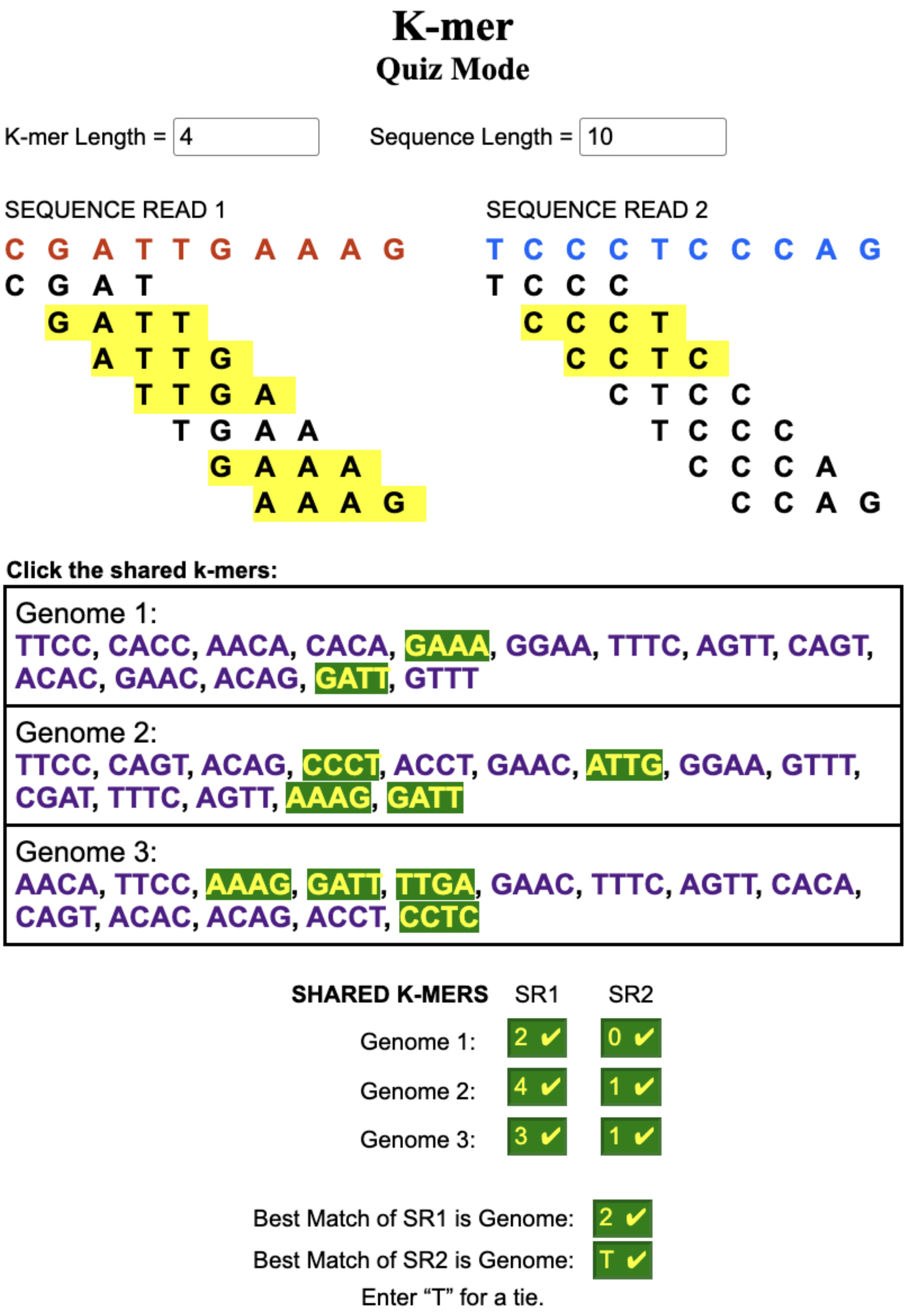


K-mer

Quiz Mode

K-mer Length = 4   Sequence Length = 10

```
SEQUENCE READ 1                  SEQUENCE READ 2
C G A T T G A A A G              T C C C T C C C A G
C G A T                          T C C C
  G A T T                          C C C T
    A T T G                          C C T C
      T T G A                          C T C C
        T G A A                          T C C C
          G A A A                          C C C A
            A A A G                          C C A G
```

Click the shared k-mers:

Genome 1:
TTCC, CACC, AACA, CACA, GAAA, GGAA, TTTC, AGTT, CAGT, ACAC, GAAC, ACAG, GATT, GTTT

Genome 2:
TTCC, CAGT, ACAG, CCCT, ACCT, GAAC, ATTG, GGAA, GTTT, CGAT, TTTC, AGTT, AAAG, GATT

Genome 3:
AACA, TTCC, AAAG, GATT, TTGA, GAAC, TTTC, AGTT, CACA, CAGT, ACAC, ACAG, ACCT, CCTC

SHARED K-MERS	SR1	SR2
Genome 1:	2 ✓	0 ✓
Genome 2:	4 ✓	1 ✓
Genome 3:	3 ✓	1 ✓

Best Match of SR1 is Genome: 2 ✓

Best Match of SR2 is Genome: T ✓

Enter “T” for a tie.

De Bruijn Graph Exercise

Purpose: Students will create independent de Bruijn graphs, then merge the graphs. Once created, then the students will then attempt Eulerian walks to assemble the sequences.

Learning Objectives:

1) Learn how to create simple de Bruijn graphs with DNA sequence reads.
2) Learn the principles of graphing and walking.
3) Learn how to combine graphs from multiple reads to create an assembly (a contig).
4) Learn how sequencing gaps and repeat regions affect assembly.
5) Learn how increasing k-mer length affects de Bruijn graph creation.

STEP 1: Create k-mers and k-1 overlaps for each sequence.
STEP 2: Make De Bruijn graphs for each sequence separately.
STEP 3: Merge the 2 graphs.
STEP 4: Write out the resulting assembled sequence.
STEP 5: Scan or take a photo or your (legible and easy to read) answers. Post all three graphs to Canvas (no more than 3 files).

GRAPH PROBLEM (use other paper if necessary.)

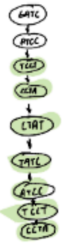
Use a k-mer length of 5 (k-1=4) to make de Bruijn graphs and assemble the following reads.

2 Reads:

GATCCTATCCTAT

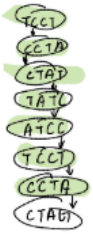
K=5 K-1=4

		L	R
GATCC	→	GATC	ATCC
ATCCT	→	ATCC	TCCT
TCCTA	→	TCCT	CCTA
CCTAT	→	CCTA	CTAT
CTATC	→	CTAT	TATC
TATCC	→	TATC	ATCC
ATCCT	→	ATCC	TCCT
TCCTA	→	TCCT	CCTA
CCTAT	→	CCTA	CTAT

