

Exercise Part 1: Command-line Perl

1. Identifying duplicates in two FASTA files (awk)

Say we have two FASTA files that contain some duplicate sequences (exact match) but with different IDs [F1.fasta](#) and [F2.fasta](#).

```
[uzi@quince-srv2 ~/oneliners$ cat f1.fasta
>F1
GTGTCAGCCCGCCGGTATACAGGGGTGCCAAGCCTTGTCCGGATTTACTGGGTAAAGGTCGCCAGGCGGACTTAT
AAGTCGGGGTTAAATCCATGTCTTAAACACATGCAAGGCTTCGGATACTAGGCTCTAGAGTCTCGAAGTCCGGTGA
ACGCTGGAAATGCTAGATATCGGAAGAACAACCGGTGCCAAGGCAGTCTTCTGTCGAGAACTGACGCTCAGGCACGAA
AGCCTGGGGAGCAAACAGGATTAGATACCCCTAGTAGTC
>F2
GTGTCAGCCCGCCGGTATACAGGGGTGCCAAGCCTTGTCCGGATTTACTGGGTAAAGGTCGCCAGGCGGACTTAT
AAGTCGGGGTTAAATCCATGTCTTAAACACATGCAAGGCTTCGGATACTAGGCTCTAGAGTCTCGAAGTCCGGTGA
ACGCTGGAAATGCTAGATATCGGAAGAACAACCGGTGCCAAGGCAGTCTTCTGTCGAGAACTGACGCTCAGGCACGAA
AGCCTGGGGAGCAAACAGGATTAGATACCCCTAGTAGTC
>F3
GTGTCAGCCCGCCGGTATACAGGGGTGCCAAGCCTTGTCCGGATTTACTGGGTAAAGGTCGCCAGGCGGACTTAT
AAGTCGGGGTTAAATCCATGTCTTAAACACATGCAAGGCTTCGGATACTAGGCTCTAGAGTCTCGAAGTCCGGTGA
ACGCTGGAAATGCTAGATATCGGAAGAACAACCGGTGCCAAGGCAGTCTTCTGTCGAGAACTGACGCTCAGGCACGAA
AGCCTGGGGAGCAAACAGGATTAGATACCCCTAGTAGTC
>F4_c1
GTGTCAGCCCGCCGGTATACAGGGGTGCCAAGCCTTGTCCGGATTTACTGGGTAAAGGTCGCCAGGCGGACTTAT
AAGTCGGGGTTAAATCCATGTCTTAAACACATGCAAGGCTTCGGATACTAGGCTCTAGAGTCTCGAAGTCCGGTGA
ACGCTGGAAATGCTAGATATCGGAAGAACAACCGGTGCCAAGGCAGTCTTCTGTCGAGAACTGACGCTCAGGCACGAA
AGCCTGGGGAGCAAACAGGATTAGATACCCCTAGTAGTC
>F5_c2
GTGTCAGCCCGCCGGTATACAGGGGTGCCAAGCCTTGTCCGGATTTACTGGGTAAAGGTCGCCAGGCGGACTTAT
AAGTCGGGGTTAAATCCATGTCTTAAACACATGCAAGGCTTCGGATACTAGGCTCTAGAGTCTCGAAGTCCGGTGA
ACGCTGGAAATGCTAGATATCGGAAGAACAACCGGTGCCAAGGCAGTCTTCTGTCGAGAACTGACGCTCAGGCACGAA
AGCCTGGGGAGCAAACAGGATTAGATACCCCTAGTAGTC
[uzi@quince-srv2 ~/oneliners$ cat f2.fasta
>C1_F4
GTGTCAGCCCGCCGGTATACAGGGGTGCCAAGCCTTGTCCGGATTTACTGGGTAAAGGTCGCCAGGCGGACTTAT
AAGTCGGGGTTAAATCCATGTCTTAAACACATGCAAGGCTTCGGATACTAGGCTCTAGAGTCTCGAAGTCCGGTGA
ACGCTGGAAATGCTAGATATCGGAAGAACAACCGGTGCCAAGGCAGTCTTCTGTCGAGAACTGACGCTCAGGCACGAA
AGCCTGGGGAGCAAACAGGATTAGATACCCCTAGTAGTC
>C2_F5
GTGTCAGCCCGCCGGTATACAGGGGTGCCAAGCCTTGTCCGGATTTACTGGGTAAAGGTCGCCAGGCGGACTTAT
AAGTCGGGGTTAAATCCATGTCTTAAACACATGCAAGGCTTCGGATACTAGGCTCTAGAGTCTCGAAGTCCGGTGA
ACGCTGGAAATGCTAGATATCGGAAGAACAACCGGTGCCAAGGCAGTCTTCTGTCGAGAACTGACGCTCAGGCACGAA
AGCCTGGGGAGCAAACAGGATTAGATACCCCTAGTAGTC
>C3
GTGTCAGCCCGCCGGTATACAGGGGTGCCAAGCCTTGTCCGGATTTACTGGGTAAAGGTCGCCAGGCGGACTTAT
AAGTCGGGGTTAAATCCATGTCTTAAACACATGCAAGGCTTCGGATACTAGGCTCTAGAGTCTCGAAGTCCGGTGA
ACGCTGGAAATGCTAGATATCGGAAGAACAACCGGTGCCAAGGCAGTCTTCTGTCGAGAACTGACGCTCAGGCACGAA
AGCCTGGGGAGCAAACAGGATTAGATACCCCTAGTAGTC
>C4
GTGTCAGCCCGCCGGTATACAGGGGTGCCAAGCCTTGTCCGGATTTACTGGGTAAAGGTCGCCAGGCGGACTTAT
AAGTCGGGGTTAAATCCATGTCTTAAACACATGCAAGGCTTCGGATACTAGGCTCTAGAGTCTCGAAGTCCGGTGA
ACGCTGGAAATGCTAGATATCGGAAGAACAACCGGTGCCAAGGCAGTCTTCTGTCGAGAACTGACGCTCAGGCACGAA
AGCCTGGGGAGCAAACAGGATTAGATACCCCTAGTAGTC
```

