WGS metagenomics de novo assembly

Fernando Freire

April 27, 2019

Contents

1 Whole Genome Shotgun metagenomics: de novo Assembly

1.1 Summary Pipeline

We have two *fastqc* files to process, corresponding to the paired ends reads of the virome *Vir1_100k* containing 100, 000 paired end reads from the same saliva sample. After purification of viral particles.

Note 1: modify the MINLEN argument considering the original read length.

Note 2: try to optimize de novo assembly using metaspades.py and comparing at least 4 kmer strategies with QUAST. You do not have a reference genome in this case.

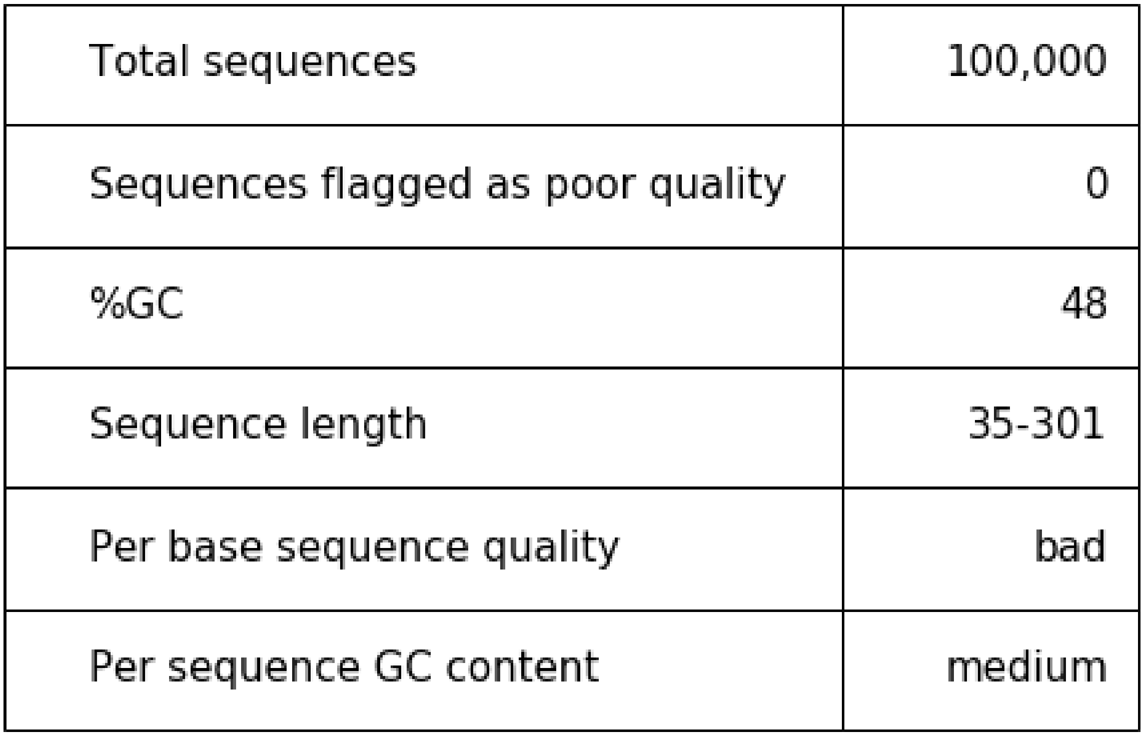
The detailed pipeline is developed in chapter 2.

Here we show the relevant information.

1.1.1 Preprocessing and quality check

Script 1.1.1 (python)

```python
plt.figure(figsize=(2,5))
plt.axis('off')
table = plt.table(cellText=df_qual.values, rowLabels=df_qual.index, colWidths = [0.3]*len(df_qual.columns),
          loc='top', cellLoc = 'right', rowLoc = 'left', bbox=[0,0,1,1]);
table.auto_set_font_size(False)
table.set_fontsize(15)
```



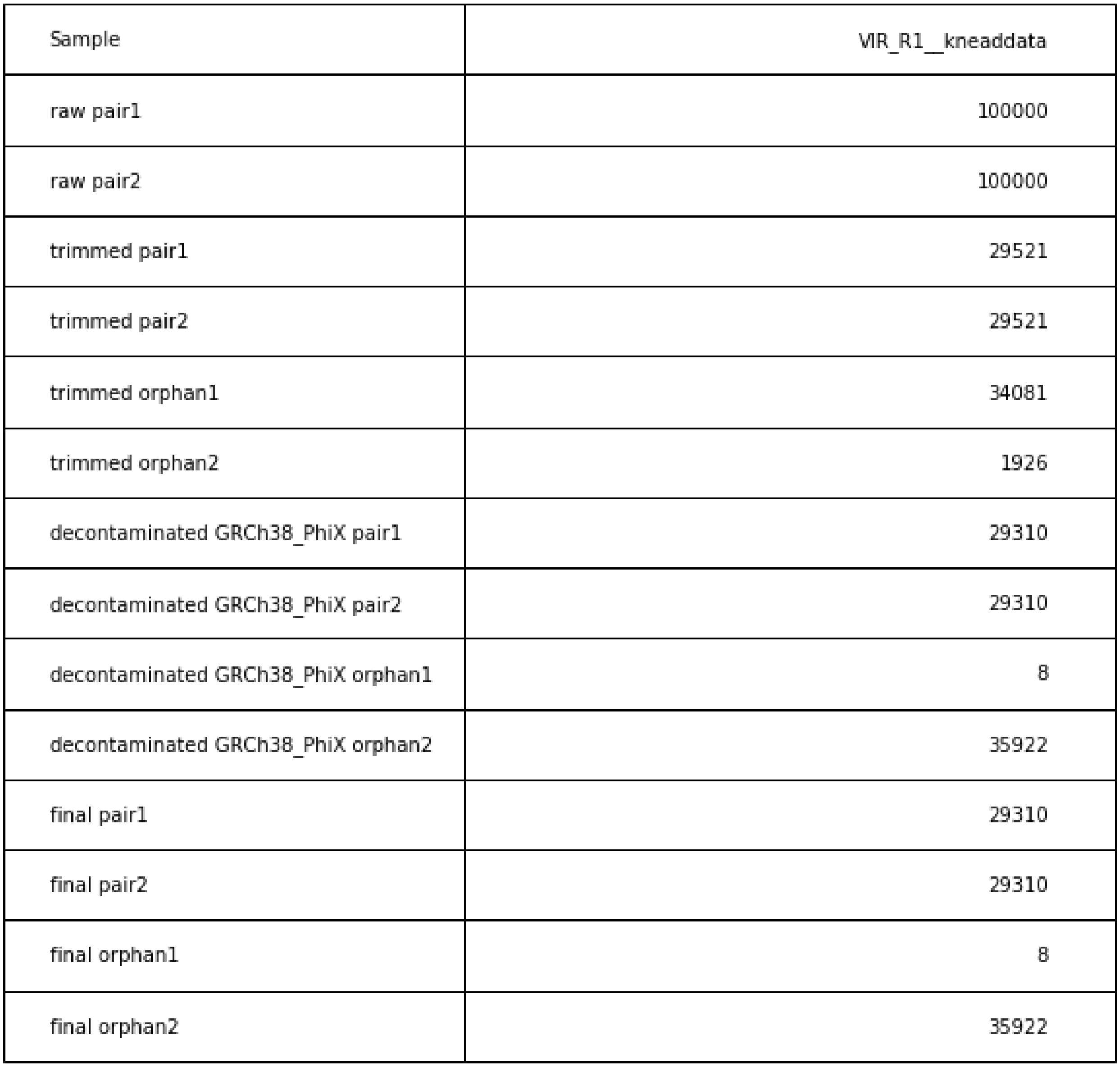
Total sequences	100,000
Sequences flagged as poor quality	0
%GC	48
Sequence length	35-301
Per base sequence quality	bad
Per sequence GC content	medium

1.1.2 Trimming and decontaminating

Kneaddata summary

Script 1.1.2 (python)

```python
plt.figure(figsize=(3,5))
plt.axis('off')
plt.table(cellText=df_knead.values, rowLabels=df_knead.index, colWidths = [1]*len(df_knead.columns), loc='top', cellLoc = 'right', rowLoc = 'left', bbox=[0,0,2,2]);
```

