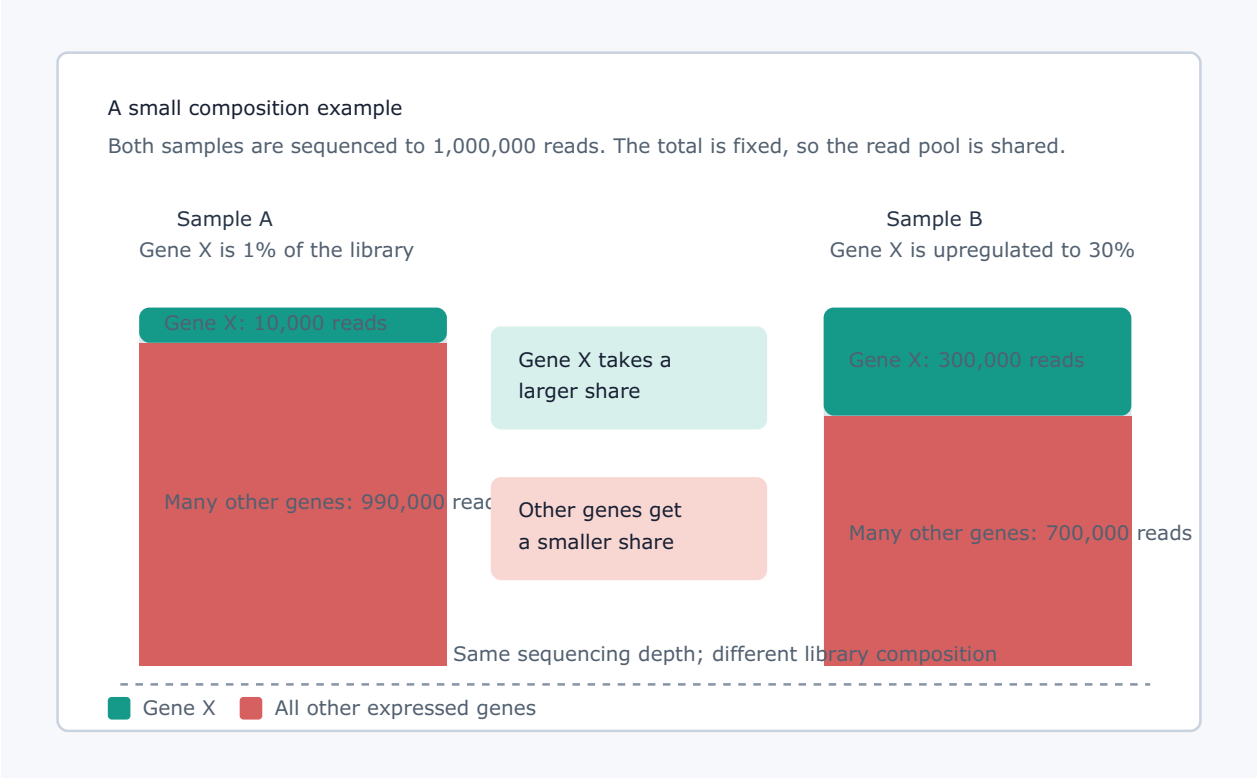


Figure: raw counts can change because sequencing depth changes, and they can also shift because one highly abundant transcript changes the composition of the finite sequencing library.

A Small Composition Example

Imagine two samples sequenced to exactly 1,000,000 reads. Gene X is upregulated in sample B. The library is fixed size, so other genes occupy a smaller fraction of the library even if their absolute molecule numbers did not change.



Gene group	Sample A reads	Sample B reads	Naive interpretation
Gene X	10,000	300,000	Upregulated
Many unchanged genes	990,000	700,000	Appears downregulated

This is why RNA-Seq normalization is not just "divide by total reads." Composition-aware methods try to estimate the sample scaling factor from genes that are not dominated by extreme expression changes.

---

4. QC Before DEG Analysis

Before DEG testing, we need to check that the count matrix and sample metadata are consistent, that library sizes are reasonable, and that biological replicates cluster together. If

these checks fail, you should investigate and fix the problem before trusting downstream results.

If QC is skipped, DEG results are often misleading regardless of method.

Wrangling our count matrix

Different tools may expect our data to be in different formats. Does the method / tool require certain column names, a specific file type, or a particular orientation of the matrix? Check the documentation for your method of choice before starting. You may need to transpose the matrix, rename columns, or convert to a different file type using excel, R, or Python. You can also use genAI to help you write a script to reformat your count matrix.

Pre-DEG QC Checklist

1. **Check sample metadata first**
    - Confirm sample labels match the count matrix.
    - Verify condition names, replicate IDs, and batch variables.
    - Make sure there are enough biological replicates (preferably at least 3 per group).

Replicate #

The number of replicates is of greater importance than the sequencing depth when applying differential expression methods. If you have a choice between sequencing more deeply or adding more replicates, adding replicates is usually the better choice. See Rapaport et al. 2013 *(https://link.springer.com/article/10.1186/gb-2013-14-9-r95)*. Aim for 3-5 replicates per condition for a standard bulk RNA-seq experiment. More replicates are better, especially if you expect high biological variability.

2. **Check sequencing depth and mapping summary**
    - Review total reads and mapped reads per sample.
    - Flag samples with unusually low depth or low mapping rate.
    - Ensure no sample is an extreme outlier in library size.
3. **Check low-count burden**
    - Count how many genes have near-zero counts across all samples.
    - Apply low-count filtering before DE testing.
    - Record filtering criteria for reproducibility.
4. **Check sample relationships**
    - Use PCA or MDS plots to inspect clustering.
    - Use sample-to-sample correlation or distance heatmaps.
    - Confirm biological replicates cluster more closely than unrelated samples.The experimental condition should be a major driver of separation in PCA or clustering.

Note

> PCA, clustering, and correlation heatmaps are almost never run on raw counts. Raw counts have a mean-variance relationship that distorts distances between samples and can make clustering misleading, being largely driven by a few highly expressed genes. Instead, these plots should be built on log-transformed or variance-stabilized values (log2CPM, VST, or rlog). See Section 5 for how these are calculated.

5. **Check for batch effects and outliers**
    - Ask whether batch, lane, date, or prep kit could drive separation in PCA.
    - Investigate outlier samples before removing them.
    - If batch effects are present, include batch terms in the design matrix.
6. **Check normalization assumptions**
    - Ask whether most genes are expected to be unchanged.
    - If massive global shifts are expected, note this as a limitation and consider spike-ins or external controls.
7. **Check model readiness**
    - Confirm raw counts are used for DESeq2/edgeR/limma-voom workflows.
    - Confirm contrast definitions are biologically meaningful.
    - Confirm the final design matrix matches the experimental question.

Before we trust p-values, we trust QC. DEG is not a button click. If sample labels are wrong, replicates do not cluster, or batch dominates the data, your method choice will not save the analysis.

---

5. Normalization Strategies

Normalization adjusts for technical differences so that biological comparisons are more meaningful. It does not remove the need for replication, good experimental design, or statistical modeling.

Low-Count Filtering.

Filtering is not a normalization method, but like normalization, it is a preprocessing step that affects downstream results.

Before testing, low-count genes are usually filtered. Genes with almost no reads across the experiment provide little statistical power and increase the multiple-testing burden.

Filtering is not just cosmetic. It changes the set of genes being tested, which also affects downstream enrichment backgrounds.

CPM

Counts per million (CPM) divides each gene count by the total library size and multiplies by one million.

Use CPM for:

- quick expression summaries;
- low-count filtering;
- exploratory plots after appropriate transformations.

Do not treat simple CPM scaling as a complete DE model.

RPKM, FPKM, and TPM

RPKM/FPKM and TPM normalize for both gene length and sequencing depth.

- **RPKM** (Reads Per Kilobase per Million) and **FPKM** (Fragments Per Kilobase per Million) are length- and depth-normalized measures; FPKM is the paired-end version of RPKM.
- **TPM** (Transcripts Per Million) is similar but rescales values so all genes in a sample sum to one million, making within-sample comparisons more interpretable.

Use TPM-like values for:

- describing relative transcript abundance within a sample;
- comparing expression of different genes cautiously within one sample;
- some visualization and downstream signature-scoring contexts.

Do not use RPKM and FPKM, and **avoid using RPKM/FPKM/TPM as direct input to DESeq2 or edgeR**. Those packages are designed for count data and include their own between-sample normalization inside the model.

TMM

Trimmed Mean of M-values (TMM) is used by edgeR and is commonly used before limma-voom. It estimates scaling factors after trimming genes with extreme log fold changes or extreme abundance.

TMM is useful when library composition differs across samples because it avoids letting a small number of highly expressed genes define the scaling factor.

Median-of-Ratios / RLE

DESeq2 uses a median-of-ratios size factor approach. Each sample is compared to a pseudo-reference based on gene-wise geometric means, and the median ratio estimates the sample scaling factor.

This approach works well when most genes are not changing strongly in one direction. If nearly all genes truly shift in the same direction, any global-scaling method becomes difficult to interpret without spike-ins or an external reference.

Transformations for Plots

These are types of transformations not normalizations.

Normalization vs. transformation. - *Normalization* adjusts for differences in sequencing depth/library size between samples (e.g., converting to counts per million, CPM). - *Transformation* (log2, VST, rlog) stabilizes variance across the range of expression so that highly-expressed genes don't dominate downstream clustering.

Variance-stabilizing transformations (`DESeq2::vst()`), regularized log transforms (`DESeq2::rlog()`), and log2(CPM + c) (`edgeR::cpm(log=TRUE)`) are often used for:

- PCA;
- clustering;
- heatmaps;
- correlation plots.

These are usually visualization and exploratory-analysis transformations. They are not substitutes for the DE model used for p-values.

> **Note**
>
> rlog > vst > log2CPM in terms of variance stabilization, but rlog is slower to compute and can be memory-intensive for large datasets. VST is faster and more scalable, while log2CPM is the simplest and fastest, but less effective at stabilizing variance for low-count genes. All 3 are valid, and you can compare how well they stabilize variance in your own data by examining correlation scatterplots and mean-variance plots.

Want more information?

See https://hbctraining.github.io/Training-modules/planning_successful_rnaseq/lessons/sample_level_QC.html *(https://hbctraining.github.io/Training-modules/planning_successful_rnaseq/lessons/sample_level_QC.html)* for a nice table comparing normalization strategies.

---

6. Meet the Tools

Before going further into theory, it helps to know what's actually out there. Three tools dominate RNA-Seq differential expression analysis in practice. All three are free, open-source, peer-reviewed, and distributed through **Bioconductor *(https://bioconductor.org/)***, the main repository for genomics software in R. All three tools also have excellent documentation and active support communities, so consider reading the vignettes and tutorials for each tool for an

in-depth understanding on how they work and how to use them. We only scratch the surface here.

> **Programming vs GUI**
>
> All three tools are R packages, but you do not need to know R to use them. iDEP (Integrated Differential Expression & Pathway analysis) provides a GUI that uses some of these tools under the hood. Running them directly in R gives you more flexibility, but requires some programming knowledge. You can start with iDEP and move to R later if you want more control. We will also see that iDEP provides R code snippets for each analysis, so you can learn the R commands while using the GUI and customize the code later if needed.

DESeq2

One of the two most widely used DE tools. Known for being relatively beginner-friendly with sensible defaults, strong documentation, and a very active support community.

- **Bioconductor page:** bioconductor.org/packages/DESeq2 *(https://bioconductor.org/packages/DESeq2/)*
- **How it works:** DESeq2 uses a "median-of-ratios" normalization strategy and models counts with a Negative Binomial distribution. It estimates per-gene dispersion and shrinks noisy estimates toward a global trend.
- **Good first use case:** A standard bulk RNA-seq comparison with a simple design and clear biological replicates.

edgeR

One of the two most widely used *count-based* DE tools, alongside DESeq2. Offers more manual control over the modeling process, but because of this, has a slightly steeper learning curve for beginners.

- **Bioconductor page:** bioconductor.org/packages/edgeR *(https://bioconductor.org/packages/edgeR/)*
- **How it works:** edgeR uses TMM normalization, fits Negative Binomial models with empirical Bayes dispersion estimation (often quasi-likelihood testing), and offers the most manual control for flexible, count-based modeling.
- **Good first use case:** Small-to-moderate studies where count-based modeling is preferred and the user wants flexibility.

limma-voom

Originally built for microarray data, extended to RNA-Seq counts via an added step called "voom." Particularly strong for complex experimental designs.

- **Bioconductor page:** bioconductor.org/packages/limma *(https://bioconductor.org/packages/limma/)*

- **How it works:** limma-voom typically uses TMM normalization then voom to convert counts to logCPM with precision weights, followed by weighted linear models with empirical Bayes moderation, making it especially strong for complex designs and multiple contrasts.
- **Good first use case:** Complex designs such as paired samples, batch-adjusted analyses, or multi-factor experiments.

Summary Table

Use this narrower 4-column version in Lesson8.md, replacing the current Summary Table.

Method	Normalization + model (high level)	Best use case	Main tradeoff
DESeq2	Median-of-ratios normalization; Negative Binomial GLM with shrinkage of dispersion (and optional log2FC shrinkage)	Standard two-group bulk RNA-seq with clear replicates	Strong defaults and robustness, but assumes most genes are not globally shifted
edgeR	TMM normalization; Negative Binomial modeling with empirical Bayes dispersion estimation (often quasi-likelihood testing)	Small-to-moderate studies needing flexible count-based modeling	Very flexible, but more manual choices can be harder for beginners
limma-voom	Usually TMM first, then voom converts counts to logCPM with precision weights; weighted linear models with empirical Bayes moderation	Complex designs (paired, batch-adjusted, multi-factor) and many contrasts	Powerful and fast, but depends on good mean-variance trend estimation and preprocessing

Important Assumptions: All samples are correctly labeled and biologically independent; the input is raw count data; replication is available so biological variability can be estimated; and, via whichever normalization step each one relies on (median-of-ratios for DESeq2, TMM for edgeR and, typically, for limma-voom), that most genes are not changing dramatically in the same direction.

What Do the Comparison Studies Say?

No single method wins across every study, dataset, and sample size; multiple independent method-comparison papers agree on that much. EdgeR, DESeq2, and limma-voom all perform well in practice, as there are inconsistencies in benchmarking across studies, and most methods tend to converge on similar results at larger sample sizes. The choice of method is often less important than good experimental design, replication, and QC.

Here are a few sources to peruse if you want to dig deeper into the literature:

- Soneson & Delorenzi 2013 *(https://pmc.ncbi.nlm.nih.gov/articles/PMC3608160/)* — simulation-based comparison of 11 DE methods

- Rapaport et al. 2013 *(https://genomebiology.biomedcentral.com/articles/10.1186/gb-2013-14-9-r95)* — real-data benchmark using SEQC/ENCODE datasets.
- Seyednasrollah et al. 2015 *(https://academic.oup.com/bib/article/16/1/59/240754)* — practical pipeline comparison of 8 packages
- Schurch et al. 2016 *(https://doi.org/10.1261/rna.053959.115)* — 48-replicate yeast benchmark, with a focus on how many replicates you actually need
- Love 2016 blog post *(https://mikelove.wordpress.com/2016/09/28/deseq2-or-edger/)* — a short, even-handed take from one of DESeq2's own authors on the DESeq2-vs-edgeR question

Why not just use a t-test?

RNA-Seq counts are discrete, non-negative integers with a large spike near zero and a long right tail, rather than the continuous, roughly bell-shaped data a t-test assumes. The simplest count model (Poisson, which assumes equal mean and variance) doesn't work either, because biological replicates vary more than Poisson allows, a property called **overdispersion**. DESeq2, edgeR, and limma-voom do not use exactly the same model, but they do solve the same RNA-Seq problems: extra variability beyond Poisson, small replicate numbers, and many near-zero genes. DESeq2 and edgeR use Negative Binomial count models, limma-voom uses weighted linear modeling of log-counts, and all three improve stability by sharing information across genes and by filtering sparse low-count features.

Why can't I find edgeR in iDEP?

DESeq2, edgeR, and limma-voom are the three tools you'll see referenced everywhere in the RNA-Seq literature, and all three are described above because you should know them. But when we move to **iDEP in Lesson 2, you'll only find DESeq2, limma-voom, and limma-trend as DE testing options; edgeR is not one of them**. iDEP does, however, quietly call edgeR's `cpm()` function under the hood for low-count filtering, so edgeR is present in the codebase, just not as a DE testing method you can select.

This isn't a statistical judgment against edgeR. The developers were explicit about it: the original iDEP publication states plainly that DE testing is done with limma-trend, limma-voom, and DESeq2, and specifically notes that *"other methods such as edgeR may be incorporated in the future." (https://link.springer.com/article/10.1186/s12859-018-2486-6)*

If your workflow specifically calls for edgeR, for example, matching a lab protocol, a collaborator's prior analysis, or a published method you're replicating, you'll need to run it separately in R or use a GUI that implements edgeR (e.g., Degust *(https://degust.erc.monash.edu)*). iDEP remains an excellent way to get a fast, GUI-driven DESeq2 or limma-voom analysis, but it is not a drop-in replacement if edgeR specifically is required.

What About limma-trend?

You may see **limma-trend** listed as a fourth option in iDEP (and elsewhere). It's a close cousin of limma-voom, built on the same underlying limma package, but meant for a slightly different situation.

- **limma-voom** takes raw counts, estimates the mean-variance trend, and turns it into a precision weight for every individual observation. It needs raw (or library-size-proportional) counts to do this correctly. Feeding it already-normalized values like TPM or CPM breaks its ability to infer the correct precision.
- **limma-trend** takes the same kind of mean-variance trend but folds it into the statistical testing step itself (via the empirical Bayes moderation), rather than into per-observation weights. This makes it appropriate for **already-normalized, log-scale expression data** — including microarray data, or RNA-seq counts that have already been transformed to logCPM.

When to reach for limma-trend specifically: it's the simplest and most robust choice when **sequencing depth (library size) is reasonably consistent across your samples**. If library sizes vary a lot between samples, limma-voom is the more powerful choice, since it models that variability directly through its per-observation weights. When library sizes are similar, the two methods tend to give comparable results, and limma-trend is the lighter-weight option. *(https://link.springer.com/article/10.1186/gb-2014-15-2-r29#Sec13)*

What If My Design Doesn't Fit Any of These?

DESeq2, edgeR, and limma-based methods cover most standard bulk RNA-Seq comparisons, but they're not the whole story. Specialized situations (e.g., single-cell RNA-Seq, longitudinal or repeated-measures designs, transcript-level (rather than gene-level) analysis, or experiments with very few or no replicates) often call for purpose-built methods beyond the scope of this lesson.

A good starting point for surveying what's out there is **Conesa et al. 2016, "A survey of best practices for RNA-seq data analysis," Genome Biology *(https://doi.org/10.1186/s13059-016-0881-8)***. This review walks through experimental design, differential expression, and single-cell RNA-Seq considerations, and is a reasonable jumping-off point before searching for a method tailored to your specific design.

---

7. Going Further: R and Python

iDEP (Lesson 2) is a great way to run a first differential expression analysis without writing code. But a GUI can only take you so far, custom experimental designs, batch correction, multi-omics integration, and full reproducibility all eventually call for running DESeq2, edgeR, or limma-voom yourself, in R or Python.

If you're new to R, or want a refresher before tackling DE analysis in code, BTEP offers introductory training:

- *Introductory R for Novices 2025 (https://bioinformatics.ccr.cancer.gov/docs/r_for_novices/)*
- *Data Visualization with R (https://bioinformatics.ccr.cancer.gov/docs/data-visualization-with-r/index.html)*
- *Data Wrangling with R (https://bioinformatics.ccr.cancer.gov/docs/data-wrangle-with-r/)*

We also have singular lessons on specific topics like ClusterProfiler, ComplexHeatmap, ggpubr, Quarto, etc. in the BTEP Coding Club documentation *(https://bioinformatics.ccr.cancer.gov/docs/btep-coding-club/)*. I recommend searching for "R Programming" in the BTEP Coding Club documentation to find relevant lessons.

Roughly, here's the path:

1. Learn base R (or brush up) — vectors, data frames, reading in files.
2. Learn to load a count matrix and sample metadata into R.
3. Follow a DESeq2 or edgeR vignette on your own data (see the Bioconductor links in Section 3).

Python users can reach similar results with packages such as `PyDESeq2` *(https://pydeseq2.readthedocs.io/en/stable/)*, though the R ecosystem remains the most mature and best-documented path for RNA-Seq DE analysis today.

For an introduction to python, BTEP offers the Python Introductory Education Series *(https://bioinformatics.ccr.cancer.gov/docs/python-introductory-series/)*.

---

8. Introducing iDEP

In the next lesson, we'll put everything from this lesson into practice using **iDEP** (integrated Differential Expression and Pathway analysis), a free, web-based tool that runs DESeq2, limma-voom, and limma-trend behind a point-and-click interface, so you can go from a count matrix to differential expression results and pathway enrichment without writing any code.

iDEP was built by the Ge Lab at South Dakota State University and is available as a public web app at bioinformatics.sdstate.edu/idep *(https://bioinformatics.sdstate.edu/idep/)*. For this training, we'll instead be running iDEP directly on **Biowulf**, NIH's HPC cluster, using version **iDEP 2.01**.

Running iDEP on Biowulf *(https://hpc.nih.gov/apps/idep.html)* rather than the public web app has a few advantages for our purposes:

1. Your data stays on NIH systems rather than being uploaded to an external server.
2. You get dedicated compute resources rather than sharing a public server with other users worldwide.

> **Files and iDEP on Biowulf**
>
> The file browser in iDEP on Biowulf accesses your local file system (on your laptop or workstation) rather than the Biowulf file system. You can either download the count matrix to your local machine and upload it to iDEP, or mount the Biowulf file system on your local machine *(https://hpc.nih.gov/docs/hpcdrive.html)* and upload the count matrix from there.

What You'll Need

- An active Biowulf account (see https://hpc.nih.gov/docs/accounts.html *(https://hpc.nih.gov/docs/accounts.html)* if you don't have one yet). Student accounts are available to a limited number of registrants for this training.
- Access to the NIH network or VPN.
- Your count matrix file (from Section 1 — `RSEM.genes.expected_count.all_samples.txt` or your own data, formatted the same way).

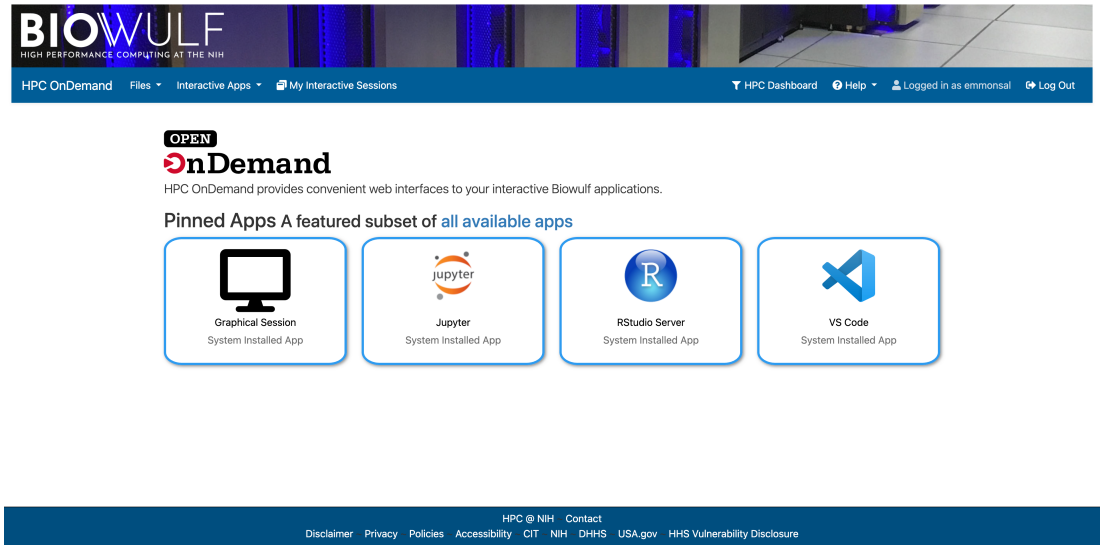
Getting the Data

You can download the count matrix here.

9. Accessing iDEP via HPC Open OnDemand on Biowulf

Step 1 — Log in to HPC Open OnDemand

- Navigate to https://hpcondemand.nih.gov/ *(https://hpcondemand.nih.gov/)* (NIH network or VPN required).
- Authenticate with your NIH credentials (PIV/MFA).



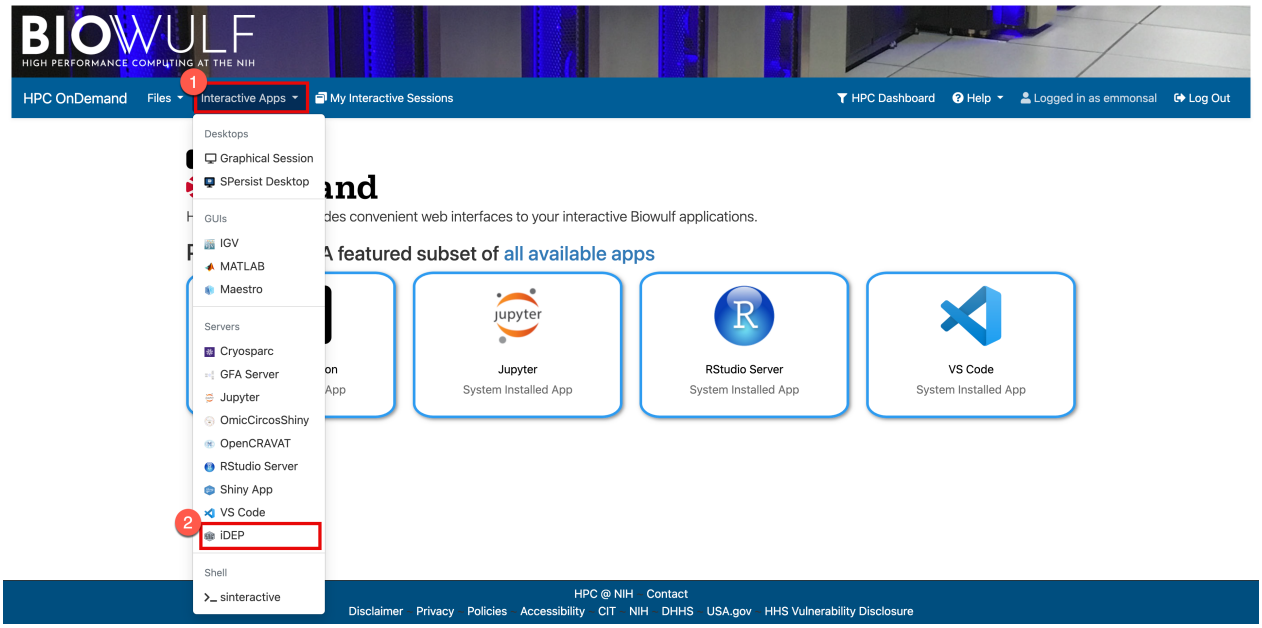
HPC Open OnDemand Dashboard at login.