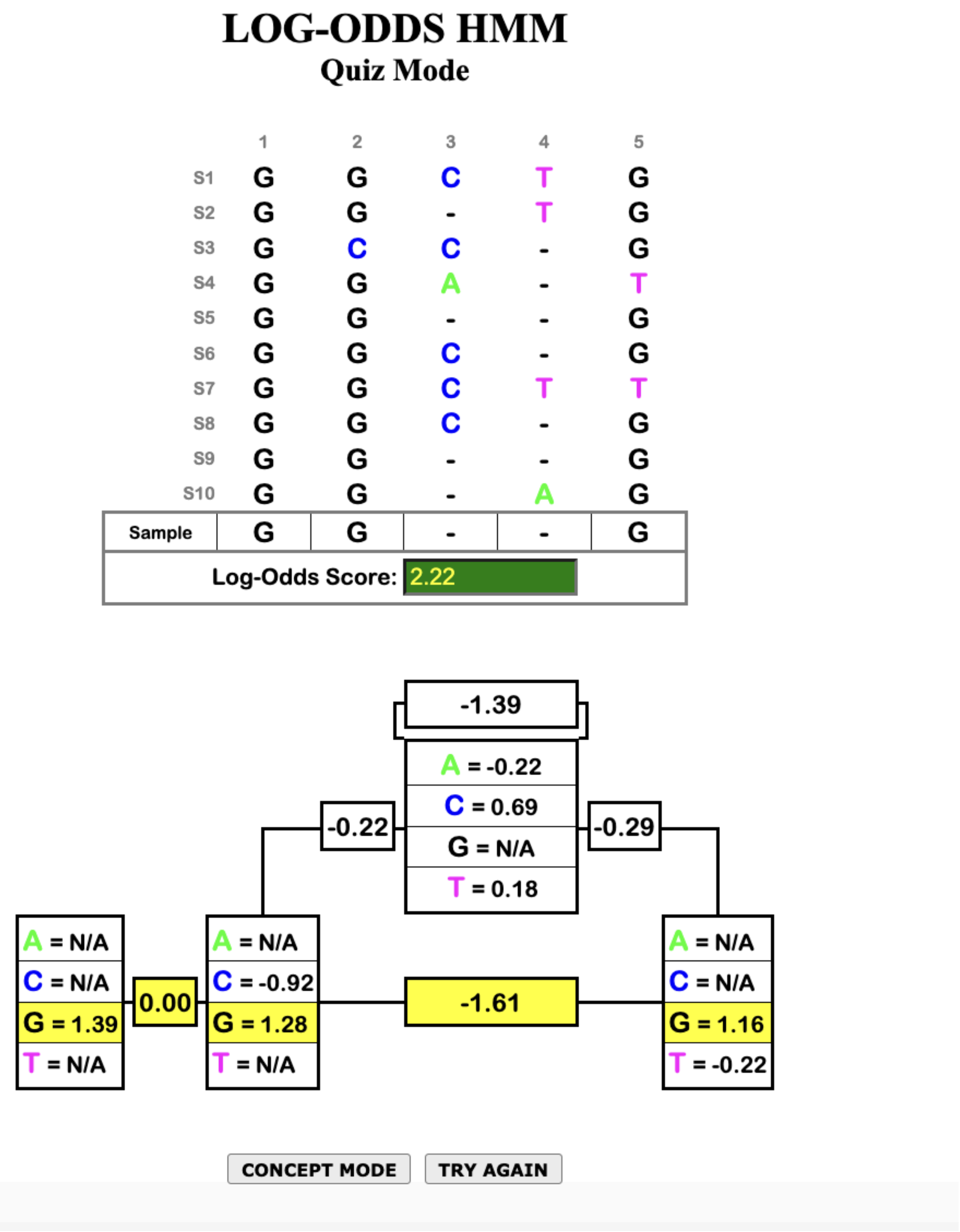


TCCTATCCTAGG

	L	R
TCCTA	TCCT	CCTA
CCTAT	CCTA	CTAT
CTATC	CTAT	TATC
TATCC	TATC	ATCC
ATCCT	ATCC	TCCT
TCCTA	TCCT	CCTA
CCTAG	CCTA	CTAG
CTAGG	CTAG	TAGG



GATCCTATCCTAG

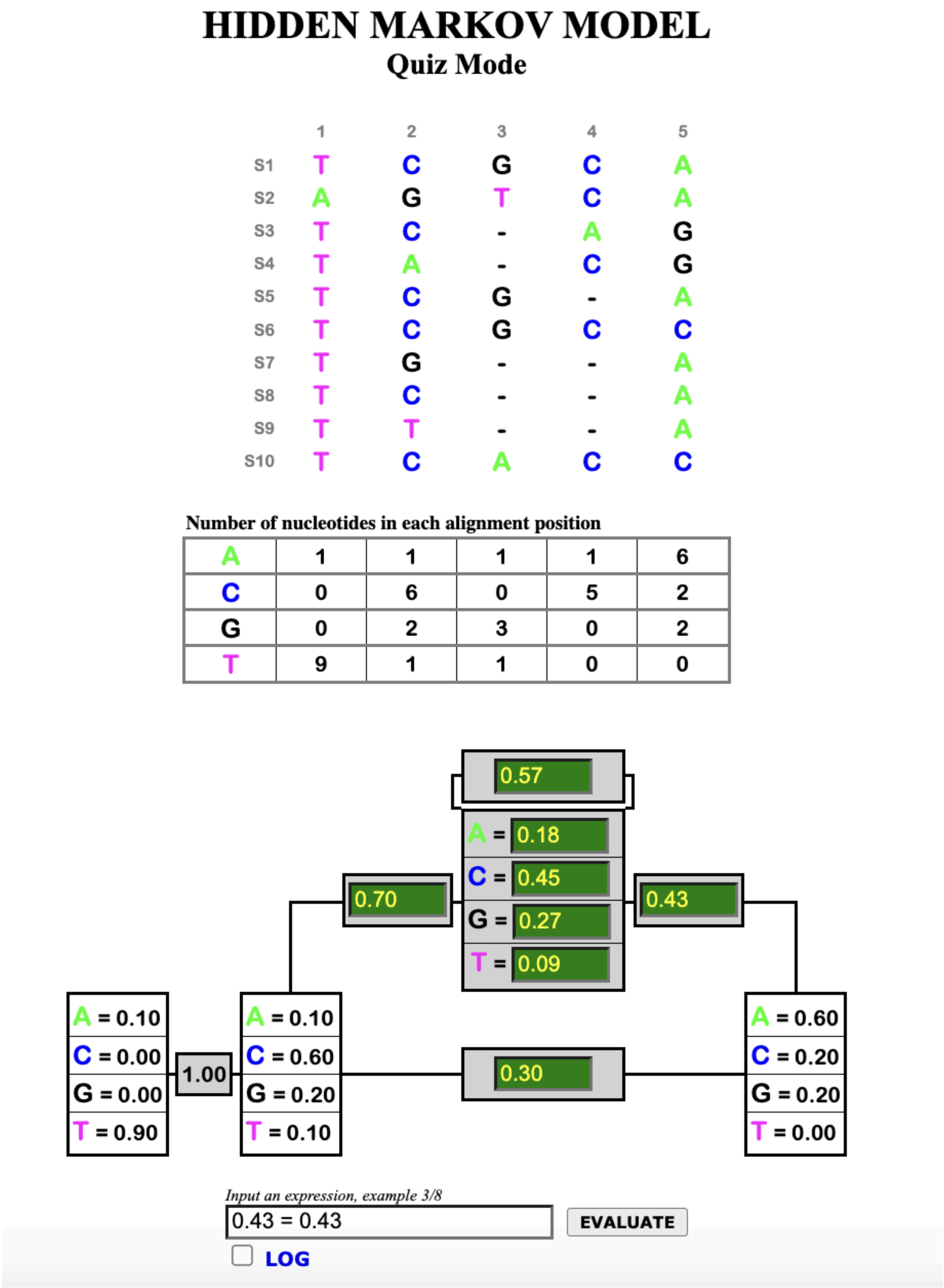


LOG-ODDS HMM

Quiz Mode

	1	2	3	4	5
S1	G	G	C	T	G
S2	G	G	-	T	G
S3	G	C	C	-	G
S4	G	G	A	-	T
S5	G	G	-	-	G
S6	G	G	C	-	G
S7	G	G	C	T	T
S8	G	G	C	-	G
S9	G	G	-	-	G
S10	G	G	-	A	G
Sample	G	G	-	-	G

Log-Odds Score: 2.22



HIDDEN MARKOV MODEL

Quiz Mode

	1	2	3	4	5
S1	T	C	G	C	A
S2	A	G	T	C	A
S3	T	C	-	A	G
S4	T	A	-	C	G
S5	T	C	G	-	A
S6	T	C	G	C	C
S7	T	G	-	-	A
S8	T	C	-	-	A
S9	T	T	-	-	A
S10	T	C	A	C	C