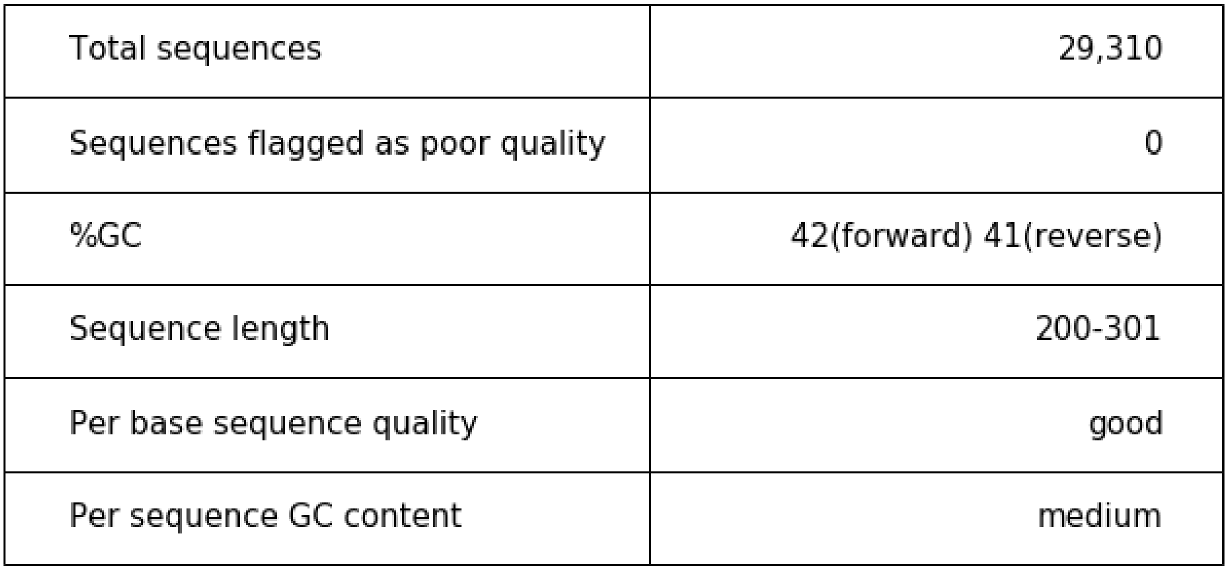


Sample	VIR_R1__kneaddata
raw pair1	100000
raw pair2	100000
trimmed pair1	29521
trimmed pair2	29521
trimmed orphan1	34081
trimmed orphan2	1926
decontaminated GRCh38_PhiX pair1	29310
decontaminated GRCh38_PhiX pair2	29310
decontaminated GRCh38_PhiX orphan1	8
decontaminated GRCh38_PhiX orphan2	35922
final pair1	29310
final pair2	29310
final orphan1	8
final orphan2	35922

Quality check

Script 1.1.3 (python)

```python
plt.figure(figsize=(5,5))
plt.axis('off')
table = plt.table(cellText=df_high_qual.values, rowLabels=df_high_qual.index, colWidths = [0.3]*len(df_high_qual.columns),
          loc='top', cellLoc = 'right', rowLoc = 'left', bbox=[0,0,1,1]);
table.auto_set_font_size(False)
table.set_fontsize(15)
```



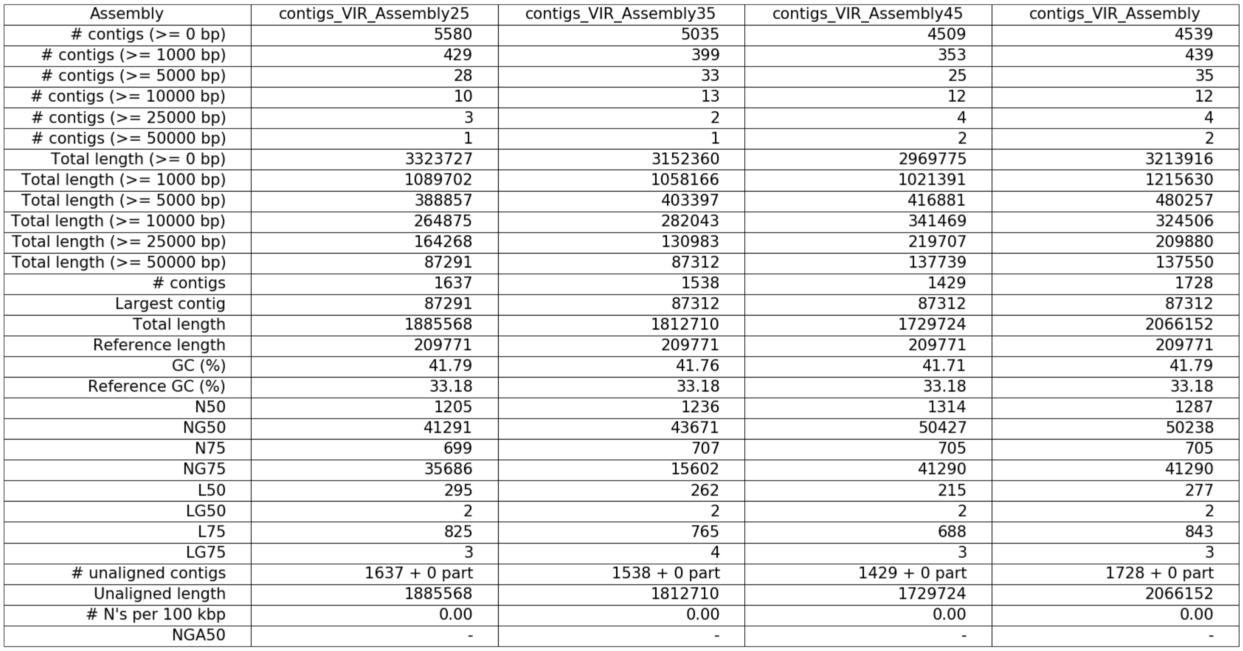
Total sequences	29,310
Sequences flagged as poor quality	0
%GC	42(forward) 41(reverse)
Sequence length	200-301
Per base sequence quality	good
Per sequence GC content	medium

1.1.3 Assembly(spades). Quast results.

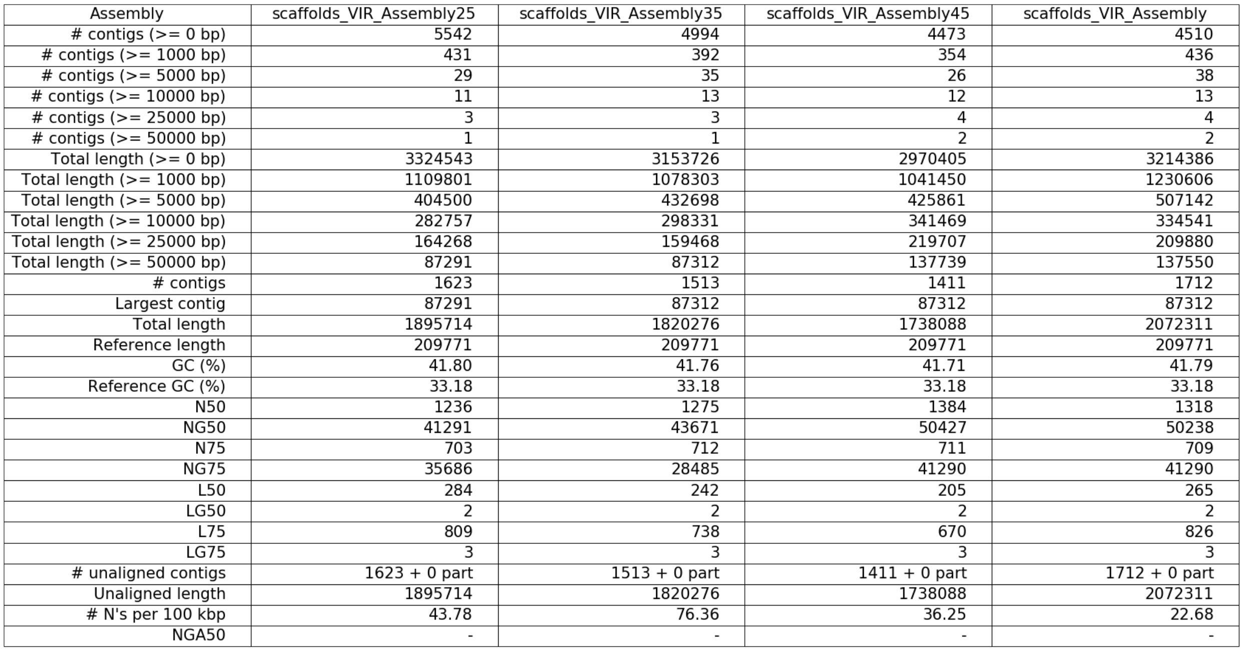
These are the quast results for the spades process comparing assemblies of k-mers of length 25, 35 and 45, and not informing k-mer to spades program.

