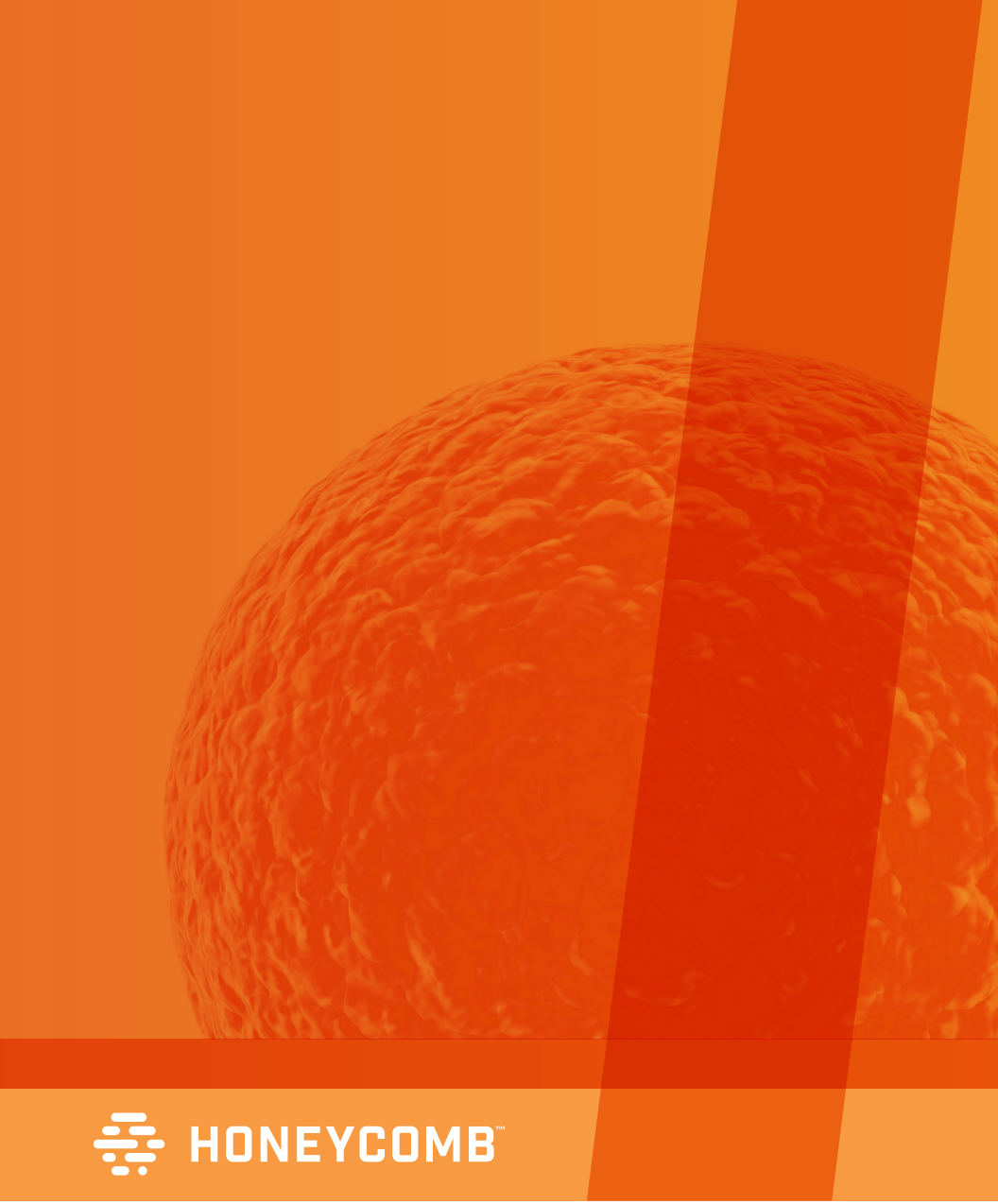


HONEYCOMB™

HIVE™ scRNAseq BeeNet™ Software Guide

Single Cell RNA Sequencing Analysis with HIVE CALL

BeeNet Software Version v2.X.X¬¬

This product is for research use only.
Not for use in diagnostic procedures.