Number of nucleotides in each alignment position

A	1	1	1	1	6
C	0	6	0	5	2
G	0	2	3	0	2
T	9	1	1	0	0

0.57

A = 0.18
C = 0.45
G = 0.27
T = 0.09

0.70

0.43

A = 0.10
C = 0.00
G = 0.00
T = 0.90

1.00

A = 0.10
C = 0.60
G = 0.20
T = 0.10

0.30

A = 0.60
C = 0.20
G = 0.20
T = 0.00

Input an expression, example 3/8

0.43 = 0.43

EVALUATE

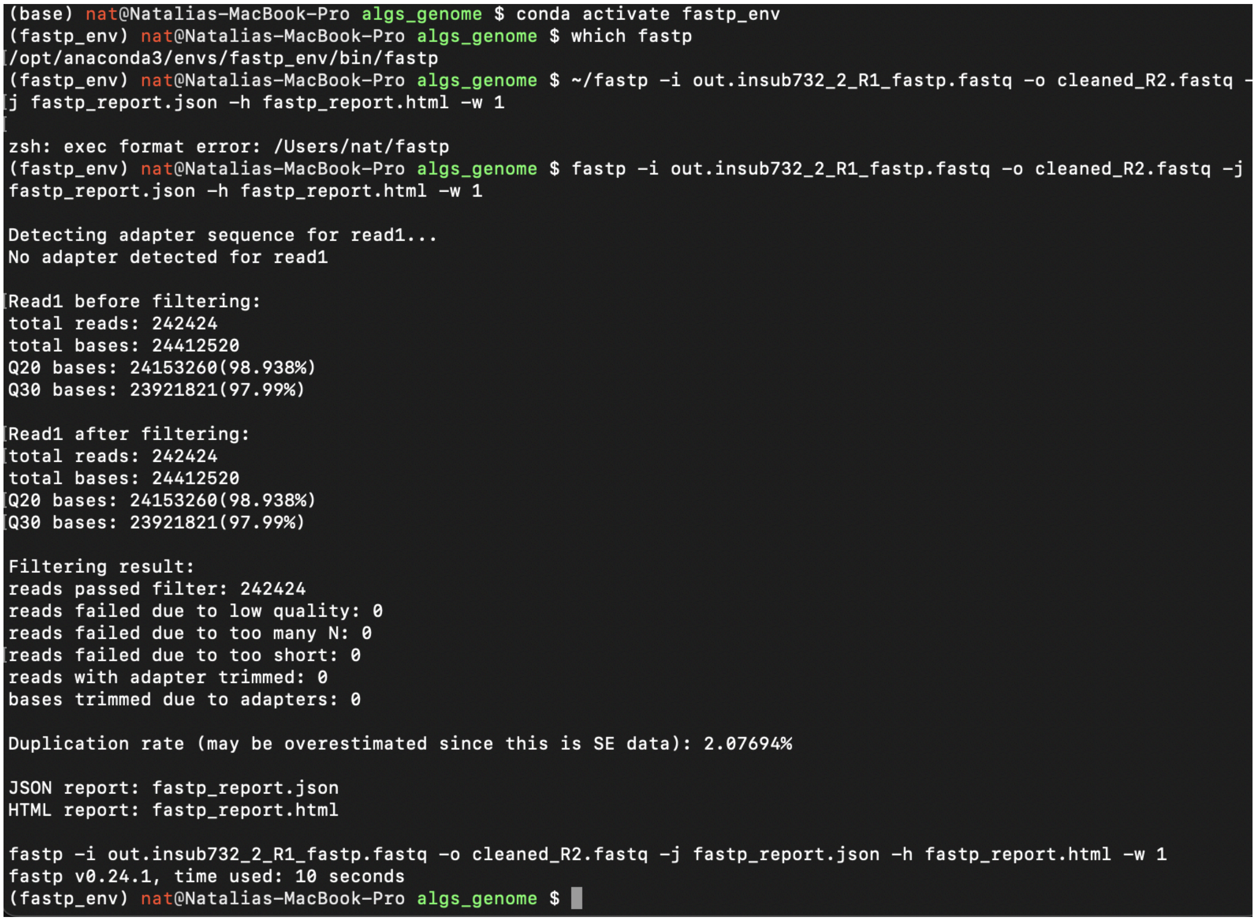
□ LOG

Fastp Installation:

I ran “conda install -c bioconda fastp” however I did not take a screenshot of this command. Essentially, since I already installed anaconda, I easily got fastp installed. I then exported the path to be able to use in any directory.

Process pair-read ends:

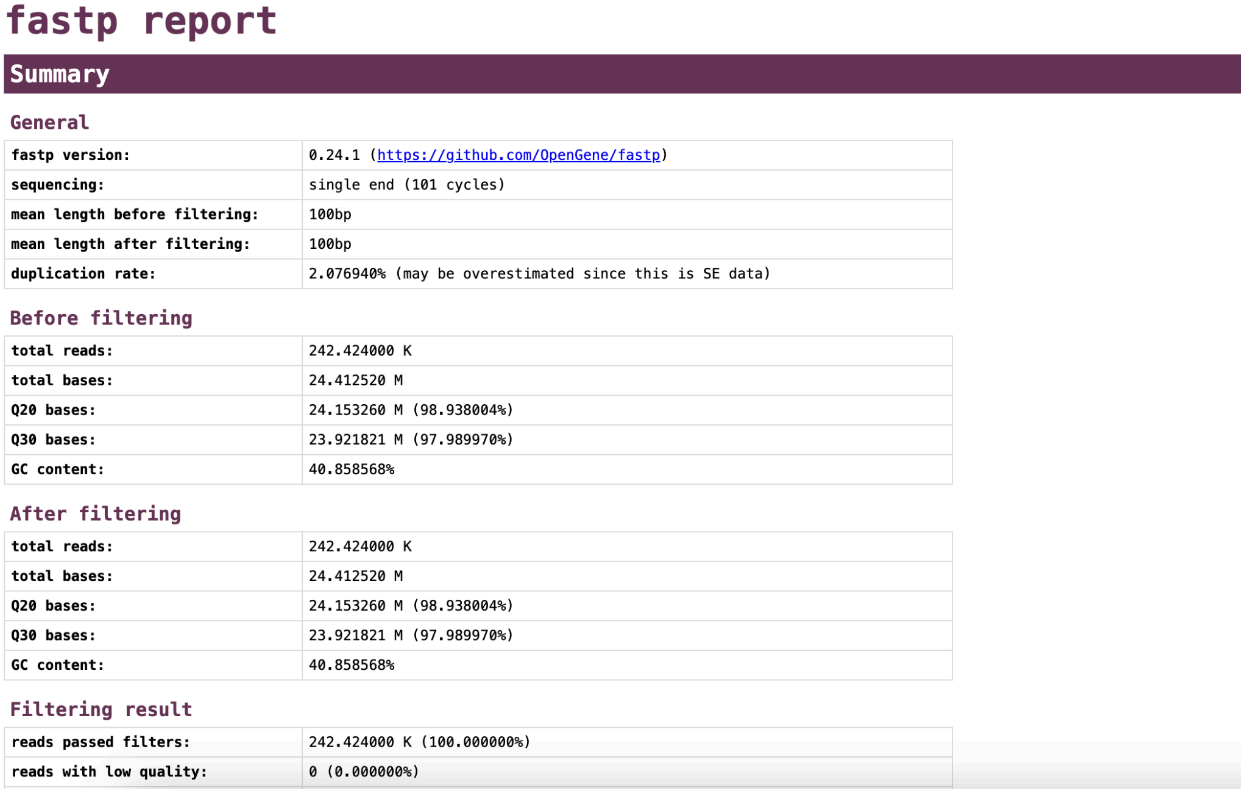
The code below is essentially running fastp, taking out.insub732_2_R1_fastp.fastq as the input, and specifying cleaned_R2.fastq as the output file (since the assignment said we were given R2, however the the file we got says R1). It also generates json and html reports for viewing. Fastp takes the fastq file containing short read and performs preprocessing and quality control analysis.



```
(base) nat@Natalias-MacBook-Pro algs_genome $ conda activate fastp_env
(fastp_env) nat@Natalias-MacBook-Pro algs_genome $ which fastp
/opt/anaconda3/envs/fastp_env/bin/fastp
(fastp_env) nat@Natalias-MacBook-Pro algs_genome $ ~/fastp -i out.insub732_2_R1_fastp.fastq -o cleaned_R2.fastq -
j fastp_report.json -h fastp_report.html -w 1

zsh: exec format error: /Users/nat/fastp
(fastp_env) nat@Natalias-MacBook-Pro algs_genome $ fastp -i out.insub732_2_R1_fastp.fastq -o cleaned_R2.fastq -j
fastp_report.json -h fastp_report.html -w 1

Detecting adapter sequence for read1...
No adapter detected for read1

Read1 before filtering:
total reads: 242424
total bases: 24412520
Q20 bases: 24153260(98.938%)
Q30 bases: 23921821(97.99%)

Read1 after filtering:
total reads: 242424
total bases: 24412520
Q20 bases: 24153260(98.938%)
Q30 bases: 23921821(97.99%)

Filtering result:
reads passed filter: 242424
reads failed due to low quality: 0
reads failed due to too many N: 0
reads failed due to too short: 0
reads with adapter trimmed: 0
bases trimmed due to adapters: 0

Duplication rate (may be overestimated since this is SE data): 2.07694%

JSON report: fastp_report.json
HTML report: fastp_report.html

fastp -i out.insub732_2_R1_fastp.fastq -o cleaned_R2.fastq -j fastp_report.json -h fastp_report.html -w 1
fastp v0.24.1, time used: 10 seconds
(fastp_env) nat@Natalias-MacBook-Pro algs_genome $
```

Fastp Output:

This is a Html output file containing statistics regarding the preprocessing and quality control of the R2 file. It gives information about how many reads were trimmed, quality scores, gc content and informative graphs, assessing overall read quality post-trimming. Additionally, the k-mer content graph reveals the abundance of specific k-mers across the dataset, darker regions indicate higher k-mer frequency, which can help identify overrepresented sequences.



fastp report

Summary

General

fastp version:	0.24.1 (https://github.com/OpenGene/fastp)
sequencing:	single end (101 cycles)
mean length before filtering:	100bp
mean length after filtering:	100bp
duplication rate:	2.076940% (may be overestimated since this is SE data)

Before filtering

total reads:	242.424000 K
total bases:	24.412520 M
Q20 bases:	24.153260 M (98.938004%)
Q30 bases:	23.921821 M (97.989970%)
GC content:	40.858568%