Script 1.1.4 (python)

```python
df_quast_contigs = df_quast.iloc[:,0:5]
fig = plt.figure(figsize=(15,8))
ax = plt.subplot(111)
ax.axis('off')
table = plt.table(cellText=df_quast_contigs.values, colLabels=df_quast_contigs.columns,
          colWidths = [2]*len(df_quast_contigs.columns),
          loc='top',
          cellLoc = 'right', rowLoc = 'left',
          bbox=[0,0,2,2]);

table.auto_set_font_size(False)
table.set_fontsize(21)

df_quast_scaffolds = df_quast.iloc[:,[0,5,6,7,8]]
plt.figure(figsize=(15,8))
plt.axis('off')
table = plt.table(cellText=df_quast_scaffolds.values, colLabels=df_quast_scaffolds.columns,
          colWidths = [2]*len(df_quast_scaffolds.columns),
          loc='top',
                 cellLoc = 'right', rowLoc = 'left',
                 bbox=[0,0,2,2]);

table.auto_set_font_size(False)
table.set_fontsize(21)
```



Assembly	contigs_VIR_Assembly25	contigs_VIR_Assembly35	contigs_VIR_Assembly45	contigs_VIR_Assembly
# contigs (>= 0 bp)	5580	5035	4509	4539
# contigs (>= 1000 bp)	429	399	353	439
# contigs (>= 5000 bp)	28	33	25	35
# contigs (>= 10000 bp)	10	13	12	12
# contigs (>= 25000 bp)	3	2	4	4
# contigs (>= 50000 bp)	1	1	2	2
Total length (>= 0 bp)	3323727	3152360	2969775	3213916
Total length (>= 1000 bp)	1089702	1058166	1021391	1215630
Total length (>= 5000 bp)	388857	403397	416881	480257
Total length (>= 10000 bp)	264875	282043	341469	324506
Total length (>= 25000 bp)	164268	130983	219707	209880
Total length (>= 50000 bp)	87291	87312	137739	137550
# contigs	1637	1538	1429	1728
Largest contig	87291	87312	87312	87312
Total length	1885568	1812710	1729724	2066152
Reference length	209771	209771	209771	209771
GC (%)	41.79	41.76	41.71	41.79
Reference GC (%)	33.18	33.18	33.18	33.18
N50	1205	1236	1314	1287
NG50	41291	43671	50427	50238
N75	699	707	705	705
NG75	35686	15602	41290	41290
L50	295	262	215	277
LG50	2	2	2	2
L75	825	765	688	843
LG75	3	4	3	3
# unaligned contigs	1637 + 0 part	1538 + 0 part	1429 + 0 part	1728 + 0 part
Unaligned length	1885568	1812710	1729724	2066152
# N's per 100 kbp	0.00	0.00	0.00	0.00
NGA50	-	-	-	-



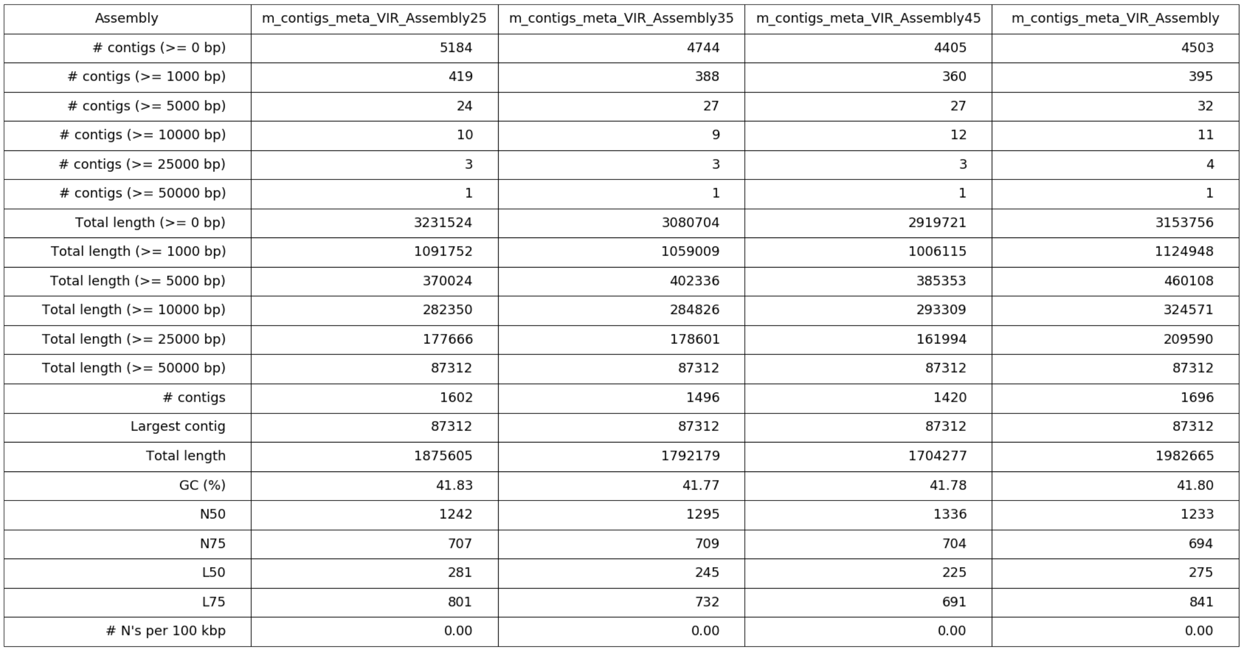
Assembly	scaffolds_VIR_Assembly25	scaffolds_VIR_Assembly35	scaffolds_VIR_Assembly45	scaffolds_VIR_Assembly
# contigs (>= 0 bp)	5542	4994	4473	4510
# contigs (>= 1000 bp)	431	392	354	436
# contigs (>= 5000 bp)	29	35	26	38
# contigs (>= 10000 bp)	11	13	12	13
# contigs (>= 25000 bp)	3	3	4	4
# contigs (>= 50000 bp)	1	1	2	2
Total length (>= 0 bp)	3324543	3153726	2970405	3214386
Total length (>= 1000 bp)	1109801	1078303	1041450	1230606
Total length (>= 5000 bp)	404500	432698	425861	507142
Total length (>= 10000 bp)	282757	298331	341469	334541
Total length (>= 25000 bp)	164268	159468	219707	209880
Total length (>= 50000 bp)	87291	87312	137739	137550
# contigs	1623	1513	1411	1712
Largest contig	87291	87312	87312	87312
Total length	1895714	1820276	1738088	2072311
Reference length	209771	209771	209771	209771
GC (%)	41.80	41.76	41.71	41.79
Reference GC (%)	33.18	33.18	33.18	33.18
N50	1236	1275	1384	1318
NG50	41291	43671	50427	50238
N75	703	712	711	709
NG75	35686	28485	41290	41290
L50	284	242	205	265
LG50	2	2	2	2
L75	809	738	670	826
LG75	3	3	3	3
# unaligned contigs	1623 + 0 part	1513 + 0 part	1411 + 0 part	1712 + 0 part
Unaligned length	1895714	1820276	1738088	2072311
# N's per 100 kbp	43.78	76.36	36.25	22.68
NGA50	-	-	-	-