///// Introduction

BeeNet™ is a custom software designed to process data from paired-end Illumina® sequencing of single-cell RNA-seq libraries produced by the HIVE™ line of products. The software consists of a set of programs that receives demultiplexed FASTQ file inputs and yields a transcript and gene count matrix (CM), aligned BAM file, and a quality metric (QC) file.

This document, along with the accompanying video tutorials **BeeNet Download** and **BeeNet Running Analysis**, detail how to download, install, and use the software to analyze HIVE scRNAseq data. This document is oriented to support a user familiar with Linux command line interface.

BeeNet is also hosted on Terra.bio, a cloud-native platform. Terra's graphic user interface (GUI) allows users from any background to run and automate workflows without prior knowledge of command-line tools or cloud computing. **BeeNet hosted on Terra.bio** document and **Running Analysis with BeeNet on Terra.bio** video protocol details how to use the Terra.bio implementation of BeeNet.

///// HIVE™ scRNAseq

HIVE™ scRNAseq is a picowell array technology that enables users to collect, store, and process single cells into NGS libraries without specialized instrumentation. HIVE™ scRNAseq libraries are indexed with a sample-specific identifier and Illumina® adapters. Each molecule within a sample is labeled with a unique cell barcode to delineate the cell of origin.

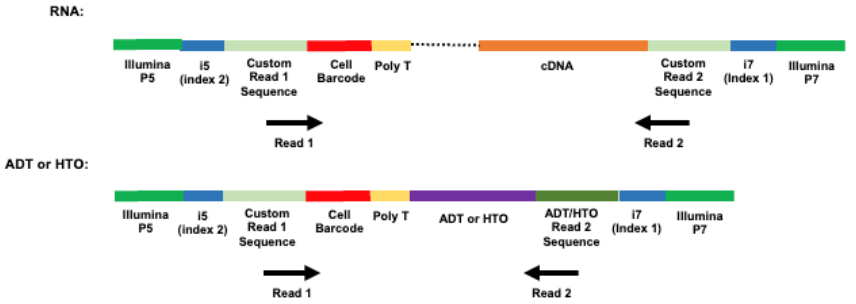


Figure 1: shows the structure of a single RNA or ADT/HTO library molecule (Illumina P5: Illumina P5 adaptor, i5: index 2, i7: index 1, Illumina P7: Illumina P7 adapter).

///// BeeNet Software

BeeNet is an integrated set of programs that will process Read 1 and Read 2 FASTQ files to create Count Matrices as well as aligned bam and QC files. To run analysis with BeeNet, you will need a Linux system and access to the command-line with read/write/execute permissions.

Setup of BeeNet is accomplished through two steps:

1. Download and install the software to your system
2. Download appropriate genome reference files

These steps do not need to be repeated for subsequent analyses on the same system unless a different reference file is required.

Running BeeNet to generate count matrices from RNA FASTQ files requires a single command 'analyze' to perform the analysis workflow.

BeeNet automated RNAseq analysis follows three main steps:

- QC and pre-processing of raw FASTQ files and cell barcodes
- Alignment and annotation of the reads – outputs a bam file
- Molecule counting and Count Matrix (CM) creation

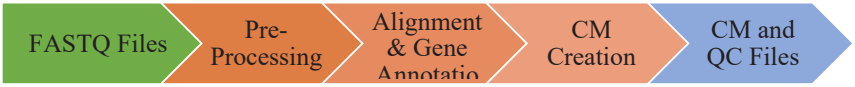


Figure 2.0 shows the scRNAseq analysis workflow

///// System Requirements

BeeNet software should be run on Linux systems that meet the requirements listed below. The software has been validated using Ubuntu and Debian distributions. Contact your local IT services for instructions on how to set up a Linux terminal on a system with the appropriate requirements.