> **Student Accounts**
>
> A limited number of student accounts are available for this lesson. If you are using a student account, navigate to https://hpcclass.cit.nih.gov/ *(https://hpcclass.cit.nih.gov/)*. If you were not able to grab a student account and you do not have a Biowulf account, feel free to use the public iDEP server *(https://bioinformatics.sdstate.edu/idep/)*.

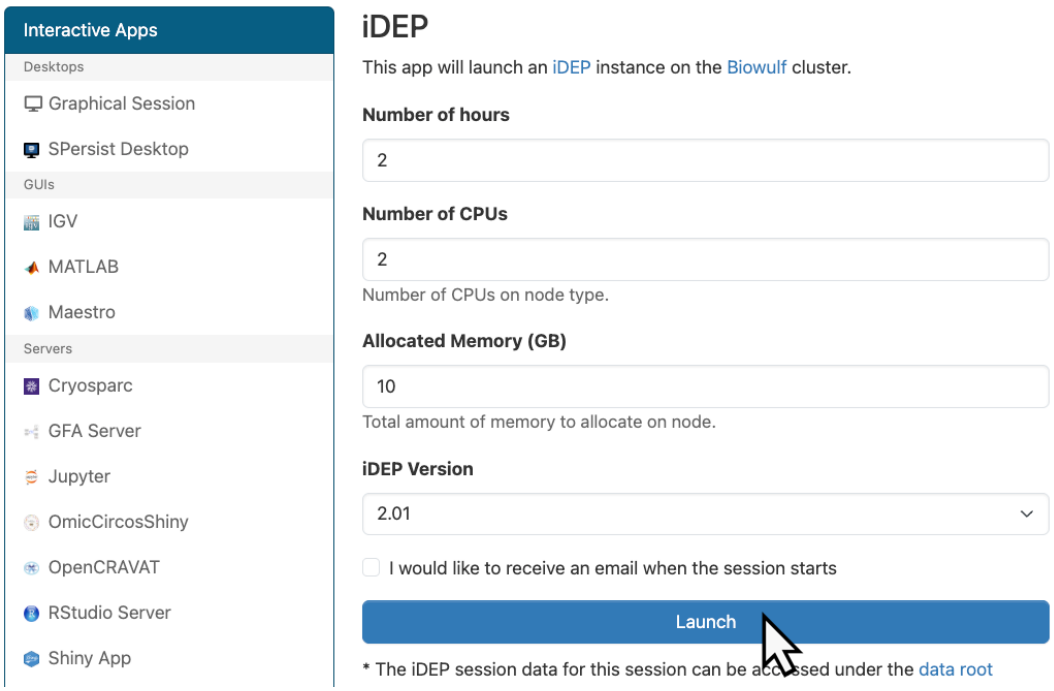
Step 2 — Launch the iDEP app

iDEP is not among the four pinned apps on the OnDemand homepage, but it is available in the full list of interactive applications.

- From the Open OnDemand dashboard, locate the "Interactive Apps" tab and select **iDEP**.



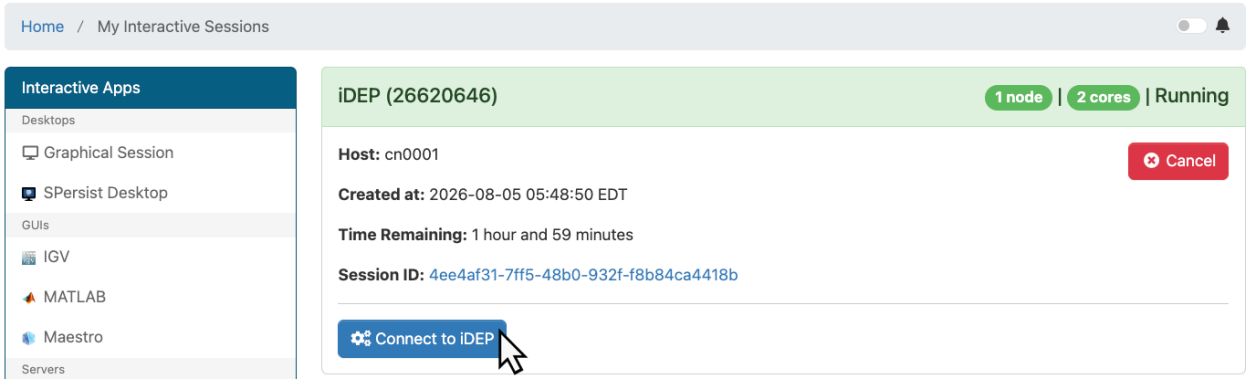
- Select appropriate resource parameters for your session (CPUs, memory, walltime)
- Launch the session and wait for the job to be allocated on a compute node.



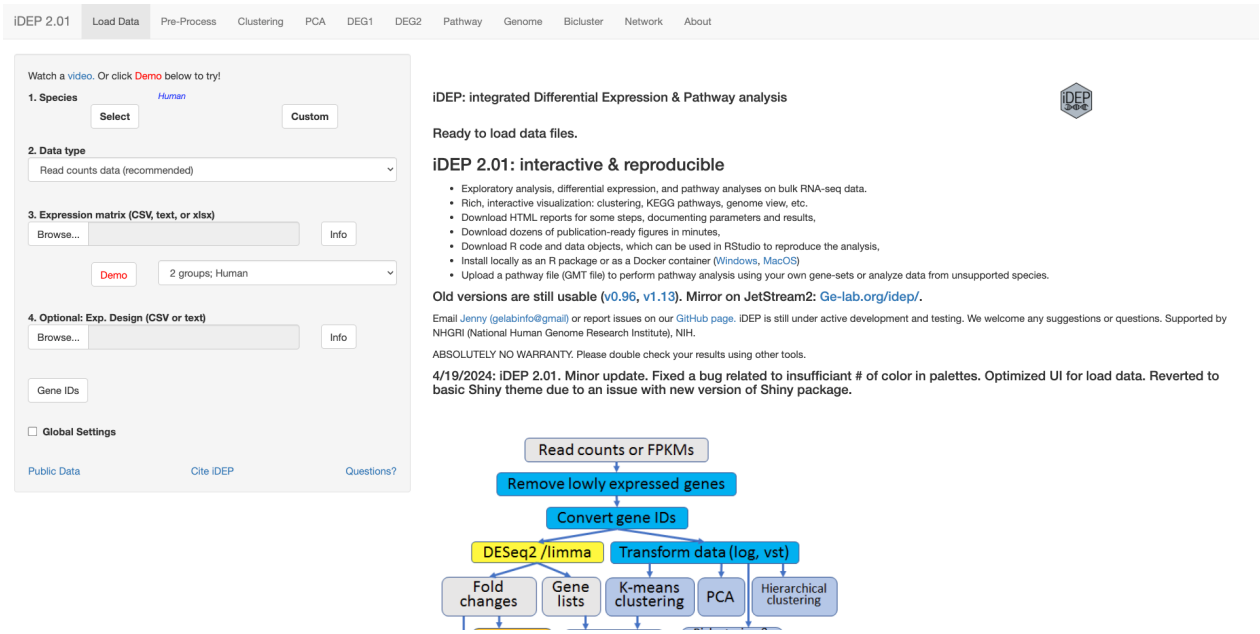
Select resources for your iDEP session and launch the job.

Step 3 — Connect to your running iDEP session

- Once the job starts, Open OnDemand will provide a **Connect to iDEP** button/link.
- This opens the iDEP Shiny interface directly in your browser — no manual SSH tunneling required.



Step 4 — Orient yourself in the iDEP interface



Step 5 — Upload your count matrix

There is no way to save the current iDEP session, so you will need to upload your count matrix (or other accepted files) each time you start a new session. We will walk through the upload process in Lesson 2, but here are some important points to keep in mind:

1. The gene expression matrix must be a tab-delimited or comma separated value file with rows as genes and columns as samples. The first column should contain gene identifiers, and the first row should contain sample identifiers. See the Data Format section of the iDEP documentation for details: https://idepsite.wordpress.com/data-format/ *(https://idepsite.wordpress.com/data-format/)*.
2. iDEP can convert gene identifiers to Ensembl IDs for many species, but it is best to use Ensembl IDs in your input file if possible.
3. Ensemble IDs should not include version numbers (e.g., ENSG00000123456.1). If your input file has version numbers, you can remove them in R or Python before uploading.

4. The raw count matrix must contain integer counts. RSEM outputs non-integer expected counts, so you will need to round them to integers before uploading. iDEP will not accept non-integer counts.

How can I wrangle my input file without programming experience?

- You can use Excel or Google Sheets to remove version numbers, transpose the matrix, or rename columns.
- You can also use genAI to reformat your count matrix.

Example prompt:

```
I have a tab-delimited count matrix that needs to be reformatted for iDEP differen
Please apply the following changes and save the result as [output_filename].txt:

1. Remove the GeneName column entirely.
2. Keep the gene_id column and its header, but strip any version suffixes from the
(e.g. ENSG00000276871.1 → ENSG00000276871).
3. Round all numeric data columns to integers.

The file will always contain gene_id and GeneName columns. The number and names of
data columns may vary. Please make no other changes to the file.

The input file is attached: [input_filename].txt.
```

Alternatively, you can use AI to generate a script in R or Python that performs the same operations on your count matrix, and use that script as a reusable tool for future datasets and a record for reproducibility.

```
Please draft an R script that can implement the changes that were applied to test.t
```

Version differences between iDEP on Biowulf and the public web app

iDEP on Biowulf is currently version 2.01, which may differ slightly in menu layout from the public web app at bioinformatics.sdstate.edu/idep, which is on a newer release. If something doesn't match a tutorial you find online, check the version number first.

In lesson 2, we will start with a wrangled count matrix and walk through the iDEP interface to run a full differential expression analysis. You can find our wrangled count matrix here.

---

10. Quick Knowledge Check

1. Why is a raw count of 200 in one sample not automatically higher expression than a raw count of 100 in another sample?

Library size, composition, and other technical factors can differ across samples.

2. Why are TPM values useful for description but not ideal as direct input to DESeq2 or edgeR?

TPM values are useful for viewing relative abundance, but count-based DE methods expect raw counts and do their own normalization.

3. Why should QC be done before DEG testing?

QC catches problems (mislabels, outliers, batch effects, poor depth) that can invalidate downstream results.

4. What does a positive log2 fold change mean?

The gene is estimated to be higher in the comparison group than in the reference group.

5. Why do we adjust p-values when testing thousands of genes?

Testing many genes creates false positives by chance, so p-values must be adjusted.

6. Name one sign from PCA or sample correlation that tells you DEG results may not be trustworthy yet.

Example answers: replicates do not cluster together, one sample is an extreme outlier, or samples separate by batch instead of biology.

7. Name the three major DE tools and one distinguishing "good first use case" for each.

DESeq2 (standard two-group comparisons), edgeR (flexible count modeling for small-to-moderate studies), limma-voom (complex/multi-factor designs).

Which of the three major DE tools is *not* available as a DE testing method in iDEP, and why?

edgeR. iDEP's developers built DE testing around limma-trend, limma-voom, and DESeq2, and explicitly left edgeR as a possible future addition.

Take-Home Messages

- **DE analysis tests whether expression differences are larger than chance** — it is not a ranking of raw counts. Interpret effect size and adjusted p-value together.
- **DESeq2, edgeR, and limma-voom are the three tools you'll meet everywhere** — all free, peer-reviewed, and available through Bioconductor.
- **Always use raw counts** as input to DESeq2, edgeR, and limma-voom. TPM and FPKM values are not appropriate inputs for these tools.
- **Normalization assumes most genes are not changing.** QC before testing is the best way to check that this assumption holds.
- **No statistical method can rescue poor experimental design or missing replicates.**

- **iDEP (next lesson) gets you a full analysis with no code** — using DESeq2 and limma-based methods. If your workflow specifically requires edgeR, you'll need to run it separately in R.
- **R or Python gives you full flexibility** once you're ready for it, including access to every method mentioned in this lesson.

Recommended Reviews and Tutorials by Section

Section	Useful review articles	Useful tutorials and guides
1. Start Here: What Can We Do with a Count Matrix?	Conesa et al. 2016 RNA-seq best practices *(https://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-0881-8)*	Bioconductor rnaseqGene workflow *(https://bioconductor.org/packages/release/workflows/vignettes/rnaseqGene/inst/doc/rnaseqGene.html)*; RNA-seq analysis is easy as 1-2-3 with limma, Glimma and edgeR *(https://www.bioconductor.org/packages/devel/workflows/vignettes/RNAseq123/inst/doc/limmaWorkflow.html)*
2. The Core Question	Conesa et al. 2016 RNA-seq best practices *(https://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-0881-8)*; Soneson and Delorenzi 2013 DEG method comparison *(https://pmc.ncbi.nlm.nih.gov/articles/PMC3608160/)*	Bioconductor rnaseqGene workflow *(https://bioconductor.org/packages/release/workflows/vignettes/rnaseqGene/inst/doc/rnaseqGene.html)*; Harvard FAS bulk RNA-seq DE tutorial *(https://informatics.fas.harvard.edu/resources/tutorials/differential-expression-analysis/)*
3. Meet the Tools	Soneson and Delorenzi 2013 method comparison *(https://pmc.ncbi.nlm.nih.gov/articles/PMC3608160/)*; Rapaport et al. 2013 comprehensive DEG-method evaluation *(https://genomebiology.biomedcentral.com/articles/10.1186/gb-2013-14-9-r95)*; Love et al. 2014 DESeq2 *(https://doi.org/10.1186/s13059-014-0550-8)*; Robinson et al. 2010 edgeR *(https://doi.org/*	RNAseq123 workflow *(https://www.bioconductor.org/packages/devel/workflows/vignettes/RNAseq123/inst/doc/limmaWorkflow.html)*; DESeq2 vignette *(https://www.bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html)*

Section	Useful review articles	Useful tutorials and guides
	10.1093/bioinformatics/btp616); Law et al. 2014 voom *(https://doi.org/10.1186/gb-2014-15-2-r29)*	
4. Why Absolute Abundances Are Tricky	Robinson and Oshlack 2010 TMM normalization *(https://genomebiology.biomedcentral.com/articles/10.1186/gb-2010-11-3-r25)*; Zhao et al. 2020 misuse of RPKM/TPM *(https://pmc.ncbi.nlm.nih.gov/articles/PMC7373998/)*	DESeq2 vignette: count transformations and normalized counts *(https://www.bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html)*; RNAseqGene exploratory analysis sections *(https://bioconductor.org/packages/release/workflows/vignettes/rnaseqGene/inst/doc/rnaseqGene.html)*
5. QC Before DEG Analysis	Conesa et al. 2016 RNA-seq best practices *(https://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-0881-8)*	Bioconductor rnaseqGene exploratory workflow *(https://bioconductor.org/packages/release/workflows/vignettes/rnaseqGene/inst/doc/rnaseqGene.html)*; Harvard FAS bulk RNA-seq QC and DE tutorial *(https://informatics.fas.harvard.edu/resources/tutorials/differential-expression-analysis/)*
6. Normalization Strategies	Evans et al. 2017 normalization assumptions *(https://pmc.ncbi.nlm.nih.gov/articles/PMC6171491/)*; Dillies et al. 2013 normalization-method evaluation *(https://doi.org/10.1093/bib/bbs046)*	edgeR user guide *(https://bioconductor.org/packages/release/bioc/vignettes/edgeR/inst/doc/edgeRUsersGuide.pdf)*; DESeq2 vignette *(https://www.bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html)*
7. Going Further: R and Python	R or Python: Which should I learn? *(https://bioinformatics.ccr.cancer.gov/btep/r-or-python-which-should-i-learn/)*; Learning R with BTEP *(https://bioinformatics.ccr.cancer.gov/btep/learning-r-with-btep/)*	*Introductory R for Novices 2025 (https://bioinformatics.ccr.cancer.gov/docs/r_for_novices/)*; *Data Visualization with R (https://bioinformatics.ccr.cancer.gov/docs/data-visualization-with-r/index.html)*; *Data Wrangling with*

Section	Useful review articles	Useful tutorials and guides
		R (https://bioinformatics.ccr.cancer.gov/docs/data-wrangle-with-r/); Python Introductory Education Series *(https://bioinformatics.ccr.cancer.gov/docs/python-introductory-series/)*
8. Introducing iDEP	Ge, Son & Yao 2018 iDEP paper *(https://doi.org/10.1186/s12859-018-2486-6)*	iDEP public web app *(https://bioinformatics.sdstate.edu/idep/)*
9. Accessing iDEP via HPC Open OnDemand		NIH HPC Open OnDemand documentation *(https://hpc.nih.gov/ondemand/)*; iDEP on Biowulf *(https://hpc.nih.gov/apps/idep.html)*
10. Practice and check	Conesa et al. 2016 RNA-seq best practices *(https://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-0881-8)*	Harvard FAS bulk RNA-seq DE tutorial *(https://informatics.fas.harvard.edu/resources/tutorials/differential-expression-analysis/)*; Bioconductor RNA-seq workflows collection *(https://bioconductor.org/help/workflows/)*

Resources Used

- Love MI, Huber W, Anders S. 2014. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biology. https://doi.org/10.1186/s13059-014-0550-8
- DESeq2 Bioconductor package and vignette: https://bioconductor.org/packages/DESeq2/
- Robinson MD, McCarthy DJ, Smyth GK. 2010. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics. https://doi.org/10.1093/bioinformatics/btp616
- edgeR Bioconductor package and user guide: https://bioconductor.org/packages/edgeR/
- Law CW, Chen Y, Shi W, Smyth GK. 2014. voom: precision weights unlock linear model analysis tools for RNA-seq read counts. Genome Biology. https://doi.org/10.1186/gb-2014-15-2-r29
- Smyth GK. 2004. Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. Statistical Applications in Genetics and Molecular Biology. https://doi.org/10.2202/1544-6115.1027

- limma Bioconductor package: https://bioconductor.org/packages/limma/
- Brintha VPB. LinkedIn post: "DESeq2, edgeR, and limma-voom are among the most widely used methods..." — Accessible visual summary of the three-stage comparison framework (variance modeling, expression modeling, statistical testing). https://www.linkedin.com/posts/brintha-vpb_deseq2-edger-and-limma-voom-are-among-the-share-7469552632911683584-OnDu/
- Robinson MD, Oshlack A. 2010. A scaling normalization method for differential expression analysis of RNA-seq data. Genome Biology. https://doi.org/10.1186/gb-2010-11-3-r25
- Zhao S et al. 2020. Misuse of RPKM or TPM normalization when comparing across samples and sequencing protocols. RNA. https://doi.org/10.1261/rna.074922.120
- iDEP web application documentation, verified June 8, 2026: https://bioinformatics.sdstate.edu/idep/
- Ge SX, Son EW, Yao R. 2018. iDEP: an integrated web application for differential expression and pathway analysis of RNA-Seq data. BMC Bioinformatics 19:534. https://doi.org/10.1186/s12859-018-2486-6 (states that DE testing in iDEP uses limma-trend, limma-voom, and DESeq2, noting edgeR may be incorporated in the future)
- iDEP on Biowulf (NIH HPC), version 2.01, verified July 2026: https://hpc.nih.gov/apps/idep.html
- NIH HPC Open OnDemand documentation, verified July 2026: https://hpc.nih.gov/ondemand/
- Conesa A et al. 2016. A survey of best practices for RNA-seq data analysis. Genome Biology. https://doi.org/10.1186/s13059-016-0881-8
- Soneson C, Delorenzi M. 2013. A comparison of methods for differential expression analysis of RNA-seq data. BMC Bioinformatics. https://pmc.ncbi.nlm.nih.gov/articles/PMC3608160/
- Evans C, Hardin J, Stoebel DM. 2018. Selecting between-sample RNA-Seq normalization methods from the perspective of their assumptions. Briefings in Bioinformatics. https://pmc.ncbi.nlm.nih.gov/articles/PMC6171491/
- Dillies MA et al. 2013. A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. Briefings in Bioinformatics. https://doi.org/10.1093/bib/bbs046
- Rapaport F et al. 2013. Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data. Genome Biology. https://doi.org/10.1186/gb-2013-14-9-r95
- RNA-seq workflow: gene-level exploratory analysis and differential expression. Bioconductor. https://bioconductor.org/packages/release/workflows/vignettes/rnaseqGene/inst/doc/rnaseqGene.html
- RNA-seq analysis is easy as 1-2-3 with limma, Glimma and edgeR. Bioconductor. https://www.bioconductor.org/packages/devel/workflows/vignettes/RNAseq123/inst/doc/limmaWorkflow.html
- edgeR quasi-likelihood RNA-seq workflow. Bioconductor. https://bioconductor.posit.co/packages/3.19/workflows/vignettes/RnaSeqGeneEdgeRQL/inst/doc/edgeRQL.html
- Harvard FAS Informatics Group. Bulk RNA-seq DE analysis tutorial. https://informatics.fas.harvard.edu/resources/tutorials/differential-expression-analysis/

- Anders S, Huber W. 2010. Differential expression analysis for sequence count data. Genome Biology. https://doi.org/10.1186/gb-2010-11-9-r106
- McCarthy DJ, Chen Y, Smyth GK. 2012. Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. Nucleic Acids Research. https://doi.org/10.1093/nar/gks042
- Bourgon R, Gentleman R, Huber W. 2010. Independent filtering increases detection power for high-throughput experiments. PNAS. https://doi.org/10.1073/pnas.0914005107
- HBC Training. Introduction to DGE (archived). https://hbctraining.github.io/DGE_workshop/lessons/01_DGE_setup_and_overview.html

Lesson 9: Differential Expression Analysis with iDEP

Learning Objectives

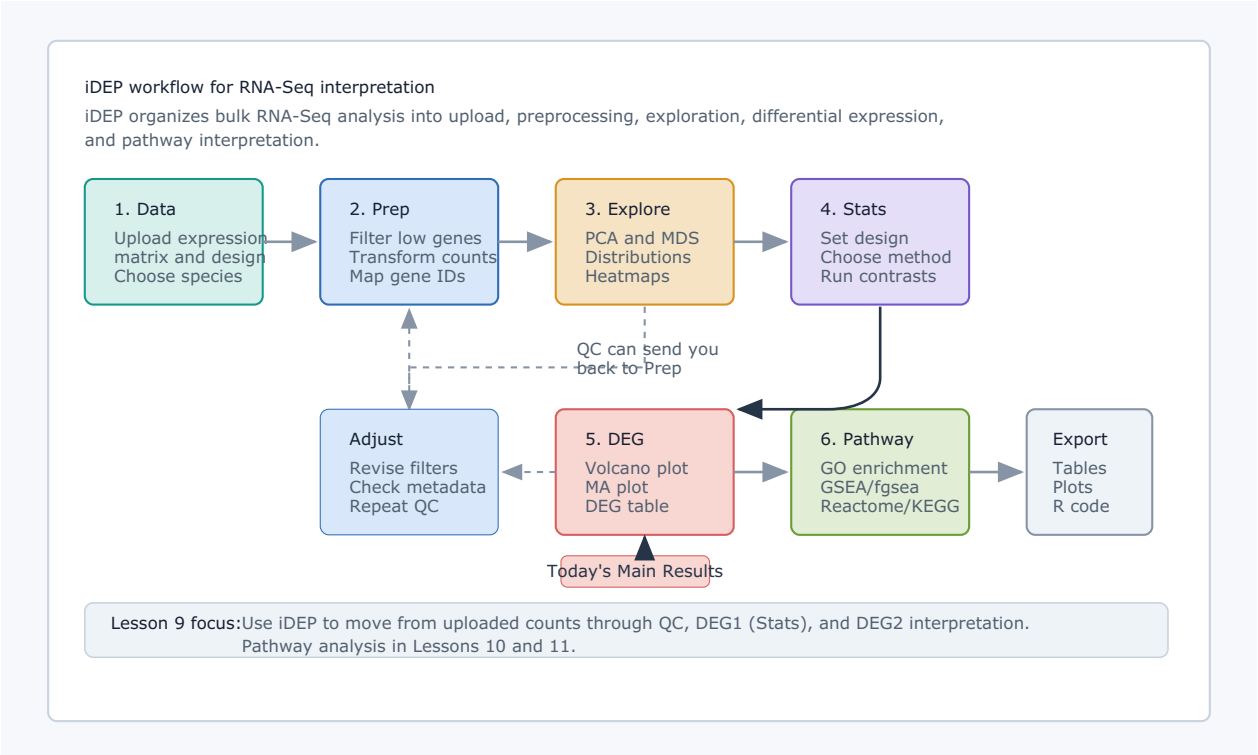
By the end of this lesson, learners should be able to:

- upload an RNA-Seq expression matrix and sample design into iDEP;
- run preprocessing, QC, and differential expression analysis in iDEP;
- interpret PCA, heatmap, volcano plot, MA plot, and DEG table outputs;
- explain what the selected contrast means;
- export results and identify settings needed for reproducibility.

Introduction to iDEP

iDEP stands for Integrated Differential Expression and Pathway analysis. It is a web application for interactive RNA-Seq analysis that brings together data upload, preprocessing, visualization, differential expression, gene-set enrichment, pathway analysis, and reproducible output in one interface.

For this module, iDEP is useful because it lets learners see the analysis workflow without writing R code first. The goal is not to hide the statistics, but to make the sequence of decisions visible: upload the matrix, check the data, define the comparison, run a model, inspect the DEG output, and then ask pathway-level questions.