1.1.4 Assembly(metaspades). Quast results.

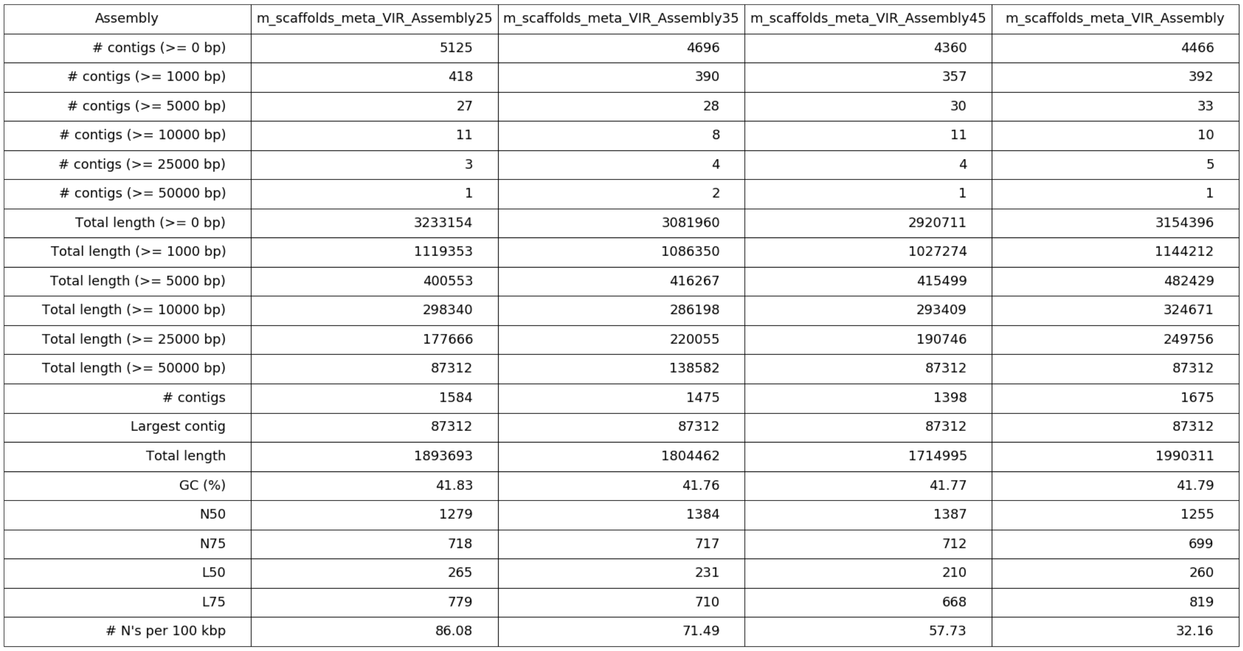
These are the quast results for the meta-spades process comparing assemblies of k-mers of length 25, 35 and 45, and not informing k-mer to spades program.

Script 1.1.5 (python)

```python
df_quast_contigs = df_quast_meta.iloc[:,0:5]
fig = plt.figure(figsize=(15,8))
ax = plt.subplot(111)
ax.axis('off')
table = plt.table(cellText=df_quast_contigs.values, colLabels=df_quast_contigs.columns,
          colWidths = [2]*len(df_quast_contigs.columns),
          loc='top',
          cellLoc = 'right', rowLoc = 'left',
          bbox=[0,0,2,2]);

table.auto_set_font_size(False)
table.set_fontsize(18)

df_quast_scaffolds = df_quast_meta.iloc[:,[0,5,6,7,8]]
fig = plt.figure(figsize=(15,8))
ax = plt.subplot(111)
ax.axis('off')
table = plt.table(cellText=df_quast_scaffolds.values, colLabels=df_quast_scaffolds.columns,
          colWidths = [2]*len(df_quast_scaffolds.columns),
          loc='top',
          cellLoc = 'right', rowLoc = 'left',
          bbox=[0,0,2,2]);

table.auto_set_font_size(False)
table.set_fontsize(18)
```



Assembly	m_contigs_meta_VIR_Assembly25	m_contigs_meta_VIR_Assembly35	m_contigs_meta_VIR_Assembly45	m_contigs_meta_VIR_Assembly
# contigs (>= 0 bp)	5184	4744	4405	4503
# contigs (>= 1000 bp)	419	388	360	395
# contigs (>= 5000 bp)	24	27	27	32
# contigs (>= 10000 bp)	10	9	12	11
# contigs (>= 25000 bp)	3	3	3	4
# contigs (>= 50000 bp)	1	1	1	1
Total length (>= 0 bp)	3231524	3080704	2919721	3153756
Total length (>= 1000 bp)	1091752	1059009	1006115	1124948
Total length (>= 5000 bp)	370024	402336	385353	460108
Total length (>= 10000 bp)	282350	284826	293309	324571
Total length (>= 25000 bp)	177666	178601	161994	209590
Total length (>= 50000 bp)	87312	87312	87312	87312
# contigs	1602	1496	1420	1696
Largest contig	87312	87312	87312	87312
Total length	1875605	1792179	1704277	1982665
GC (%)	41.83	41.77	41.78	41.80
N50	1242	1295	1336	1233
N75	707	709	704	694
L50	281	245	225	275
L75	801	732	691	841
# N's per 100 kbp	0.00	0.00	0.00	0.00



Assembly	m_scaffolds_meta_VIR_Assembly25	m_scaffolds_meta_VIR_Assembly35	m_scaffolds_meta_VIR_Assembly45	m_scaffolds_meta_VIR_Assembly
# contigs (>= 0 bp)	5125	4696	4360	4466
# contigs (>= 1000 bp)	418	390	357	392
# contigs (>= 5000 bp)	27	28	30	33
# contigs (>= 10000 bp)	11	8	11	10
# contigs (>= 25000 bp)	3	4	4	5
# contigs (>= 50000 bp)	1	2	1	1
Total length (>= 0 bp)	3233154	3081960	2920711	3154396
Total length (>= 1000 bp)	1119353	1086350	1027274	1144212
Total length (>= 5000 bp)	400553	416267	415499	482429
Total length (>= 10000 bp)	298340	286198	293409	324671
Total length (>= 25000 bp)	177666	220055	190746	249756
Total length (>= 50000 bp)	87312	138582	87312	87312
# contigs	1584	1475	1398	1675
Largest contig	87312	87312	87312	87312
Total length	1893693	1804462	1714995	1990311
GC (%)	41.83	41.76	41.77	41.79
N50	1279	1384	1387	1255
N75	718	717	712	699
L50	265	231	210	260
L75	779	710	668	819
# N's per 100 kbp	86.08	71.49	57.73	32.16

1.2 Summary

In the first step we check the files in order to detect any error:expected read length (100,000), file format (fastq). Also we substitute spaces by underscores in fastq heads, because of the programs, cut the headers after the first space avoiding to distinguish between forward and reverse.

After we perform a fast quality check, with the results printed in (1.1.1). Per base sequence quality is no good, so we expect so we go to the next step: trimming and decontaminating.

We use *kneaddata* over the forward(R1) and reverse(R2) pairs files. First it runs *trimmomatic* to trim and crop the reads and to remove adapters. We use the trimming sliding window option: starts scanning at the 5-end and clips the read once the average quality within the window falls below a threshold. And also the *MINLEN* parameter: drop the read if it is below a specified length. We use a conservative length of 200 bp.

After that it runs *bowtie2* to delete the contaminant sequences: we use a reference database to delete reads that map to the human or PhiX genomes. PhiX is a control frequently used during Illumina sequencing runs.

After this execution (stats on 1.1.2.1) we end by 29,310 sequences. We check the coherence number of reads on resulting files with the statistics.

We pass a quality check and that evidences a improvement of the per base sequence quality. Aldo a certain loss of %GC but not relevant.

We use *spades* for the assembly in single cell parameter with a reference genome, with four executions with different assembly lengths: 25, 35, 55 and let the parameter free to the program.

We use *quast* for comparison of genome assembly outputs (1.1.3). The best N50 score correspond to a contig length of 45.

We perform a second run of assembly with *metaspades*, with results on(1.1.4) and now without a reference genome. We have a lesser detection of contigs and scaffolds. The scores (N50 and so on) are very similar to spades.

Script 1.2.1 (python)

```python
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import matplotlib.pyplot as plt
#FILE_ID = "ECTV"
#FASTQ_STR = "@HWUSI-EAS1752R"
#MIN_LEN = "70"

FILE_ID = "VIR"
FASTQ_STR = "@M02255"
MIN_LEN = "200"
```

1.3 Detailed pipeline

1.3.1 Preprocessing and quality check

Script 1.3.1 (bash)

```bash
%%bash -s "$FILE_ID" "$FASTQ_STR"
ssh microbioinf@192.168.56.101 env FILE_ID=$1 FASTQ_STR=$2 2>/dev/null /bin/bash <<"EOT"
export PATH=$PATH:/home/microbioinf/miniconda3/bin
echo "#### Check files FILE_ID=${FILE_ID}, FASTQ_STR=$FASTQ_STR"
cd Documentos/Tema_3
head -4 ${FILE_ID}*fastq
grep -c $FASTQ_STR ${FILE_ID}*fastq

```

```
echo "#### Compute quality"
mkdir ${FILE_ID}_Quality
fastqc ${FILE_ID}_R1.fastq -o ${FILE_ID}_Quality/
fastqc ${FILE_ID}_R2.fastq -o ${FILE_ID}_Quality/

echo "#### Replace ' ' by '_' in header"
head -n 1 ${FILE_ID}*fastq
cat ${FILE_ID}_R1.fastq | sed 's/ /_/g' > ${FILE_ID}_R1_.fastq
cat ${FILE_ID}_R2.fastq | sed 's/ /_/g' > ${FILE_ID}_R2_.fastq
head -n 1 ${FILE_ID}*fastq
EOT
```