- RAM: 64GB (note: 128GB is recommended for "mixed-species" genomes)
- Free disk space (after downloading reference files): at least two times the size of your FASTQ files.

///// BeeNet Setup

Download and Install BeeNet

(video protocol – **BeeNet Download**)

Color coding for the rest of the document:
command function flag argument(s)

To download and install the BeeNet software, register at **https://download.honeycomb.bio/** by entering your full name, institution, institutional email and accepting the license agreement. You will be emailed a link to download the software, which you can use from a Linux Terminal to install (wget) and give permission to execute commands (chmod).

```
$ wget [your unique link]
$ chmod +x beenet
```

Additionally, the **syscheck** function can be used to check your system configuration details and available resources for running BeeNet, to verify system requirements for BeeNet are met.

```
$ ./beenet syscheck
```

Download reference files

The **download-ref** function can be run without any arguments to view a list of available references. You can download the appropriate reference files based on your sample type. References are large (30 GB+) and will take between 5-15 minutes to downloaded based on your internet speed.

```
$ ./beenet download-ref
```

Available reference files

Human Genome GRCh38

The reference files were created using the below FASTA and gtf:

FASTA: http://ftp.ensembl.org/pub/release-104/fasta/homo_sapiens/dna/Homo_sapiens.GRCh38.dna.primary_assembly.fa.gz

gtf: http://ftp.ensembl.org/pub/release-104/gtf/homo_sapiens/Homo_sapiens.GRCh38.104.gtf.gz

Source: Ensembl

```
$ ./beenet download-ref 20210603_GRCh38.104
```

Human Genome hg19

The reference files were created using the below FASTA and gtf:

FASTA:http://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_19/GRCh37.p13.genome.fa.gz

gtf:http://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_19/gencode.v19.annotation.gtf.gz

Source: Gencode

```
$ ./beenet download-ref 20210608_GRCh37.p13
```

Mouse Genome mm10

FASTA:http://ftp.ensembl.org/pub/release-104/gtf/mus_musculus/dna_index/Mus_musculus.GRCm39.dna.toplevel.fa.gz

gtf:http://ftp.ensembl.org/pub/release-104/gtf/mus_musculus/Mus_musculus.GRCm39.104.gtf.gz

Source: Ensembl

```
$ ./beenet download-ref 20210714_mm10.104
```

Custom reference files can also be made using the **make-ref** function outlined in the **Advanced Functions** section of this document.

//// BeeNet RNAseq Analysis Workflow

BeeNet Analyze

The **beenet analyze** function performs a complete end-to-end analysis of a single sample, incorporating the individual steps of the pipeline into a single invocation. It will process the demultiplexed FASTQ files for a single sample or library and output count matrix and summary files, as well as a BAM file of the aligned and annotated sample data. A typical use of **beenet analyze**, where the input FASTQ forward (R1) and reverse (R2) pairs are provided as arguments, looks like:

```
$ ./beenet analyze --sample-name=... --ref=... --num-barcodes=# <fq.gz files>
```

beenet analyze flags:

--sample-name=[name of the sample or library] **(required)**: Used, with a timestamp automatically appended, as a filename prefix for the output files. Only alphanumeric characters, hyphen, period, and underscore allowed (no leading period or hyphen).

--num-barcodes=[integer]**(required)**: This is the expected number of barcodes in a given sample. Count matrix generation uses this, after sorting by count of reads mapped to genes, as the number of cell barcodes in the sample. Must be an integer

--ref=[absolute path to reference folder] **(required)**: The absolute path to the STAR-indexed genome reference. BeeNet searches the tree provided by the path for the file named genomeParameters.txt and uses this file to determine STAR version compatibility as well as a reference point for the rest of the required STAR index files. If it finds multiple STAR indices it will report all the ones found.