Resources for using iDEP:

- Add the paper reference here https://link.springer.com/article/10.1186/s12859-018-2486-6 *(https://link.springer.com/article/10.1186/s12859-018-2486-6)*
- Wordpress docs: https://idepsite.wordpress.com/ *(https://idepsite.wordpress.com/)*
- Github page: https://github.com/gexijin/idepGolem *(https://github.com/gexijin/idepGolem)*

Helpful Things to Know About iDEP:

- if the analysis hangs at any point, try refreshing the page and starting over. You will need to start again from the "Load Data" tab. While annoying, it shouldn't take too long to get back to the same point.

Get Started with HPC OnDemand

If you are following along with this lesson, go ahead and open a browser to the NIH HPC OnDemand portal: https://hpcondemand.nih.gov/ *(https://hpcondemand.nih.gov/)* and log in with your NIH credentials. Then open an iDEP session using the instructions outlined in Lesson 8.

> **NIH HPC OnDemand Student Accounts**
>
> If you are using one of the available student accounts, you will need to navigate to https://hpcclass.cit.nih.gov/ *(https://hpcclass.cit.nih.gov/)*. There are a limited number of student accounts available, so please be considerate of other students. If you are unable to access the student accounts, you can still follow along with the lesson using a personal Biowulf account of the public iDEP server *(https://bioinformatics.sdstate.edu/idep/)*. New features and

updates may be available on the public server that are not yet available on the Biowulf iDEP installation. For a similar experience, you can select v2.01 from the public iDEP server homepage.

Get the Files

In Lesson 1, we downloaded the primary RENEE output file `RSEM.genes.expected_count.all_samples.txt`. For this iDEP demo, we will use a modified version of that file ready for import rsem_modified.txt.

More about RSEM.genes.expected_count.all_samples.txt

This matrix comes from RSEM's own per-sample rollup (transcript-level counts summed to genes), not from `tximport`. That's fine for most analyses, but it means the matrix carries no per-sample transcript-length correction. Gene-level counting implicitly assumes each gene's effective transcript length is constant across samples; if isoform usage shifts between conditions, that assumption breaks, since a sample using a longer isoform of a gene will generate more reads at the same expression level. `tximport` uses transcript-level estimates to compute a per-sample, length-weighted offset that corrects for this (Soneson, Love, and Robinson 2015). If your samples/conditions are unlikely to involve major isoform-usage shifts, RENEE's gene-level matrix is a reasonable, standard input for DESeq2/edgeR/limma-voom as-is.

Reference: Soneson C, Love MI, Robinson MD. Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences. F1000Research. 2015;4:1521. https://doi.org/10.12688/f1000research.7563.1

Sample Names and Design File

If you look at our modified count matrix, the sample names are: hcc1395_normal_rep1, hcc1395_normal_rep2, hcc1395_normal_rep3, hcc1395_tumor_rep1, hcc1395_tumor_rep2, and hcc1395_tumor_rep3.

iDEP uses the sample names to define the sample groups. The rep number should be placed at the end, with the condition preceding it. With simple experimental designs (e.g., two groups, no batch effects, etc.), an experimental design file is optional. See the iDEP documentation for specifics on the sample IDs and design file format: https://idepsite.wordpress.com/data-format/ *(https://idepsite.wordpress.com/data-format/)*.

Continuous Variables / Covariates

There are seemingly no options or documented examples in iDEP for non-factorial (continuous) designs. The experimental design interface appears to be built around factor-based columns. However, all three major DE tools discussed in Lesson 8 (DESeq2, edgeR, and limma-voom) can accept numeric covariates directly in their design formulas or model matrices. Trainees wanting more flexible designs will likely need to move to R for that work. See the Bioconductor Support threads below (and others not listed) for more discussion on incorporating continuous covariates in DESeq2 and limma-voom.

Further reading (Bioconductor Support threads):

- DESeq2: Continuous and discrete variables in the design (#126713) *(https://support.bioconductor.org/p/126713/)*

- DESeq2 with continuous variables — predictors and/or outcomes (#97262) *(https://support.bioconductor.org/p/97262/)*
- Limma-voom with both categorical and continuous confounders (#104580) *(https://support.bioconductor.org/p/104580/)*
- Design matrix for continuous independent variable in limma (#76638) *(https://support.bioconductor.org/p/76638/)*

Consider consulting with a statistician *(https://bioinformatics.ccr.cancer.gov/btep/the-advanced-biomedical-computational-science-abcs-group/)* when modeling complex experimental designs, especially if you have continuous covariates or multiple confounders. iDEP is a great tool for simple designs, but it does not expose the full flexibility of the underlying DESeq2 or limma-voom workflows.

iDEP Orientation

In version 2.01, iDEP has eleven primary tabs. Latest versions of iDEP have a cleaner UI with fewer tabs, but the underlying workflow is the same. The following table summarizes the purpose of each tab.

Tab	Teaching purpose
Load Data	Upload expression matrix, fold changes, design matrix, or gene choose species, choose data type
Pre-process	Filter low-expression genes, transform data, inspect processed matrix, examine useful QC plots
Clustering / PCA	Explore sample and gene structure before formal testing
DEG	Define design and run differential expression models
DEG2	Inspect genes, heatmaps, volcano plots, MA plots, and DEG enrichment
Pathway	Run pathway analysis methods

There are two additional tabs: Bicluster and Network, which require larger sample sizes and are beyond the scope of this lesson.

iDEP includes many opportunities to download data, plots, and R code for reproducibility..

Upload the Data

Demo script:

1. Choose species: `Homo sapiens`.
2. Select Data type - In this case we are using the "Read counts data" as this will provide the most robust differential expression analysis.
3. Upload `rsem_modified.txt` and define the experimental design.

4. Consider including an experiemental design file if you have one. For this demo, we are not including a design file. **If you do include a design file, the sample names in the design file must match the sample names in the count matrix.**
5. Modify global settings if needed. This is your chance to include some customizations like plot colors, themes, and ID types.

Gene ID Mapping

iDEP can map gene IDs to a common set of identifiers for each species. This is useful if your input data uses gene symbols, Entrez IDs, or other identifiers. For this demo, we are using Ensembl gene IDs. Note, for version 2.01, if you choose "Do not convert gene IDs", iDEP will not attempt to map gene symbols for downstream plots.

Preprocess the Data

The pre-processing step is important for filtering and transformation prior to QC and exploratory data analysis. iDEP provides a number of options for filtering and transformation. The default settings are usually reasonable, but you can adjust them if needed.

Filtering

Before differential expression analysis, iDEP filters out lowly-expressed genes, which add noise without adding biological signal. iDEP uses CPM-based filtering for count data. CPM is calculated by scaling each gene's raw count by that sample's total library size:

CPM = (raw count / total library size) * 1,000,000

A common rule is to keep genes that pass a minimum CPM threshold (CPM > 0.5) in at least one or more samples.

Why CPM?

A fixed cutoff like "10 reads" treats every sample the same, even if some were sequenced much more deeply than others, making the filter too lenient for shallow samples and too strict for deep ones. CPM adjusts for library size first, so the threshold reflects the same relative expression level regardless of sequencing depth. This makes it a depth-aware filter.

The "n libraries" setting controls how many samples must clear the CPM threshold. A common practice is to set this to match your number of replicates per group, so a gene isn't kept based on one noisy sample alone.

Transformation

Before clustering or PCA, count data needs to be transformed to stabilize variance across the range of expression levels. This is to avoid genes with the highest counts dominating the analysis. iDEP offers three options:

- **edgeR-style log2(CPM + c)** *(default)*: log2 transformation with a small pseudo-count (c=1–10, default c=4) added to avoid taking log(0). Fast and works well in most cases — a good default for beginners.
- **VST (variance stabilizing transform)**: fast even with many samples; applies a strong, uniform transformation across expression levels. Best when library sizes are fairly similar across samples.
- **rlog (regularized log)**: more robust when library sizes vary a lot between samples, since it shrinks low-count genes more carefully. Slower than VST, so only recommended for fewer than 10 samples.

Rule of thumb: many samples or similar library sizes → VST; fewer samples and/or uneven library sizes → rlog.

A few practical notes:. - Transformed data is used for exploratory analysis (clustering, PCA) and for limma-trend DEG analysis. DESeq2 and limma-voom instead use the raw counts directly.
- If library sizes vary more than ~3-fold between samples, limma-trend isn't recommended, as this method works best for samples with similar sequencing depths.
- iDEP checks for this automatically, flagging a warning if sequencing depth differs significantly between groups, since this can confound the analysis.

Filtering and Transformation Summary

Following filtering and transformation, we will receive a summary of what was filtered: **"1453 genes in 6 samples. 627 genes passed filter, 605 were converted to Ensembl gene IDs in our database. The remaining 22 genes were kept in the data using original IDs."**

QC and Exploratory Analysis

After transformation, iDEP provides several diagnostic plots to help you verify the transformation worked well and to explore your dataset. Nicely, we can easily compare transformation methods by switching between them in the UI.

Warning

On heavily sub-setted or very small gene sets (like this chr22 practice dataset), VST's dispersion-trend fitting step can run very slowly or stall, since it's designed to fit trends across genome-wide data. If VST hangs, started log or rlog are reliable alternatives for small datasets like this one.

The following plots are included here:

1. Barplot of library sizes
2. Boxplot of transformed expression distributions
3. Density plot of transformed expression distributions
4. Scatterplots of pairwise sample correlations
5. Dispersion plot of transformed expression distributions
6. QC plot with the number of genes by type
7. Gene plot comparing gene expression by condition or sample

Plots 2-5 are about verifying the transformation worked before moving on to downstream clustering and PCA. Plots 6-7 are more about exploring whether the data makes sense biologically.

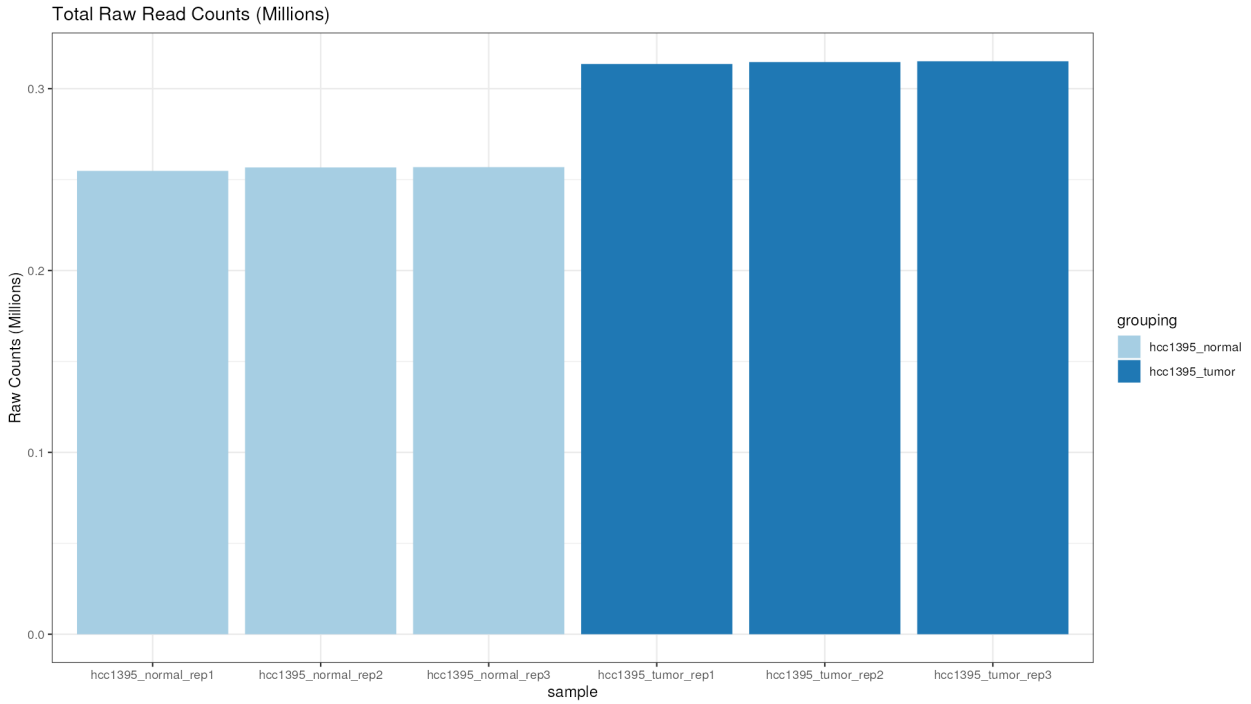
Barplot of library sizes

Are samples similarly sequenced? Do any samples have unusually small or large libraries?

iDEP checks whether sequencing depth (library size) differs significantly between your comparison groups, using an ANOVA test (flagged if $P < 0.01$). It also reports the ratio of max/min total counts across groups, with <1.5 generally considered acceptable.

> We received "Warning! Sequencing depth bias detected. Total read counts are significantly different among sample groups (p= 2.10e-07) based on ANOVA. Total read counts max/min = 1.23."

This warning is a prompt to check whether depth and biology are entangled, not an automatic sign that something is wrong. Stay under a max/min ratio of 1.5, and use your knowledge of the samples to judge whether it's worth acting on. Not that if depth varies more than ~3-fold across samples, avoid limma-trend and use DESeq2 or limma-voom instead, since they handle depth differences more robustly.



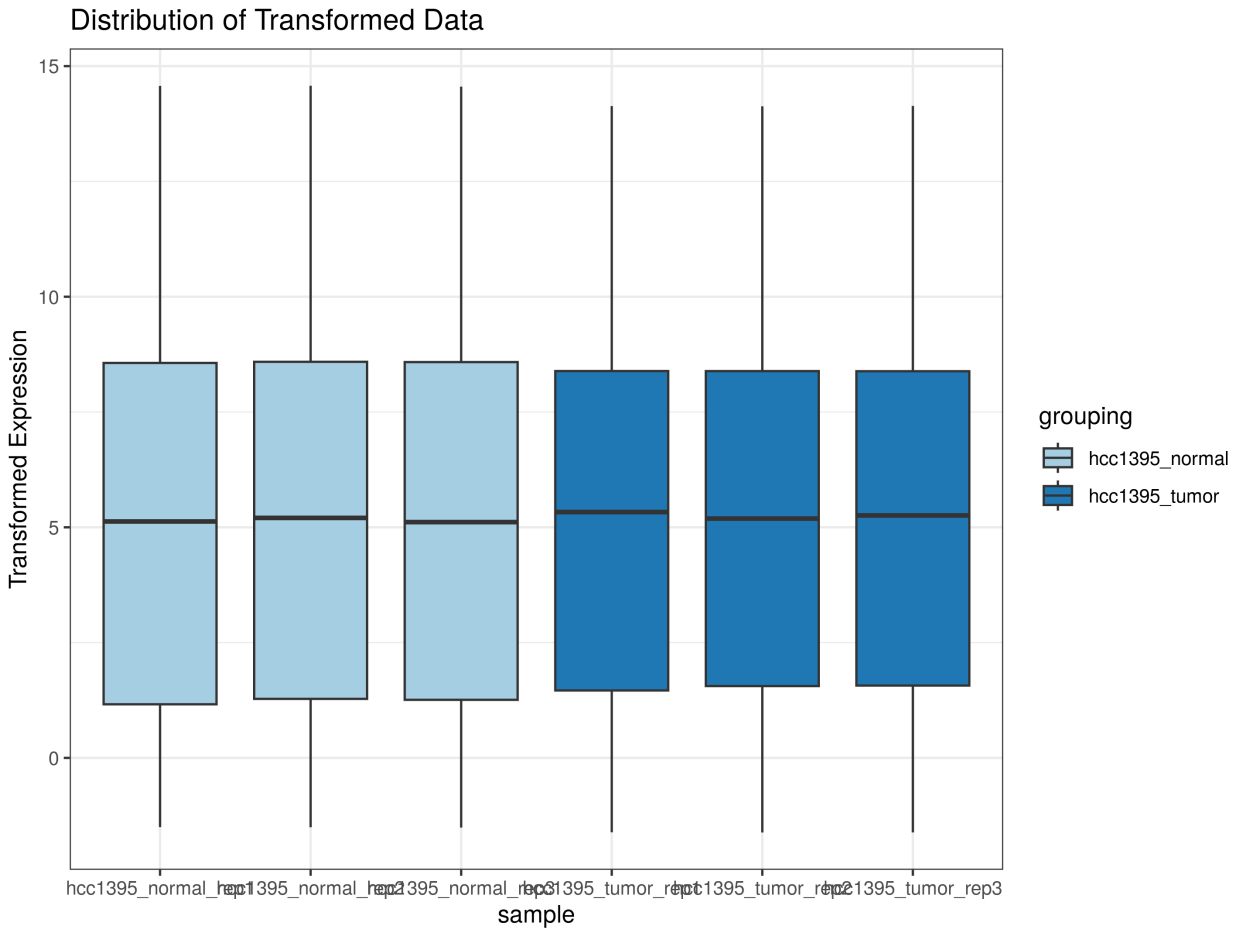
Small dataset for teaching

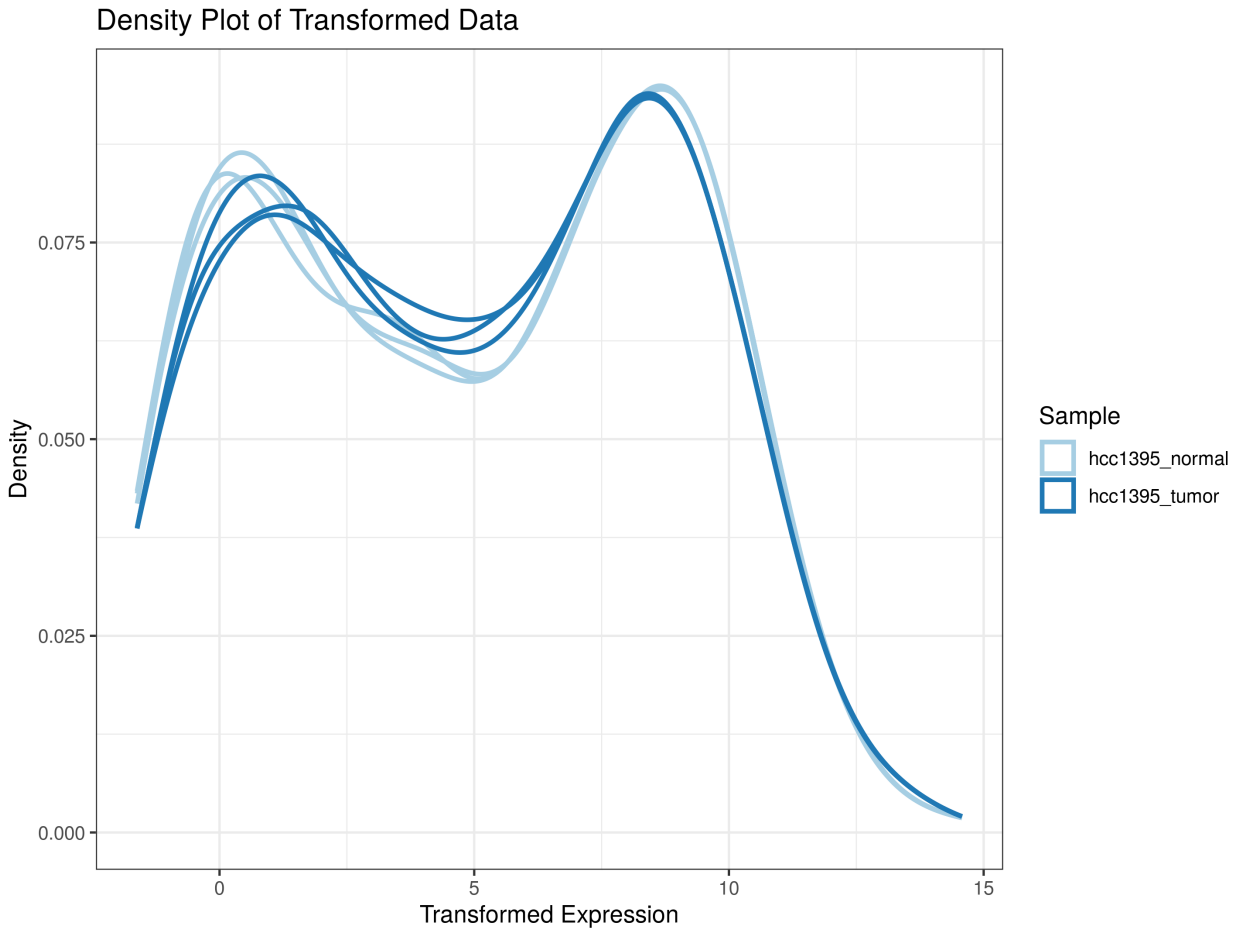
The HCC1395 dataset is a small, pre-filtered subset of a larger RNA-seq experiment. It is intended for teaching purposes only. The small size allows for fast runtimes in iDEP, but it does not represent the full complexity of a real RNA-seq experiment. In practice, you would typically work with larger datasets with more replicates and more genes.

Boxplot and Density plots of transformed expression distributions

Do transformed expression distributions look comparable?

The boxplot shows the spread (median, IQR, range) of expression per sample, while the desnity plot shows the shape of each sample's distribution. For the boxplot, you should look for boxes and medians aligned across samples, indicating consistent normalization. For the density plot, look for overlapping curves across samples, indicating similar distributions. A bimodal shape (one peak of low/noise genes, one peak of expressed genes) is typical and expected for RNA-seq data.

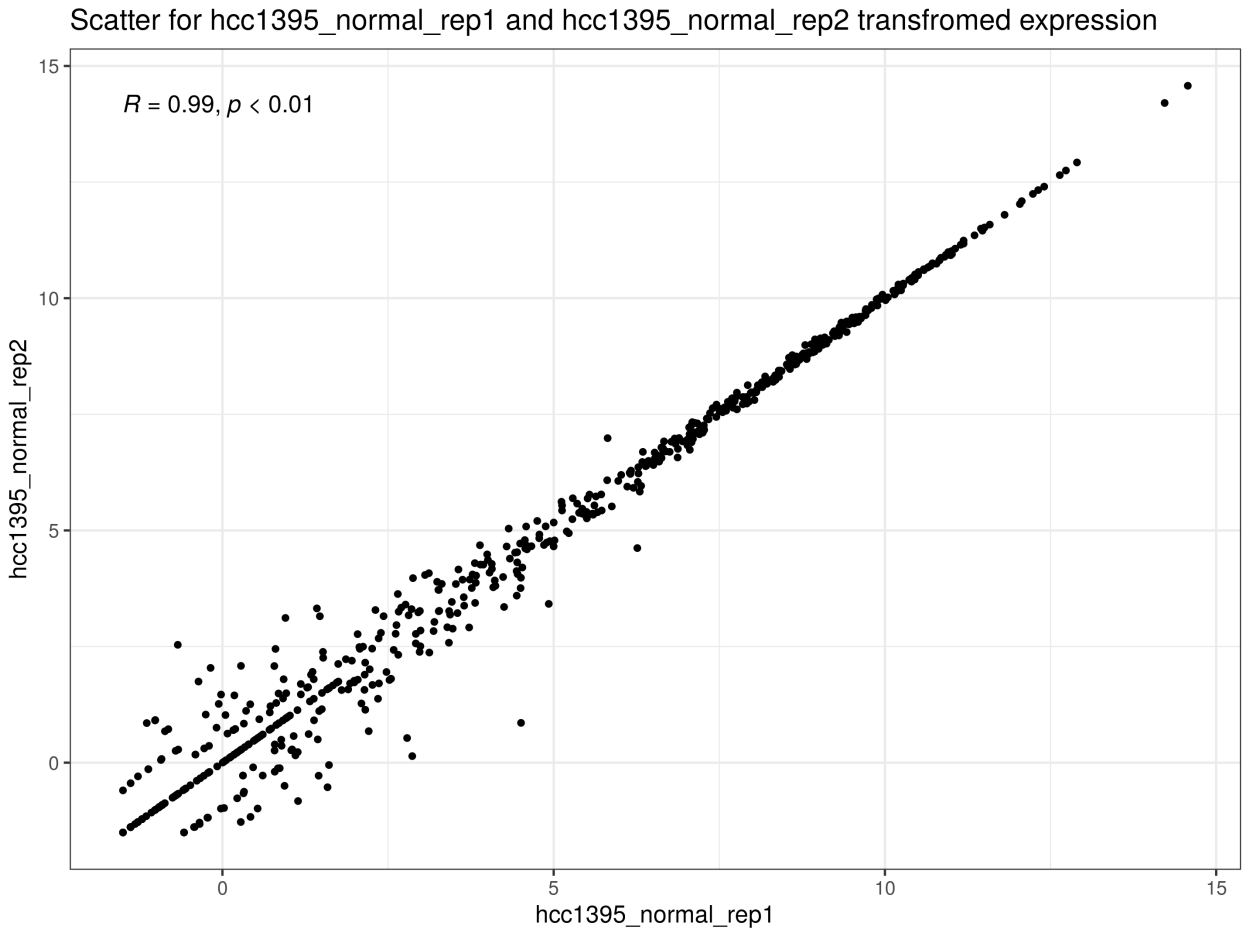

Distribution of Transformed Data
Transformed Expression
15
10
5
0
hcc1395_normal_rep1
hcc1395_normal_rep2
hcc1395_normal_rep3
hcc1395_tumor_rep1
hcc1395_tumor_rep2
hcc1395_tumor_rep3
sample
grouping
hcc1395_normal
hcc1395_tumor



Scatterplots of pairwise sample correlations

Do technical or biological replicates resemble each other?

These scatter plots compare transformed expression between two samples directly. You can compare replicates within a condition and replicates between conditions. You should see tight clustering along the diagonal and a high correlation (R close to 1) between replicates within a condition, indicating good reproducibility between replicates.