Output

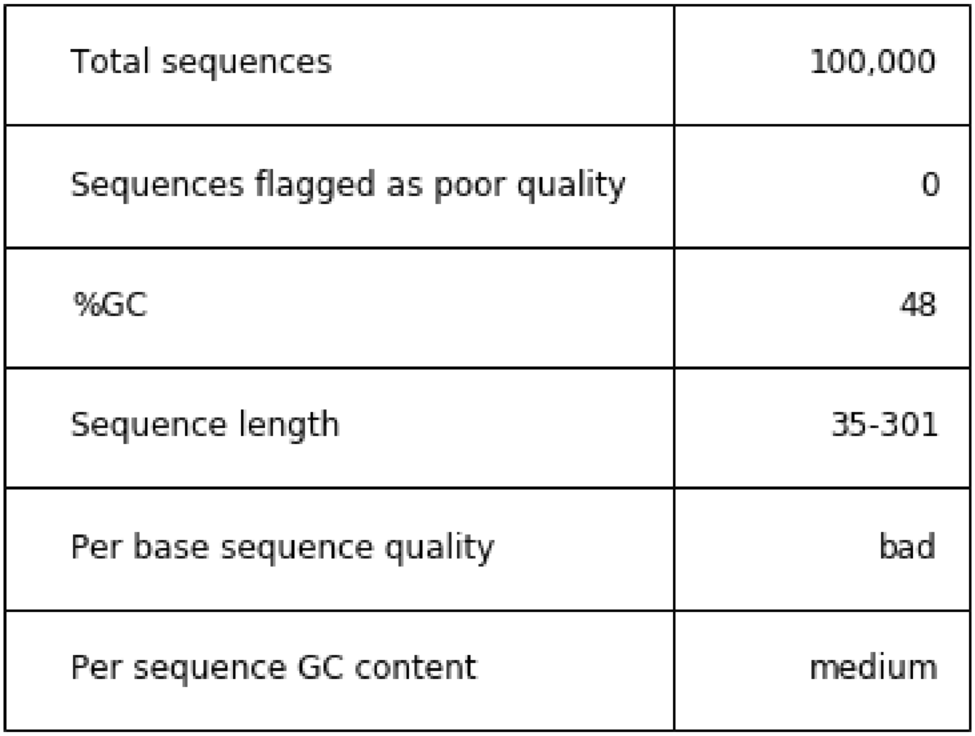
```
#### Check files FILE_ID=VIR, FASTQ_STR=@M02255
==> VIR_R1_.fastq <==
@M02255:131:000000000-AJC6R:1:1105:23249:10170_1:N:0:AGTCAA
AACTGGCGTTACATGAAGGGCTCTGAGTTGATTGATGCTTTGGAGGAGTACCTGTGAAATGGCCGTCTGAGAAGGTTGTTAATGCGACCGTAA⌟
 ↪   AGTATGGTGGTGTCGTGTTGAGACGTGGACCGTACGCATATTTCGATAAGGGGGGCATTCGATTGTGTGCTACAAGGCTTGGTCTCTCT⌟
 ↪   TCATATATTGTGGAGAGTGATGATTGTGGTCCTGAGATTTATAGTGAGGATGGTATGATTGAGTTGGTGACGTCTTTATGATTCCTGTT⌟
 ↪   ACCGAGACTATCCTGAAAACTGCTTACCAT
+
CCCCCGGGGGGGGGGGGGGGGGGGGGGFGGGGGGGGGGGGGGGGGGFGGGGGGGGGGGGGGGGGGGGGFAGGGGGGGGGGGGGGGGGGGGGGG⌟
 ↪   GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGEFGGGGGGGGGGGGGFGGGGGGGGGGGGGGFGGGGGGGGGGGGGGGCF@EGGGG9CE⌟
 ↪   GGGFFGGGC?FGECFFGGGG9CFGGFEFCGGEGGCF?FG7EGCFGC:FFGF6<FGFGGFGGGFFFFFEGFF@7@)8@FFFBAFA=F<FD⌟
 ↪   FF<157526))4?<39>B9?><?BAA?2>F

==> VIR_R1.fastq <==
@M02255:131:000000000-AJC6R:1:1105:23249:10170_1:N:0:AGTCAA
AACTGGCGTTACATGAAGGGCTCTGAGTTGATTGATGCTTTGGAGGAGTACCTGTGAAATGGCCGTCTGAGAAGGTTGTTAATGCGACCGTAA⌟
 ↪   AGTATGGTGGTGTCGTGTTGAGACGTGGACCGTACGCATATTTCGATAAGGGGGGCATTCGATTGTGTGCTACAAGGCTTGGTCTCTCT⌟
 ↪   TCATATATTGTGGAGAGTGATGATTGTGGTCCTGAGATTTATAGTGAGGATGGTATGATTGAGTTGGTGACGTCTTTATGATTCCTGTT⌟
 ↪   ACCGAGACTATCCTGAAAACTGCTTACCAT
+
CCCCCGGGGGGGGGGGGGGGGGGGGGGFGGGGGGGGGGGGGGGGGGFGGGGGGGGGGGGGGGGGGGGGFAGGGGGGGGGGGGGGGGGGGGGGG⌟
 ↪   GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGEFGGGGGGGGGGGGGFGGGGGGGGGGGGGGFGGGGGGGGGGGGGGGCF@EGGGG9CE⌟
 ↪   GGGFFGGGC?FGECFFGGGG9CFGGFEFCGGEGGCF?FG7EGCFGC:FFGF6<FGFGGFGGGFFFFFEGFF@7@)8@FFFBAFA=F<FD⌟
 ↪   FF<157526))4?<39>B9?><?BAA?2>F

==> VIR_R2_.fastq <==
@M02255:131:000000000-AJC6R:1:1105:23249:10170_2:N:0:AGTCAA
GATGAAATTCTGAAGCAACGGACTAATGATCGGCAGCGGCATGCTTCCTCCTCAATTTCTCCTTCAGGAATATGATTGTCCCGATTTCTGTCA⌟
 ↪   ATTGAATATCGACCTGTTCAAAAGTGCACTGCCAGAGATCCTCCTTAATTCTAATAATATCCACGAAGCGGTTTCCTGAATTAATGCAT⌟
 ↪   GCAGTAGCATTATCAGGGAAAAAGATGTGCCACCTGAAATGGTAAGCAGTTTTCAGGATAGTCTCGGTAACAGGAATCATAAAGACGTC⌟
 ↪   ACCAACTCAATCATATCAANCTCACTATA
+
CCCCCGGGGGGGGGGGGGGGGGGGGGGGGFGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG⌟
 ↪   GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGFGGGGGGGGGGGGGDGGEGGGGGGGGGADFGCDGG⌟
 ↪   C?F8@FGFGGDGFGGGFGFFFFFGGGFFGFFGF*96FFAFFCFFFF@FF;A?FA<?EA=4@F478A2>F@@CFBDD@B9EFFC).//4/⌟
 ↪   8?EF0:@AEE?=;?.).5)#/(/6(6265
```

```
==> VIR_R2.fastq <==
@M02255:131:000000000-AJC6R:1:1105:23249:10170_2:N:0:AGTCAA
GATGAAATTCTGAAGCAACGGACTAATGATCGGCAGCGGCATGCTTCCTCCTCAATTTCTCCTTCAGGAATATGATTGTCCCGATTTCTGTCA⌋
 ↪  ATTGAATATCGACCTGTTCAAAAGTGCACTGCCAGAGATCCTCCTTAATTCTAATAATATCCACGAAGCGGTTTCCTGAATTAATGCAT⌋
 ↪  GCAGTAGCATTATCAGGGAAAAAGATGTGCCACCTGAAATGGTAAGCAGTTTTCAGGATAGTCTCGGTAACAGGAATCATAAAGACGTC⌋
 ↪  ACCAACTCAATCATATCAANCTCACTATA
+
CCCCCGGGGGGGGGGGGGGGGGGGGGGGGFGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG⌋
 ↪  GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGFGGGGGGGGGGGGGDGGEGGGGGGGGGADFGCDGG⌋
 ↪  C?F8@FGFGGDGFGGGFGFFFFFGGGFFGFFGF*96FFAFFCFFFF@FF;A?FA<?EA=4@F478A2>F@@CFBDD@B9EFFC).//4/⌋
 ↪  8?EF0:@AEE?=;?.).5)#/(/6(6265
VIR_R1_.fastq:100000
VIR_R1.fastq:100000
VIR_R2_.fastq:100000
VIR_R2.fastq:100000
#### Compute quality
Analysis complete for VIR_R1.fastq
Analysis complete for VIR_R2.fastq
#### Replace ' ' by '_' in header
==> VIR_R1_.fastq <==
@M02255:131:000000000-AJC6R:1:1105:23249:10170_1:N:0:AGTCAA

==> VIR_R1.fastq <==
@M02255:131:000000000-AJC6R:1:1105:23249:10170_1:N:0:AGTCAA

==> VIR_R2_.fastq <==
@M02255:131:000000000-AJC6R:1:1105:23249:10170_2:N:0:AGTCAA

==> VIR_R2.fastq <==
@M02255:131:000000000-AJC6R:1:1105:23249:10170_2:N:0:AGTCAA
==> VIR_R1_.fastq <==
@M02255:131:000000000-AJC6R:1:1105:23249:10170_1:N:0:AGTCAA

==> VIR_R1.fastq <==
@M02255:131:000000000-AJC6R:1:1105:23249:10170_1:N:0:AGTCAA

==> VIR_R2_.fastq <==
@M02255:131:000000000-AJC6R:1:1105:23249:10170_2:N:0:AGTCAA

==> VIR_R2.fastq <==
@M02255:131:000000000-AJC6R:1:1105:23249:10170_2:N:0:AGTCAA
```

