

GENOME 569

Class 4: Workflow automation tools

Essential UNIX

Essential commands

Command	Function
ls	List the contents of a directory
sort	Sort a file line by line
grep	Search inside files for pattern
awk	Write programs that process files line by line
sed	Transform a file line by line
cut	Cut out selected columns of a file
join	Join two files together based on columns
datamash	Multi-purpose tools to manipulate tab-delimited files
bedtools	Multi-purpose tool to manipulate BED files
samtools	Multi-purpose tool to manipulate SAM/BAM files

grep

Problem: search for strings in other strings

Solution: `grep <regex> file`

Regular Expression Syntax

- Most characters match themselves
The regular expression “test” matches the string ‘test’, and only that string
- [x] matches any *one* of a list of characters
“[abc]” matches ‘a’, ‘b’, or ‘c’
- [^x] matches any *one* character that is not included in x
“[^abc]” matches any single character *except* ‘a’, ‘b’, or ‘c’

Regular Expression Syntax

- “.” matches any single character
- Parentheses can be used for grouping
“(abc)+” matches ‘abc’, ‘abcabc’, ‘abcabcabc’, etc.
- x|y matches x or y
“this|that” matches ‘this’ and ‘that’, but not ‘thisthat’.

Regular Expression Syntax

- x^* matches zero or more x 's
“ a^* ” matches $'$, $'a'$, $'aa'$, etc.
- x^+ matches one or more x 's
“ a^+ ” matches $'a'$, $'aa'$, $'aaa'$, etc.
- $x?$ matches zero or one x 's
“ $a?$ ” matches $'$ or $'a'$
- $x\{m, n\}$ matches i x 's, where $m \leq i \leq n$
“ $a\{2,3\}$ ” matches $'aa'$ or $'aaa'$

Example: email addresses

How can we easily tell these two apart?

coletrap@uw.edu

spam@go.away

Here's a pattern to match simple email addresses:

`\w+@(\w+\.)+(com|org|net|edu)`

Regular Expression Syntax

- “\d” matches any digit; “\D” any non-digit
- “\s” matches any whitespace character; “\S” any non-whitespace character
- “\w” matches any alphanumeric character; “\W” any non-alphanumeric character
- “^” matches the beginning of the string; “\$” the end of the string
- “\b” matches a word boundary; “\B” matches a character that is not a word boundary

Sample problem 1

Download “grep_sed_example1.txt”

Find the lines in the notes that mention 'cell'

Find the lines in the notes that talk about 'A549'

Find the lines that talk about either A549 or MCF7 (use regex)

Find the lines that end with a cell line id (i.e. A549 or MCF7, use regex!)

Sample problem 2

Download “grep_sed_example1.txt”

Change instances of A549 to MCF7

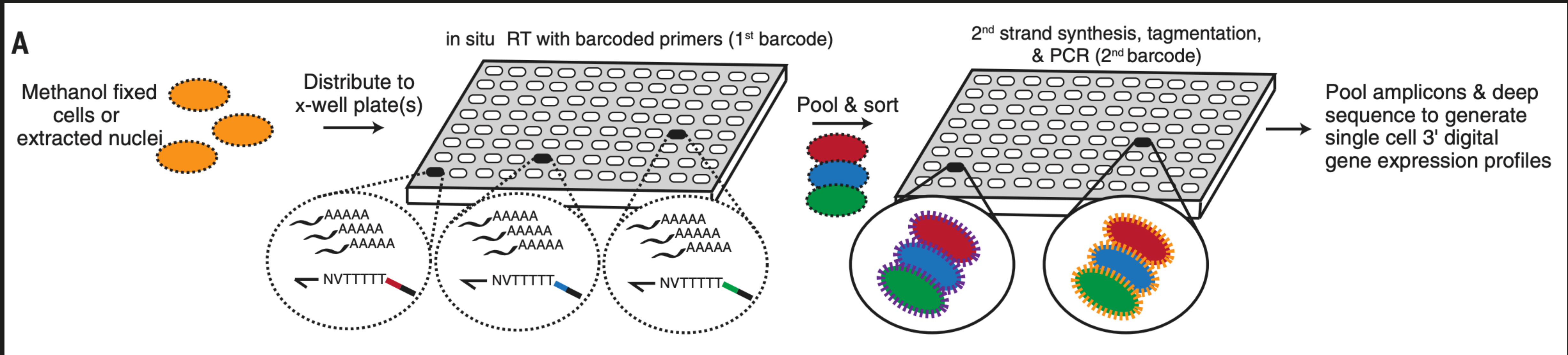
Change only the second instance of A549 to A549_LUNG

Change instances of A549 to MCF7, but without stating "A549" (i.e. use regex)

How do we make all these changes at the same time?