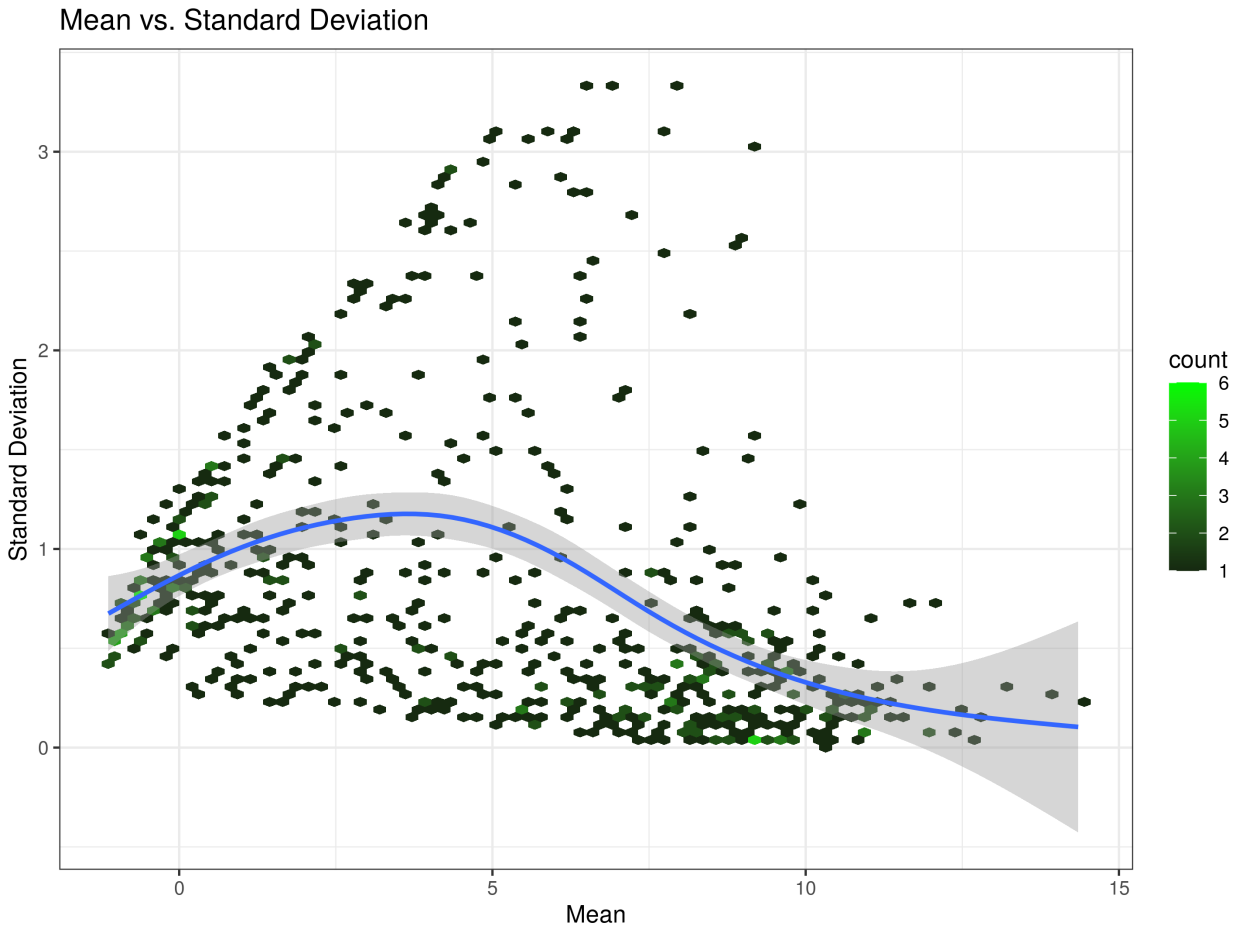


We can also see here how the rlog transformation has stabilized variance across the range of expression levels, as the points are more tightly clustered along the diagonal compared to the log2(CPM + c) transformation. Compare this to the idep documentation for more examples of how the different transformations affect the scatter plots.

Mean vs standard deviation plot of transformed expression distributions

Did the transformation successfully stabilized variance across expression levels?

This dispersion plot can also provide information on how well the transformation stabilized variance across expression levels. Ideally, we would like to see a relatively flat trend since transformation methods like VST and rlog aim to reduce the relationship between mean expression and variance (making the data more homoskedastic, or having consistent variance across the mean, rather than heteroskedastic, where variance changes with expression level). Some residual trend is normal; no transformation removes it completely.

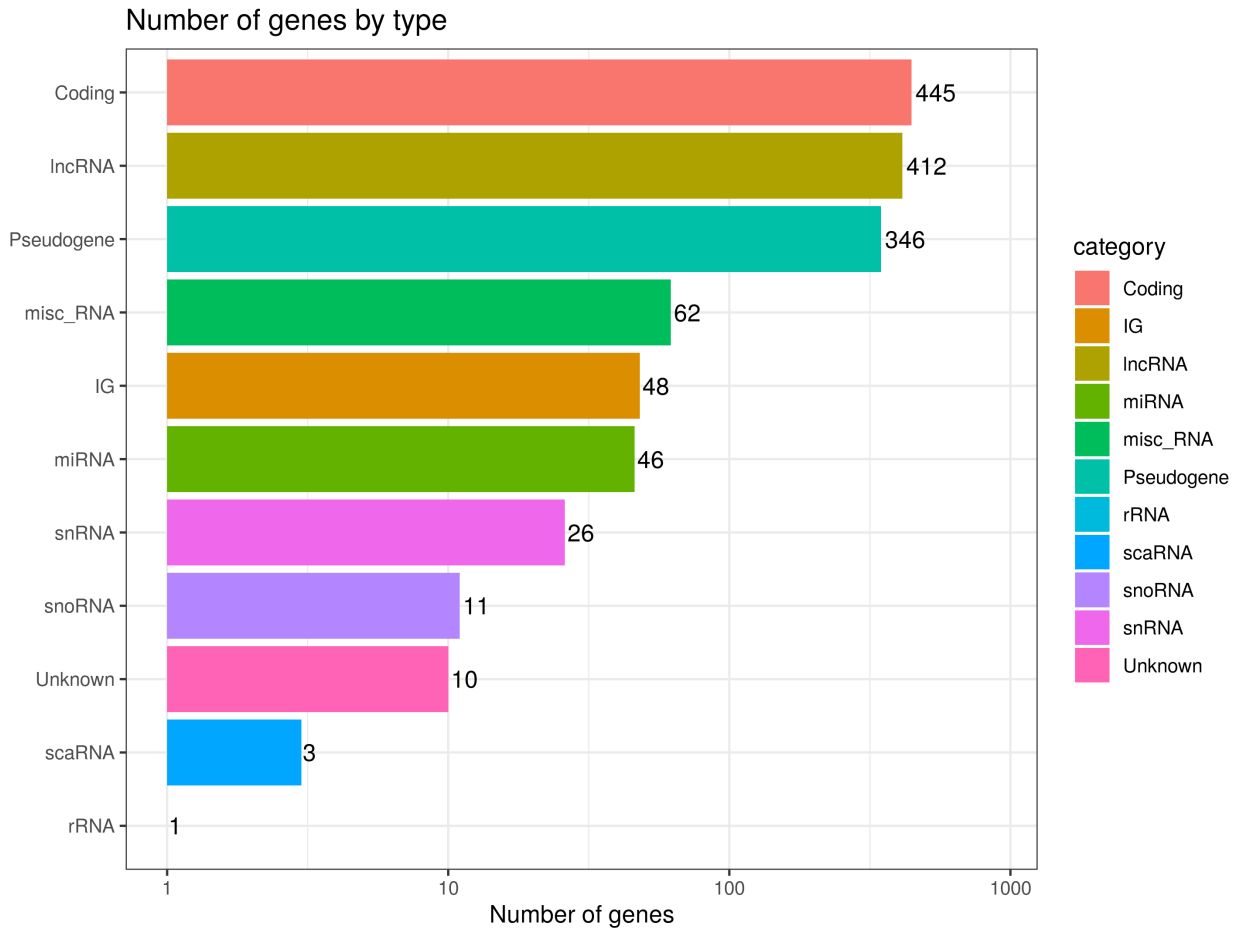


Read more on this here *(https://bioconductor.org/packages/release/workflows/vignettes/rnaseqGene/inst/doc/rnaseqGene.html#exploratory-analysis-and-visualization)*.

QC plot with the number of genes by type

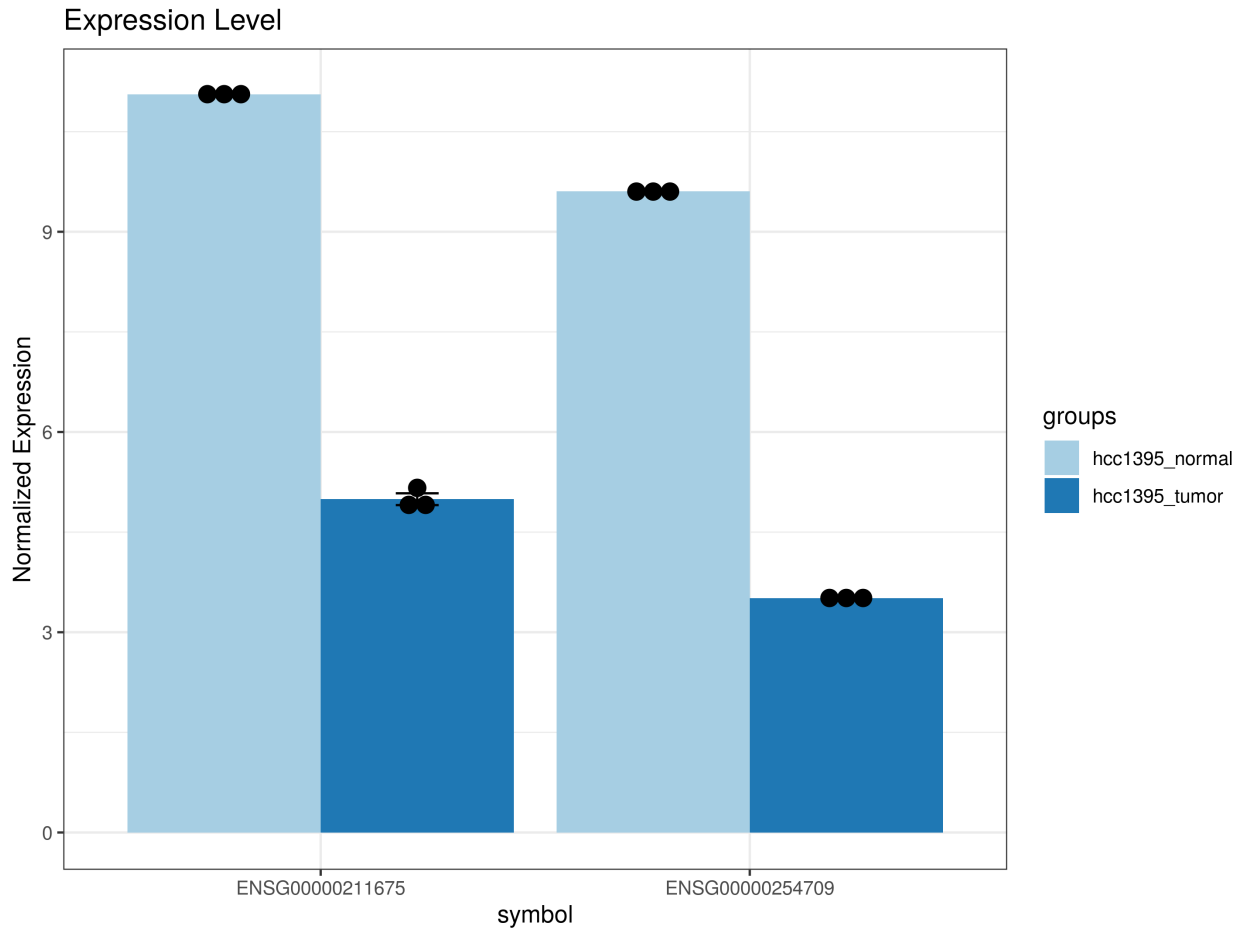
Are the expected gene types present?

This is basically an annotation summary of your filtered gene set by type (protein-coding, lncRNA, pseudogene, etc.), based on iDEP's Ensembl-derived annotation. We want to make sure the composition matches our expectations. We should see more coding genes than non-coding genes, and a reasonable number of lncRNAs and pseudogenes. If the composition is unexpected, it may indicate a problem with the input data or filtering.



Gene plot comparing gene expression by condition or sample: Are the expected genes expressed?

Shows normalized expression for specific genes you select, split by group or sample with replicate points overlaid. Useful for checking known marker genes against expectations, or for visually confirming genes that came up as significant in your DEG results.



Clustering and Heatmap

Hierarchical clustering of samples and genes.

Do samples cluster as expected? Are genes grouped into coherent patterns?

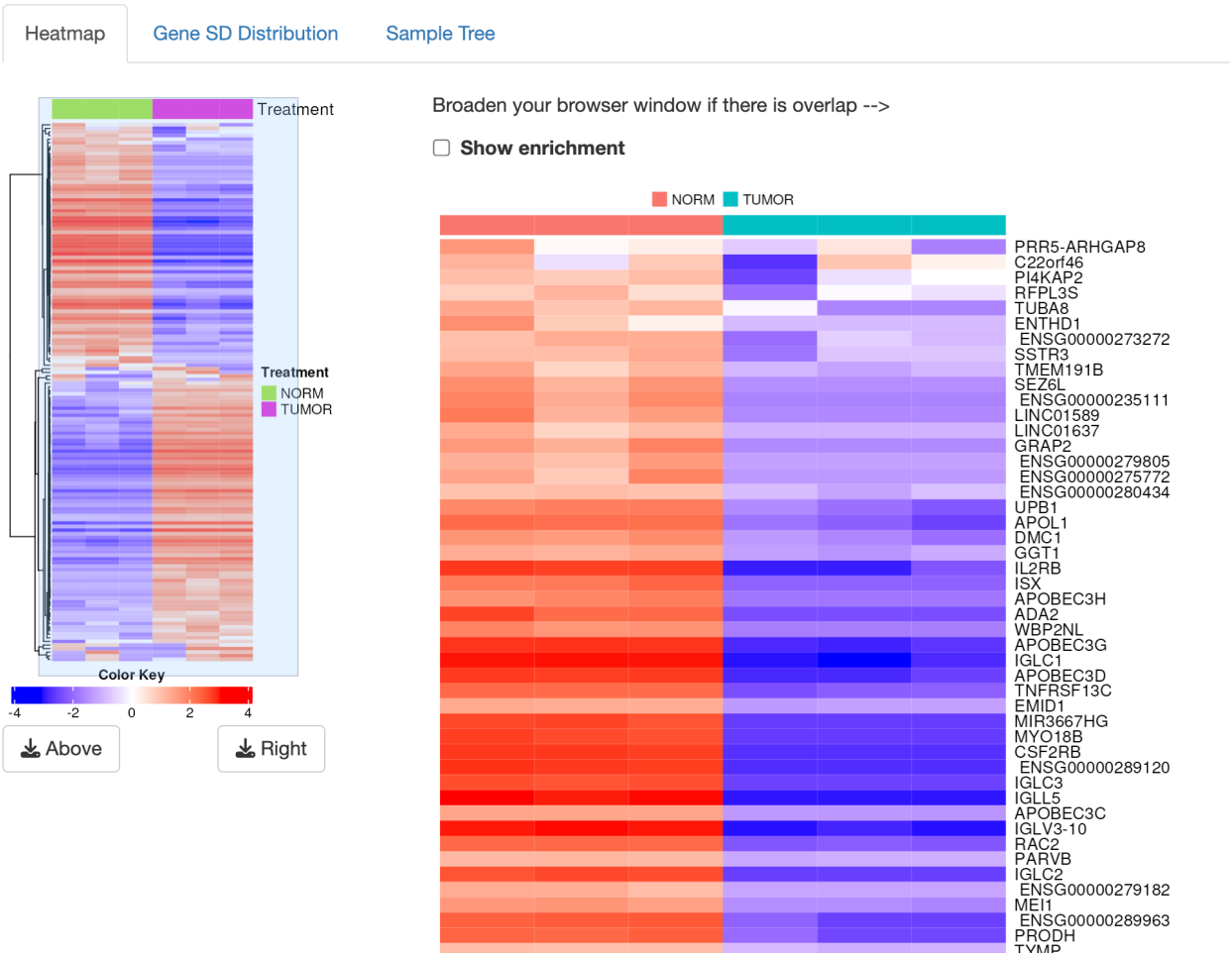
The Clustering tab gives you a visual overview of your dataset through hierarchical clustering of both samples and genes, allowing further exploratory data analysis. Rather than jumping straight into formal statistics, this tab lets you eyeball the overall structure in your data first. While this step will not tell you which genes are significant, it can catch problems that would otherwise go unnoticed for a long time: sample mix ups, labeling errors, or contamination. The heatmap will let you see whether your samples group together as expected, and whether the genes you selected for clustering show coherent patterns.

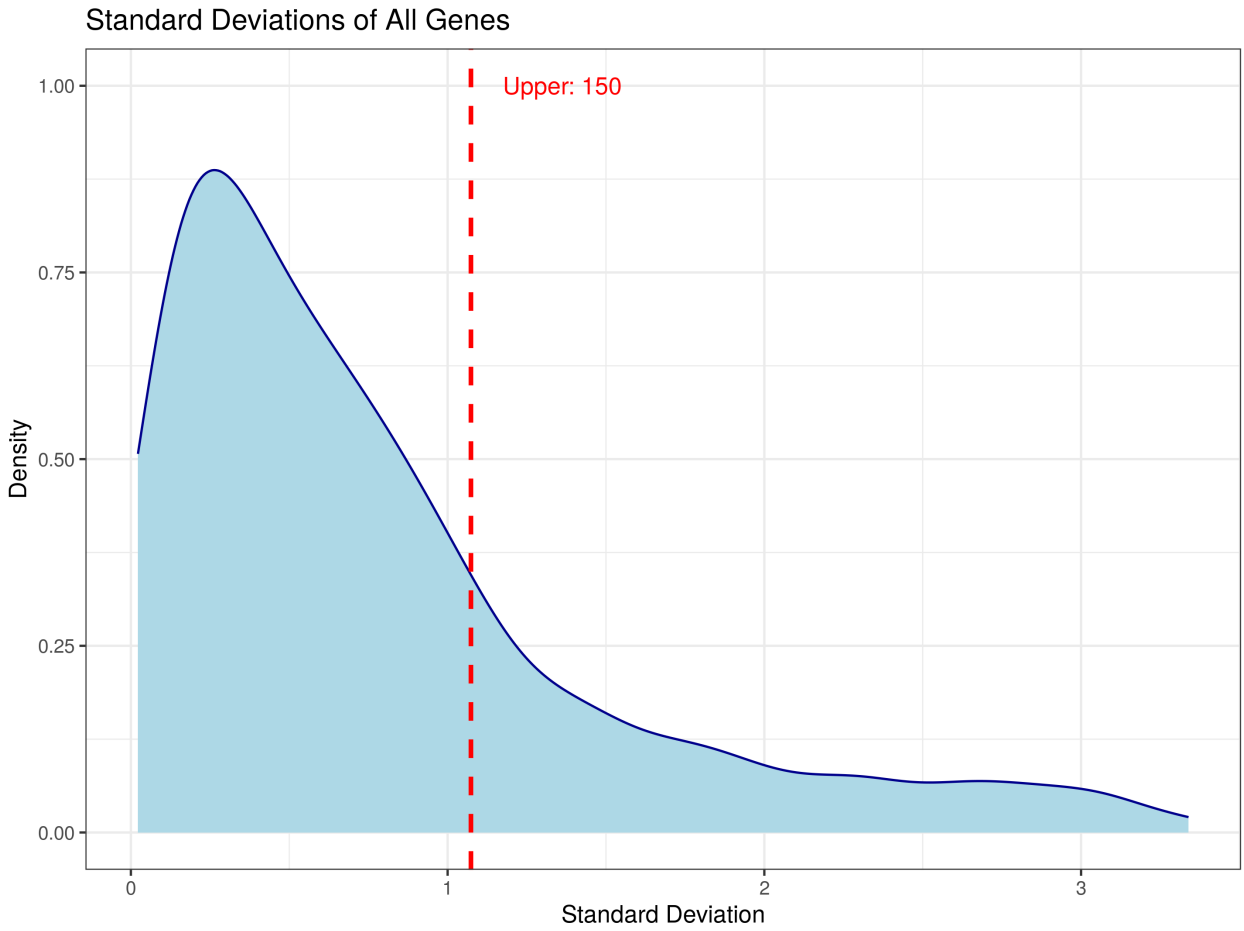
By default, iDEP identifies the genes with the greatest variability across your samples (typically the top 2,000, ranked by standard deviation on the transformed data) and clusters using those. Genes that barely change across any sample carry little information for distinguishing groups, so they are excluded from this view. Expression values are also centered per gene, meaning the colors you see represent whether a sample is above or below that gene's own average, not its absolute expression level. Clustering itself relies on correlation between samples or genes, so two samples with similar expression patterns end up placed near each other in the dendrogram

regardless of overall magnitude. Importantly, your experimental group labels play no role in how the clustering is calculated. They only appear afterward as a color strip for reference. Read more in the docs *(https://idepsite.wordpress.com/heatmap/)*.

The Gene SD Distribution can be used to select the number of genes to include in the heatmap. The default is 2,000, but you can adjust this number to include more or fewer genes. The heatmap will update automatically when you change this setting. You want to shoot for the right skewed tail, which includes the genes with the most variation across samples.

Otherwise, the heatmap uses sensible defaults including Pearson for the distance metric and complete linkage for the clustering method. You can change these settings if you want to explore different clustering methods, but only if you have an understanding of what those methods do and how they affect the results *(https://bioinformatics.ccr.cancer.gov/docs/data-visualization-with-r/Lesson5_intro_to_ggplot/#distance-clustering-and-scaling)*.





K-means clustering of genes

The K-means clustering method is an alternative to hierarchical clustering. It partitions the genes into a pre-specified number of clusters (k) based on their expression patterns across samples. The user can choose the number of clusters, and iDEP will assign each gene to one of these clusters. This method is useful for identifying groups of genes that share similar expression profiles, which may indicate co-regulation or shared biological functions, and iDEP will perform enrichment analysis on each cluster to identify over-represented pathways or gene sets. Read more in the docs *(https://idepsite.wordpress.com/k-means-clustering/)*.

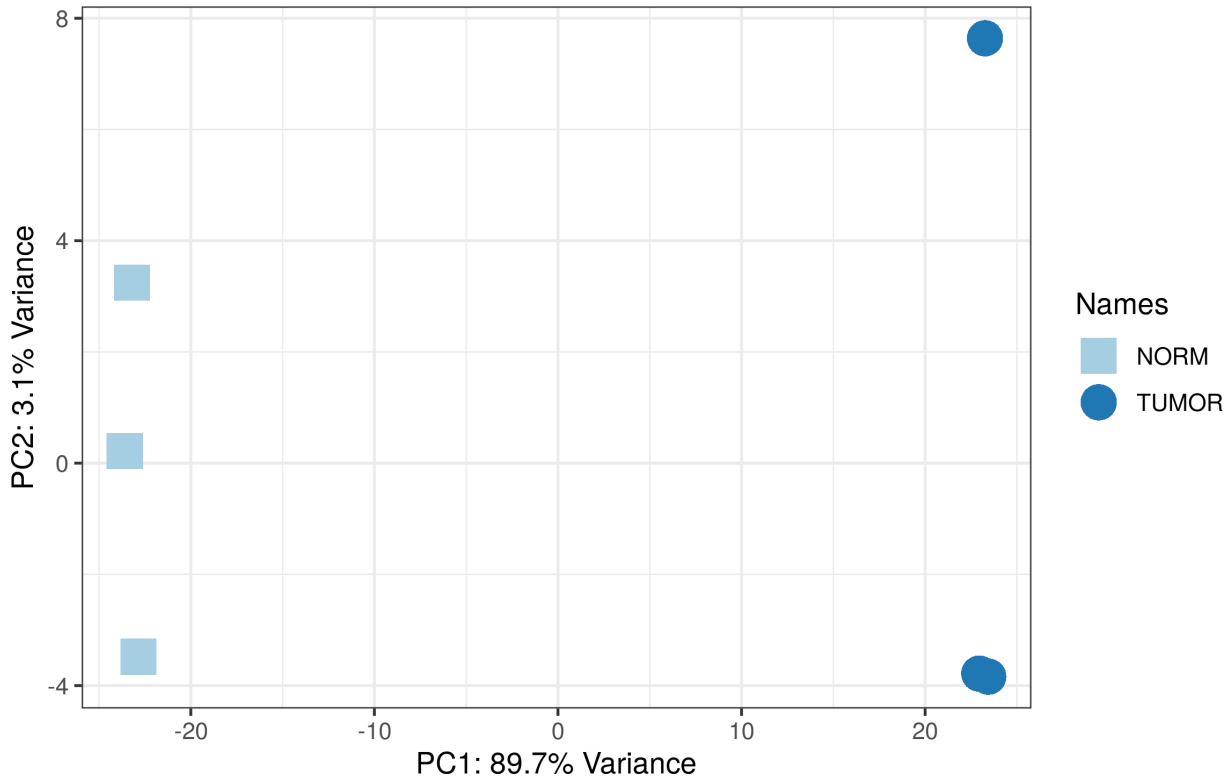
This analysis is beyond the scope of this lesson, but you can explore it on your own if you are interested. You can learn more about why this method is useful and how to interpret the results in the iDEP documentation *(https://idepsite.wordpress.com/k-means/)* and in this chapter *(https://link.springer.com/protocol/10.1007/978-1-0716-1307-8_22)*.

Clustering Methods

If you would like a deeper understanding of clustering methods, check out this ABCS Training on clustering: https://bioinfo-abcc.ncifcrf.gov/training/event/introduction-to-clustering-building-549-executive-board-room-nci-frederick-1103251200 *(https://bioinfo-abcc.ncifcrf.gov/training/event/introduction-to-clustering-building-549-executive-board-room-nci-frederick-1103251200)*.

PCA

PCA allows us to explore the overall structure of our dataset and see whether samples cluster as expected. PCA reduces the dimensionality of the data by finding the directions (principal components) that capture the most variance in the dataset. The first principal component (PC1) captures the most variance, followed by PC2, and so on. By plotting the samples in this reduced space, we can visualize how they relate to each other based on their expression profiles.



PCAtools provides us with some additional information including the PCA loadings and a scree plot. The PCA loadings tell us which genes contribute the most to each principal component, while the scree plot shows us how much variance is explained by each principal component.