--fastqs=[absolute path to text file] As an alternative to passing in long lists of FASTQ file paths on the command line, this parameter specifies a plain text file containing paths to FASTQ files, one pair on each line, R1 and R2 separated by whitespace.

--out=[folder name] Specifies an output directory for output files, this will also contain the temporary STARtmp directory. A timestamp is appended to this directory name so that subsequent runs of BeeNet don't overwrite previous results. If nothing specified, it will output files to working directory.

--mixed=PREFIX1, PREFIX2: Run BeeNet in "mixed genome" mode (differential genome analysis). This will generate separate count matrices and bam files for the indicated species. Note a mixed genome reference files can be used without calling this option but all genes will be outputted in a single CM and bam file.

--introns: Include intronic reads in count matrix generation. By default, introns are included and annotated in the BAM file, but addition of this flag will trigger use of intronic reads in barcode sorting and count matrix generation.

Example Analysis:

A test dataset is available for download for users who want to do a test run with BeeNet. You can download the hosted test data using wget and decompress it using tar

```
$ wget https://storage.googleapis.com/resources.honeycomb.bio/test-data/HC-TestSample1-FBL.tar.gz
$ tar -zxvf HC-TestSample1-FBL.tar.gz
```

Example Inputs

- The sample is named "**MySample**": --sample-name=MySample
- There are **5000** cell barcodes in your sample: --num-barcodes=5000
- Using Honeycomb-provided human reference bundle (see: download-refs): --ref=**20210603_GRCh38.104**
- There are 2 lanes worth of paired-end compressed fastq files (...fq.gz) for the sample in a directory called **fastqs/HC-TestSample1-FBL/** put them in the arguments in pairs

To initiate the analysis using the test dataset, you can run the below command:

```
$ ./beenet analyze --sample-name=MySample \
        --num-barcodes=5000 --ref=20210603_GRCh38.104 \
    fastqs/HC-TestSample1-FBL/HC-TestSample1-FBL_S1_L001_R1.fastq.gz \
    fastqs/HC-TestSample1-FBL/HC-TestSample1-FBL_S1_L001_R2.fastq.gz \
    fastqs/HC-TestSample1-FBL/HC-TestSample1-FBL_S1_L002_R1.fastq.gz \
    fastqs/HC-TestSample1-FBL/HC-TestSample1-FBL_S1_L002_R2.fastq.gz
```

BeeNet Compatibility with Cite-seq and Hashing Protocols (HIVE CALL)

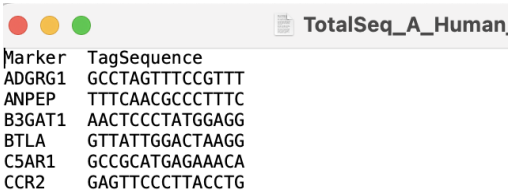
BeeNet™ (version 2.X.X or later) also performs end-to-end analysis of HIVE libraries that are generated from Hashtag Oligo Labeling (HTO) and Antibody-derived tag labeling (ADT) experuments. BeeNet™ receives demultiplexed FASTQ file inputs and yields a read count matrix (CM) and quality metric (QC) file. These files can be important into downstream workflows for hash-based demultiplexing and protein expression analyses.

///// BeeNet HIVE CALL Analysis Workflows

BeeNet AnalyzeCALL

The **beenet analyzeCALL** function performs a complete end-to-end analysis of a single sample or library, incorporating the individual steps of the pipeline into a single invocation. It will process the demultiplexed FASTQ files for a single sample and output a read count matrix and summary file.

The **analyzeCALL** function requires an input sheet containing the tag sequences for alignment. The Panel Info sheet must contain the information for the protein/marker name and tag sequence. For more accurate results, only include tags that were included in your experiment. Several panel Info Sheets for commercially available panels are available for download on the Honeycomb Biotechnologies Support Page. If you used a custom panel, please create a custom Info sheet. A template of a custom Info Sheet can be found on the Honeycomb Biotechnologies Support Page; see below a screen shot of an example of an ADT Panel Sheet.