After filtering

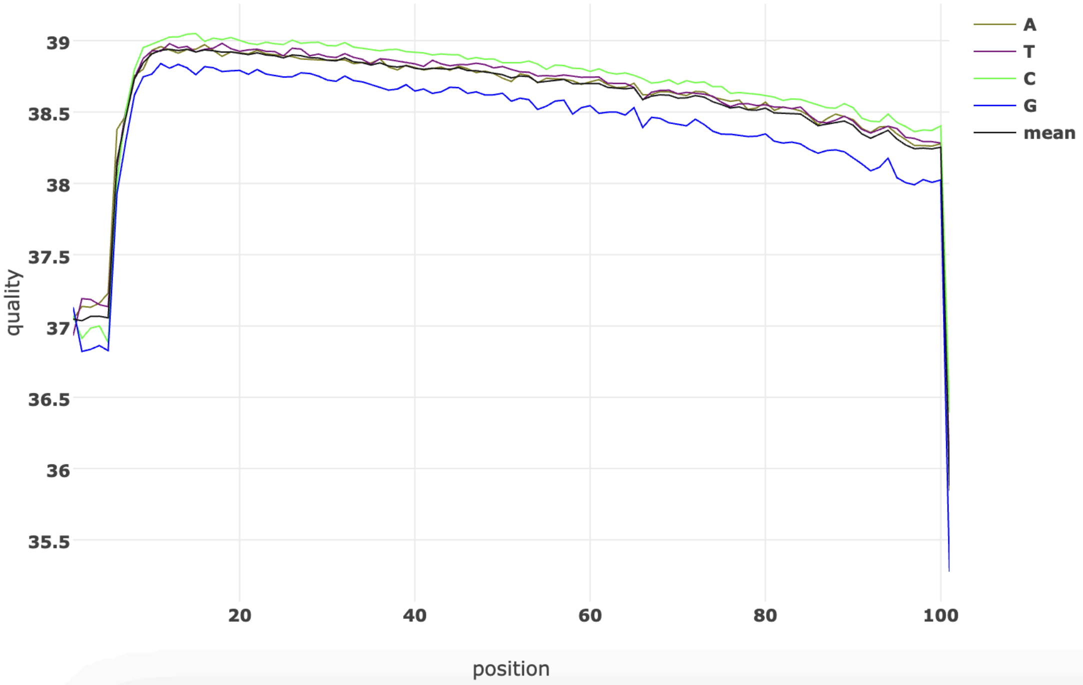
total reads:	242.424000 K
total bases:	24.412520 M
Q20 bases:	24.153260 M (98.938004%)
Q30 bases:	23.921821 M (97.989970%)
GC content:	40.858568%

Filtering result

reads passed filters:	242.424000 K (100.000000%)
reads with low quality:	0 (0.000000%)

After filtering: read1: quality

Value of each position will be shown on mouse over.

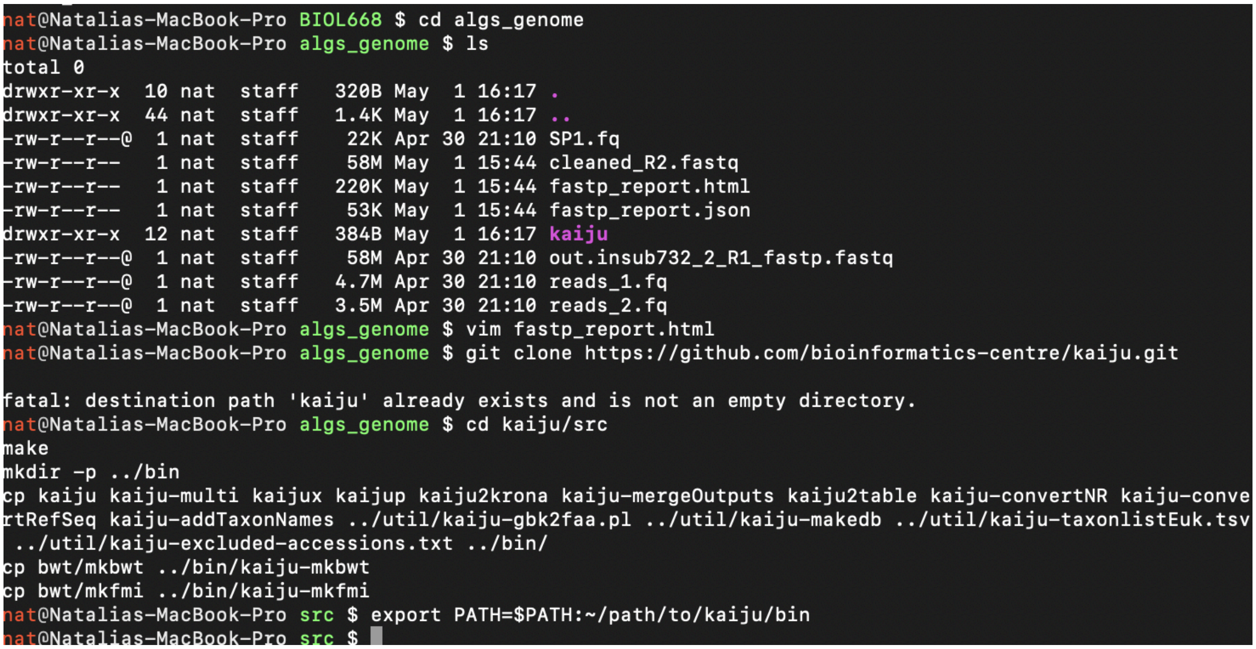


Darker background means larger counts. The count will be shown on mouse over.