MDS is a non-linear method available alongside PCA. It can serve as a useful cross-check against PCA, since a fairly different mathematical approach that still shows similar sample groupings adds confidence that the pattern is real. iDEP also offers t-SNE as another non-linear projection option, though it is not covered in the PCA documentation section itself. You can learn more about various dimensionality reduction methods in this ABCS training *(https://bioinfo-abcc.ncifcrf.gov/training/event/visualizing-high-dimensional-data-mds-pca-t-sne-and-umap)*.

What to look for in PCA and clustering?

- If Norm lanes cluster together and Tumor lanes cluster together, the global expression pattern supports the planned comparison.

- If a replicate is far from its group, pause before interpreting DEG results. Discuss sample swap, batch effect, low quality, or biological heterogeneity.

Batch Correction in RNA-Seq

Batch Correction in RNA-Seq

Batch effects are technical differences that can look like biology. Common batch variables include library preparation date, sequencing run, lane, technician, site, extraction batch, or patient/donor pairing.

We can address batch effects in two ways:

- **Adjusting the statistical model**: include a known batch variable in the DEG model, such as `~ Batch + Treatment`. This is generally preferred for differential expression because the raw counts still go into DESeq2, edgeR, or limma-voom, and the model estimates the treatment effect while accounting for batch.
- **Creating a corrected expression matrix**: remove or regress out batch signal from transformed/log expression values for visualization, clustering, PCA, or downstream exploratory work. These corrected matrices should be used cautiously and are usually not the input for count-based DEG testing.

In iDEP: known batch variables can be added to the design file and included as block factors in the Stats tab (e.g., `~ Batch + Treatment`). For methods like `ComBat`, `ComBat_seq`, `removeBatchEffect`, or `RUVSeq`, export the iDEP results/R code and continue in R.

Resources:

- iDEP Stats documentation on block factors and batch variables: https://bioinformatics.sdstate.edu/idep/
- DESeq2 vignette, design formulas and transformations: https://www.bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html
- limma `removeBatchEffect()` documentation: https://rdrr.io/bioc/limma/man/removeBatchEffect.html
- sva Bioconductor package, including surrogate variables and ComBat: https://www.bioconductor.org/packages/release/bioc/html/sva.html
- RUVSeq vignette: https://bioconductor.org/packages/release/bioc/vignettes/RUVSeq/inst/doc/RUVSeq.html

Run Differential Expression in DGE1

The DGE1 tab is where we define the comparison and run the differential expression analysis. iDEP supports three methods: DESeq2, limma-voom, and limma-trend (Read more here *(https://idepsite.wordpress.com/degs/)*).

For this demo, we will use DESeq2, since we are working with a small sample size and a simple two group comparison. It is also the recommended option by the iDEP developers, and includes more options for additional variables within iDEP. While DESeq2 is slower than limma-voom and has been shown to be more conservative in some cases, it is generally considered a robust choice for small-sample RNA-seq experiments.

iDEP recommendations for method selection

- DESeq2 (default): robust for most RNA-seq experiments and recommended whenever you have access to raw counts. Modelling counts directly gives the method full information about sequencing depth and dispersion.

- limma-voom: useful for larger studies or when mean–variance trends need extra modeling.
- limma-trend: recommended for already normalized expression data such as microarrays. Avoid it when library sizes differ by more than ~3×. - iDEP internal docs *(https://bioinformatics.sdstate.edu/idep/)*

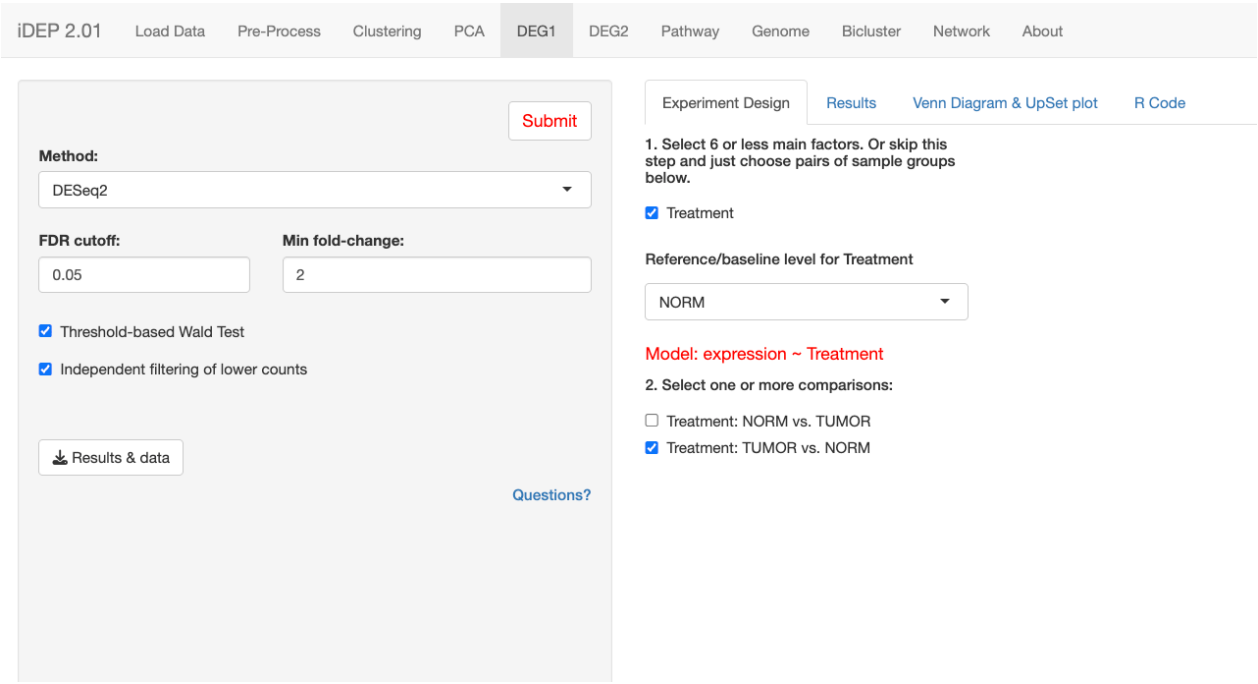
Pay close attention to the output script

There may be things that you assume are happening in the background that are not. For example, iDEP's default DESeq2 script does not apply a separate shrinkage step to the fold change estimates themselves (via lfcShrink), so the log2 fold changes reported here are the direct maximum likelihood estimates, not shrunken values. The shrunken values are particularly useful for smaller sample sizes.

Also, as I have already mentioned, iDEP offers less flexibility for all methods than you would have in R. Only two factors are allowed for limma-voom, whereas more complex designs can be handled in R. If you are working with a more complex design, you will need to export the data and run the analysis in R. It is possible that newer versions of iDEP have added more flexibility, so check the documentation for the version you are using.

Experimental Design

Your view in the experimental design section will depend on your design file. Again, for this demo, we are using a simple two-group comparison with no batch effects, so we will only have one factor, `Treatment`, and one contrast, `Tumor vs Norm`.



Recommended Settings for a simple two-group comparison in iDEP

Setting	Suggested value
Factor	`Treatment`
Comparison	`Tumor-Norm` or equivalent B-A contrast
Method	`DESeq2`

Setting	Suggested value
FDR cutoff	0.05
Fold-change cutoff	2, equivalent to abs(log2FC) >= 1

Comparisons

Choosing between norm vs tumor or tumor vs norm comes down to how you want to think and communicate about direction, up in tumor versus up in normal tissue, whichever framing matches the biological question you're asking. If you are looking for genes activated by tumor progression, TUMOR vs. NORM (positive values = upregulated in tumor) is probably the more intuitive framing to present in a paper or figure, even though NORM vs. TUMOR is technically valid and just mirrored.

Other Options

The following options are optional and result in a smaller and more trustworthy set of DEGs:

- Threshold-based Wald Test *(https://www.bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html#tests-of-log2-fold-change-above-or-below-a-threshold)*.
- Independent filtering of lower counts *(https://www.bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html#indfilttheory)*.

Threshold-based Wald test changes DESeq2's significance test so it asks whether a gene's fold change exceeds a chosen cutoff, rather than just whether it differs from zero, building your effect size requirement directly into the p-value. Independent filtering of lower counts removes genes with unreliably low expression before multiple testing correction is applied, which avoids wasting statistical power.

You can find more about the differential expression test settings in iDEP here: https://idepsite.wordpress.com/degs/ *(https://idepsite.wordpress.com/degs/)*. To better understand the model set up and experimental design, especially for more complex designs, see this helpful doc from iDEP: https://rpubs.com/ge600/deseq2 *(https://rpubs.com/ge600/deseq2)*.

DEG Results Walkthrough

The DEG results are provied in the DEG1 tab along with the R Code that was used. You should download all reports at each step in the iDEP workflow to ensure reproducibility. You will also find Venn diagrams and UpSet plots for visualizing the overlap of DEGs across multiple comparisons. *This is not as useful for our simple two-group comparison, but it is a nice feature for more complex designs.*

The DEG2 tab shows the results for all comparisons across different plots: heatmaps, volcano plots, and scatter plots. It also provides enrichment results, which we will discuss more in Lesson 11. Unfortunately, the DEG2 tab does not provide the R code used to generate the plots, so the plots have to be recreated in R if you want to reproduce / customize them. You can

find the R code used to generate these plots on Github: https://github.com/gexijin/idepGolem *(https://github.com/gexijin/idepGolem)*.

> **Data Export: DEG Visualizations**
>
> iDEP DOES provide the data and R code needed to recreate the DEG visualizations in newer versions of iDEP.

DEG Table

The output DEG table includes the following columns as well as the processed data. Note: the raw counts were used as input for the DESeq2 model, but the processed data includes the rlog transformed values for visualization and clustering.

Column	Meaning
symbol	Feature being tested
ensembl_ID	Ensembl gene ID for the feature being tested
User_ID	User-provided gene ID
log2 fold change for selected contrast	Estimated direction and magnitude
adjPval	Multiple-testing adjusted significance

symbol	ensembl_ID	User_ID	TUMOR-NORM_log2FC	TUMOR-NORM_adjPval	Processed data:	hcc1395_nor	hcc1395_nor	hcc1395_nor	hcc1395_tun	hcc1395_tun	hcc1395_tumor_rep3
TIMP3	ENSG000001	ENSG00000100234	9.348961356	1.53976941031754e-224		6.41296798	6.47281265	6.37422183	11.9817802	11.9685724	11.9802294
ATF4	ENSG000001	ENSG00000128272	1.900478793	1.21034253469066e-275		11.4498133	11.4982751	11.4718215	12.7774774	12.7851344	12.7805059
LGALS1	ENSG000001	ENSG00000100097	0.763593361	1		12.8991968	12.9230822	12.9059677	13.4147846	13.4455566	13.4370117
XBP1	ENSG000001	ENSG00000100219	1.886770923	7.26444351635543e-191		11.00027	10.9419215	10.9915705	12.2885937	12.2731735	12.2675452
RBFOX2	ENSG000001	ENSG00000100320	7.389438863	2.57089149326174e-279		6.57740231	6.50397915	6.5200166	11.2323728	11.2104037	11.2624371
RPL3	ENSG000001	ENSG00000100316	-0.631496116	1		14.5717748	14.5741946	14.5551422	14.1334967	14.1282479	14.1370727
MYH9	ENSG000001	ENSG00000100345	-0.703005406	1		14.2222822	14.2034387	14.1927362	13.7310292	13.7190355	13.7212925
IGLC1	ENSG000002	ENSG00000211675	-11.4487589	1.42535967660486e-57		11.0603412	11.0667111	11.0562963	4.95150056	4.86494429	5.16328136
XRCC6	ENSG000001	ENSG00000196419	0.479406347	1		12.4039565	12.4019085	12.3902036	12.7039657	12.7276041	12.7554272
RAC2	ENSG000001	ENSG00000128340	-7.145520356	0		11.144074	11.1500747	11.1302689	6.53477301	6.61939373	6.53193086
EIF3L	ENSG000001	ENSG00000100129	0.361925962	1		12.6391073	12.6500338	12.638198	12.905149	12.8911136	12.8809843
SPECC1L	ENSG000001	ENSG00000100014	1.680374034	9.67064906816197e-78		10.5067939	10.5649086	10.5068504	11.7135587	11.6547604	11.6845291
APOBEC3G	ENSG000002	ENSG00000239713	-9.941088382	1.42969652846885e-83		10.5877273	10.6037897	10.593906	4.86831768	4.78581147	5.00750751
MICALL1	ENSG000001	ENSG00000100139	2.049186302	2.9166803782132e-106		9.57143109	9.57941837	9.48645176	10.9686504	10.9526983	10.9496651
PARVB	ENSG000001	ENSG00000188677	-3.291986812	0		10.9655073	10.9876042	10.9420366	8.67471232	8.74825639	8.72926614
CELSR1	ENSG000000	ENSG00000075275	6.072904927	2.42434221882402e-192		6.31972754	5.95892385	6.25918905	10.1706432	10.1716857	10.1825927

You can also download a list of up and down regulated genes based on your FDR and fold-change cutoffs and your contrasts. This is useful for downstream analysis, such as pathway enrichment or gene set enrichment analysis.

Interpreting the DEG Tab Plots

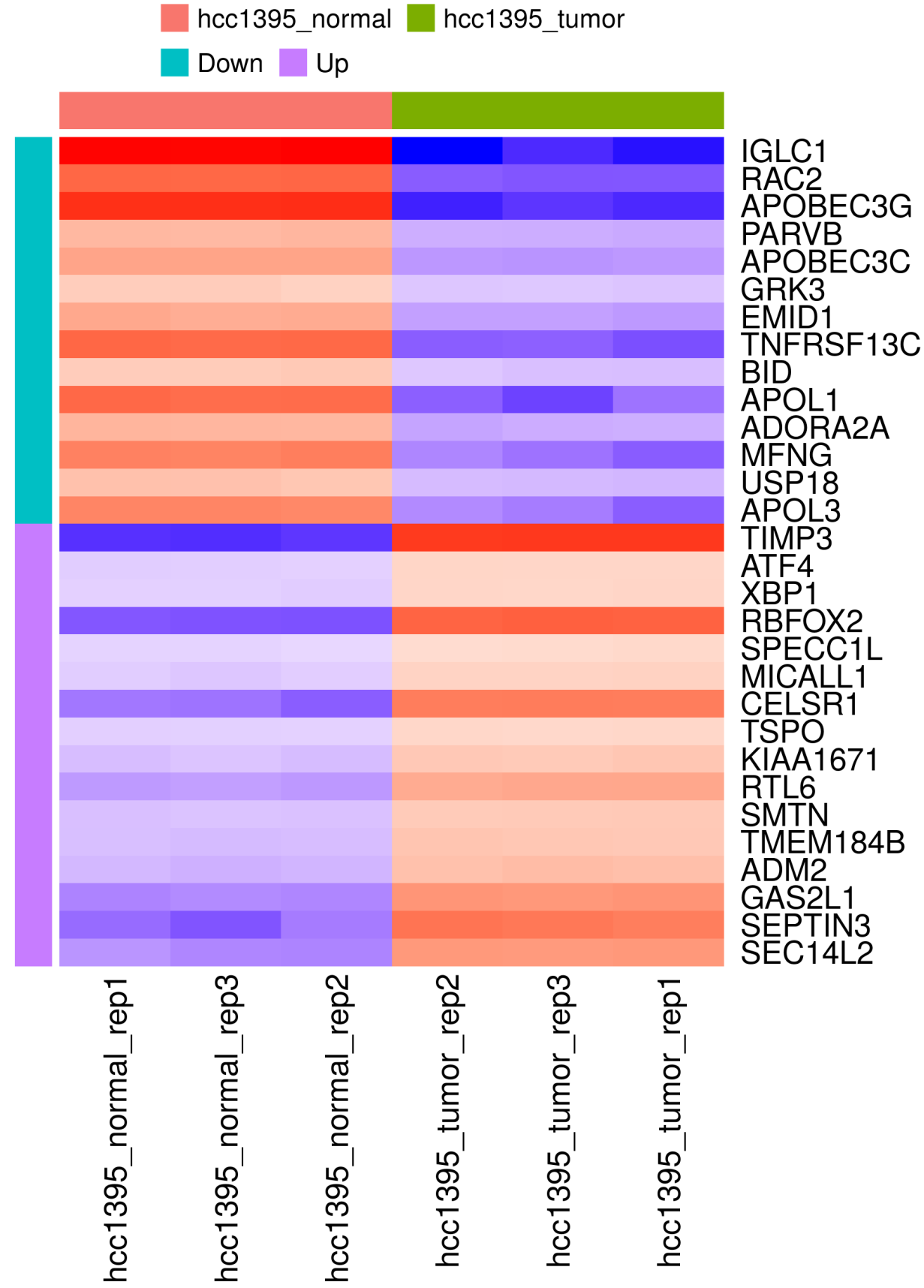
Heatmap

What it's for: Unlike a QC heatmap (which checks overall sample/replicate clustering), this heatmap visualizes your curated list of top DEGs, letting you confirm each hit looks consistent across replicates and spot genes with similar expression patterns. It's a visual companion to

your results table, not a data quality check. You choose how many genes to show and whether to sort by fold change or FDR.

What you see here: Normal and tumor samples cluster cleanly into two groups, with genes split into "Down" and "Up" blocks. There is a distinct lack of documentation regarding this plot and how it was constructed. It seems likely that the values are **mean-centered**, but there is no dendrogram, so the clustering method is unclear.

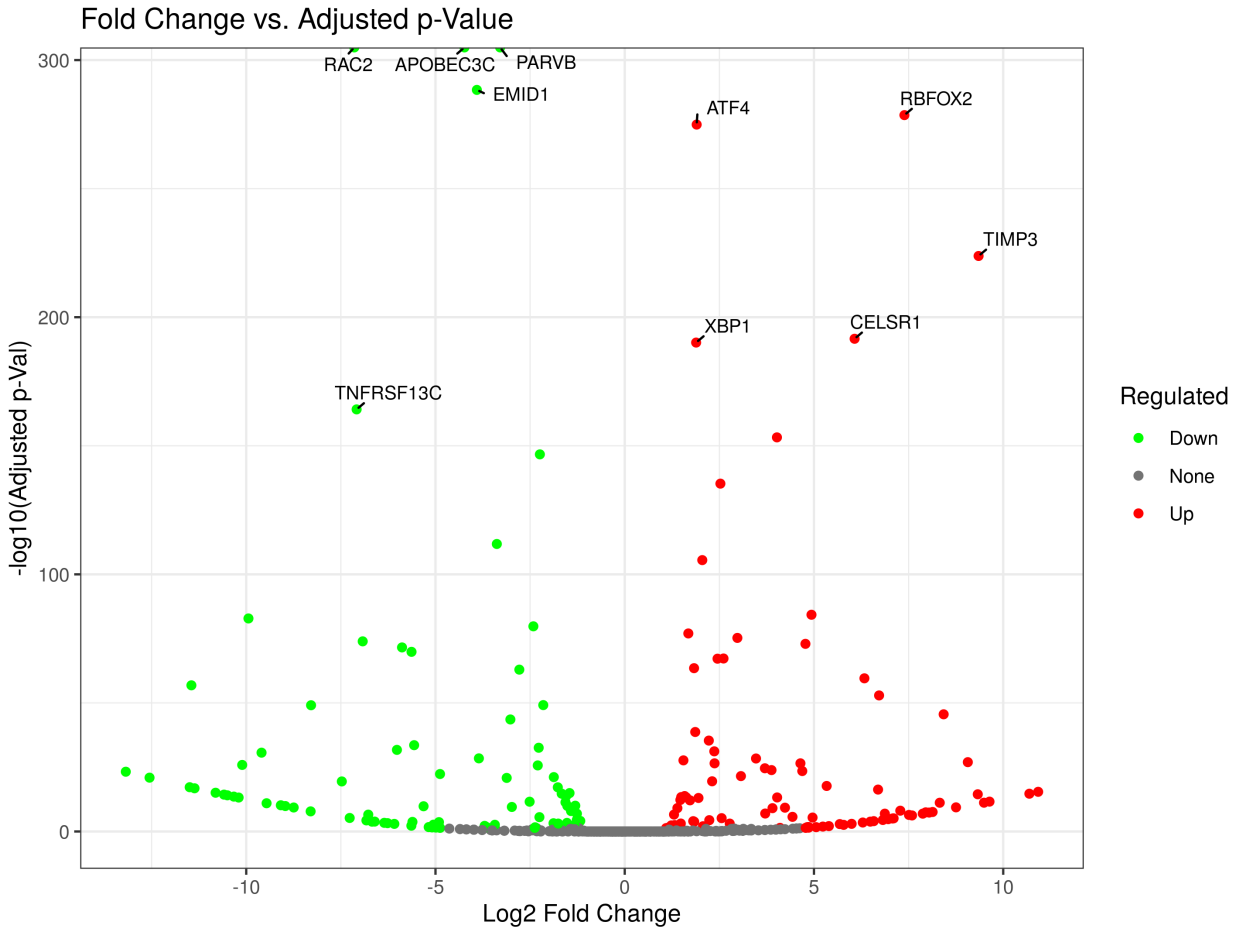
You can subset the plot and hover over points to see the gene ID, value, and whether it is up or down regulated. You can also download the heatmap as a PDF, PNG, or SVG.


hcc1395_normal
hcc1395_tumor
Down
Up
IGLC1
RAC2
APOBEC3G
PARVB
APOBEC3C
GRK3
EMID1
TNFRSF13C
BID
APOL1
ADORA2A
MFNG
USP18
APOL3
TIMP3
ATF4
XBP1
RBFOX2
SPECC1L
MICALL1
CELSR1
TSPO
KIAA1671
RTL6
SMTN
TMEM184B
ADM2
GAS2L1
SEPTIN3
SEC14L2
hcc1395_normal_rep1
hcc1395_normal_rep3
hcc1395_normal_rep2
hcc1395_tumor_rep2
hcc1395_tumor_rep3
hcc1395_tumor_rep1

Volcano plot

What it's for: Highlights genes with large fold change and strong statistical support (log2FC vs. -log10 adjusted p-value). The y-axis uses -log10(p) so smaller p-values plot higher (more intuitive) and tiny p-values get spread out instead of crushed near zero. Labeling controls let you tag specific genes.

What you see here: Red = Up, green = Down, gray = None, for the current contrast. Direction follows the "B-A" baseline convention, so Tumor-Norm means that the Up-regulated points are associated with Tumor samples. We are looking for genes with large fold change and strong significance (top corners of the plot). You can also download the volcano plot as a PDF, PNG, or SVG.



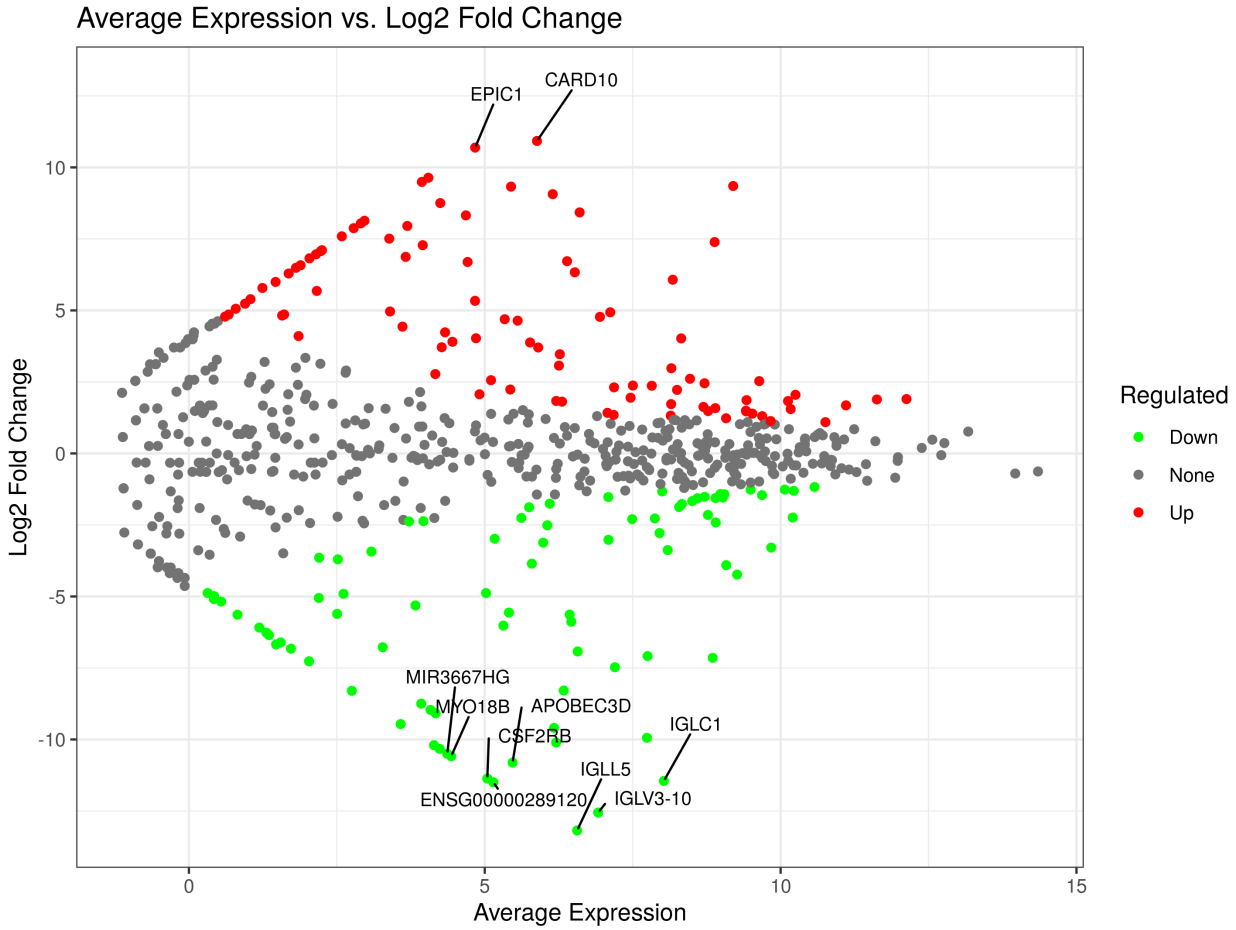
MA plot

An MA plot shows fold change vs. average expression across conditions, which can be useful for catching intensity dependent artifacts (unreliable fold changes at low expression).