The Panel Info Sheet must 2 columns:

Marker: The name of the marker. The name may contain dashes and underscores, but may not contain spaces or other special characters.

TagSequence: The tag sequence. This must be all capital letters and only contain A, T, C, G. These sequences are provided by the manufacturer.

A typical use of **beenet analyzeCALL**, where the input FASTQ forward (R1) and reverse (R2) pairs are provided as arguments, looks like:

```
$ ./beenet analyzeCALL --panel=... --sample-name=... --num-barcodes=# <fq.gz files>
```

beenet analyzeCALL flags:

--panel=[absolute path to panel file] (required): BeeNet will search for tag sequences within this panel in the R2 FASTQ to determine presence of oligo tags. Panel sheets are required for both HTO and ADT analyses. The Panel name will be appended as a file name prefix for the output files.

--sample-name=[name of the sample or library] (required): Used, with a timestamp automatically appended, as a filename prefix for the output files. Only alphanumeric characters, hyphen, period, and underscore allowed (no leading period or hyphen).

--num-barcodes=[integer](required): This is the expected number of barcodes in a given sample. Count matrix generation uses this, after sorting by count of reads mapped to genes, as the number of cell barcodes in the sample. Must be an integer.

--fastqs=[absolute path to text file]: As an alternative to passing in long lists of FASTQ file paths on the command line, this parameter specifies a plain text file containing paths to FASTQ files, one pair on each line, with R1 and R2 separated by whitespace.

--out=[folder name]: Specifies an output directory for output files, this will also contain the temporary STARtmp directory. A timestamp is appended to this directory name so that subsequent runs of BeeNet don't overwrite previous results. If nothing specified, output files will be written to the working directory.

--hamming-bc=#: Barcode hamming reduction distance, 0 or 1 (default: 1). With a hamming reduction value of 1, BeeNet combines cell barcodes that are mismatched by one base pair.

--hamming-tag=#: Sequence tag hamming reduction distance, 0 or 1 (default: 1). With a hamming reduction value of 1, BeeNet will align a read to a tag sequence if it is mismatched by one base pair.

Example Analysis- Hashtag Oligos (HTO)

A test dataset is available for download for users who want to do a HashTag Oligo test run with BeeNet. You can download the hosted test data using wget and decompress it using tar

```
$ wget https://storage.googleapis.com/resources.honeycomb.bio/test-data/HC-TestSample1-FBL.tar.gz
$ tar -zxvf HC-TestSample1-FBL.tar.gz
```

IMPORTANT: You will notice RNA-seq and Hashtag Oligo FASTQs associated with this test dataset. For any panel analysis, BeeNet must be run twice, once on the RNA library using **beenet analyze**, and once on the HTO library using **beenet analyzeCALL**.

Example Inputs

- The RNA-seq library is named "MyRNALibrary" --sample-name=**MyRNALibary**
- There are **30000** cell barcodes in your sample: --num-barcodes=30000
- Using Honeycomb-provided hashtag Panel Info Sheet: --panel=panel.txt
- There are 2 lanes worth of paired-end compressed FASTQ files (...fq.gz) for the sample in a directory called **fastqs/HC-TestSampleRNA01-FBL/** put them in the arguments in pairs

To initiate the RNAseq analysis using the test dataset, you can run the below command:

```
$ ./beenet analyze --sample-name=MyRNALibrary \
        --num-barcodes=30000 --ref=20210603_GRCh38.104 \
    fastqs/HC-TestSampleRNA01-FBL/HC-TestSampleRNA01-FBL_S1_L001_R1.fastq.gz \
    fastqs/HC-TestSampleRNA01-FBL/HC-TestSampleRNA01-FBL_S1_L001_R2.fastq.gz \
    fastqs/HC-TestSampleRNA01-FBL/HC-TestSampleRNA01-FBL_S1_L002_R1.fastq.gz \
    fastqs/HC-TestSampleRNA01-FBL/HC-TestSampleRNA01-FBL_S1_L002_R2.fastq.gz
```