Script 1.3.2 (python)

```python
df_qual = pd.DataFrame(data=['100,000', '0', '48', '35-301', 'bad', 'medium'],
                       index = ['Total sequences', 'Sequences flagged as poor quality', '%GC⌋
',
                      'Sequence length', 'Per base sequence quality','Per sequence GC
                       ↪  content'])
plt.figure(figsize=(2,5))
plt.axis('off')
```

```python
table = plt.table(cellText=df_qual.values, rowLabels=df_qual.index, colWidths = [0.3]*len(df_qual.columns),
          loc='top', cellLoc = 'right', rowLoc = 'left', bbox=[0,0,1,1]);
table.auto_set_font_size(False)
table.set_fontsize(12)
```



Total sequences	100,000
Sequences flagged as poor quality	0
%GC	48
Sequence length	35-301
Per base sequence quality	bad
Per sequence GC content	medium

1.3.2 Trimming and decontaminating

Trimming poor quality ends and short sequences (**Trimmomatic**) and removal of reads aligning to the human and phiX174 genomes (***bowtie2**). The later one is a contaminant used as spike by Illumina kits to control quality of the sequencing process.

We are only filtering only R1 files because forward reads have usually better quality than reverse reads.

Process

Script 1.3.3 (bash)

```bash
%%bash -s "$FILE_ID" "$FASTQ_STR" "$MIN_LEN"
ssh microbioinf@192.168.56.101 env FILE_ID=$1 FASTQ_STR=$2 MIN_LEN=$3 2>/dev/null /bin/bash <<"EOT"
export PATH=$PATH:/home/microbioinf/miniconda3/bin
```

```
cd Documentos/Tema_3
echo "#### Trimming and decontaminating FILE_ID=${FILE_ID} MIN_LEN=${MIN_LEN}"
kneaddata -i ${FILE_ID}_R1_.fastq -i ${FILE_ID}_R2_.fastq \
-o kneaddata_out_${FILE_ID} -db /home/shared/bowtiedb/GRCh38_PhiX \
--trimmomatic /home/microbioinf/miniconda3/pkgs/trimmomatic-0.38-1/share/trimmomatic-0.38-1/
\
-t 2 --trimmomatic-options "SLIDINGWINDOW:4:20 MINLEN:${MIN_LEN}" \
--bowtie2-options "--very-sensitive --dovetail" --remove-intermediate-output
EOT
```

Output

```
#### Trimming and decontaminating FILE_ID=VIR MIN_LEN=200
Initial number of reads ( /home/microbioinf/Documentos/Tema_3/VIR_R1_.fastq ): 100000
Initial number of reads ( /home/microbioinf/Documentos/Tema_3/VIR_R2_.fastq ): 100000
Running Trimmomatic ...
Total reads after trimming (
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.1.fastq
 ↪  ): 29521
Total reads after trimming (
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.2.fastq
 ↪  ): 29521
Total reads after trimming ( /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kn ⌋
 ↪  eaddata.trimmed.single.1.fastq ):
 ↪  34081
Total reads after trimming ( /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kn ⌋
 ↪  eaddata.trimmed.single.2.fastq ):
 ↪  1926
Decontaminating ...
Running bowtie2 ...
Total reads after removing those found in reference database ( /home/microbioinf/Documentos/T ⌋
 ↪  ema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_PhiX_bowtie2_paired_clean_1.fastq ):
 ↪  29310
Total reads after removing those found in reference database ( /home/microbioinf/Documentos/T ⌋
 ↪  ema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_PhiX_bowtie2_paired_clean_2.fastq ):
 ↪  29310
Total reads after merging results from multiple databases (
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_paired_1.fastq ):
 ↪  29310
Total reads after merging results from multiple databases (
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_paired_2.fastq ):
 ↪  29310
Total reads after removing those found in reference database ( /home/microbioinf/Documentos/T ⌋
 ↪  ema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_PhiX_bowtie2_unmatched_1_clean.fastq ):
 ↪  8
Total reads after merging results from multiple databases (
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_unmatched_1.fastq
 ↪  ): 8
Total reads after removing those found in reference database ( /home/microbioinf/Documentos/T ⌋
 ↪  ema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_PhiX_bowtie2_unmatched_2_clean.fastq ):
 ↪  35922
```

```
Total reads after merging results from multiple databases (/home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_unmatched_2.fastq): 35922

Final output files created:
/home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_paired_1.fastq
/home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_paired_2.fastq
/home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_unmatched_1.fastq
/home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_unmatched_2.fastq
```

Process statistics

Script 1.3.4 (bash)

```bash
%%bash -s "$FILE_ID" "$FASTQ_STR" "$MIN_LEN"
ssh microbioinf@192.168.56.101 env FILE_ID=$1 FASTQ_STR=$2 MIN_LEN=$3 2>/dev/null /bin/bash <<"EOT"
export PATH=$PATH:/home/microbioinf/miniconda3/bin
cd Documentos/Tema_3
cd kneaddata_out_${FILE_ID}/
cat ${FILE_ID}_R1*log
kneaddata_read_count_table --input ./ --output kneaddata_read_counts.txt
grep -c $FASTQ_STR ${FILE_ID}*fastq
```