What you see here: Gray points form a funnel, wider at low expression, narrower at high expression, as expected. Significant genes (red/green) mostly sit at low to moderate expression, which is typical and not a red flag on its own.

When to use it: Great for quality control, after normalization, most points should center around M = 0 (the M axis is the y-axis), since most genes aren't differentially expressed. It also reveals systematic bias. If genes seem to be almost entirely upregulated in one condition with no biological reason to expect that, suspect a technical issue rather than biology.

Resources: Check out this detailed explanation from Biostatsquid: https://biostatsquid.com/how-to-interpret-ma-plots/ *(https://biostatsquid.com/how-to-interpret-ma-plots/)*.



Scatter plot

What it's for: Compares average expression between the two conditions in your contrast (NORM vs. TUMOR), so you can see directly which condition a gene is higher in, something the MA plot can't tell you on its own, since its x-axis combines both conditions.

What you see here: Points above the diagonal are up-regulated in TUMOR, points below are up-regulated in NORM. Genes sitting near one axis (e.g., NORM near 0, TUMOR high) suggest a gene that's essentially off in one condition and on in the other, distinct pattern worth flagging separately from genes that are simply higher in one group. It's also a useful sanity check for your top MA/volcano plot hits, confirming the underlying values make sense rather than being driven by a dropout in one condition.

Note on the docs: newer iDEP documentation describes this scatter plot as comparing fold changes between two contrasts (e.g., Drug-Control vs. Drug2-Control) to reveal shared responses across treatments, useful when you have multiple contrasts and want to see which genes respond similarly (or oppositely) to two different conditions. That view isn't what's shown here since this dataset only has one contrast, but is something to be aware of.

Genome Tab

"The Genome tab shows where your fold-changes of genes along each chromosome. It also scans for genomic regions where many neighboring genes change together, highlighting potential co-regulation or shared copy-number events. The controls on the left let you choose the comparison, gene filters, and window settings used in these plots." - iDEP documentation: https://bioinformatics.sdstate.edu/idep/ *(https://bioinformatics.sdstate.edu/idep/)*

Reproducibility Checklist

Lastly, before ending any analysis session, make sure to download all the relevant files and reports for reproducibility or keep a record of any important parameters or details not provided. This includes:

- uploaded input file names;
- species and ID type;
- filtering settings;
- transformation settings;
- design factor and contrast;
- DEG method;
- FDR and fold-change thresholds;
- DEG table;
- key plots;
- iDEP report and R code from the Stats tab when available.

Resources Used

- iDEP main documentation site: https://idepsite.wordpress.com/
- iDEP current web application and embedded documentation, verified June 8, 2026: https://bioinformatics.sdstate.edu/idep/
- Ge SX, Son EW, Yao R. 2018. iDEP: an integrated web application for differential expression and pathway analysis of RNA-Seq data. BMC Bioinformatics. https://doi.org/10.1186/s12859-018-2486-6
- DESeq2 Bioconductor package and vignette: https://bioconductor.org/packages/DESeq2/
- Love MI, Huber W, Anders S. 2014. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biology. https://doi.org/10.1186/s13059-014-0550-8
- limma Bioconductor package and user guide (covers both limma-trend and limma-voom): https://bioconductor.org/packages/limma/
- Law CW, Chen Y, Shi W, Smyth GK. 2014. voom: precision weights unlock linear model analysis tools for RNA-seq read counts. Genome Biology. https://doi.org/10.1186/gb-2014-15-2-r29
- RNA-seq analysis is easy as 1-2-3 with limma, Glimma and edgeR (limma-voom workflow vignette). Bioconductor. https://www.bioconductor.org/packages/devel/workflows/vignettes/RNAseq123/inst/doc/limmaWorkflow.html

Introduction to Gene Ontology and Pathway Analysis

Objectives

1. Explain what functional enrichment analysis and pathway analysis are, describe how the two terms are used (often interchangeably, sometimes with a narrower distinction), and distinguish a GO term, a gene set, and a pathway from one another.
2. Describe the structure of the Gene Ontology, including its three main sub-ontologies (molecular function, cellular component, biological process).
3. Compare the four general approaches to functional enrichment and pathway analysis (over-representation analysis, functional class scoring, pathway topology, and single-sample gene set scoring) and identify which kind of question each approach is suited to answer.
4. Identify the major knowledgebases used in enrichment and pathway analysis (e.g., GO, KEGG, Reactome, DO, MSigDB, STRING) and explain what distinguishes a curated database from a meta-database or aggregator tool.
5. Explain what a background gene set is and why its choice matters for over-representation analysis, and describe the risks of testing a gene list against many databases or collections at once.

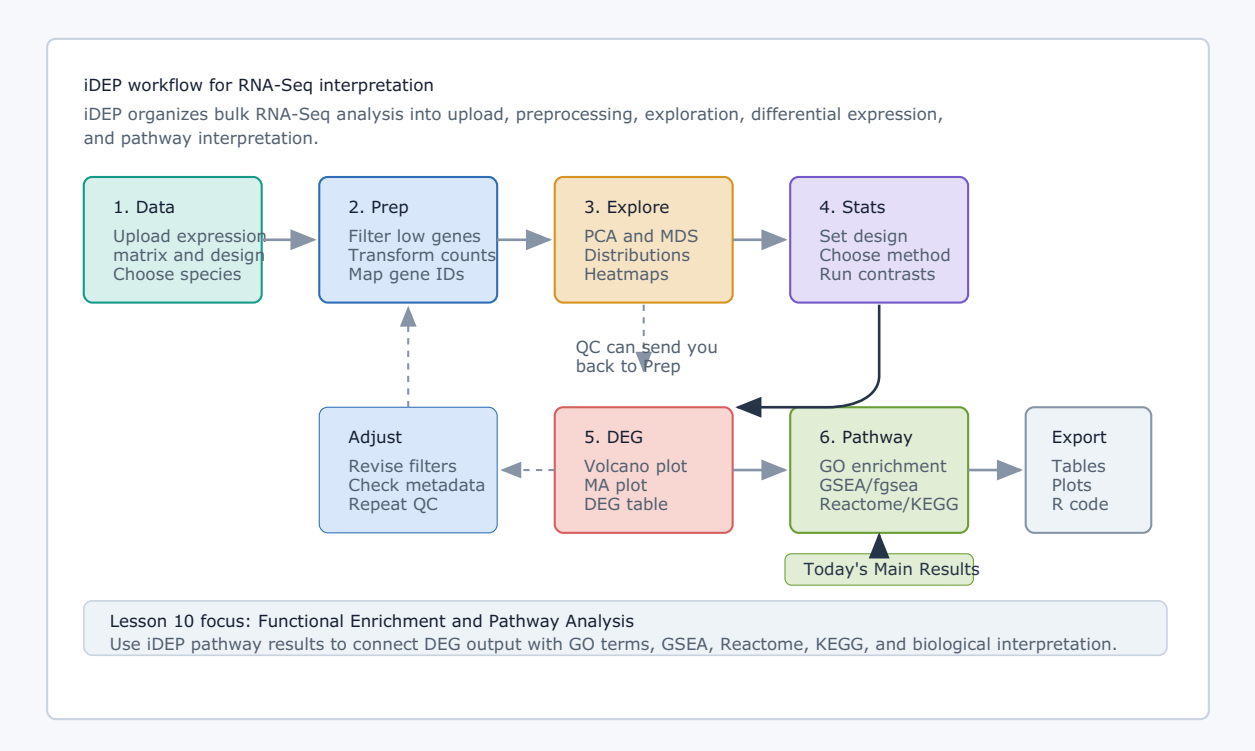
Where have we been and where are we going?

Thus far we have:

Downloaded raw RNA-Seq data (.fastq files), examined raw data quality, performed adapter and quality trimming, aligned the raw sequences to a reference genome, viewed and compared alignments, and generated a gene count matrix using CCBR's RENEE pipeline. We also performed differential expression analysis with iDEP.

You now have a potentially large list of differentially expressed genes. Now what? If you are like most biologists, you want to understand these genes within their biological context — what processes they participate in, where they act, and how they relate to one another.

To do that, we turn to gene ontology and functional enrichment or pathway analysis.



What do we mean by gene ontology, functional enrichment, and pathway analysis?

Before going further, it's worth defining these three terms at a high level. We'll get more precise about each one as the lesson goes on, but a rough map up front will make everything else easier to place.

Gene Ontology (GO) is a specific, structured knowledgebase: a controlled vocabulary of terms describing gene product function and the annotations linking real genes to those terms. It's one particular resource among several (KEGG and Reactome are others) that the methods below can be run against.

Functional enrichment analysis is the general statistical operation of asking whether a gene list shows more overlap with some functionally defined group of genes than you'd expect by chance. It's an umbrella term. It doesn't specify which knowledgebase or which statistical method you're using, only that you're testing for over-representation or coordinated behavior of some functional grouping within your data.

Pathway analysis is often used as a near-synonym for functional enrichment analysis, and in casual usage the two terms are largely interchangeable. You will see papers describe an analysis run entirely on GO terms as "pathway analysis," even though, strictly, a GO term isn't a pathway (more on that distinction below). Where a firmer line exists, "pathway analysis" more properly refers to enrichment testing against databases of actual pathways (KEGG, Reactome) rather than more general functional groupings like GO terms.

Gene sets, GO terms, and pathways — what's the difference?

These terms get used loosely and interchangeably in the literature, so it's worth pinning them down before going further.

A gene set is simply a collection of genes grouped by some shared property, as defined by a reference knowledge base. Gene sets are "formed on the basis of shared biological or functional properties as defined by a reference knowledge base," and knowledge bases are "database collections of molecular knowledge which may include molecular interactions, regulation, molecular product(s) and even phenotype associations" (Mathur et al. 2018) *(https://biodatamining.biomedcentral.com/articles/10.1186/s13040-018-0166-8)*. A gene set doesn't imply that its members physically interact. These can be arbitrary. For example, genes on the same chromosome arm, or genes that share a GO annotation, both count as gene sets.

A GO term is a single node in the Gene Ontology's controlled vocabulary (e.g., "DNA repair" (GO:0006281)). A GO term comes with an accession number, a definition, and a position in the ontology's hierarchy. The set of all genes annotated to a GO term (plus its descendants, since annotations propagate up the hierarchy) is one particular kind of gene set known as a "GO gene set." A few other things worth knowing about GO terms: the vocabulary itself is species-agnostic, though the annotations linking real genes to terms are species-specific; both the ontology and its annotations are updated on an ongoing basis; and terms are computer-readable, which is what makes large-scale automated enrichment analysis possible in the first place.

A pathway is more specific than a generic gene set: it is not just a list of genes but includes an *interaction component*, a description of how the members relate to one another mechanistically (e.g., a signaling cascade, a metabolic pathway with defined reaction steps and directionality). KEGG,Reactome, WikiPathways and PANTHER pathways are pathways in this stricter sense, while a GO Biological Process term, is treated like a pathway but is less so.

Quick Definitions.

GO term → a controlled-vocabulary label.
Gene set → any group of genes sharing a property (GO terms define one kind, but there are many others — e.g., co-expression clusters, chromosomal location, curated disease genes). These may or may not represent a pathway.
Pathway → a gene set with added structural/mechanistic information about how the members interact.

Which one your analysis returns depends on the database and method you choose, which will affect how much you can say about *mechanism* versus simple *association*.

Why do gene set / pathway analysis?

Gene set and pathway analysis help answer questions like:

- Are related genes changing together?
- Which biological processes are over-represented among your DE genes?
- Are subtle, coordinated changes visible even when no individual gene is significant on its own?
- Which hypotheses are worth prioritizing for follow-up experiments?

These methods are useful because they:

- Increase the statistical power of your analysis
- Ease interpretation by reducing dimensionality — genes → processes
- Predict new roles for genes based on the company they keep
- Better integrate data from different high-throughput methods (Khatri et al. 2012) *(https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1002375#pcbi-1002375-t001)* {target=_blank}

A caveat worth stating up front: **pathway analysis reduces complexity, but it doesn't prove causality or pathway activation on its own. Treat a hit as a lead, not a conclusion.** This is largely a hypothesis generating step, and follow-up experiments are needed to confirm any mechanistic claims.

What is gene ontology?

Many functional enrichment tools use GO terms, examining GO enrichment. What do we mean by GO?

> The Gene Ontology (GO) provides a framework and set of concepts for describing the functions of gene products from all organisms. — https://www.ebi.ac.uk/ols/ontologies/go *(https://www.ebi.ac.uk/ols/ontologies/go)*

GO is manually curated by the GO Consortium, a distributed group of biocurators, and is updated regularly.

There are two core parts to the gene ontology (check out this video *(https://www.youtube.com/watch?v=6Am2VMbyTm4)* for a more detailed overview):

1. **The ontology** — the GO terms and their hierarchical relationships, forming a directed, acyclic graph (nodes = GO terms, edges = relationships like "is a" or "part of").
2. **The annotations** — the links between real gene products and GO terms, each backed by a documented evidence code.

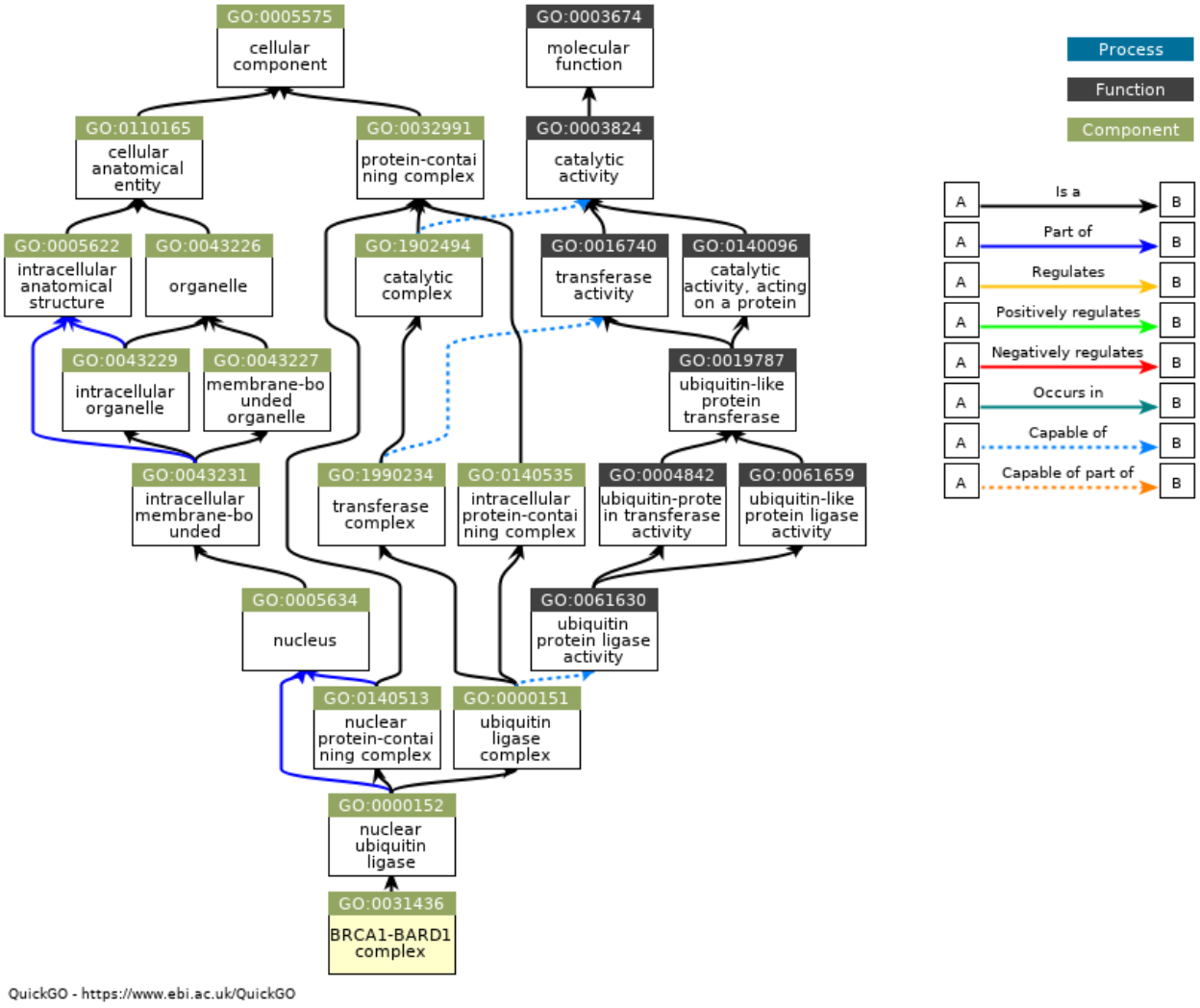


Image from https://www.ebi.ac.uk/QuickGO/GTerm?id=GO:0031436 *(https://www.ebi.ac.uk/QuickGO/GTerm?id=GO:0031436)*

GO integrates information about gene product function across three domains (the "three main ontologies"):

1. **Molecular function (MF)** — the molecular activity of a gene product (e.g., kinase activity)
2. **Cellular component (CC)** — where the gene product is active (e.g., mitochondrion)
3. **Biological process (BP)** — the larger pathway or process the gene product's activity contributes to (e.g., transport)

Biological Process is the domain most commonly used for pathway-style enrichment analysis, since it aligns most closely with the everyday sense of "pathway."

> **Note**
>
> A single gene product is very often annotated to multiple GO terms across all three domains, since one protein can have several functions, live in more than one location, and participate in multiple processes.

What is GO-CAM?

GO-CAM (Causal Activity Modeling) is a newer addition to the GO knowledgebase. Where a standard annotation links one gene product to one GO term, a GO-CAM model chains multiple annotations together using formally defined relations to represent how several gene products' activities causally connect into a larger biological process, essentially building a computable "pathway diagram" out of GO terms. This narrows the historical gap between "GO is a bag of terms" and "a pathway has real structure," and it's increasingly used as an input for enrichment and network analysis tools. You can browse existing models at geneontology.org *(https://geneontology.org)* under the GO-CAM section.

GO by the numbers

As of the most recent GO knowledgebase update, the ontology contains nearly 40,000 terms (biological process, molecular function, and cellular component combined), and the annotation corpus has grown to include experimental evidence drawn from over 180,000 published papers, represented as over 1,000,000 experimentally-supported annotations, plus several million more computationally or phylogenetically inferred annotations across thousands of species. See more stats here *(https://geneontology.org/stats.html)*.

What is GO Slim? Reducing semantic similarity

Pathway analysis results are often highly redundant with dozens of overlapping, near-synonymous terms coming back as "significant." Tools like DAVID's Functional Annotation Clustering, REVIGO, and GOSemSim reduce this redundancy computationally. Alternatively, you can work with **GO Slim** terms, which are simplified, broader subsets of the full ontology, for a high-level summary of function without wading through hundreds of overlapping child terms.

Check out geneontology.org *(https://geneontology.org/)* to explore GO directly, including the ontology browser, the annotation search, GO-CAM models, and tools/guides for enrichment analysis. The GO evidence codes page *(https://geneontology.org/docs/guide-go-evidence-codes/)* is also worth a look, since not all annotations carry the same weight; for example, an experimentally validated annotation (e.g., direct assay) is stronger evidence than one inferred computationally or by sequence similarity.

Approaches to functional enrichment / pathway analysis

Functional enrichment and pathway analysis are broad, loosely defined terms, and many people use them interchangeably. Khatri et al. 2012 *(https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1002375#pcbi-1002375-t001)* organized the field into three

general approaches, and we'll add a fourth that has become common enough since then to warrant its own category:

1. Over-Representation Analysis (ORA)
2. Functional Class Scoring (FCS)
3. Pathway Topology (PT)
4. Single-Sample Gene Set Scoring

Over-representation analysis (ORA)

> statistically evaluates the fraction of genes in a particular pathway found among the set of genes showing changes in expression — Khatri et al. 2012 *(https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1002375#pcbi-1002375-t001)*.

ORA asks: "Are there more genes from this pathway in my list than I'd expect by chance?"

Things to know:

- Uses tests based on the hypergeometric, Fisher's exact, chi-square, or binomial distribution to determine the probability that the overlap between your gene list and a gene set arose by chance.
- Assumes independence between genes — an assumption that rarely holds biologically, since genes in a pathway are, by definition, related.
- Requires an appropriate **background gene set** for comparison (see below).
- Uses an arbitrary, user-determined threshold (e.g., |fold change| > 2 and adjusted p < 0.05) to define the input list — a major source of variability between analyses.
- Doesn't require expression magnitude data, just gene identifiers — this also means it can be applied to gene lists from non-transcriptomic experiments.
- Performs best with gene lists larger than roughly 50 genes; in benchmarking comparisons, ORA methods have tended to show a higher false-positive rate than FCS methods. See Nguyen et al. 2019 *(https://link.springer.com/article/10.1186/s13059-019-1790-4)*.
- Example tools: DAVID *(https://bioinformatics.ccr.cancer.gov/docs/btep-coding-club/CC2024/DAVID/DAVID/)*, Qiagen IPA, g:Profiler, ToppGene, PANTHER, Enrichr.

To better understand the statstics behind ORA, see this guide *(https://www.pathwaycommons.org/guide/primers/statistics/fishers_exact_test/)* from Pathway Commons and this video *(https://www.youtube.com/watch?v=H1cUs6pql9s&list=PLRGSLMDpRxCrTybN8g8GjNk39lmzJ2UTF&index=2)* from BioStatsQuid.

Background gene sets

Every ORA method needs to compare your list of "hits" against a background, which includes the universe of genes that *could* have been called significant. Common choices include:

- All genes in the organism of interest
- All protein-coding genes
- Only the genes measured on the platform (e.g., all probes on a microarray)
- **Only the genes actually expressed in your RNA-Seq experiment *(https://genomebiology.biomedcentral.com/articles/10.1186/s13059-015-0761-7)*** — generally the recommended default for RNA-Seq
- Genes present in a specific gene set collection you're testing against

The background should represent "the universe of possible genes that could be called as significantly regulated in the experiment." *(https://link.springer.com/article/10.1186/s13059-015-0761-7)* Using the whole genome as background when you only measured, say, 12,000 expressed genes will systematically bias your p-values. This is one of the most common and avoidable errors in ORA.

Commercially licensed tools

QIAGEN Ingenuity Pathway Analysis (IPA) uses over-representation analysis, but its main selling point is a large, manually curated proprietary knowledge base built from the literature. Partek Flow and Qlucore are other commercial options that offer enrichment functionality, typically GSEA-based. See more about these licenses here *(https://bioinformatics.ccr.cancer.gov/btep/resources/scientific-software/)*.

Functional Class Scoring (FCS)

> Includes 'gene set scoring' methods such as GSEA, which first compute DE scores for all genes measured, and subsequently compute gene set scores by aggregating the scores of contained genes. — Geistlinger et al. 2021 *(https://academic.oup.com/bib/article/22/1/545/5722384?login=true)*

FCS methods are generally more sensitive than ORA, because they don't discard information by applying an arbitrary threshold, every measured gene contributes.

GSEA (Gene Set Enrichment Analysis) is the classic example:

- Does not pre-select genes; it considers the entire ranked list of gene expression changes. You need identifiers and a ranking metric (typically magnitude of change and significance) for every measured gene, not just the "significant" ones.
- Top of the ranking = upregulated + significant
- Bottom = downregulated + significant
- Middle = non-significant

- Asks whether members of a gene set cluster toward the top or bottom of that ranking, rather than being randomly scattered throughout.
- Builds a running-sum "enrichment score" as it walks down the ranked list, and evaluates significance using a permutation-based approach.
- Originally Broad Institute desktop software, but the same logic is implemented in R (`fgsea`, `clusterProfiler`), other web tools, and commercial platforms like Qlucore.
- FCS is best understood as a strategy encompassing many related methods (not just GSEA), which can further be split into **self-contained** methods (test whether a gene set is associated with a phenotype on its own) versus **competitive** methods (test whether a gene set is more associated with a phenotype than the rest of the genome is).

Check out this video overview of GSEA *(https://www.youtube.com/watch?v=egO7Lt92gDY)*, and the official GSEA User Guide *(https://docs.gsea-msigdb.org/#GSEA/GSEA_User_Guide/)* for a deep dive into reading enrichment plots (peak enrichment score, leading-edge subset, and the ranking-metric track). You may also be interested in this GSEA with R tutorial *(https://bioinformatics.ccr.cancer.gov/docs/btep-coding-club/CC2025/GSEA/GSEA/)* from a past BTEP Coding Club.

MSigDB and its relationship to GSEA

The Molecular Signatures Database (MSigDB), maintained by the Broad Institute, is the standard gene set collection paired with GSEA, though its gene sets are widely reused by other tools too.

- Covers both human and mouse.
- The human collection now contains roughly 35,000+ gene sets (this number grows with each release, so treat it as an order of magnitude rather than a fixed count).
- Organized into major thematic collections, including:
    - **H (Hallmark)** — a curated, low-redundancy summary of well-defined biological states; a strong default choice for many studies.
    - **C2** — curated gene sets from publications and pathway databases (including KEGG and Reactome).
    - **C5** — the Gene Ontology collection.
    - **C7** — immunologic signatures, useful for immunology-focused studies.
    - **C9** — a newer collection (added in early 2026) worth checking if your project touches its focus area; consult the MSigDB collections page *(https://www.gsea-msigdb.org/gsea/msigdb/collections.jsp)* for the current, versioned list, since collections continue to be added.

You can also build and submit your own custom gene sets for use with GSEA.

Pathway Topology (PT)

ORA and FCS both discard structural information. Even when the gene set in question represents a "real" pathway, these methods ignore how the genes interact, their position in the pathway, and their type (e.g., activator vs. inhibitor). Pathway Topology methods try to fix this by modeling the pathway's actual network structure.