	AA	AT	AC	AG	TA	TT	TC	TG	CA	CT	CC	CG	GA	GT	GC	GG
AAA	AAAAA	AAAAT	AAAAC	AAAAG	AAATA	AAATT	AAATC	AAATG	AAACA	AAACT	AAACC	AAACG	AAAGA	AAAGT	AAAGC	AAAGG
AAT	AATAA	AATAT	AATAC	AATAG	AATTA	AATTT	AATTC	AATTG	AATCA	AATCT	AATCC	AATCG	AATGA	AATGT	AATGC	AATGG
AAC	AACAA	AACAT	AACAC	AACAG	AACTA	AACTT	AACTC	AACTG	AACCA	AACCT	AACCC	AACCG	AACGA	AACGT	AACGC	AACGG
AAG	AAGAA	AAGAT	AAGAC	AAGAG	AAGTA	AAGTT	AAGTC	AAGTG	AAGCA	AAGCT	AAGCC	AAGCG	AAGGA	AAGGT	AAGGC	AAGGG
ATA	ATAAA	ATAAT	ATAAC	ATAAG	ATATA	ATATT	ATATC	ATATG	ATACA	ATACT	ATACC	ATACG	ATAGA	ATAGT	ATAGC	ATAGG
ATT	ATTAA	ATTAT	ATTAC	ATTAG	ATTTA	ATTTT	ATTTC	ATTTG	ATTCA	ATTCT	ATTCC	ATTCG	ATTGA	ATTGT	ATTGC	ATTGG
ATC	ATCAA	ATCAT	ATCAC	ATCAG	ATCTA	ATCTT	ATCTC	ATCTG	ATCCA	ATCCT	ATCCC	ATCCG	ATCGA	ATCGT	ATCGC	ATCGG
ATG	ATGAA	ATGAT	ATGAC	ATGAG	ATGTA	ATGTT	ATGTC	ATGTG	ATGCA	ATGCT	ATGCC	ATGCG	ATGGA	ATGGT	ATGGC	ATGGG
ACA	ACAAA	ACAAT	ACAAC	ACAAG	ACATA	ACATT	ACATC	ACATG	ACACA	ACACT	ACACC	ACACG	ACAGA	ACAGT	ACAGC	ACAGG
ACT	ACTAA	ACTAT	ACTAC	ACTAG	ACTTA	ACTTT	ACTTC	ACTTG	ACTCA	ACTCT	ACTCC	ACTCG	ACTGA	ACTGT	ACTGC	ACTGG
ACC	ACCAA	ACCAT	ACCAC	ACCAG	ACCTA	ACCTT	ACCTC	ACCTG	ACCCA	ACCCT	ACCCC	ACCCG	ACCGA	ACCGT	ACCGC	ACCGG
ACG	ACGAA	ACGAT	ACGAC	ACGAG	ACGTA	ACGTT	ACGTC	ACGTG	ACGCA	ACGCT	ACGCC	ACGCG	ACGGA	ACGGT	ACGGC	ACGGG
AGA	AGAAA	AGAAT	AGAAC	AGAAG	AGATA	AGATT	AGATC	AGATG	AGACA	AGACT	AGACC	AGACG	AGAGA	AGAGT	AGAGC	AGAGG
AGT	AGTAA	AGTAT	AGTAC	AGTAG	AGTTA	AGTTT	AGTTC	AGTTG	AGTCA	AGTCT	AGTCC	AGTCG	AGTGA	AGTGT	AGTGC	AGTGG
AGC	AGCAA	AGCAT	AGCAC	AGCAG	AGCTA	AGCTT	AGCTC	AGCTG	AGCCA	AGCCT	AGCCC	AGCCG	AGCGA	AGCGT	AGCGC	AGCGG
AGG	AGGAA	AGGAT	AGGAC	AGGAG	AGGTA	AGGTT	AGGTC	AGGTG	AGGCA	AGGCT	AGGCC	AGGCG	AGGGA	AGGGT	AGGGC	AGGGG
TAA	TAAAA	TAAAT	TAAAC	TAAAG	TAATA	TAATT	TAATC	TAATG	TAACA	TAACT	TAACC	TAACG	TAAGA	TAAGT	TAAGC	TAAGG
TAT	TATAA	TATAT	TATAC	TATAG	TATTA	TATTT	TATTC	TATTG	TATCA	TATCT	TATCC	TATCG	TATGA	TATGT	TATGC	TATGG
TAC	TACAA	TACAT	TACAC	TACAG	TACTA	TACTT	TACTC	TACTG	TACCA	TACCT	TACCC	TACCG	TACGA	TACGT	TACGC	TACGG
TAG	TAGAA	TAGAT	TAGAC	TAGAG	TAGTA	TAGTT	TAGTC	TAGTG	TAGCA	TAGCT	TAGCC	TAGCG	TAGGA	TAGGT	TAGGC	TAGGG
TTA	TTAAA	TTAAT	TTAAC	TTAAG	TTATA	TTATT	TTATC	TTATG	TTACA	TTACT	TTACC	TTACG	TTAGA	TTAGT	TTAGC	TTAGG
TTT	TTTAA	TTTAT	TTTAC	TTTAG	TTTTA	TTTTT	TTTTC	TTTTG	TTTCA	TTTCT	TTTCC	TTTCG	TTTGA	TTTGT	TTTGC	TTTGG
TTC	TTCAA	TTCAT	TTCAC	TTCAG	TTCTA	TTCTT	TTCTC	TTCTG	TTCCA	TTCCT	TTCCC	TTCCG	TTCGA	TTCGT	TTCGC	TTCGG
TTG	TTGAA	TTGAT	TTGAC	TTGAG	TTGTA	TTGTT	TTGTC	TTGTG	TTGCA	TTGCT	TTGCC	TTGCG	TTGGA	TTGGT	TTGGC	TTGGG
TCA	TCAAA	TCAAT	TCAAC	TCAAG	TCATA	TCATT	TCATC	TCATG	TCACA	TCACT	TCACC	TCACG	TCAGA	TCAGT	TCAGC	TCAGG
TCT	TCTAA	TCTAT	TCTAC	TCTAG	TCTTA	TCTTT	TCTTC	TCTTG	TCTCA	TCTCT	TCTCC	TCTCG	TCTGA	TCTGT	TCTGC	TCTGG
TCC	TCCAA	TCCAT	TCCAC	TCCAG	TCCTA	TCCTT	TCCTC	TCCTG	TCCCA	TCCCT	TCCCC	TCCCG	TCCGA	TCCGT	TCCGC	TCCGG
TCG	TCGAA	TCGAT	TCGAC	TCGAG	TCGTA	TCGTT	TCGTC	TCGTG	TCGCA	TCGCT	TCGCC	TCGCG	TCGGA	TCGGT	TCGGC	TCGGG
TGA	TGAAA	TGAAT	TGAAC	TGAAG	TGATA	TGATT	TGATC	TGATG	TGACA	TGACT	TGACC	TGACG	TGAGA	TGAGT	TGAGC	TGAGG
TGT	TGTAA	TGTAT	TGTAC	TGTAG	TGTTA	TGTTT	TGTTC	TGTTG	TGTCA	TGTCT	TGTCC	TGTCG	TGTGA	TGTGT	TGTGC	TGTGG
TGC	TGCAA	TGCAT	TGCAC	TGCAG	TGCTA	TGCTT	TGCTC	TGCTG	TGCCA	TGCCT	TGCCC	TGCCG	TGCGA	TGCGT	TGCGC	TGCGG
TGG	TGGAA	TGGAT	TGGAC	TGGAG	TGGTA	TGGTT	TGGTC	TGGTG	TGGCA	TGGCT	TGGCC	TGGCG	TGGGA	TGGGT	TGGGC	TGGGG
CAA	CAAAA	CAAAT	CAAAC	CAAAG	CAATA	CAATT	CAATC	CAATG	CAACA	CAACT	CAACC	CAACG	CAAGA	CAAGT	CAAGC	CAAGG
CAT	CATAA	CATAT	CATAC	CATAG	CATTA	CATTT	CATTC	CATTG	CATCA	CATCT	CATCC	CATCG	CATGA	CATGT	CATGC	CATGG
CAC	CACAA	CACAT	CACAC	CACAG	CACTA	CACTT	CACTC	CACTG	CACCA	CACCT	CACCC	CACCG	CACGA	CACGT	CACGC	CACGG
CAG	CAGAA	CAGAT	CAGAC	CAGAG	CAGTA	CAGTT	CAGTC	CAGTG	CAGCA	CAGCT	CAGCC	CAGCG	CAGGA	CAGGT	CAGGC	CAGGG
CTA	CTAAA	CTAAT	CTAAC	CTAAG	CTATA	CTATT	CTATC	CTATG	CTACA	CTACT	CTACC	CTACG	CTAGA	CTAGT	CTAGC	CTAGG
CTT	CTTAA	CTTAT	CTTAC	CTTAG	CTTTA	CTTTT	CTTTC	CTTTG	CTTCA	CTTCT	CTTCC	CTTCG	CTTGA	CTTGT	CTTGC	CTTGG
CTC	CTCAA	CTCAT	CTCAC	CTCAG	CTCTA	CTCTT	CTCTC	CTCTG	CTCCA	CTCCT	CTCCC	CTCCG	CTCGA	CTCGT	CTCGC	CTCGG
CTG	CTGAA	CTGAT	CTGAC	CTGAG	CTGTA	CTGTT	CTGTC	CTGTG	CTGCA	CTGCT	CTGCC	CTGCG	CTGGA	CTGGT	CTGGC	CTGGG
CCA	CCAAA	CCAAT	CCAAC	CCAAG	CCATA	CCATT	CCATC	CCATG	CCACA	CCACT	CCACC	CCACG	CCAGA	CCAGT	CCAGC	CCAGG
CCT	CCTAA	CCTAT	CCTAC	CCTAG	CCTTA	CCTTT	CCTTC	CCTTG	CCTCA	CCTCT	CCTCC	CCTCG	CCTGA	CCTGT	CCTGC	CCTGG
CCC	CCCAA	CCCAT	CCCAC	CCCAG	CCCTA	CCCTT	CCCTC	CCCTG	CCCCA	CCCCT	CCCCC	CCCCG	CCCGA	CCCGT	CCCGC	CCCGG
CCG	CCGAA	CCGAT	CCGAC	CCGAG	CCGTA	CCGTT	CCGTC	CCGTG	CCGCA	CCGCT	CCGCC	CCGCG	CCGGA	CCGGT	CCGGC	CCGGG
CGA	CGAAA	CGAAT	CGAAC	CGAAG	CGATA	CGATT	CGATC	CGATG	CGACA	CGACT	CGACC	CGACG	CGAGA	CGAGT	CGAGC	CGAGG
CGT	CGTAA	CGTAT	CGTAC	CGTAG	CGTTA	CGTTT	CGTTC	CGTTG	CGTCA	CGTCT	CGTCC	CGTCG	CGTGA	CGTGT	CGTGC	CGTGG
CGC	CGCAA	CGCAT	CGCAC	CGCAG	CGCTA	CGCTT	CGCTC	CGCTG	CGCCA	CGCCT	CGCCC	CGCCG	CGCGA	CGCGT	CGCGC	CGCGG
CGG	CGGAA	CGGAT	CGGAC	CGGAG	CGGTA	CGGTT	CGGTC	CGGTG	CGGCA	CGGCT	CGGCC	CGGCG	CGGGA	CGGGT	CGGGC	CGGGG
GAA	GAAAA	GAAAT	GAAAC	GAAAG	GAATA	GAATT	GAATC	GAATG	GAACA	GAACT	GAACC	GAACG	GAAGA	GAAGT	GAAGC	GAAGG
GAT	GATAA	GATAT	GATAC	GATAG	GATTA	GATTT	GATTC	GATTG	GATCA	GATCT	GATCC	GATCG	GATGA	GATGT	GATGC	GATGG
GAC	GACAA	GACAT	GACAC	GACAG	GACTA	GACTT	GACTC	GACTG	GACCA	GACCT	GACCC	GACCG	GACGA	GACGT	GACGC	GACGG
GAG	GAGAA	GAGAT	GAGAC	GAGAG	GAGTA	GAGTT	GAGTC	GAGTG	GAGCA	GAGCT	GAGCC	GAGCG	GAGGA	GAGGT	GAGGC	GAGGG
GTA	GTAAA	GTAAT	GTAAC	GTAAG	GTATA	GTATT	GTATC	GTATG	GTACA	GTACT	GTACC	GTACG	GTAGA	GTAGT	GTAGC	GTAGG
GTT	GTTAA	GTTAT	GTTAC	GTTAG	GTTTA	GTTTT	GTTTC	GTTTG	GTTCA	GTTCT	GTTCC	GTTCG	GTTGA	GTTGT	GTTGC	GTTGG
GTC	GTCAA	GTCAT	GTCAC	GTCAG	GTCTA	GTCTT	GTCTC	GTCTG	GTCCA	GTCCT	GTCCC	GTCCG	GTCGA	GTCGT	GTCGC	GTCGG
GTG	GTGAA	GTGAT	GTGAC	GTGAG	GTGTA	GTGTT	GTGTC	GTGTG	GTGCA	GTGCT	GTGCC	GTGCG	GTGGA	GTGGT	GTGGC	GTGGG
GCA	GCAAA	GCAAT	GCAAC	GCAAG	GCATA	GCATT	GCATC	GCATG	GCACA	GCACT	GCACC	GCACG	GCAGA	GCAGT	GCAGC	GCAGG
GCT	GCTAA	GCTAT	GCTAC	GCTAG	GCTTA	GCTTT	GCTTC	GCTTG	GCTCA	GCTCT	GCTCC	GCTCG	GCTGA	GCTGT	GCTGC	GCTGG
GCC	GCCAA	GCCAT	GCCAC	GCCAG	GCCTA	GCCTT	GCCTC	GCCTG	GCCCA	GCCCT	GCCCC	GCCCG	GCCGA	GCCGT	GCCGC	GCCGG
GCG	GCGAA	GCGAT	GCGAC	GCGAG	GCGTA	GCGTT	GCGTC	GCGTG	GCGCA	GCGCT	GCGCC	GCGCG	GCGGA	GCGGT	GCGGC	GCGGG
GGA	GGAAA	GGAAT	GGAAC	GGAAG	GGATA	GGATT	GGATC	GGATG	GGACA	GGACT	GGACC	GGACG	GGAGA	GGAGT	GGAGC	GGAGG
GGT	GGTAA	GGTAT	GGTAC	GGTAG	GGTTA	GGTTT	GGTTC	GGTTG	GGTCA	GGTCT	GGTCC	GGTCG	GGTGA	GGTGT	GGTGC	GGTGG
GGC	GGCAA	GGCAT	GGCAC	GGCAG	GGCTA	GGCTT	GGCTC	GGCTG	GGCCA	GGCCT	GGCCC	GGCCG	GGCGA	GGCGT	GGCGC	GGCGG
GGG	GGGAA	GGGAT	GGGAC	GGGAG	GGGTA	GGGTT	GGGTC	GGGTG	GGGCA	GGGCT	GGGCC	GGGCG	GGGGA	GGGGT	GGGGC	GGGGG