For the equivalent HTO library:

- The HTO library is named "MyHTOLibrary" --sample-name=**MyHTOLibary**
- There are **30000** cell barcodes in your sample: --num-barcodes=30000
- Using Honeycomb-provided hashtag Panel Info Sheet: --panel=panel.txt
- There are 2 lanes worth of paired-end compressed FASTQ files (...fq.gz) for the sample in a directory called **fastqs/HC-TestSampleHTO01-FBL/** put them in the arguments in pairs

To initiate the HTO analysis using the test dataset, you can run the below command:

```
$ ./beenet analyzeCALL --panel= panel.txt --sample-name=MyHTOLibrary \
        --num-barcodes=30000 \
    fastqs/HC-TestSampleHTO01-FBL/HC-TestSampleHTO01-FBL_S1_L001_R1.fastq.gz \
    fastqs/HC-TestSampleHTO01-FBL/HC-TestSampleHTO01-FBL_S1_L001_R2.fastq.gz \
    fastqs/HC-TestSampleHTO01-FBL/HC-TestSampleHTO01-FBL_S1_L002_R1.fastq.gz \
    fastqs/HC-TestSampleHTO01-FBL/HC-TestSampleHTO01-FBL_S1_L002_R2.fastq.gz
```

Example Analysis- Antibody-Derived Tags (ADT)

A test dataset is available for download for users who want to do an Antibody Derived Tag test run with BeeNet. You can download the hosted test data using wget and decompress it using tar

```
$ wget https://storage.googleapis.com/resources.honeycomb.bio/test-data/HC-TestSample1-FBL.tar.gz
$ tar -zxvf HC-TestSample1-FBL.tar.gz
```

Example Inputs

- The RNA-seq library is named "MyRNALibrary" --sample-name=MyRNALibary
- There are 30000 cell barcodes in your sample: --num-barcodes=30000
- Using Honeycomb-provided hashtag Panel Info Sheet: --panel=panel.txt
- There are 2 lanes worth of paired-end compressed FASTQ files (...fq.gz) for the sample in a directory called fastqs/HC-TestSampleRNA01-FBL/ put them in the arguments in pairs

To initiate the RNAseq analysis using the test dataset, you can run the below command:

```
$ ./beenet analyze --sample-name=MyRNALibrary \
            --num-barcodes=30000 --ref=20210603_GRCh38.104 \
      fastqs/HC-TestSampleRNA01-FBL/HC-TestSampleRNA01-FBL_S1_L001_R1.fastq.gz \
      fastqs/HC-TestSampleRNA01-FBL/HC-TestSampleRNA01-FBL_S1_L001_R2.fastq.gz \
      fastqs/HC-TestSampleRNA01-FBL/HC-TestSampleRNA01-FBL_S1_L002_R1.fastq.gz \
      fastqs/HC-TestSampleRNA01-FBL/HC-TestSampleRNA01-FBL_S1_L002_R2.fastq.gz
```

For the equivalent ADT library:

- The ADT library is named "MyADTLibrary" --sample-name=**MyADTLibary**
- There are **30000** cell barcodes in your sample: --num-barcodes=30000
- Using Honeycomb-provided ADT Panel Info Sheet for your antibody panel: --panel=TBNK.txt
- There are 2 lanes worth of paired-end compressed FASTQ files (...fq.gz) for the sample in a directory called **fastqs/HC-TestSampleHTO01-FBL/** put them in the arguments in pairs