Example — Impact analysis (iPathwayGuide):

> constructs a mathematical model that captures the entire topology of the pathway and uses it to calculate a perturbation for each gene. Then, these gene perturbations are combined into a total perturbation for the entire pathway and a p-value is calculated by comparing the observed value with what is expected by chance. - https://advaitabio.com/ipathwayguide/more-accurate-pathway-rankings-using-impact-analysis-instead-of-enrichment/ *(https://advaitabio.com/ipathwayguide/more-accurate-pathway-rankings-using-impact-analysis-instead-of-enrichment/)*

Other tools: Pathway-Express, SPIA, NetGSA, TPEA. See Nguyen et al. 2019 *(https://genomebiology.biomedcentral.com/articles/10.1186/s13059-019-1790-4)* for a review.

PT methods tend to be more mechanistically informative, but they "require experimental evidence for pathway structures and gene–gene interactions, which is largely unavailable for many organisms." *(https://www.sciencedirect.com/science/article/pii/S0168952523000185?via%3Dihub#bb0055)* This is also, in general, the most methodologically variable and least standardized of the three approaches. You should approach results from any single PT tool as one line of evidence, not a final answer.

Single-Sample Gene Set Scoring

ORA, FCS, and PT all answer a version of the same question: *is this pathway different between my groups?* Single-sample scoring methods ask a different question entirely: *how active is this pathway in this one sample?*

Rather than testing for a difference between conditions, these methods transform your expression matrix into a new matrix of scores (one score per sample, per gene set) describing where each individual sample falls on that pathway's activity spectrum, independent of any comparison group. This reframes "pathway activity" as a derived feature of each sample, much like a gene's expression value is a feature of that sample, except now the feature describes a whole pathway rather than one gene.

That reframing is what makes these methods useful in situations the other three categories aren't built for:

- Comparing pathway activity across **more than two groups**, or along a **continuous variable** (dose, time, disease severity), without needing a single pairwise contrast.
- Feeding pathway-level scores into **downstream analyses** — clustering samples, survival analysis, or machine learning models — the same way you would use gene-level expression values.
- Working with **single-cell data**, where "the sample" is a cell and you want a per-cell pathway activity score for visualization or clustering, rather than one enrichment statistic for an entire dataset.

Common methods include:

- **GSVA (Gene Set Variation Analysis)** — estimates each gene's relative expression rank within a sample, then aggregates within a gene set using a Kolmogorov–Smirnov-like running-sum statistic (conceptually related to GSEA, but computed per sample rather than across a ranked group comparison).
- **ssGSEA (single-sample GSEA)** — a direct per-sample adaptation of the original GSEA algorithm; computes an enrichment score for each sample/gene-set pair based on the rank of gene set members within that one sample's expression profile.
- **PLAGE (Pathway Level Analysis of Gene Expression)** — uses singular value decomposition on standardized expression data to derive a single representative "eigengene" score per pathway per sample.
- **Combined z-score / M-scores** — simpler, faster approaches that aggregate standardized (z-scored) expression values across a gene set's members.
- Newer, faster methods (e.g., **singscore**, **UCell**, **AUCell**, **PLAID**) have been developed largely to handle the scale of single-cell datasets, where GSVA and ssGSEA can become computationally impractical.

For a deeper dive, see the following:

- Toro-Domínguez D, Wang C, Ellson-Lancho I, et al. Benchmarking single-sample gene set scoring methods for application in precision medicine. *Briefings in Bioinformatics*. 2025;26(6):bbaf684. https://doi.org/10.1093/bib/bbaf684 *(https://doi.org/10.1093/bib/bbaf684)* (freely available via PMC *(https://pmc.ncbi.nlm.nih.gov/articles/PMC12710473/)*)

- GSVA package vignette from Bioconductor - https://bioconductor.org/packages/release/bioc/vignettes/GSVA/inst/doc/GSVA.html *(https://bioconductor.org/packages/release/bioc/vignettes/GSVA/inst/doc/GSVA.html)*

- Dave Tang's Blog on GSVA *(https://davetang.org/muse/2025/01/06/gene-set-variation-analysis/)*

What tools are available?

This is neither a comprehensive list of tools nor an endorsement of certain tools, but rather a list of semi-popular tools with different approaches.

Note

There are a ton of tools out there. Be aware of the background methods used and the quality of results returned. Also, check to see when the tool was last updated.

Gene set analysis tools

Tool	Author	Year	Citations[1]	Availability	Gene sets	Methods[2]
WEBGESTALT	Zhang *et al.* [73]	2005	1423	Web server	GO, KEGG, +20 more	ORA, GSEA
GOSTATS	Falcon and Gentleman [74]	2007	1437	R package	GO	ORA
G:PROFILER	Reimand *et al.* [75]	2007	534	Web server	GO, KEGG, +7 more	ORA
GENETRAIL	Backes *et al.* [76]	2007	360	Web server	GO, KEGG, +28 more	ORA, GSEA
DAVID	Huang *et al.* [8]	2009	19 569	Web server	GO, KEGG, +38 more	ORA
GORILLA	Eden *et al.* [77]	2009	1881	Web server	GO	ORA
TOPPGENE	Chen *et al.* [78]	2009	1200	Web server	GO, KEGG, +45 more	ORA
CLUSTER-PROFILER	Yu *et al.* [10]	2012	1305	R package	GO, KEGG, +8 more	ORA, GSEA
PANTHER	Mi *et al.* [79]	2013	1405	Web server	GO, +2 more	ORA, GSEA
ENRICHR	Chen *et al.* [9]	2013	1246	Web server	GO, KEGG, +33 more	ORA

[1]Google Scholar, July 2019.
[2]Detailed summary of implemented methods in Supplementary Methods S1.2.

Table from Geistlinger et al. 2021 *(https://academic.oup.com/bib/article/22/1/545/5722384)*

Other and related tools

- Gene Set Enrichment Analysis (GSEA) *(http://software.broadinstitute.org/gsea)*
- EnrichmentMap *(https://apps.cytoscape.org/apps/enrichmentmap)*
- REVIGO *(http://revigo.irb.hr/)* (reducing and visualizing gene ontology)
- Pathview *(https://pathview.uncc.edu/)*
- iPathwayGuide *(https://advaitabio.com/ipathwayguide/)* (proprietary)
- Qiagen IPA *(https://btep.ccr.cancer.gov/docs/resources-for-bioinformatics/software/ipa/)* (proprietary, CCR license)
- Qlucore *(https://btep.ccr.cancer.gov/docs/resources-for-bioinformatics/software/qlucore/)* (proprietary, CCR license)
- GeneMANIA *(https://apps.cytoscape.org/apps/genemania)*
- CellNetAnalyzer *(http://www2.mpi-magdeburg.mpg.de/projects/cna/cna.html)*

- PARADIGM *(http://paradigm.five3genomics.com)*

BTEP Tutorials on Specific Tools

- GSEA (via ClusterProfiler) - https://bioinformatics.ccr.cancer.gov/docs/btep-coding-club/CC2025/GSEA/GSEA/ *(https://bioinformatics.ccr.cancer.gov/docs/btep-coding-club/CC2025/GSEA/GSEA/)*
- ClusterProfiler - https://bioinformatics.ccr.cancer.gov/docs/btep-coding-club/CC2023/FunctionalEnrich_clusterProfiler/ *(https://bioinformatics.ccr.cancer.gov/docs/btep-coding-club/CC2023/FunctionalEnrich_clusterProfiler/)*
- DAVID - https://bioinformatics.ccr.cancer.gov/docs/btep-coding-club/CC2024/DAVID/DAVID/ *(https://bioinformatics.ccr.cancer.gov/docs/btep-coding-club/CC2024/DAVID/DAVID/)*

Databases used by enrichment and pathway tools

There are many databases devoted to relating genes and gene products to pathways, processes, and other phenomenon. Again, the following is not meant to be a comprehensive list.

Gene Ontology (GO) *(https://geneontology.org/)*

Covered in detail above — the controlled vocabulary and annotation source behind most "GO enrichment" results.

Kyoto Encyclopedia of Genes and Genomes (KEGG) *(https://www.genome.jp/kegg/)*

- Curated database of biological pathways and molecular interaction networks.
- Well-known for its detailed, manually drawn pathway maps: metabolic pathways, disease pathways, drug-target interactions, and more.
- Emphasizes system-level integration of genomic and chemical information.
- Free for academic use through the KEGG API/FTP is limited; the full, most current dataset requires a subscription license (this has been the case for some time and is worth double-checking against KEGG's current terms before building a pipeline around it).

Reactome *(https://reactome.org/)*

> The Reactome Knowledgebase systematically links human proteins to their molecular functions, providing a resource that functions both as an archive of biological processes and as a tool for discovering novel functional relationships in data such as gene expression studies or catalogs of somatic mutations in tumor

cells. — Jassal et al. 2019 *(https://academic.oup.com/nar/article/48/D1/D498/5613674)*

- Curated database covering metabolism, signaling, and other processes; human-specific (though includes some inferred orthologous data for other species).
- Includes disease super-pathways.
- Comes with several built-in analysis tools directly on reactome.org (pathway browser, over-representation and expression analysis tools, and, as of recently, an AI-assisted chat interface for navigating the database).

Pathway Commons *(https://www.pathwaycommons.org/)*

A meta-database that aggregates and standardizes pathway data from many other pathway databases into one queryable format.

PANTHER *(https://pantherdb.org/)*

> The PANTHER (Protein ANalysis THrough Evolutionary Relationships) Classification System was designed to classify proteins (and their genes) in order to facilitate high-throughput analysis. The core of PANTHER is a comprehensive, annotated "library" of gene family phylogenetic trees. — pantherdb.org *(http://pantherdb.org/about.jsp)*

Especially useful when evolutionary relationships between gene families matter to your question. PANTHER also powers the enrichment tool built directly into the GO Consortium's own website.

STRING *(https://string-db.org/cgi/help?sessionId=b89rCT8VxMBR)*

A database of known and predicted protein-protein interactions, combining evidence from experiments, curated pathway databases, co-expression, and text-mining into a single confidence-scored network. STRING is one of the most widely used resources for visualizing how a gene list's products physically or functionally connect. This database is often useful for Pathway Topology methods.

Enrichr *(https://maayanlab.cloud/Enrichr/#libraries)*

Not a single curated knowledgebase but a meta-tool: a fast, web-based ORA interface that lets you test a gene list against 200+ separate gene set libraries in one submission — GO, KEGG, ChEA (transcription factor targets), disease and drug-perturbation signatures, cell-type markers, and more. Its breadth is exactly what makes it popular for quick exploration, and exactly why it deserves the caution described in the callout below: testing against that many libraries at once multiplies your chances of a "significant" hit by chance alone.

WikiPathways *(https://www.wikipathways.org/)*

A community-curated, wiki-style meta-database of pathways. WikiPathways is a good source for newer or less-established pathways that haven't yet made it into more conservative curated databases like KEGG or Reactome.

NDEx *(https://www.ndexbio.org/)*

> The NDEx Project provides an open-source framework where scientists and organizations can store, share, manipulate, and publish biological network knowledge. — ndexbio.org

HumanCyc *(https://humancyc.org/)* (See BioCyc *(https://biocyc.org/?sid=biocyc17-3995986138onboard/onboard.shtml)*)

> HumanCyc provides an encyclopedic reference on human metabolic pathways, the human genome, and human metabolites. — humancyc.org

Free BioCyc accounts allow browsing, but bulk data access, APIs, and the desktop Pathway Tools software require a paid BioCyc subscription (pricing starts in the low thousands of dollars per year) — check current terms before designing a pipeline around it.

Disease Ontology (DO) *(https://disease-ontology.org/)*

A structured, controlled vocabulary for human disease terms, designed to standardize disease naming and support integration across genomic and clinical datasets. DO is used by disease-enrichment tools (e.g., the `DOSE` R package) the same way GO is used for functional enrichment. It lets you ask "are my differentially expressed genes enriched for association with any particular disease?" It is often paired with gene-disease association resources.

DisGeNET *(https://disgenet.com/)*

A knowledgebase of gene–disease associations, aggregating curated repositories, GWAS catalogs, animal model data, and literature-derived associations. Worth flagging: DisGeNET has moved to a tiered access model — a free academic/non-commercial plan is available, but higher tiers (with text-mined associations, drug/chemical annotations, and full API access) require a paid license. Confirm your access level before building a pipeline that depends on it.

Check Versions for tools and databases

- **Not all databases survive.** Check when a database was last updated before relying on it. Pathguide *(https://pathguide.org/)*, a once-popular searchable index of pathway databases, is a good illustration of the point it exists to make: it hasn't been substantially updated since 2017, so treat any list it returns as a starting point for further checking, not a current recommendation.

- **Not all tools stay maintained.** An unmaintained tool is quietly running on stale annotation versions, even if its web interface still loads. When in doubt, check the tool's own changelog or publication history.

The Multiple Testing Trap

Modern tools make it easy to test one gene list against many collections at once (GO, KEGG, Reactome, MSigDB collections, disease/drug libraries), sometimes in a single click. The statistical catch is multiplicity: as the number of tests increases, the chance of at least one false-positive finding also increases.

Most enrichment workflows report adjusted p-values (often BH/FDR) for the set of hypotheses tested *within that run or collection*. If you run many collections and then focus on only one favorable result, your overall error control is weaker than those per-collection adjusted values may suggest.

A second risk is selective reporting: trying many databases and highlighting only the one that supports your hypothesis (even unintentionally). More attempts create more opportunities for a spurious but convenient hit.

Good practice helps: - Pre-specify which databases/collections you will test (confirmatory set) before seeing results. - Report all tested collections, not only the ones with significant hits. - Treat broad multi-database scans as exploratory, and label them clearly as such. - Give more weight to signals that replicate across independent resources and analysis methods.

Importance of Gene IDs

To run any of these tools, you need a list of annotated genes, and gene ID systems don't all agree with each other. GO annotations, for instance, are commonly cross-referenced to Entrez, Ensembl, and official HGNC gene symbols, but converting cleanly between systems is not always one-to-one.

A few practical notes:

- **Genome builds matter.** Coordinates and even gene models can differ between reference builds, which affects ID mapping. See this tutorial *(https://github.com/hbctraining/Training-modules/blob/master/DGE-functional-analysis/lessons/Genomic_annotations.md)* from the Harvard Chan Bioinformatics Core for more detail.
- **Be careful with Excel.** Spreadsheet software has a well-documented history of silently "autocorrecting" gene symbols into dates (e.g., turning "SEPT1" into a date). This has caused real errors and even paper corrections in the literature. Avoid opening raw gene ID lists in Excel without disabling autoformatting, or better, avoid Excel for this step entirely.
- **Double-check gene identity**, not just the ID string. Confusing nomenclature and symbol reuse across species or gene families has led to retractions. Make sure the gene you're discussing in a paper is actually the gene you think it is.

Tools to help with ID conversion:

- g:Convert *(https://biit.cs.ut.ee/gprofiler/convert)*
- Ensembl BioMart.

- AnnotationHub *(https://bioconductor.org/packages/release/bioc/html/AnnotationHub.html)* (Bioconductor)
- DAVID Gene ID Conversion *(https://davidbioinformatics.nih.gov/conversion.jsp)*

Other considerations

- Describe your method(s) clearly in any resulting publication, including exact parameters, software versions, and the release date/version of any database used (e.g., "GO release 2026-XX-XX," "MSigDB v2026.1").
- For ORA, choose an appropriate, defensible background gene set and state it explicitly.
- Correct p-values for multiple comparisons whenever testing more than one gene set, to control the false discovery rate, and remember this correction generally does not extend across separate databases (see the callout above).
- Many classic enrichment methods were built specifically for transcriptomic data. Other -omics types (proteomics, metabolomics, single-cell RNA-seq, GWAS) have different statistical properties, and using a transcriptomics-native tool on them without adjustment can produce misleading results. See Zhao and Rhee 2023 *(https://www.sciencedirect.com/science/article/pii/S0168952523000185?via%3Dihub#bb0055)* for a review of methods suited to specific -omics types.
- Emerging area to watch: LLM-assisted interpretation of enrichment results (several databases, including Reactome, have begun adding AI chat interfaces, and DisGeNET now offers a natural-language query assistant). These can speed up literature triage but don't replace checking the underlying statistics and evidence codes yourself.

Functional enrichment and pathway analysis in iDEP

In the next lesson, you will explore functional enrichment and pathway analysis with **iDEP**, a web-based tool that wraps many of the R packages mentioned above behind a point-and-click interface. As you work through it, it's worth knowing which of our four categories — ORA, FCS, PT, or single-sample scoring — each iDEP option falls into, since that tells you what kind of input it expects and how to interpret its output.

iDEP tool	Category
Functional Enrichment tab	ORA
GAGE	FCS (fold-change/t-test based)
GSEA (fgsea)	FCS (rank-based)
PGSEA	FCS (parametric)
ReactomePA	FCS (Reactome-scoped)
GSVA / ssGSEA / PLAGE	Single-Sample Gene Set Scoring

iDEP tool	Category
Pathview	Visualization (not a statistical method)

References / Resources

- Mathur et al., "Gene set analysis methods: a systematic comparison," *BioData Mining*, 2018 — https://biodatamining.biomedcentral.com/articles/10.1186/s13040-018-0166-8 *(https://biodatamining.biomedcentral.com/articles/10.1186/s13040-018-0166-8)*
- Gene Ontology definition, EBI Ontology Lookup Service — https://www.ebi.ac.uk/ols/ontologies/go *(https://www.ebi.ac.uk/ols/ontologies/go)*
- GO overview video — https://www.youtube.com/watch?v=6Am2VMbyTm4 *(https://www.youtube.com/watch?v=6Am2VMbyTm4)*
- ClusterProfiler tutorial *(https://bioinformatics.ccr.cancer.gov/docs/btep-coding-club/CC2023/FunctionalEnrich_clusterProfiler/)*
- Functional enrichment and comparison with R *(https://alexslemonade.github.io/refinebio-examples/03-rnaseq/pathway-analysis_rnaseq_01_ora.html)*
- ClusterProfiler, pathview, and good introductory information *(https://github.com/hbctraining/Training-modules/blob/master/DGE-functional-analysis/lessons/functional_analysis_2019.md)*
- Article on the impact of the evolving GO *(https://www.nature.com/articles/s41598-018-23395-2)*
- Ten Years of Pathway Analysis: Current Approaches and Outstanding Challenges, PLOS Computational Biology, 2012 *(https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1002375)*
- Nguyen et al., "A comprehensive survey of tools and software for active subnetwork identification," reviewed in the context of pathway topology methods and ORA benchmarking, *Genome Biology*, 2019 — https://genomebiology.biomedcentral.com/articles/10.1186/s13059-019-1790-4 *(https://genomebiology.biomedcentral.com/articles/10.1186/s13059-019-1790-4)*
- Pathway Commons guide to Fisher's exact test — https://www.pathwaycommons.org/guide/primers/statistics/fishers_exact_test/ *(https://www.pathwaycommons.org/guide/primers/statistics/fishers_exact_test/)*
- BioStatsQuid, video overview of Fisher's exact test and ORA statistics — https://www.youtube.com/watch?v=H1cUs6pql9s&list=PLRGSLMDpRxCrTybN8g8GjNk39lmzJ2UTF&index=2 *(https://www.youtube.com/watch?v=H1cUs6pql9s&list=PLRGSLMDpRxCrTybN8g8GjNk39lmzJ2UTF&index=2)*
- Timmons, J.A., Szkop, K.J. & Gallagher, I.J. Multiple sources of bias confound functional enrichment analysis of global -omics data. Genome Biol 16, 186 (2015). https://doi.org/10.1186/s13059-015-0761-7 *(https://doi.org/10.1186/s13059-015-0761-7)*
- Reimand, J., Isserlin, R., Voisin, V. et al. Pathway enrichment analysis and visualization of omics data using g:Profiler, GSEA, Cytoscape and EnrichmentMap. Nat Protoc 14, 482–

517 (2019). https://doi.org/10.1038/s41596-018-0103-9 *(https://doi.org/10.1038/s41596-018-0103-9)*

- GSEA overview video — https://www.youtube.com/watch?v=egO7Lt92gDY *(https://www.youtube.com/watch?v=egO7Lt92gDY)*
- Official GSEA User Guide — https://docs.gsea-msigdb.org/#GSEA/GSEA_User_Guide/ *(https://docs.gsea-msigdb.org/#GSEA/GSEA_User_Guide/)*
- Toward a gold standard for benchmarking gene set enrichment analysis, Briefings in Bioinformatics, 2021 *(https://academic.oup.com/bib/article/22/1/545/5722384)* (also the source of the functional enrichment tools comparison table, sometimes cited by its 2020 preprint/online-first date)
- Zhao and Rhee, "Interpreting omics data with pathway enrichment analysis," *Trends in Genetics*, 2023 — https://www.sciencedirect.com/science/article/pii/S0168952523000185 *(https://www.sciencedirect.com/science/article/pii/S0168952523000185)*
- Introductory lectures on functional enrichment and R from DIY Transcriptomics *(https://diytranscriptomics.com/project/lecture-10)*
- The Gene Ontology Consortium. The Gene Ontology knowledgebase in 2026. Nucleic Acids Res. 2025 Dec 18;gkaf1292. DOI: https://doi.org/10.1093/nar/gkaf1292 *(https://doi.org/10.1093/nar/gkaf1292)*
- Brad T Sherman, Ganesh Panzade, Thoai Dotrang, Ming Hao, Lei Xu, Xuan Li, Michael W Baseler, H Clifford Lane, Tomozumi Imamichi, Weizhong Chang, DAVID: a web server for functional annotation and functional enrichment analysis of gene lists (2025 update), Nucleic Acids Research, Volume 54, Issue W1, 7 July 2026, Pages W367–W373, https://doi.org/10.1093/nar/gkag470 *(https://doi.org/10.1093/nar/gkag470)*
- Damian Szklarczyk, Katerina Nastou, Mikaela Koutrouli, Rebecca Kirsch, Farrokh Mehryary, Radja Hachilif, Dewei Hu, Matteo E Peluso, Qingyao Huang, Tao Fang, Nadezhda T Doncheva, Sampo Pyysalo, Peer Bork, Lars J Jensen, Christian von Mering, The STRING database in 2025: protein networks with directionality of regulation, Nucleic Acids Research, Volume 53, Issue D1, 6 January 2025, Pages D730–D737, https://doi.org/10.1093/nar/gkae1113 *(https://doi.org/10.1093/nar/gkae1113)*
- Toro-Domínguez et al., Benchmarking single-sample gene set scoring methods for application in precision medicine, *Briefings in Bioinformatics*, 2025 — https://doi.org/10.1093/bib/bbaf684 *(https://doi.org/10.1093/bib/bbaf684)*
- GSVA package vignette, Bioconductor — https://bioconductor.org/packages/release/bioc/vignettes/GSVA/inst/doc/GSVA.html *(https://bioconductor.org/packages/release/bioc/vignettes/GSVA/inst/doc/GSVA.html)*
- Dave Tang's Blog on GSVA *(https://davetang.org/muse/2025/01/06/gene-set-variation-analysis/)*
- BTEP: Pathways and gene sets — what is functional enrichment analysis? *(https://bioinformatics.ccr.cancer.gov/btep/pathways-and-gene-sets-what-is-functional-enrichment-analysis/)*

Lesson 11: Pathway Analysis and Pre-ranked GSEA with iDEP

Learning Objectives

By the end of this lesson, learners should be able to:

- Identify the pathway analysis methods available in iDEP and the general category (ORA, FCS, or single-sample scoring) each belongs to;
- Run DEG functional enrichment and Pathway tab analyses in iDEP, and navigate the resulting tables, trees, networks, heatmaps, and KEGG diagrams;
- Accurately interpret enrichment and pathway analysis results, including common pitfalls in iDEP's output columns, without overstating biological significance;
- Evaluate when iDEP's tools are sufficient for a given question versus when to use another tool.

Version Differences with iDEP

While Biowulf is restricted to iDEP v2.01, the current iDEP web application is v2.5.0. The latest version includes many UI improvements that make it easier to run, explore, and interpret pathway results. I recommend the latest version of iDEP for the most up-to-date features and user experience. The Biowulf version is still functional and will be used for this lesson, but the screenshots and instructions may differ slightly from the current web application. If you are using the current iDEP web application, please refer to the official documentation for any differences in workflow or output.

What Functional Enrichment / Pathway Analysis Is Available in iDEP?

iDEP offers both Over-Representation Analysis (ORA) and Functional Class Scoring (FCS) methods, as well as single-sample scoring and visualization. Here's a quick overview of the main options:

- Functional Enrichment - iDEP uses functional enrichment to perform ORA on a list of significant DE genes. It counts overlaps with gene sets from GO, KEGG, and other databases, and tests for over-representation against a background via hypergeometric distribution *(https://www.pathwaycommons.org/guide/primers/statistics/distributions/#hypergeometric)*. "For the selected contrast, you can run GO, pathway, and TF/miRNA target enrichment" (iDEP Internal Documentation). Functional enrichment analysis is also available post clustering with k-means (See the Cluster tab) or biclustering (See the Bicluster tab). Note: the recommended sample size for biclustering is 15 or more samples and more than 2 experimental conditions.

- Pathway Analysis - iDEP offers several FCS methods or single-sample scoring methods that use the full ranked list of genes or the full expression matrix to test whether a pathway's genes show coordinated shifts. These include GAGE, GSEA/fgsea, PGSEA, ReactomePA, GSVA, ssGSEA, and PLAGE.