Kaiju Installation:

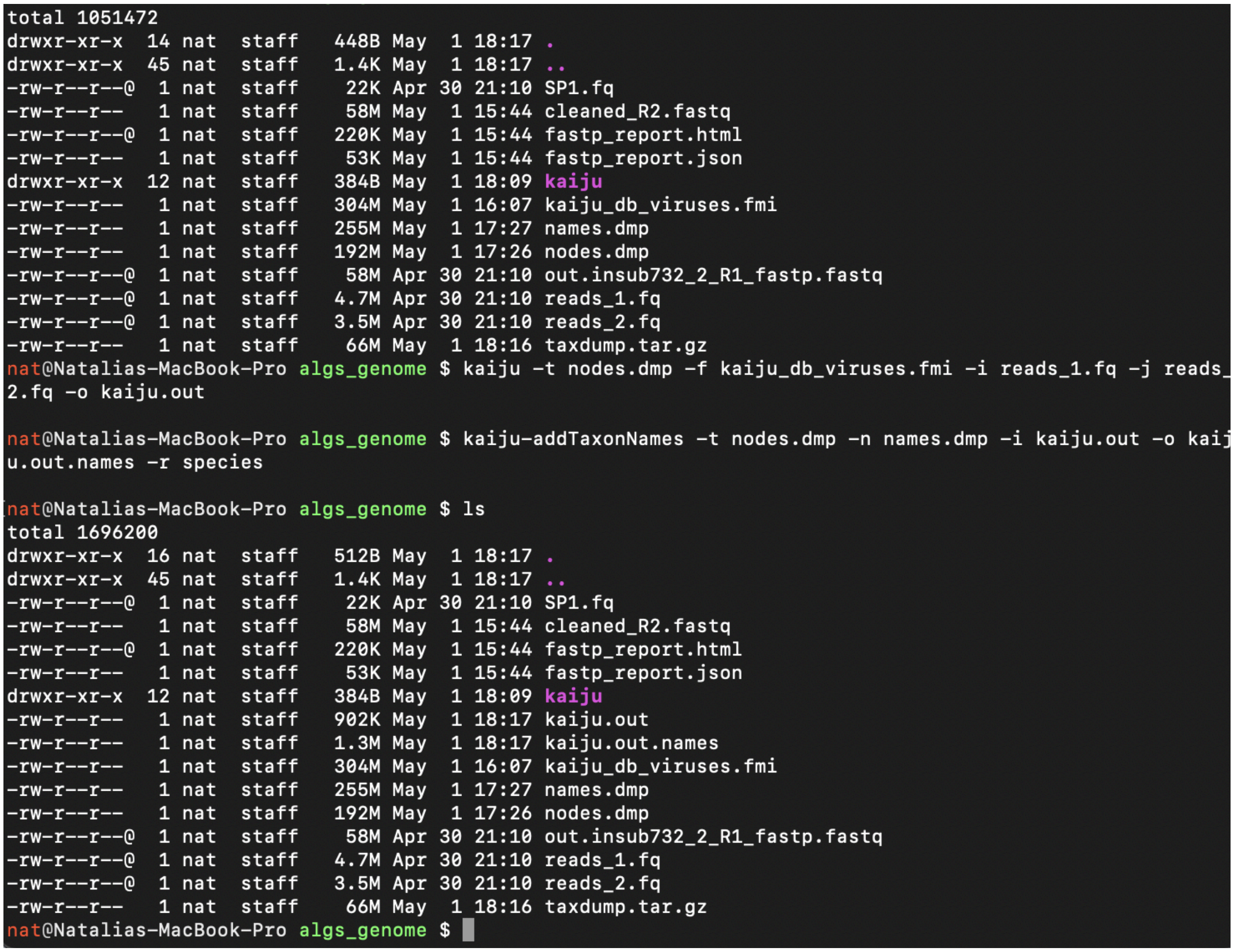
To install Kaiju I used the command git clone, moved into the src directory which contains most of the source code. Then exported the path to be able to use anywhere, particularly mkbwet, mkfmi, makedb, etc.



```
nat@Natalias-MacBook-Pro BIOL668 $ cd algs_genome
nat@Natalias-MacBook-Pro algs_genome $ ls
total 0
drwxr-xr-x  10 nat  staff   320B May  1 16:17 .
drwxr-xr-x  44 nat  staff   1.4K May  1 16:17 ..
-rw-r--r--@  1 nat  staff    22K Apr 30 21:10 SP1.fq
-rw-r--r--   1 nat  staff    58M May  1 15:44 cleaned_R2.fastq
-rw-r--r--   1 nat  staff   220K May  1 15:44 fastp_report.html
-rw-r--r--   1 nat  staff    53K May  1 15:44 fastp_report.json
drwxr-xr-x  12 nat  staff   384B May  1 16:17 kaiju
-rw-r--r--@  1 nat  staff    58M Apr 30 21:10 out.insub732_2_R1_fastp.fastq
-rw-r--r--@  1 nat  staff   4.7M Apr 30 21:10 reads_1.fq
-rw-r--r--@  1 nat  staff   3.5M Apr 30 21:10 reads_2.fq
nat@Natalias-MacBook-Pro algs_genome $ vim fastp_report.html
nat@Natalias-MacBook-Pro algs_genome $ git clone https://github.com/bioinformatics-centre/kaiju.git

fatal: destination path 'kaiju' already exists and is not an empty directory.
nat@Natalias-MacBook-Pro algs_genome $ cd kaiju/src
make
mkdir -p ../bin
cp kaiju kaiju-multi kaijux kaijup kaiju2krona kaiju-mergeOutputs kaiju2table kaiju-convertNR kaiju-conve
rtRefSeq kaiju-addTaxonNames ../util/kaiju-gbk2faa.pl ../util/kaiju-makedb ../util/kaiju-taxonlistEuk.tsv
 ../util/kaiju-excluded-accessions.txt ../bin/
cp bwt/mkbwt ../bin/kaiju-mkbwt
cp bwt/mkfmi ../bin/kaiju-mkfmi
nat@Natalias-MacBook-Pro src $ export PATH=$PATH:~/path/to/kaiju/bin
nat@Natalias-MacBook-Pro src $ 
```