Output

```
04/27/2019 11:36:19 AM - kneaddata.knead_data - INFO: Running kneaddata v0.6.1
04/27/2019 11:36:19 AM - kneaddata.knead_data - INFO: Output files will be written to: /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR
04/27/2019 11:36:19 AM - kneaddata.knead_data - DEBUG: Running with the following arguments:
verbose = False
bmtagger_path = None
minscore = 50
bowtie2_path = /home/microbioinf/miniconda3/bin/bowtie2
maxperiod = 500
no_discordant = False
serial = False
fastqc_start = False
bmtagger = False
cat_final_output = False
log_level = DEBUG
log = /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.log
max_memory = 500m
remove_intermediate_output = True
fastqc_path = None
output_dir = /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR
trf_path = None
remove_temp_output = True
reference_db = /home/shared/bowtiedb/GRCh38_PhiX
```

```
input = /home/microbioinf/Documentos/Tema_3/VIR_R1_.fastq
 ↪  /home/microbioinf/Documentos/Tema_3/VIR_R2_.fastq
pi = 10
reorder = False
pm = 80
trimmomatic_path = /home/microbioinf/miniconda3/pkgs/trimmomatic-0.38-1/share/trimmomatic-0.3 ⌟
 ↪  8-1/trimmomatic.jar
store_temp_output = False
cat_pairs = False
mismatch = 7
threads = 2
delta = 7
bowtie2_options = --very-sensitive --dovetail --phred33
bypass_trim = False
processes = 1
trimmomatic_quality_scores = -phred33
fastqc_end = False
trf = False
trimmomatic_options = SLIDINGWINDOW:4:20 MINLEN:200
output_prefix = VIR_R1__kneaddata
match = 2

04/27/2019 11:36:19 AM - kneaddata.utilities - INFO: READ COUNT: raw pair1 : Initial number
 ↪  of reads ( /home/microbioinf/Documentos/Tema_3/VIR_R1_.fastq ): 100000
04/27/2019 11:36:19 AM - kneaddata.utilities - INFO: READ COUNT: raw pair2 : Initial number
 ↪  of reads ( /home/microbioinf/Documentos/Tema_3/VIR_R2_.fastq ): 100000
04/27/2019 11:36:19 AM - kneaddata.utilities - DEBUG: Checking input file to Trimmomatic :
 ↪  /home/microbioinf/Documentos/Tema_3/VIR_R1_.fastq
04/27/2019 11:36:19 AM - kneaddata.utilities - DEBUG: Checking input file to Trimmomatic :
 ↪  /home/microbioinf/Documentos/Tema_3/VIR_R2_.fastq
04/27/2019 11:36:19 AM - kneaddata.utilities - INFO: Running Trimmomatic ...
04/27/2019 11:36:19 AM - kneaddata.utilities - INFO: Execute command: java -Xmx500m -d64 -jar
 ↪  /home/microbioinf/miniconda3/pkgs/trimmomatic-0.38-1/share/trimmomatic-0.38-1/trimmomatic ⌟
 ↪  .jar PE -threads 2 -phred33 /home/microbioinf/Documentos/Tema_3/VIR_R1_.fastq
 ↪  /home/microbioinf/Documentos/Tema_3/VIR_R2_.fastq
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.1.fastq
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.single.1. ⌟
 ↪  fastq
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.2.fastq
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.single.2. ⌟
 ↪  fastq SLIDINGWINDOW:4:20
 ↪  MINLEN:200
04/27/2019 11:36:22 AM - kneaddata.utilities - DEBUG: TrimmomaticPE: Started with arguments:
 -threads 2 -phred33 /home/microbioinf/Documentos/Tema_3/VIR_R1_.fastq
  ↪  /home/microbioinf/Documentos/Tema_3/VIR_R2_.fastq
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.1.fastq
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.single.1 ⌟
  ↪  .fastq
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.2.fastq
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.single.2 ⌟
  ↪  .fastq SLIDINGWINDOW:4:20
  ↪  MINLEN:200
```

```
Input Read Pairs: 100000 Both Surviving: 29521 (29,52%) Forward Only Surviving: 34081
 ↪  (34,08%) Reverse Only Surviving: 1926 (1,93%) Dropped: 34472 (34,47%)
TrimmomaticPE: Completed successfully

04/27/2019 11:36:22 AM - kneaddata.utilities - DEBUG: Checking output file from Trimmomatic :
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.1.fastq
04/27/2019 11:36:22 AM - kneaddata.utilities - DEBUG: Checking output file from Trimmomatic :
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.single.1. ⌋
 ↪  fastq
04/27/2019 11:36:22 AM - kneaddata.utilities - DEBUG: Checking output file from Trimmomatic :
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.2.fastq
04/27/2019 11:36:22 AM - kneaddata.utilities - DEBUG: Checking output file from Trimmomatic :
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.single.2. ⌋
 ↪  fastq
04/27/2019 11:36:22 AM - kneaddata.utilities - INFO: READ COUNT: trimmed pair1 : Total reads
 ↪  after trimming (
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.1.fastq
 ↪  ): 29521
04/27/2019 11:36:22 AM - kneaddata.utilities - INFO: READ COUNT: trimmed pair2 : Total reads
 ↪  after trimming (
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.2.fastq
 ↪  ): 29521
04/27/2019 11:36:22 AM - kneaddata.utilities - INFO: READ COUNT: trimmed orphan1 : Total
 ↪  reads after trimming ( /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__knea ⌋
 ↪  ddata.trimmed.single.1.fastq ):
 ↪  34081
04/27/2019 11:36:22 AM - kneaddata.utilities - INFO: READ COUNT: trimmed orphan2 : Total
 ↪  reads after trimming ( /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__knea ⌋
 ↪  ddata.trimmed.single.2.fastq ):
 ↪  1926
04/27/2019 11:36:22 AM - kneaddata.run - INFO: Decontaminating ...
04/27/2019 11:36:22 AM - kneaddata.utilities - DEBUG: Checking input file to bowtie2 :
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.1.fastq
04/27/2019 11:36:22 AM - kneaddata.utilities - DEBUG: Checking input file to bowtie2 :
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.2.fastq
04/27/2019 11:36:22 AM - kneaddata.utilities - DEBUG: Checking input file to bowtie2 : /home/ ⌋
 ↪  microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.single.1.fastq
04/27/2019 11:36:22 AM - kneaddata.utilities - DEBUG: Checking input file to bowtie2 : /home/ ⌋
 ↪  microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.single.2.fastq
04/27/2019 11:36:22 AM - kneaddata.utilities - INFO: Running bowtie2 ...
```

```
04/27/2019 11:36:22 AM - kneaddata.utilities - INFO: Execute command:
↪  kneaddata_bowtie2_discordant_pairs --bowtie2 /home/microbioinf/miniconda3/bin/bowtie2
↪  --threads 2 -x /home/shared/bowtiedb/GRCh38_PhiX --bowtie2-options "--very-sensitive
↪  --dovetail --phred33" -1
↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.1.fastq
↪  -2
↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.2.fastq
↪  --un-pair /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_⌋
↪  PhiX_bowtie2_paired_clean_%.fastq --al-pair
↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_PhiX_bowti⌋
↪  e2_paired_contam_%.fastq -U
↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.single.1.⌋
↪  fastq,/home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata.trimmed.sin⌋
↪  gle.2.fastq --un-single
↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_PhiX_bowti⌋
↪  e2_unmatched_%_clean.fastq --al-single
↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_PhiX_bowti⌋
↪  e2_unmatched_%_contam.fastq -S
↪  /dev/null
04/27/2019 11:36:40 AM - kneaddata.utilities - DEBUG: 65528 reads; of these:
  29521 (45.05%) were paired; of these:
    29338 (99.38%) aligned concordantly 0 times
    157 (0.53%) aligned concordantly exactly 1 time
    26 (0.09%) aligned concordantly >1 times
    ----
    29338 pairs aligned concordantly 0 times; of these:
      10 (0.03%) aligned discordantly 1 time
    ----
    29328 pairs aligned 0 times concordantly or discordantly; of these:
      58656 mates make up the pairs; of these:
        58636 (99.97%) aligned 0 times
        15 (0.03%) aligned exactly 1 time
        5 (0.01%) aligned >1 times
  36007 (54.95%) were unpaired; of these:
    35914 (99.74%) aligned 0 times
    81 (0.22%) aligned exactly 1 time
    12 (0.03%) aligned >1 times
0.52% overall alignment rate
pair1_aligned : 183
pair2_aligned : 183
orphan1_unaligned : 8
orphan2_unaligned : 35922
orphan2_aligned : 113
pair2_unaligned : 29310
pair1_unaligned : 29310
orphan1_aligned : 20

04/27/2019 11:36:40 AM - kneaddata.utilities - DEBUG: Checking output file from bowtie2 :
↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_PhiX_bowti⌋
↪  e2_paired_clean_1.fastq
```