    - **GAGE (FCS method)** - GAGE takes fold-change values (or expression values) for all genes, not just a thresholded list, and uses a two-sample t-test-style comparison to ask whether a pathway's genes show a coordinated shift. It can test up- and down-regulated genes separately, which is why iDEP lets you filter by direction on the Network tab. It is conceptually closer to PAGE than to rank-based GSEA. *Asks the question: "Averaged across my two groups, did this pathway's genes shift together more than we'd expect by chance?"*

    - **GSEA (fgsea, FCS method) — iDEP's fgsea implementation orders all genes by fold change and asks whether a pathway's members cluster toward either end of that ranking, using the same running-sum/permutation logic as the original GSEA software, just a faster implementation.** *Asks the question: "Are this pathway's genes enriched at the top or bottom of my ranked list?"*

    - **PGSEA (FCS method)** - PGSEA computes a per-sample (or per-comparison) summary statistic for each pathway rather than pooling all samples into one group-level test, which is why "PGSEA w/ all samples" gives you a full sample × pathway matrix suitable for clustering. It's parametric (t-test/z-score based, in the same family as PAGE) rather than permutation-based like GSEA. *Asks the question: "In this sample, is this pathway shifted up or down relative to the average sample in my dataset?"*

    - **ReactomePA (FCS method, rank-based GSEA)** - functional enrichment scoped specifically to Reactome's pathway hierarchy rather than a generic gene set collection. iDEP runs it via ReactomePA's `gsePathway`, which ranks all genes by fold change and applies the same permutation-based running-sum logic as fgsea. The pathway relationship plots it produces are a good way to see how Reactome's mechanistic pathway structure (as opposed to GO's simpler term hierarchy) groups related processes. *Asks the question: "Within Reactome's curated pathway hierarchy, do the genes in this pathway pile up at one end of my fold-change ranking?"*

    - **GSVA / ssGSEA / PLAGE (Single-Sample Gene Set Scoring)** - Instead of testing whether a pathway is enriched in one comparison (treatment vs. control), these methods first transform your entire expression matrix into a new matrix of per-sample pathway scores — one score per sample, per gene set. That's exactly why they're well suited to comparisons across multiple groups or timepoints. iDEP then runs a per-pathway ANOVA on these scores across your groups (equivalent to a t-test for two groups), FDR-corrects the p-values, and displays the top significant pathways as a heatmap of scores. Note: GSVA-based results are considered

unreliable with 10 or fewer total samples (at least 6 samples are recommended by iDEP). *Asks the question: "What is this sample's own activity level for this pathway, and does that score differ significantly across my groups?"*

- **Pathview (Visualization)** — not an enrichment method at all, but a visualization layer. When you select KEGG in iDEP and choose Pathview, it takes your fold-change values and overlays them directly onto the official KEGG pathway diagram for whichever pathway you're inspecting, colored by direction and magnitude of change. Use it after ORA, GAGE, or GSEA has told you which pathway is interesting, to see where in that pathway the changes are concentrated.

Pathway Topology (PT) Methods

None of iDEP's built-in options are true Pathway Topology (PT) tools. PT methods like SPIA or iPathwayGuide aren't part of the standard iDEP workflow, so if network topology matters for your question, you'll need to look outside iDEP.

WGCNA

iDEP also offers Weighted Gene Co-expression Network Analysis (WGCNA), which doesn't fit any of the four categories above. Rather than testing a predefined gene set against your data, WGCNA finds its own gene groups ("modules") by clustering genes that are correlated across your samples, then checks whether any module tracks with your trait of interest. It's a data reduction/discovery method, not an enrichment test. The two connect naturally, though: once WGCNA hands you a module of interest (a data-derived gene list), you can run that list through ORA to find out what it's biologically about.

Two Places to Start

Upload Fold Changes and Adjusted P-values

For this approach, you'll upload a table of fold changes and adjusted p-values for your genes, rather than the full expression matrix. This is suitable when you already have differential expression results and want to focus on pathway-level interpretation. We can take our output from Lesson 9 and use it to explore pathway enrichment without reprocessing the raw expression data.

- Download these data here: DEG_Results_2.csv

Changes to the iDEP DEG Output Table

When you import data in iDEP, the gene ID column is automatically converted to Ensembl IDs. Therefore, only the User_ID column, which contains the original Ensembl IDs, were included in DEG_Results_2.csv. This was to avoid Gene ID conversion issues when uploading to iDEP. The processed data were also excluded, as these are not needed for pathway analysis.

Once the data is uploaded, move to the DEG1 tab to select an FDR cutoff and minimum fold-change threshold for defining differentially expressed genes. The enrichment analysis in the DEG2 tab will use the resulting gene lists to identify over-represented pathways and gene sets.

(**OUR STARTING POINT FOR THIS LESSON**)

Expression matrix

If you upload only fold changes and adjusted p-values (rather than the full expression matrix), PGSEA, PGSEA w/ all samples, GSVA, ssGSEA, and PLAGE will not be available, since these methods require per-sample expression values to compute per-sample pathway scores. GAGE, GSEA (fgsea), and ReactomePA can run from fold-change data alone.

Upload Expression Matrix and Sample Design

Proceed from the workflow from Lesson 9. Here we would start with our full expression matrix and the corresponding sample design and run the Stats step to generate fold changes and p-values. This is suitable when you want to explore pathway-level interpretation directly from the raw expression data, allowing for more flexibility in analysis and visualization.

- Download the expression matrix here: rsem_modified.txt
- Download the sample design here: Design.csv

Proceed through the workflow as outlined in Lesson 9 *(https://bioinformatics.ccr.cancer.gov/docs/bioinformatics-for-beginners-2026/Module_3_DEG_Pathway/Lesson9)*, generating differential expression results before moving on to pathway analysis.

Before opening pathway tools, confirm:

- the data and sample design were uploaded and preprocessed;
- Stats was run (in the DEG1 tab);
- the active contrast is `Tumor-Norm` or the intended comparison;
- the direction of log2FC is understood.

TrainingData

The teaching data set is from https://rnabio.org/ *(https://rnabio.org/)*. It is a practice set with 3 replicates of data from the HCC1395 breast cancer cell line and 3 replicates of data from HCC1395BL matched lymphoblastoid line, making a tumor vs normal comparative dataset. Importantly, this dataset has been pre-filtered to chromosome 22 only. Because of that, the pathway results will be limited and will likely not reflect the biological context.

DEG Functional Enrichment

Functional enrichment of DEG up and downregulated genes helps identify biological processes, pathways, and functional categories that are over-represented in your lists. This analysis is available in the DEG2 tab under Enrichment. Remember, ORA is sensitive to the

user's choices of FDR and fold-change thresholds, so be sure to consider the implications of your filtering choices on the resulting gene lists.

Under Enrichment you will find tabs for Pathways, Tree, Network, Plot, Genes, and Details. Key points from the iDEP documentation are summarized below:

- **P-value calculation:** One-sided hypergeometric test, adjusted for multiple testing via Benjamini-Hochberg, reported as FDR
- **FDR interpretation:** Likelihood the enrichment is due to chance; larger pathways tend to get smaller FDRs simply due to greater statistical power
- **Fold Enrichment (Effect size measure):** % of gene list belonging to a pathway divided by the % of background with that pathway; shows magnitude of over-representation.
- **Background gene set:** Includes filtered genes from the original list (i.e., genes passing the low-count filter in RNA-seq)
- **Redundant pathway handling:** Pathways sharing ≥90% of genes AND ≥50% of words in the pathway name are collapsed, represented by the most significant pathway.

To begin your analysis:

1. Open the Enrichment view.
2. Run enrichment separately for:
3. up-regulated genes;
4. down-regulated genes.
5. Enable `Use filtered genes as background`. *Remember that the background is the set of genes that could have been detected as DE, not all genes in the genome. This is important for accurate enrichment results.*
6. Use `Remove Redundant Gene Sets` for a cleaner view.
7. Start with a manageable number of top pathways, such as 10 to 20.
8. Export the CSV table.

Choosing Gene Sets and Databases

Your scientific question will guide which gene sets and databases are most relevant for your analysis. For example, if you are interested in transcriptional regulation, you may want to focus on transcription factor target gene sets. If you are interested in signaling pathways, you may want to focus on KEGG or Reactome pathways. If you are interested in TF binding motifs, you may want to focus on the Motif gene sets from MSigDB.

Results Interpretation

Pathways table

The pathways table summarizes the enriched pathways, including statistics such as the adjusted p-value, fold enrichment, and the pathway name, which is clickable for more

information. This table is the starting point for interpreting functional enrichment results. You should download the table as a CSV file for further analysis and record-keeping.

> "Fold Enrichment is defined as the percentage of genes in the list belonging to a pathway, divided by the corresponding percentage in the background. As a measure of effect size, Fold Enrichment indicates how drastically genes of a certain pathway is overrepresented." - Internal iDEP Documentation

The downloaded CSV file will include additional columns such as the number of genes in the overlap, the pathway size, and the genes from our DEG list that were in the pathway.

	group	FDR	nGenes	Pathway size	Fold enriche	Pathway	URL	Genes
1	Upregulated	6.80e-02	6	263	7.59066474	GO:0031668	http://amigo	ENSG00000128272, ENSG00000169184, ENSG00000100292, ENSG00000100330, ENSG00000100219, ENSG00000185340
2	Upregulated	6.80e-02	6	231	8.6421854	GO:0031669	http://amigo	ENSG00000128272, ENSG00000169184, ENSG00000100292, ENSG00000100330, ENSG00000100219, ENSG00000185340
3	Upregulated	6.80e-02	2	6	110.908046	GO:0035934	http://amigo	ENSG00000100300, ENSG00000198832
4	Upregulated	6.80e-02	14	1522	3.06053741	GO:0048699	http://amigo	ENSG00000008735, ENSG00000040608, ENSG00000075275, ENSG00000100139, ENSG00000100271, ENSG00000100300, ENSG00000100360, ENSG00000128272, ENSG000001
5	Upregulated	8.89e-02	14	1777	2.62134943	GO:0007267	http://amigo	ENSG00000073150, ENSG00000100211, ENSG00000213923, ENSG00000008735, ENSG00000075275, ENSG00000099957, ENSG00000100300, ENSG00000100302, ENSG000001
6	Upregulated	8.89e-02	14	1757	2.65118835	GO:0022008	http://amigo	ENSG00000008735, ENSG00000040608, ENSG00000075275, ENSG00000100139, ENSG00000100271, ENSG00000100300, ENSG00000100360, ENSG00000128272, ENSG000001
7	Upregulated	8.89e-02	13	1449	2.98510269	GO:0030182	http://amigo	ENSG00000008735, ENSG00000040608, ENSG00000100139, ENSG00000100271, ENSG00000100300, ENSG00000100360, ENSG00000128272, ENSG00000130540, ENSG000001
8	Upregulated	8.89e-02	7	481	4.84213922	GO:0031667	http://amigo	ENSG00000128272, ENSG00000169184, ENSG00000100292, ENSG00000100300, ENSG00000100330, ENSG00000100219, ENSG00000185340
9	Upregulated	8.89e-02	3	42	23.7660099	GO:0031670	http://amigo	ENSG00000169184, ENSG00000100292, ENSG00000100219
10	Upregulated	8.89e-02	3	56	17.8245074	GO:0035176	http://amigo	ENSG00000008735, ENSG00000251322, ENSG00000184702
11	Downregulat	1.11e-09	7	25	93.1627586	GO:0006213	http://amigo	ENSG00000025708, ENSG00000239713, ENSG00000244509, ENSG00000100298, ENSG00000128394, ENSG00000243811, ENSG00000100024
12	Downregulat	1.11e-09	7	27	86.2618135	GO:0009164	http://amigo	ENSG00000093072, ENSG00000239713, ENSG00000244509, ENSG00000100298, ENSG00000128394, ENSG00000243811, ENSG00000100024
13	Downregulat	2.73e-09	8	59	45.1151373	GO:0009116	http://amigo	ENSG00000093072, ENSG00000025708, ENSG00000239713, ENSG00000244509, ENSG00000100298, ENSG00000128394, ENSG00000243811, ENSG00000100024
14	Downregulat	2.73e-09	7	32	72.7834052	GO:0034656	http://amigo	ENSG00000093072, ENSG00000239713, ENSG00000244509, ENSG00000100298, ENSG00000128394, ENSG00000243811, ENSG00000100024
15	Downregulat	2.73e-09	6	17	117.432049	GO:0046135	http://amigo	ENSG00000239713, ENSG00000244509, ENSG00000100298, ENSG00000128394, ENSG00000243811, ENSG00000100024
16	Downregulat	6.23e-09	7	39	59.7197171	GO:0072529	http://amigo	ENSG00000239713, ENSG00000244509, ENSG00000025708, ENSG00000100024, ENSG00000100298, ENSG00000128394, ENSG00000243811
17	Downregulat	7.78e-09	7	41	56.8065601	GO:1901658	http://amigo	ENSG00000093072, ENSG00000239713, ENSG00000244509, ENSG00000100298, ENSG00000128394, ENSG00000243811, ENSG00000100024
18	Downregulat	8.73e-09	6	22	90.7429467	GO:0042454	http://amigo	ENSG00000093072, ENSG00000239713, ENSG00000244509, ENSG00000100298, ENSG00000128394, ENSG00000243811
19	Downregulat	9.79e-09	5	10	166.362069	GO:0070383	http://amigo	ENSG00000100298, ENSG00000128394, ENSG00000239713, ENSG00000243811, ENSG00000244509
20	Downregulat	3.19e-08	5	13	127.970822	GO:0006216	http://amigo	ENSG00000239713, ENSG00000244509, ENSG00000100298, ENSG00000128394, ENSG00000243811

Tree and Network

The hierarchical clustering tree and network graph can be used to assess the relationship between enriched gene sets and pathways. Both are based on a distance metric derived from the percentage of overlapped genes between terms. The tree shows the hierarchical relationships among enriched terms, while the network graph visualizes the connections between pathways based on shared genes. Many of these gene sets are likely to be related or redundant, so these visualizations help identify clusters of related terms.

We can reduce redundancy by selecting the `Remove Redundant Gene Sets` option, which will help focus on the most informative terms.

Plot

Here we can create a lollipop plot, dot plot, or bar plot to visualize the top enriched pathways. The plot can be customized to show different statistics, such as adjusted p-value, fold enrichment, or gene count. This visualization helps to quickly identify the most significant pathways and their relative importance in the context of the analysis.

Genes/Details

The Genes tab lists the down and / or up-regulated genes that went into our analysis. It includes gene identifier and annotation information, as well as the location, type, and description of the gene. The Details tab provides information about the analysis that was run.

Do not over-interpret pathway results.

Pathway enrichment results indicate associations between gene lists and annotated pathways. They do not prove causal activation or inhibition of the pathways. Interpret the results as suggestive rather than definitive. These results should be used to generate hypotheses for further experimental validation.

Pathway Analysis

Under the Pathway tab, you can run GAGE, GSEA/fgsea, PGSEA, ReactomePA, GSVA, ssGSEA, or PLAGE. These methods consider the data from all genes regardless of user defined DEG thresholds. Each method has its own strengths and weaknesses, and the choice of method will depend on your scientific question and the characteristics of your data. Check out iDEP's documentation for more information on the methods and databases available: https://idepsite.wordpress.com/pathways/ *(https://idepsite.wordpress.com/pathways/)*.

P-hacking

Pathway analysis is sensitive to the choice of method, gene set database, and parameters. Avoid trying multiple methods and databases until you find a significant result, commonly known as p-hacking. Instead, choose a method and database that are appropriate for your scientific question and stick with it. If you want to compare methods, do so systematically and report all results. Comparing results from different methods can be informative, and in some cases can be used to support a finding, but it should be done in a transparent and reproducible manner. For example, consider comparing GSEA results to earlier functional enrichment results to see if many of the same pathways are identified.

To run a pathway analysis, follow these steps:

1. Select the same comparison used in DEG (if proceeding from DGE workflow).
2. Choose a pathway analysis method, e.g., `GSEA (preranked fgsea)`.
3. Choose a gene-set / pathway database. A good starting point is KEGG, Reactome, or GO Biological Process.
4. Set the pathway FDR cutoff (e.g., 0.05). This is the threshold for significance.
5. Check gene-set size limits.
6. Click Submit.
7. Inspect Significant pathways, Tree, Network, Heatmap, and KEGG tabs as available.

Gene Set Enrichment Analysis (GSEA) / fgsea

We are using the fgsea implementation of GSEA in iDEP for a demo. GSEA is a popular rank-based method that tests whether a predefined set of genes shows statistically significant, concordant differences between two biological states.

Ranking metrics

GSEA ranks genes by a metric that reflects their differential expression between two conditions. iDEP uses the log2 fold change from the DEG analysis. GSEA results are sensitive to the choice of the ranking metric. See a summary of common ranking metrics here *(https://bioinformatics.ccr.cancer.gov/docs/btep-coding-club/CC2025/GSEA/GSEA/#choosing-a-ranking-metric)*. Unfortunately, the least robust choice is generally a raw FC (the default in iDEP

with no other alternatives). Better choices include the test statitic (e.g., t-test statistic, Wald test statistic from DESeq2, Signal-to-Noise Ratio) or signed -log10(p-value), though the latter requires correction.

Gene set size limits

GSEA results can be influenced by the size of the gene sets. The normalization of enrichment scores is "not very accurate for extremely small or extremely large gene sets. For example, for gene sets with fewer than 10 genes, just 2 or 3 genes can generate significant results.".

Other reasons for limiting large gene sets include:

- The run time of the analysis will increase with larger gene sets.
- Large gene sets are more general and therefore, more difficult to interpret.
- "The GSEA null hypothesis is that the set is distributed uniformly at random along the gene ranking. For larger, structured gene sets that tends to be the case, even when there is no biological significance. In particular, artifacts from gene-gene expression correlation start to play an important role. --- Alexey Sergushichev, FGSEA author

Settings to control with iDEP:

Pathway significance cutoff (FDR): filters results to pathways with adjusted p-values at or below the chosen threshold. More options: reveals extra sliders/toggles such as gene-set size limits, number of top pathways to show, and the gene-level FDR filter applied before analysis. **Gene-set size (Min/Max):** exclude sets that are too small (often noisy) or too large (overly generic).
Number of top pathways: controls how many entries appear in the summary tables and plots.
Remove genes with big FDR: discards genes whose Stats adjusted p-values exceed the specified threshold before pathway scoring.
Use absolute values of fold changes: treats up- and down-regulation equally (useful for pathways where mixed regulation indicates activation, such as signaling cascades).
Show pathway IDs: appends identifiers (for example, GO terms or KEGG IDs) to pathway names in the output.
Report: download an HTML report of the results.
PGSEA color palette: when using PGSEA, choose the color scheme applied to the sample-by-pathway heatmap. - IDEP Internal Documentation

Results

iDEP returns a table of significant pathways / gene sets, a tree diagram of related terms, a network graph of related terms, and a heatmap visualizing the expression of genes driving each pathway. The heatmap is more relevant if proceeding to this analysis from the full expression matrix rather than the DEG results. There is also a KEGG tab that shows a pathway diagram via Pathview, if KEGG is selected as the database for the analysis.

Pathview Integration

There were no significant pathways from KEGG in the teaching dataset. To see how this option works, you can use a demo dataset from the iDEP website.

A colored KEGG diagram is a visualization of gene-level changes on a pathway map. It does not prove flux, protein activity, phosphorylation, or causal pathway activation.

iDEP does not provide a by pathway enrichment plot.

Significant Pathways Table

The significant pathways table provides the direction, pathway, NES (normalized enrichment score), Genes, and adjusted p-value (FDR). The NES is a normalized version of the enrichment score that accounts for differences in gene set size and correlations between gene sets. This allows comparability across different gene sets. Genes are the number of genes in the ranked list that are in the pathway. The adjusted p-value (FDR) is the false discovery rate, which is a measure of the expected proportion of false positives among the significant pathways.

Example following GSEA with Curated.Reactome:

Direction	GSEA analysis: TUMORNORM_log2FC	NES	Genes	adj.Pval
Down	R-HSA-168256 Immune System	-0.7335	21	5.0e-06
	R-HSA-168249 Innate Immune System	-0.7478	12	7.3e-04
	R-HSA-1643685 Disease	-0.479	21	9.3e-02

Column Names

The column names in the iDEP output are neither intuitive nor well-documented. The description of "Genes" here was gained from the code.

Leading Edge Genes

The leading edge genes are the subset of genes that contribute most to the enrichment score of a pathway. These are not included as output for the GSEA results in iDEP, but they can be obtained by running GSEA outside of iDEP. The leading edge genes are important for understanding which genes are driving the enrichment of a pathway and can provide insights into the underlying biology.

Example Interpretation of Results

Remember not to overstate results. The following is an example of how to interpret the results from GSEA with GO BP:

"Gene set enrichment analysis (fgsea) comparing tumor versus normal samples identified several immune-related Gene Ontology Biological Process terms as significantly downregulated

in tumor samples, including "immune response" (adj. p = 2.8e-04) and "innate immune response" (adj. p = 0.032). These results may suggest reduced coordinated expression of immune-related gene sets in tumor tissue relative to normal."

Common Troubleshooting

Symptom	Likely issue	Trainer response
Pathway tab has no contrast	Stats was not run or no contrast selected	Return to Stats and run the comparison
Few pathways are significant	Strict FDR/gene-set size filters, weak signal, poor ID mapping	Check mapping, relax cutoffs cautiously, inspect ranked results
Many generic GO terms dominate	GO redundancy and broad parent terms	Use redundancy removal, tree/network views, or a more focused database
KEGG tab is empty	KEGG database not selected or unavailable	Select KEGG first and rerun
Up/down interpretation is confusing	Contrast direction unclear	Revisit `Tumor-Norm` direction and log2FC sign
Different methods return different pathways	Different null hypotheses and inputs	Treat as a comparison exercise, not a failure

Beyond iDEP

iDEP is a strong teaching tool because DEG and pathway analysis live in the same interface. However, for a deeper analysis, especially when considering GSEA and the data not included (See R tutorial here *(https://bioinformatics.ccr.cancer.gov/docs/btep-coding-club/CC2025/GSEA/GSEA/)* for a comparison of expected results), consider using other tools that offer more flexibility, additional databases, and advanced visualization options. Find a few popular options below, and explore their documentation to see which best fits your needs:

Tool	Why use it
Broad GSEA	Full-featured GSEA workflows and MSigDB integration
Enrichr	Fast exploratory enrichment across many libraries
g:Profiler	Convenient multi-organism enrichment and ID mapping
Reactome Pathway Browser	Curated pathway hierarchy and diagrams
clusterProfiler	

Tool	Why use it
	Reproducible R-based GO, KEGG, Reactome, and GSEA workflows
DAVID	Classic ORA and functional annotation clustering
Cytoscape EnrichmentMap	Network visualization of enrichment results

Resources Used

iDEP

- iDEP current web application and embedded documentation: https://bioinformatics.sdstate.edu/idep/.
- iDEP pathway analysis documentation: https://idepsite.wordpress.com/pathways/.
- Ge SX, Son EW, Yao R. 2018. iDEP: an integrated web application for differential expression and pathway analysis of RNA-Seq data. BMC Bioinformatics. https://doi.org/10.1186/s12859-018-2486-6.

Method-specific papers

- Luo W, Friedman MS, Shedden K, Hankenson KD, Woolf PJ. 2009. GAGE: generally applicable gene set enrichment for pathway analysis. BMC Bioinformatics. https://doi.org/10.1186/1471-2105-10-161.
- Subramanian A et al. 2005. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. PNAS. https://doi.org/10.1073/pnas.0506580102.
- Korotkevich G, Sukhov V, Sergushichev A. 2021. Fast gene set enrichment analysis (fgsea). https://doi.org/10.1101/060012.
- Furlotte NA. PGSEA: Parametric Gene Set Enrichment Analysis. Bioconductor. https://bioconductor.org/packages/PGSEA/.
- Yu G, He QY. 2016. ReactomePA: an R/Bioconductor package for Reactome pathway analysis and visualization. Molecular BioSystems. https://doi.org/10.1039/C5MB00663E.
- Hanzelmann S, Castelo R, Guinney J. 2013. GSVA: gene set variation analysis for microarray and RNA-Seq data. BMC Bioinformatics. https://doi.org/10.1186/1471-2105-14-7.
- Barbie DA et al. 2009. Systematic RNA interference reveals that oncogenic KRAS-driven cancers require TBK1. Nature. https://doi.org/10.1038/nature08460 (ssGSEA).
- Tomfohr J, Lu J, Kepler TB. 2005. Pathway level analysis of gene expression using singular value decomposition. BMC Bioinformatics. https://doi.org/10.1186/1471-2105-6-225 (PLAGE).
- Luo W, Brouwer C. 2013. Pathview: an R/Bioconductor package for pathway-based data integration and visualization. Bioinformatics. https://doi.org/10.1093/bioinformatics/btt285.

Databases

- Reactome database: https://reactome.org/
- reactome.db Bioconductor annotation package: https://bioconductor.org/packages/reactome.db/
- KEGG database: https://www.genome.jp/kegg/

General pathway analysis guidance

- Khatri P, Sirota M, Butte AJ. 2012. Ten years of pathway analysis: current approaches and outstanding challenges. PLOS Computational Biology. https://doi.org/10.1371/journal.pcbi.1002375
- Nguyen TM et al. 2019. Identifying significantly impacted pathways: a comprehensive review and assessment. Genome Biology. https://doi.org/10.1186/s13059-019-1790-4.
- Young MD et al. 2010. Gene ontology analysis for RNA-seq: accounting for selection bias. Genome Biology. https://doi.org/10.1186/gb-2010-11-2-r14
- Conesa A et al. 2016. A survey of best practices for RNA-seq data analysis. Genome Biology. https://doi.org/10.1186/s13059-016-0881-8
- Gene Set Analysis: A Step-by-Step Guide. https://pmc.ncbi.nlm.nih.gov/articles/PMC4638147/
- Bora A, McKenzie M, Ziemann M. 2026. Ten common mistakes that could ruin your enrichment analysis. PLOS Computational Biology. https://doi.org/10.1371/journal.pcbi.1014122
- GO enrichment analysis guide: https://geneontology.org/docs/go-enrichment-analysis/
- Pathway Commons GSEA primer: https://www.pathwaycommons.org/guide/primers/data_analysis/gsea/
- GAGE/Pathview RNA-seq pathway workflow: https://www.bioconductor.org/packages/devel/bioc/vignettes/gage/inst/doc/RNA-seqWorkflow.pdf
- EnrichmentMap documentation: https://enrichmentmap.readthedocs.io/
- GSEA-MSigDB documentation: https://docs.gsea-msigdb.org/.

Internal course material

- Lesson 9 (DEG workflow): https://bioinformatics.ccr.cancer.gov/docs/bioinformatics-for-beginners-2026/Module_3_DEG_Pathway/Lesson9
- GSEA ranking metrics: https://bioinformatics.ccr.cancer.gov/docs/btep-coding-club/CC2025/GSEA/GSEA/#choosing-a-ranking-metric
- GSEA vs. iDEP comparison (Coding Club): https://bioinformatics.ccr.cancer.gov/docs/btep-coding-club/CC2025/GSEA/GSEA/

Additional Resources

Module 1: Unix

- Unix/Linux Command Reference from FOSSwire.com — a handy reference sheet of common commands and their options.