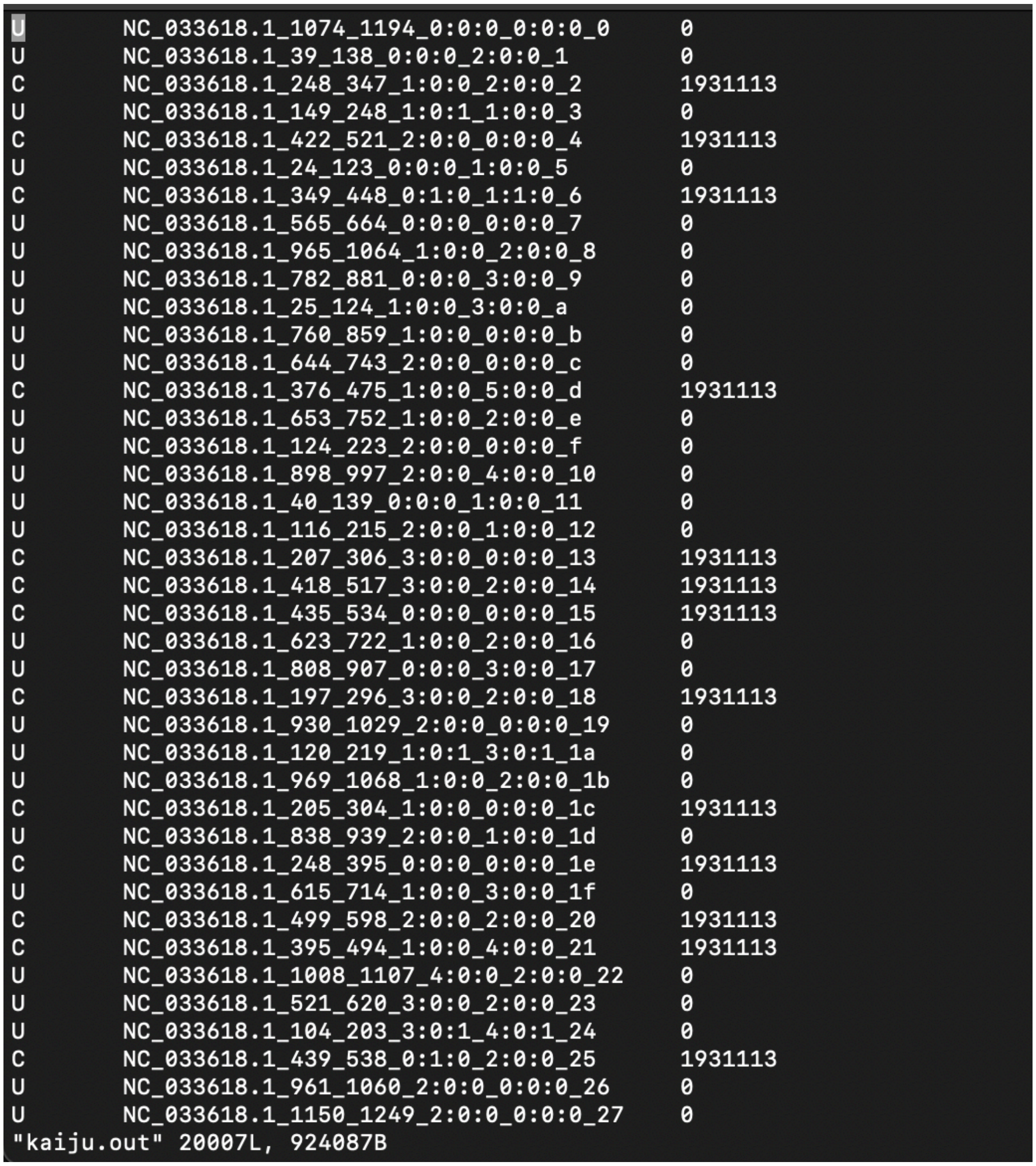
Analyze Metagenomic dataset with viruses:

The code below, kaiju -t nodes.dmp -f kaiju_db_viruses.fmi -i reads_1.fq -j reads_2.fq -o kaiju.out does the following: classifies reads against a viral reference database. The -t option specifies the nodes.dmp file, which contains the NCBI taxonomy tree necessary for mapping taxon IDs. The -f option points to the kaiju_db_viruses.fmi file, which is the indexed viral protein database used during classification. The -i and -j options indicate the forward (reads_1.fq) and reverse (reads_2.fq) paired-end read files. The -o option defines the output file (kaiju.out) that stores the classification results. Each read is evaluated and either assigned to the most probable taxonomic group based on protein sequence similarity or marked as unclassified.

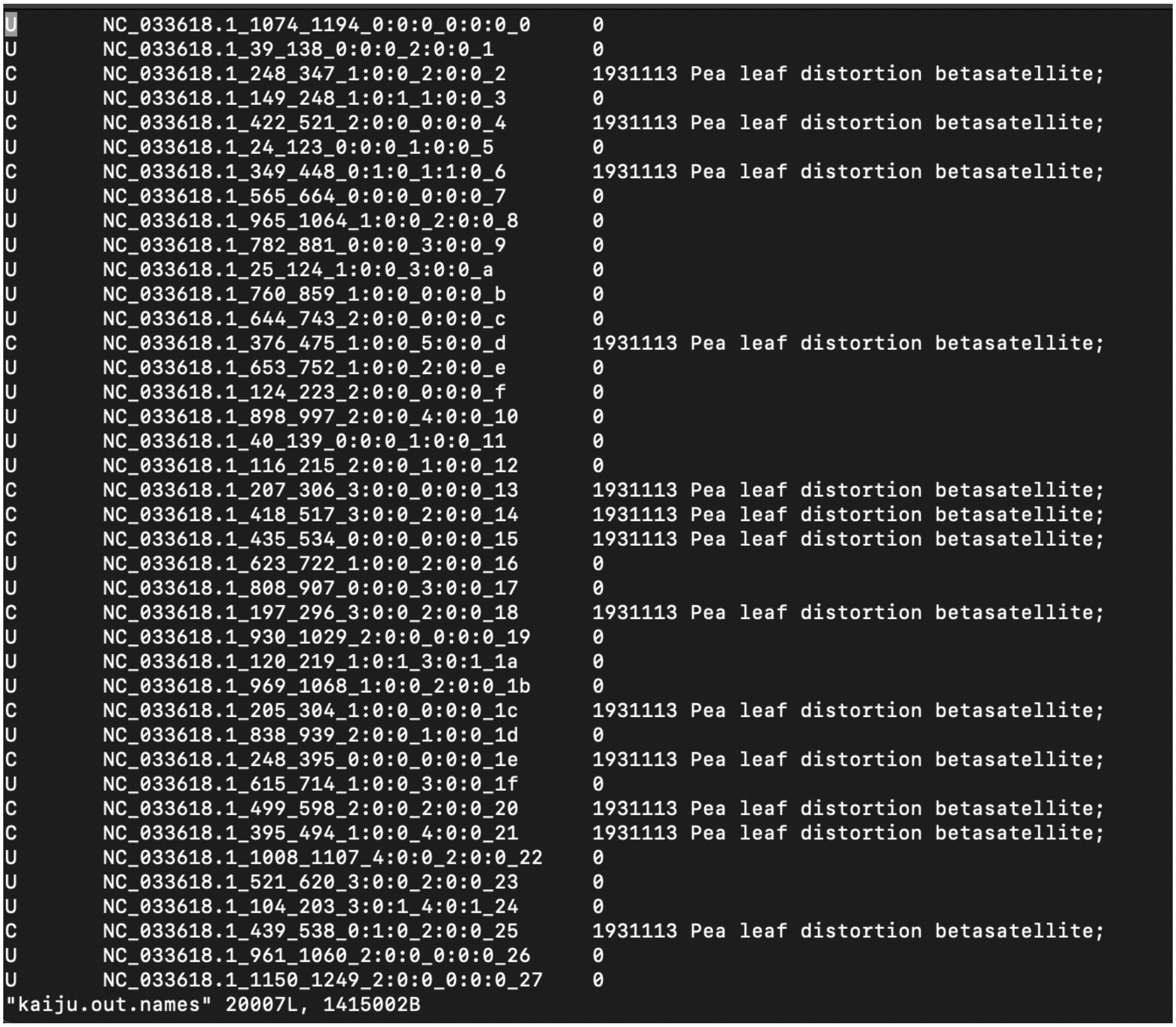


```
total 1051472
drwxr-xr-x  14 nat  staff   448B May  1 18:17 .
drwxr-xr-x  45 nat  staff   1.4K May  1 18:17 ..
-rw-r--r--@  1 nat  staff    22K Apr 30 21:10 SP1.fq
-rw-r--r--   1 nat  staff    58M May  1 15:44 cleaned_R2.fastq
-rw-r--r--@  1 nat  staff   220K May  1 15:44 fastp_report.html
-rw-r--r--   1 nat  staff    53K May  1 15:44 fastp_report.json
drwxr-xr-x  12 nat  staff   384B May  1 18:09 kaiju
-rw-r--r--   1 nat  staff   304M May  1 16:07 kaiju_db_viruses.fmi
-rw-r--r--   1 nat  staff   255M May  1 17:27 names.dmp
-rw-r--r--   1 nat  staff   192M May  1 17:26 nodes.dmp
-rw-r--r--@  1 nat  staff    58M Apr 30 21:10 out.insub732_2_R1_fastp.fastq
-rw-r--r--@  1 nat  staff   4.7M Apr 30 21:10 reads_1.fq
-rw-r--r--@  1 nat  staff   3.5M Apr 30 21:10 reads_2.fq
-rw-r--r--   1 nat  staff    66M May  1 18:16 taxdump.tar.gz
nat@Natalias-MacBook-Pro algs_genome $ kaiju -t nodes.dmp -f kaiju_db_viruses.fmi -i reads_1.fq -j reads_
2.fq -o kaiju.out

nat@Natalias-MacBook-Pro algs_genome $ kaiju-addTaxonNames -t nodes.dmp -n names.dmp -i kaiju.out -o kaij
u.out.names -r species

nat@Natalias-MacBook-Pro algs_genome $ ls
total 1696200
drwxr-xr-x  16 nat  staff   512B May  1 18:17 .
drwxr-xr-x  45 nat  staff   1.4K May  1 18:17 ..
-rw-r--r--@  1 nat  staff    22K Apr 30 21:10 SP1.fq
-rw-r--r--   1 nat  staff    58M May  1 15:44 cleaned_R2.fastq
-rw-r--r--@  1 nat  staff   220K May  1 15:44 fastp_report.html
-rw-r--r--   1 nat  staff    53K May  1 15:44 fastp_report.json
drwxr-xr-x  12 nat  staff   384B May  1 18:09 kaiju
-rw-r--r--   1 nat  staff   902K May  1 18:17 kaiju.out
-rw-r--r--   1 nat  staff   1.3M May  1 18:17 kaiju.out.names
-rw-r--r--   1 nat  staff   304M May  1 16:07 kaiju_db_viruses.fmi
-rw-r--r--   1 nat  staff   255M May  1 17:27 names.dmp
-rw-r--r--   1 nat  staff   192M May  1 17:26 nodes.dmp
-rw-r--r--@  1 nat  staff    58M Apr 30 21:10 out.insub732_2_R1_fastp.fastq
-rw-r--r--@  1 nat  staff   4.7M Apr 30 21:10 reads_1.fq
-rw-r--r--@  1 nat  staff   3.5M Apr 30 21:10 reads_2.fq
-rw-r--r--   1 nat  staff    66M May  1 18:16 taxdump.tar.gz
nat@Natalias-MacBook-Pro algs_genome $ 
```