To initiate the HTO analysis using the test dataset, you can run the below command:

```
$ ./beenet analyzeCALL --panel = TBNK.txt  --sample-name=MyADTLibrary \
        --num-barcodes=30000 \
    fastqs/HC-TestSampleHTO01-FBL/HC-TestSampleHTO01-FBL_S1_L001_R1.fastq.gz \
    fastqs/HC-TestSampleHTO01-FBL/HC-TestSampleHTO01-FBL_S1_L001_R2.fastq.gz \
    fastqs/HC-TestSampleHTO01-FBL/HC-TestSampleHTO01-FBL_S1_L002_R1.fastq.gz \
    fastqs/HC-TestSampleHTO01-FBL/HC-TestSampleHTO01-FBL_S1_L002_R2.fastq.gz
```

///// BeeNet Outputs

BeeNet Output File Naming

BeeNet analyze outputs an aligned bam file, which is used to create the count matrices for RNA or HTO/ADT libraries. In addition to the count matrix files, there are multiple summary files with QC information related to the sequencing data. Files will be named automatically based on the –sample-name flag as below:

SampleName_DateofAnalysis (YYYYMMDD)_filename.extension

e.g., Sample1_20201201_TCM.tsv.gz

file extensions:

RCM – Read Count Matrix

TCM – Transcript Count Matrix

CMSummary – Count Matrix Summary

BeeNet analyzeCALL outputs a read count matrix for the specified panel as well as a summary QC file for the library. Files will be named automatically based on the --sample-name and –panel flags as below:

SampleName_panel_DateofAnalysis (YYYYMMDD)_filenameCALL.extension

e.g., Sample1_20201201_CALL.tsv.gz

file extensions:

CALL – Count Matrix for HTO/ADT

CALLQC – Count Matrix Summary for HTO/ADT

BeeNet RNA Output Files List

Based on your analysis there will either be a single bam file for that specific species, or if the mixed species analysis was done there will be a bam file for each species type. The cell barcode for each read is contained in the **CR** tag for each read.

List of expected output files are as below:

Single species outputs:	**For mixed samples (Human & Mouse):**
Sample1_20201201.bam	Sample1_20201201_Human.bam
	Sample1_20201201_Mouse.bam
Sample1_20201201_RCM.tsv.gz	Sample1_20201201_HUMAN_RCM.tsv.gz
Sample1_20201201_TCM.tsv.gz	Sample1_20201201_HUMAN_TCM.tsv.gz
	Sample1_20201201_MOUSE_RCM.tsv.gz
	Sample1_20201201_MOUSE_TCM.tsv.gz
Sample1_20201201_CMSummary.tsv	Sample1_20201201_CMSummary.tsv
Sample1_20201201_ ReadsQC.tsv	Sample1_20201201_ ReadsQC.tsv
Sample1_20201201_ SampleQC.tsv	Sample1_20201201_ SampleQC.tsv

BeeNet Output File Descriptions

All Files with barcodes are alphabetized by the barcodes.

***RCM.tsv.gz**- displays the number of reads for each unique cell barcode that maps to a specific gene in the reference genome

- **Gene:** UniProtKB Gene Name or HGNC Symbol
- **'Headers'**: barcodes for each cell - unique cell barcode of 12 bases

***TCM.tsv.gz** - displays the number of unique molecule counts for each transcriptome that mapped to a specific gene in the reference genome

- **Gene**: UniProtKB Gene Name or HGNC Symbol
- **'Headers'**: unique transcriptome barcode of 12 bases

***CMSummary.tsv** - displays the number of total Genes and number of molecule counts for each transcriptome barcode

- **Barcode**: unique cell barcode of 12 bases
- **nGenes**: total number of genes for each cell barcode
- **nTran**: total number of molecule counts for each cell barcode

***ReadsQC.tsv** - QC metrics per cell barcode

- **TotalReads**: Total reads for an individual cell barcode
- **MappedReads**: Reads from an individual cell barcode that map to the reference genome
- **ExonReads**: Reads from an individual cell barcode that map to exons
- **IntronicReads**: Reads from an individual cell barcode that maps to introns. These reads are only included in the count matrix if the --intron flag is specified
- **FilteredReads**: Total Reads that are filtered out prior to alignment
- **PolyAReads**: Reads that contain PolyA
- **5PFReads**: Reads that were filtered out due to having adapter sequence present in 5' end
- **3PFReads**: Reads that were filtered out due to having adapter sequence present in 3' end
- **badBaseBC**: Reads that were filtered out due to 2 or more bases in cell barcode with poor phred scores