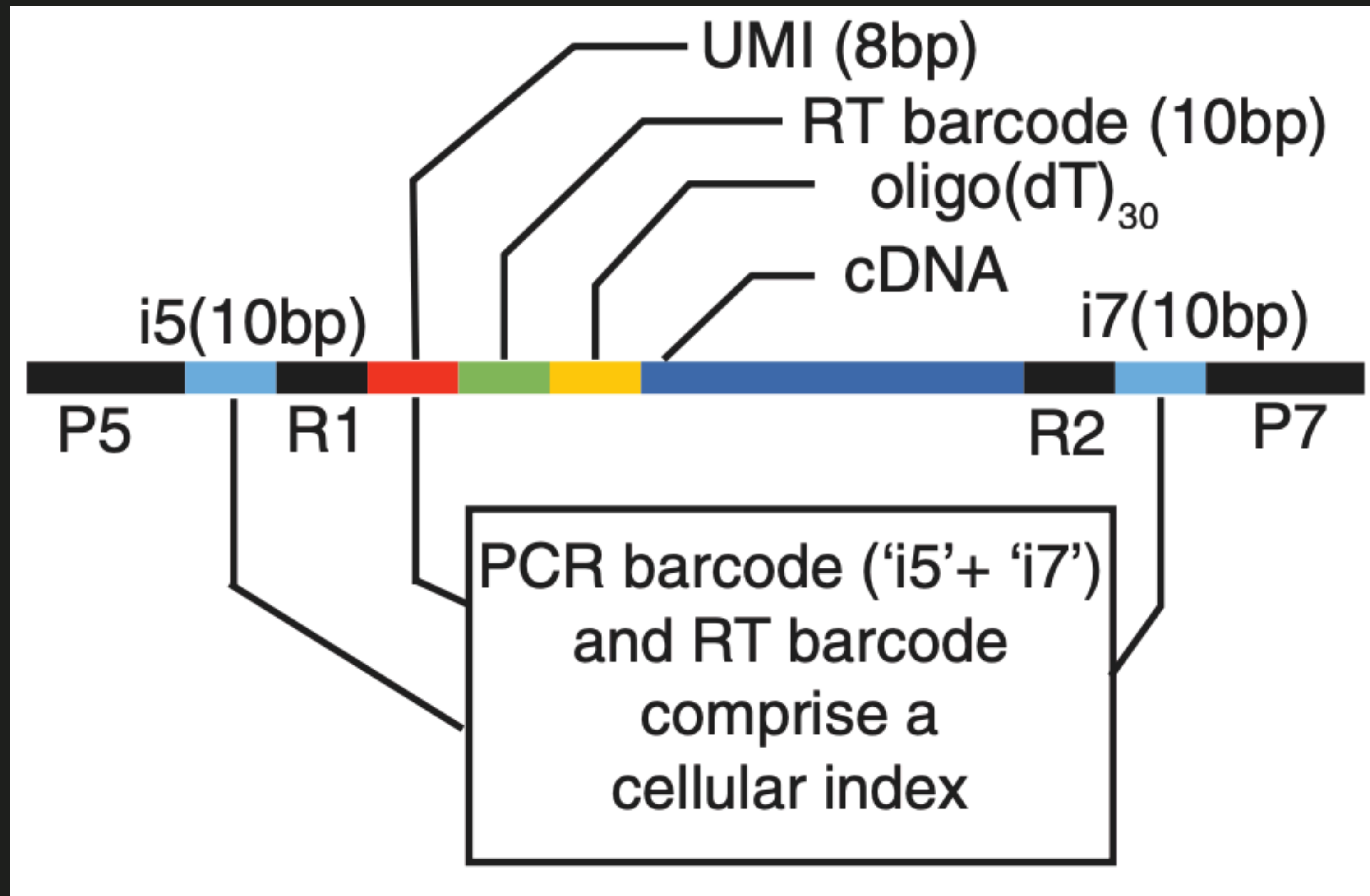
sci-RNA-seq

Single-cell RNA-seq with combinatorial cellular indexing

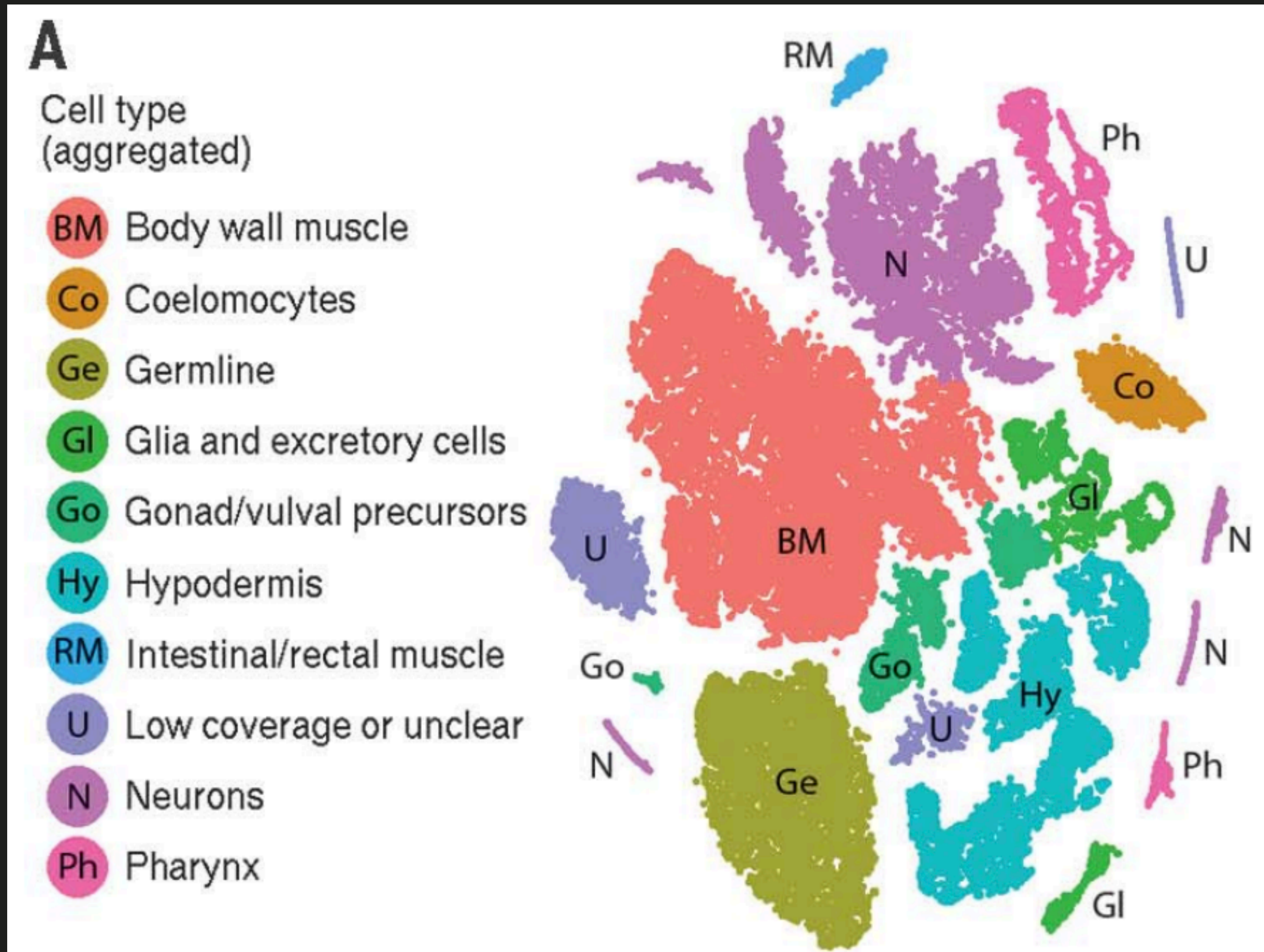


Cao & Packer et al, Science 2017

The sci-RNA-seq read layout

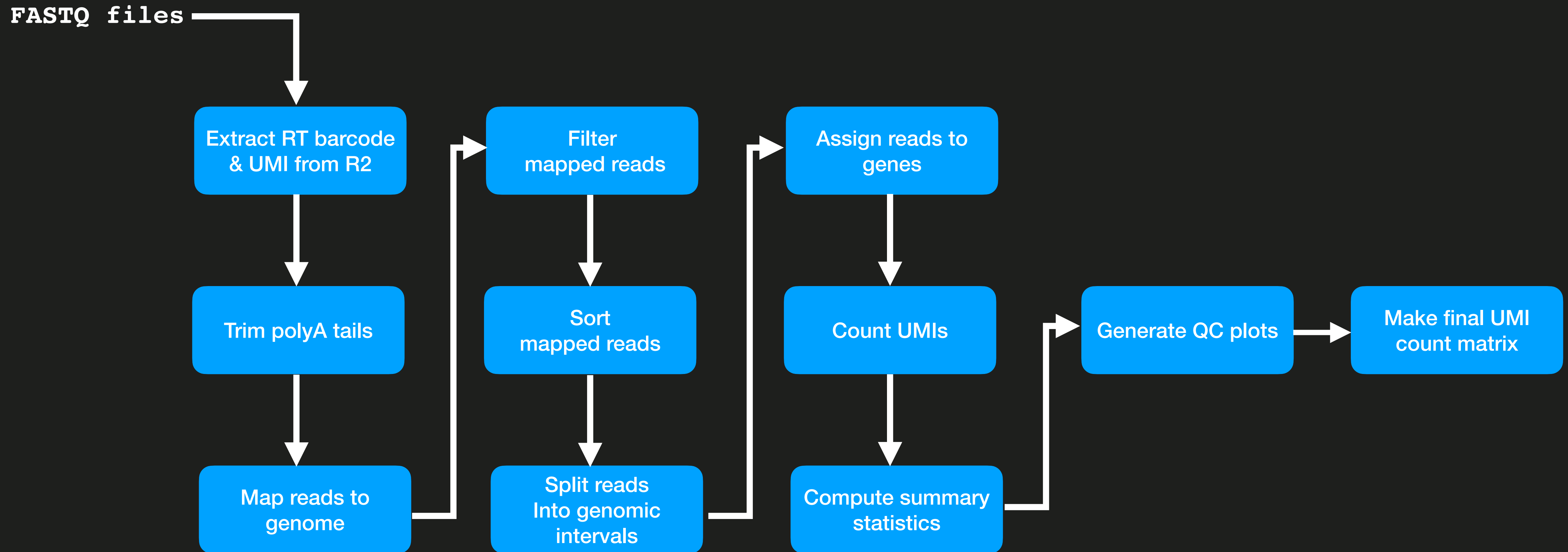


sci-RNA-seq: single-cell RNA-seq on whole animals



Anatomy of a pipeline

The sci-RNA-seq pipeline



Repo structure for Cao pipeline

▼ sci-RNA-seq-pipeline-scripts	Today at 2:05 PM	--	Folder
 assign-reads-to-genes.jun-pipeline-compatible.py	Today at 1:10 PM	4 KB	Python Source
 assign-reads-to-genes.jun-pipeline-compatible.sh	Today at 1:10 PM	772 bytes	Shell Script
 assign-reads-to-genes.py	Today at 1:10 PM	4 KB	Python Source
 assign-reads-to-genes.sh	Today at 1:10 PM	764 bytes	Shell Script
 count-rRNA-reads.sh	Today at 1:58 PM	385 bytes	Shell Script