Use the following Perl one-liner to identify a list of unique sequences per line with duplicates as comma separated list:

```
cat F1.fasta F2.fasta | awk '/^>/ {printf("\n%s\n", $0); next; } { printf("%s", $0); } END {printf("\n");}' | awk 'NR>1 { if(NR%2==0) {gsub(">", "", $1); h=$1} else {a[$1]++; b[$1]=b[$1]" , "h}} END {for (n in a) {gsub("^", "", b[n]); print b[n]}}'
```

Note: The first awk statement linearizes the FASTA file so that header and sequence alternate on separate lines. To identify duplicates only, add an extra awk statement at the end:

```
cat F1.fasta F2.fasta | awk '/^>/ {printf("\n%s\n", $0); next; } { printf("%s", $0); } END {printf("\n");}' | awk 'NR>1 { if(NR%2==0) {gsub(">", "", $1); h=$1} else {a[$1]++; b[$1]=b[$1]" , "h}} END {for (n in a) {gsub("^", "", b[n]); print b[n]}}' | awk -F, 'NF>1'
```

2. Dereplicating NGS Reads

We can also dereplicate the reads from FASTQ files and generate a sorted list of reads in [usearch](#) header format as a FASTA file:

a) Single-end FASTQ file

```
perl -MDigest::MD5=md5_hex -ne 'push @a, $ ; @a = @a[@a-4..$#a]; if ($ % 4 == 0) {chomp($a[1]); $d{uc($a[1])}++;} END {foreach my $n (sort{ $d{$b} <=> $d{$a} } keys %d) {print ">".md5_hex($n).";size=".$d{$n}."; \n". $n. "\n"}}' forward.fastq > forward_derep.fasta
```

b) Paired-end FASTQ files

```
paste forward.fastq reverse.fastq | perl -MDigest::MD5=md5_hex -ne  
'chomp($_);push @a, $_; @a = @a[@a-4..$#a]; if ($. % 4 ==  
0){@q=split("\t",$a[1]);$d{uc($q[0].".".$q[1])}++;} END  
{open(FF,">","forward_derep.fasta");open(RR,">","reverse_derep.fasta");foreach  
my $n (sort{ $d{$b} <=> $d{$a} } keys %d){$l=md5_hex($n);@r=split(":",$n);print  
FF ">".$l." 1;size=".$d{$n}.";\n".$r[0]."\n"; print RR ">".$l."  
2;size=".$d{$n}.";\n".$r[1]."\n";}}
```