Output:
One of the kaiju outputs is the kaigu.out file, it shows classification results for each read in the dataset. Each line corresponds to a single read and begins with either a "C" (classified) or "U" (unclassified). Classified reads are followed by the read ID, the NCBI taxon ID to which the read was assigned. This output provides the foundational data for understanding which organisms are present in the sample. To make the results easier to interpret, the kaiju.out file was processed using kaiju-addTaxonNames, which generates a new file called kaiju.out.names. This file replaces the numerical NCBI taxon IDs with actual species names, making it easier to understand which organisms the reads were assigned to. For example, a line beginning with "C" may show a read classified as *Escherichia coli*, providing a better description of the sample composition.



```
U       NC_033618.1_1074_1194_0:0:0_0:0:0_0     0
U       NC_033618.1_39_138_0:0:0_2:0:0_1        0
C       NC_033618.1_248_347_1:0:0_2:0:0_2       1931113
U       NC_033618.1_149_248_1:0:1_1:0:0_3       0
C       NC_033618.1_422_521_2:0:0_0:0:0_4       1931113
U       NC_033618.1_24_123_0:0:0_1:0:0_5        0
C       NC_033618.1_349_448_0:1:0_1:1:0_6       1931113
U       NC_033618.1_565_664_0:0:0_0:0:0_7       0
U       NC_033618.1_965_1064_1:0:0_2:0:0_8      0
U       NC_033618.1_782_881_0:0:0_3:0:0_9       0
U       NC_033618.1_25_124_1:0:0_3:0:0_a        0
U       NC_033618.1_760_859_1:0:0_0:0:0_b       0
U       NC_033618.1_644_743_2:0:0_0:0:0_c       0
C       NC_033618.1_376_475_1:0:0_5:0:0_d       1931113
U       NC_033618.1_653_752_1:0:0_2:0:0_e       0
U       NC_033618.1_124_223_2:0:0_0:0:0_f       0
U       NC_033618.1_898_997_2:0:0_4:0:0_10      0
U       NC_033618.1_40_139_0:0:0_1:0:0_11       0
U       NC_033618.1_116_215_2:0:0_1:0:0_12      0
C       NC_033618.1_207_306_3:0:0_0:0:0_13      1931113
C       NC_033618.1_418_517_3:0:0_2:0:0_14      1931113
C       NC_033618.1_435_534_0:0:0_0:0:0_15      1931113
U       NC_033618.1_623_722_1:0:0_2:0:0_16      0
U       NC_033618.1_808_907_0:0:0_3:0:0_17      0
C       NC_033618.1_197_296_3:0:0_2:0:0_18      1931113
U       NC_033618.1_930_1029_2:0:0_0:0:0_19     0
U       NC_033618.1_120_219_1:0:1_3:0:1_1a      0
U       NC_033618.1_969_1068_1:0:0_2:0:0_1b     0
C       NC_033618.1_205_304_1:0:0_0:0:0_1c      1931113
U       NC_033618.1_838_939_2:0:0_1:0:0_1d      0
C       NC_033618.1_248_395_0:0:0_0:0:0_1e      1931113
U       NC_033618.1_615_714_1:0:0_3:0:0_1f      0
C       NC_033618.1_499_598_2:0:0_2:0:0_20      1931113
C       NC_033618.1_395_494_1:0:0_4:0:0_21      1931113
U       NC_033618.1_1008_1107_4:0:0_2:0:0_22    0
U       NC_033618.1_521_620_3:0:0_2:0:0_23      0
U       NC_033618.1_104_203_3:0:1_4:0:1_24      0
C       NC_033618.1_439_538_0:1:0_2:0:0_25      1931113
U       NC_033618.1_961_1060_2:0:0_0:0:0_26     0
U       NC_033618.1_1150_1249_2:0:0_0:0:0_27    0
"kaiju.out" 20007L, 924087B
```



```
U       NC_033618.1_1074_1194_0:0:0_0:0:0_0     0
U       NC_033618.1_39_138_0:0:0_2:0:0_1        0
C       NC_033618.1_248_347_1:0:0_2:0:0_2       1931113 Pea leaf distortion betasatellite;
U       NC_033618.1_149_248_1:0:1_1:0:0_3       0
C       NC_033618.1_422_521_2:0:0_0:0:0_4       1931113 Pea leaf distortion betasatellite;
U       NC_033618.1_24_123_0:0:0_1:0:0_5        0
C       NC_033618.1_349_448_0:1:0_1:1:0_6       1931113 Pea leaf distortion betasatellite;
U       NC_033618.1_565_664_0:0:0_0:0:0_7       0
U       NC_033618.1_965_1064_1:0:0_2:0:0_8      0
U       NC_033618.1_782_881_0:0:0_3:0:0_9       0
U       NC_033618.1_25_124_1:0:0_3:0:0_a        0
U       NC_033618.1_760_859_1:0:0_0:0:0_b       0
U       NC_033618.1_644_743_2:0:0_0:0:0_c       0
C       NC_033618.1_376_475_1:0:0_5:0:0_d       1931113 Pea leaf distortion betasatellite;
U       NC_033618.1_653_752_1:0:0_2:0:0_e       0
U       NC_033618.1_124_223_2:0:0_0:0:0_f       0
U       NC_033618.1_898_997_2:0:0_4:0:0_10      0
U       NC_033618.1_40_139_0:0:0_1:0:0_11       0
U       NC_033618.1_116_215_2:0:0_1:0:0_12      0
C       NC_033618.1_207_306_3:0:0_0:0:0_13      1931113 Pea leaf distortion betasatellite;
C       NC_033618.1_418_517_3:0:0_2:0:0_14      1931113 Pea leaf distortion betasatellite;
C       NC_033618.1_435_534_0:0:0_0:0:0_15      1931113 Pea leaf distortion betasatellite;
U       NC_033618.1_623_722_1:0:0_2:0:0_16      0
U       NC_033618.1_808_907_0:0:0_3:0:0_17      0
C       NC_033618.1_197_296_3:0:0_2:0:0_18      1931113 Pea leaf distortion betasatellite;
U       NC_033618.1_930_1029_2:0:0_0:0:0_19     0
U       NC_033618.1_120_219_1:0:1_3:0:1_1a      0
U       NC_033618.1_969_1068_1:0:0_2:0:0_1b     0
C       NC_033618.1_205_304_1:0:0_0:0:0_1c      1931113 Pea leaf distortion betasatellite;
U       NC_033618.1_838_939_2:0:0_1:0:0_1d      0
C       NC_033618.1_248_395_0:0:0_0:0:0_1e      1931113 Pea leaf distortion betasatellite;
U       NC_033618.1_615_714_1:0:0_3:0:0_1f      0
C       NC_033618.1_499_598_2:0:0_2:0:0_20      1931113 Pea leaf distortion betasatellite;
C       NC_033618.1_395_494_1:0:0_4:0:0_21      1931113 Pea leaf distortion betasatellite;
U       NC_033618.1_1008_1107_4:0:0_2:0:0_22    0
U       NC_033618.1_521_620_3:0:0_2:0:0_23      0
U       NC_033618.1_104_203_3:0:1_4:0:1_24      0
C       NC_033618.1_439_538_0:1:0_2:0:0_25      1931113 Pea leaf distortion betasatellite;
U       NC_033618.1_961_1060_2:0:0_0:0:0_26     0
U       NC_033618.1_1150_1249_2:0:0_0:0:0_27    0
"kaiju.out.names" 20007L, 1415002B
```