 count-UMI-per-sample.sh	Today at 1:10 PM	818 bytes	Shell Script
 debug-assign-reads-to-genes.sh	Today at 1:10 PM	708 bytes	Shell Script
 kallisto.sh	Today at 1:10 PM	339 bytes	Shell Script
 knee-plot-merge-samples.R	Today at 1:10 PM	1 KB	Rez Source
 knee-plot.R	Today at 1:10 PM	1 KB	Rez Source
 make-fasta-for-kallisto.sh	Today at 1:10 PM	618 bytes	Shell Script
 Nextera.v2.0.P7.oligo.rev.comp	Today at 1:10 PM	1 KB	Document
 prelim-umi-count-stats.sh	Today at 1:10 PM	555 bytes	Shell Script
 put-read1-info-in-read2.awk	Today at 1:10 PM	2 KB	Document
 put-read1-info-in-read2.fancy-sample-mapping.awk	Today at 1:10 PM	4 KB	Document
 put-read1-info-in-read2.fancy-sample-mapping.sh	Today at 1:10 PM	678 bytes	Shell Script
 put-read1-info-in-read2.sh	Today at 1:10 PM	657 bytes	Shell Script
 rmdup-and-make-split-bed.sh	Today at 1:10 PM	532 bytes	Shell Script
 rmdup.awk	Today at 1:10 PM	1 KB	Document
 run-bcl2fastq.sh	Today at 1:10 PM	423 bytes	Shell Script
 run-trim-galore-for-c-elegans.sh	Today at 1:10 PM	370 bytes	Shell Script
 run-trim-galore.sh	Today at 1:10 PM	358 bytes	Shell Script
 samtools-filter-sort.sh	Today at 1:10 PM	363 bytes	Shell Script
 sci-RNA-seq.P5.oligo.rev.comp	Today at 1:10 PM	1 KB	Document
 sci-RNA-seq.RT.oligos	Today at 1:10 PM	1 KB	Document
 STAR-alignReads-8-cores.sh	Today at 1:10 PM	610 bytes	Shell Script
 STAR-alignReads-moreMem.sh	Today at 1:10 PM	611 bytes	Shell Script
 STAR-alignReads.sh	Today at 1:13 PM	611 bytes	Shell Script
 UMI-count-rollup.sh	Today at 2:05 PM	289 bytes	Shell Script
 sci-RNA-seq.pipeline.elegans.sh	Today at 2:02 PM	13 KB	Shell Script