***SampleQC.tsv** – QC metrics for all the reads in the fastq files for the sample

- **TotalReads**: Total reads in the FASTQ files for the sample
- **MappedReads**: Total Reads that map to the reference genome for the sample
- **ExonReads**: Total Reads that map to exons for the sample
- **IntronicReads**: Reads from an individual cell barcode that maps to introns. These reads are only included in the count matrix if the --intron flag is specified
- **FilteredReads**: Total Reads that were filtered out prior to alignment for the sample
- **PolyAReads**: Total Reads that were filtered out for containing polyA stretch
- **5PFReads**: Total Reads that were filtered out due to having adapter sequence present in 5' end
- **3PFReads**: Total Reads that were filtered out due to having adapter sequence present in 3' end
- **badBaseBC**: Total Reads that were filtered out due to 2 or more bases in cell barcode with poor phred scores

***CALL.tsv.gz**- displays the number of reads for each unique cell barcode that maps to a specific panel

- **Tag**: Tag Name from Panel Info Sheet
- **'Headers'**: barcodes for each cell - unique cell barcode of 12 bases

***CALLQC.tsv** - displays the number of reads for each unique library that maps to a specific panel

- **TotalReads**: Total reads in the FASTQ files for the sample
- **MappedReads**: Total Reads that map to the reference panel for the sample
- **UnmappedReads**: Total Reads that did not map to the reference panel for the sample
- **Merged Reads, tag**: The number of reads that were merged due to Hamming Reduction of the sequence tag
- **Merged Barcodes**: The number of cell barcodes that were merged due to Hamming Reduction
- **Merged Reads, barcode**: The number of reads that were merged due to Hamming Reduction of the barcode sequence

BeeNet Advanced Functions

make-ref make custom reference: leverages star to create index files for custom genomes by users and outputs a reference directory with the necessary index files for alignment. Note that creating references requires a min of 128GB RAM.

--ref=FASTA file of the genome you would like to create indexes for

--gtf=annotation file of the genome you would like to use, please contact support to make sure the gtf file is compatible with the annotation algorithm of Honeycomb

--out=directory where the reference files will be outputted

The individual functions listed below are automatically initiated with the beenet analyse function, but can also be run independently. Please contact support@honeycomb.bio for any questions.

bcin pre-processing of fastq files: takes compressed fastq files (...fq.gz) as parameters, extracts the cell barcode from the forward read (R1) and encodes it in the name of the reverse read (R2), and writes the single-ended result to stdout. It requires the --sample-name parameter so that it can write out FASTQ_ pipeline metrics for later use for generating the count matrix.

align perform alignment using star: calls the embedded STAR aligner with the specified reference, connecting its stdin, stdout, and stderr to BeeNet's stdin, stdout, and stderr. STAR is called with parameters such that it expects single-ended FASTQ on its stdin and produces aligned SAM records on its standard out, and uses STARtmp as a temporary directory.

bcout barcode tagging & alignment filtering: reads SAM records on stdin, filters reads with multiple alignments, decodes cell barcodes from the read name, and writes barcode-tagged SAM records to stdout.

annotate gene annotation: reads SAM records on stdin and annotates them with gene and function according to the provided annotation file (only GTF supported at this time).

count count matrix creation: reads a BAM file, and an optional FASTQ pipeline metrics file, and generates count matrices and summary files.

To reorder or learn more, visit perkinelmer-appliedgenomics.com/hive

Revvity, Inc.
940 Winter Street, Waltham, MA 02451 USA
www.revvity.com

HONEYCOMB

For a complete listing of our global offices, visit www.revvity.com