← Start here

sci-RNA-seq.pipeline.elegans.sh

Extract RT barcode
& UMI from R2

```
#-----  
# Put read 1 info (RT well, UMI) into read 2 read name  
#-----  
echo "Moving read 1 info into read 2 name"  
  
cd $WORKING_DIR  
  
mkdir combined-fastq  
mkdir file-lists-for-r1-info-munging  
mkdir put-r1-info-in-r2-logs  
  
ls $READS_DIR/fastq | grep _R1_ | grep -v Undetermined | split -l 25 -d - file-lists-for-r1-info-munging/
```

sci-RNA-seq.pipeline.elegans.sh

Extract RT barcode
& UMI from R2

```
#-----  
# Put read 1 info (RT well, UMI) into read 2 read name  
#-----  
echo "Moving read 1 info into read 2 name"  
  
cd $WORKING_DIR  
  
mkdir combined-fastq  
mkdir file-lists-for-r1-info-munging  
mkdir put-r1-info-in-r2-logs  
  
ls $READS_DIR/fastq | grep _R1_ | grep -v Undetermined | split -l 25 -d - file-lists-for-r1-info-munging/
```

“pipe”: sends the output of
one command into the next
command as input

grep: prints lines from the
input that match the regular
expression provided

split: split up a file into
chunks. In this case, chunks
of 25 lines

`sci-RNA-seq.pipeline.elegans.sh`

Extract RT barcode
& UMI from R2

```
#-----  
# Put read 1 info (RT well, UMI) into read 2 read name  
#-----  
echo "Moving read 1 info into read 2 name"
```

qsub: submit a job to the SGE scheduler. In this case, the job script is “put-read1-info-in-read2.sh”

```
ls file-lists-for-r1-info-munging | while read BATCH; do  
  qsub -P $CLUSTER_ID -j y -o $LOG_DIR -N MOVER1R2.${BATCH} $SCRIPTS_DIR/put-read1-info-in-read2.sh \  
    $READS_DIR/fastq \\  
    $WORKING_DIR/file-lists-for-r1-info-munging/$BATCH \\  
    $SCRIPTS_DIR/ \\  
    $RT_BARCODES_FILE \\  
    $CONFIG_DIR/combinatorial.indexing.key \\  
    $WORKING_DIR/combined-fastq \\  
    $WORKING_DIR/put-r1-info-in-r2-logs  
done  
qsub -hold_jid "MOVER1R2*" -sync y -b y echo "DONE"
```

← This is a special way to call qsub so it waits for all
jobs named “MOVER1R2” to finish

Put-read1-into-read2.awk

Extract RT barcode
& UMI from R2

```
BEGIN {
    read_num = 0;
    hits = 0;

    bases[1] = "A";
    bases[2] = "C";
    bases[3] = "G";
    bases[4] = "T";

    p5_row = substr(PCR_COMB0, 1, 1);
    p5_col = substr(PCR_COMB0, 2, 2);
    p7_row = substr(PCR_COMB0, 4, 1);
    p7_col = substr(PCR_COMB0, 5, 2);

    single_sample = "";
} {

XXXXXXXXXXXX

} END {
    printf("%d\t%d\t%.3f\t(RT barcode matches, total reads, proportion)\n",
           hits, read_num, hits / read_num) > "/dev/stderr";
}
```

Snakemake

Project Phase II

THE END

For now