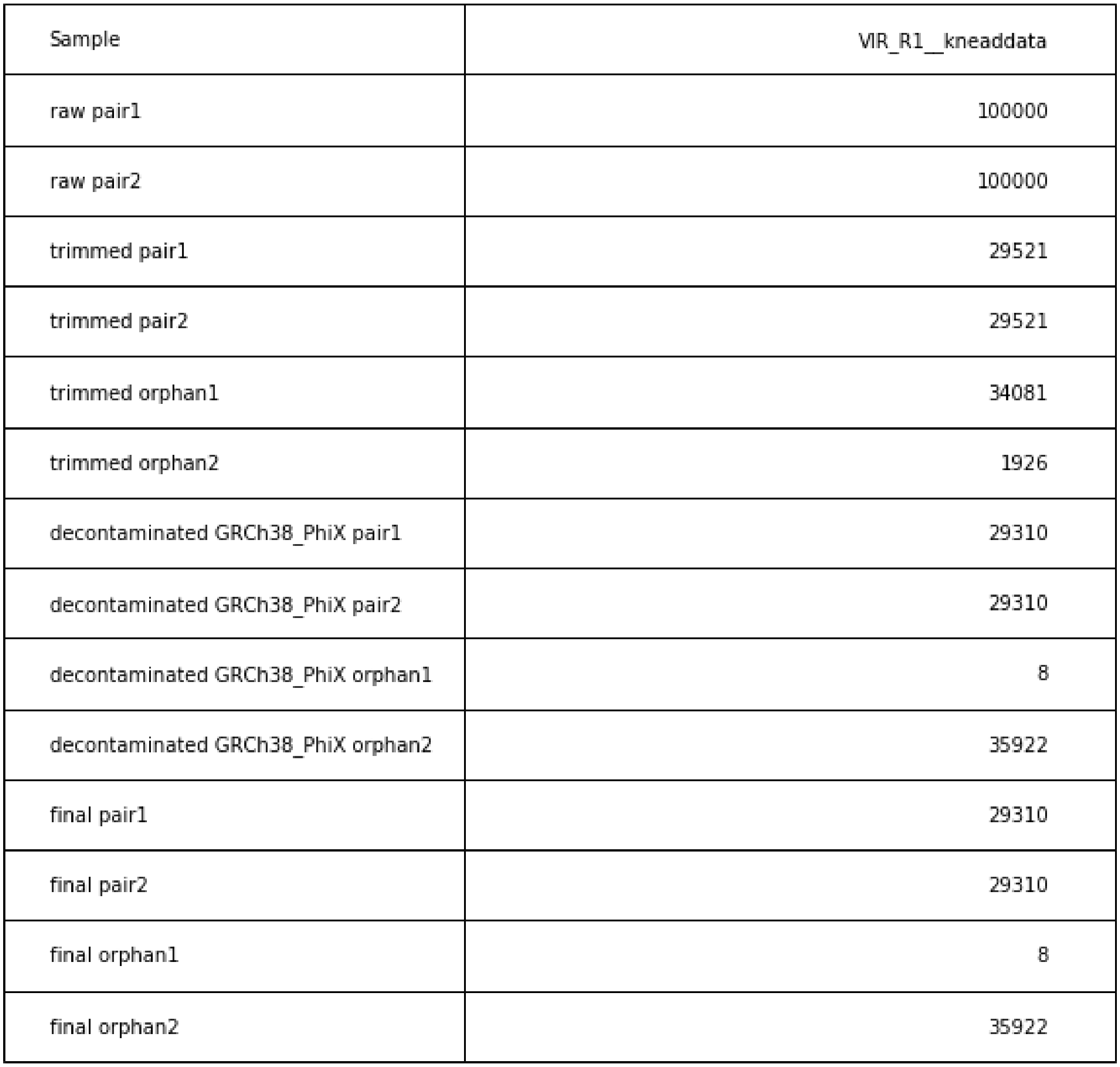
```
04/27/2019 11:36:40 AM - kneaddata.utilities - DEBUG: Checking output file from bowtie2 :
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_PhiX_bowti ⌋
 ↪  e2_paired_clean_2.fastq
04/27/2019 11:36:40 AM - kneaddata.run - INFO: Total contaminate sequences in file (
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_PhiX_bowti ⌋
 ↪  e2_paired_contam_1.fastq ) :
 ↪  183
04/27/2019 11:36:40 AM - kneaddata.run - INFO: Total contaminate sequences in file (
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_PhiX_bowti ⌋
 ↪  e2_paired_contam_2.fastq ) :
 ↪  183
04/27/2019 11:36:40 AM - kneaddata.run - INFO: Total contaminate sequences in file (
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_PhiX_bowti ⌋
 ↪  e2_unmatched_1_contam.fastq ) :
 ↪  20
04/27/2019 11:36:40 AM - kneaddata.run - INFO: Total contaminate sequences in file (
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_PhiX_bowti ⌋
 ↪  e2_unmatched_2_contam.fastq ) :
 ↪  113
04/27/2019 11:36:40 AM - kneaddata.utilities - INFO: READ COUNT: decontaminated GRCh38_PhiX
 ↪  pair1 : Total reads after removing those found in reference database (
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_PhiX_bowti ⌋
 ↪  e2_paired_clean_1.fastq ):
 ↪  29310
04/27/2019 11:36:40 AM - kneaddata.utilities - INFO: READ COUNT: decontaminated GRCh38_PhiX
 ↪  pair2 : Total reads after removing those found in reference database (
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_PhiX_bowti ⌋
 ↪  e2_paired_clean_2.fastq ):
 ↪  29310
04/27/2019 11:36:40 AM - kneaddata.utilities - INFO: READ COUNT: final pair1 : Total reads
 ↪  after merging results from multiple databases (
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_paired_1.fastq ):
 ↪  29310
04/27/2019 11:36:40 AM - kneaddata.utilities - INFO: READ COUNT: final pair2 : Total reads
 ↪  after merging results from multiple databases (
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_paired_2.fastq ):
 ↪  29310
04/27/2019 11:36:40 AM - kneaddata.utilities - WARNING: Unable to remove file:
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_PhiX_bowti ⌋
 ↪  e2_paired_clean_1.fastq
04/27/2019 11:36:40 AM - kneaddata.utilities - WARNING: Unable to remove file:
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_PhiX_bowti ⌋
 ↪  e2_paired_clean_2.fastq
04/27/2019 11:36:40 AM - kneaddata.utilities - INFO: READ COUNT: decontaminated GRCh38_PhiX
 ↪  orphan1 : Total reads after removing those found in reference database (
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_PhiX_bowti ⌋
 ↪  e2_unmatched_1_clean.fastq ):
 ↪  8
04/27/2019 11:36:40 AM - kneaddata.utilities - INFO: READ COUNT: final orphan1 : Total reads
 ↪  after merging results from multiple databases (
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_unmatched_1.fastq
 ↪  ): 8
```

```
04/27/2019 11:36:40 AM - kneaddata.utilities - WARNING: Unable to remove file: /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_PhiX_bowtie2_unmatched_1_clean.fastq
04/27/2019 11:36:40 AM - kneaddata.utilities - INFO: READ COUNT: decontaminated GRCh38_PhiX orphan2 : Total reads after removing those found in reference database ( /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_PhiX_bowtie2_unmatched_2_clean.fastq ): 35922
04/27/2019 11:36:40 AM - kneaddata.utilities - INFO: READ COUNT: final orphan2 : Total reads after merging results from multiple databases ( /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_unmatched_2.fastq ): 35922
04/27/2019 11:36:40 AM - kneaddata.utilities - WARNING: Unable to remove file: /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_GRCh38_PhiX_bowtie2_unmatched_2_clean.fastq
04/27/2019 11:36:40 AM - kneaddata.knead_data - INFO:
Final output files created:
/home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_paired_1.fastq
/home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_paired_2.fastq
/home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_unmatched_1.fastq
/home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR_R1__kneaddata_unmatched_2.fastq

Reading log: ./VIR_R1__kneaddata.log
Read count table written: kneaddata_read_counts.txt
VIR_R1__kneaddata_GRCh38_PhiX_bowtie2_paired_contam_1.fastq:183
VIR_R1__kneaddata_GRCh38_PhiX_bowtie2_paired_contam_2.fastq:183
VIR_R1__kneaddata_GRCh38_PhiX_bowtie2_unmatched_1_contam.fastq:20
VIR_R1__kneaddata_GRCh38_PhiX_bowtie2_unmatched_2_contam.fastq:113
VIR_R1__kneaddata_paired_1.fastq:29310
VIR_R1__kneaddata_paired_2.fastq:29310
VIR_R1__kneaddata_unmatched_1.fastq:8
VIR_R1__kneaddata_unmatched_2.fastq:35922
```

Script 1.3.5 (python)

```python
data = """
cat Documentos/Tema_3/kneaddata_out_%s/kneaddata_read_counts.txt
EOT
""" % FILE_ID
output = !ssh microbioinf@192.168.56.101 /bin/bash <<"EOT" {data}

data = []
# To list of lists
for row in output:
    data.append(row.split('\t'))
# To dataframe
df_knead = pd.DataFrame(data[1:], columns=data[0])
df_knead.style.hide_index().set_properties(**{'text-align': 'right', 'font-family' : 'courier', 'color' : 'darkgreen', "font-size" : "11pt"}).\
set_properties(**{'text-align': 'right', 'font-family' : 'courier', 'color' : 'darkblue', "font-size" : "12pt"}, subset=['Sample'])
```

```python
df_knead = df_knead.transpose()

fig = plt.figure(figsize=(3,5))
ax = plt.subplot(111)
ax.axis('off')
plt.table(cellText=df_knead.values, rowLabels=df_knead.index, colWidths = [1]*len(df_knead.columns),
          loc='top',
          cellLoc = 'right', rowLoc = 'left',
          bbox=[0,0,2,2]);
```



Sample	VIR_R1__kneaddata
raw pair1	100000
raw pair2	100000
trimmed pair1	29521
trimmed pair2	29521
trimmed orphan1	34081
trimmed orphan2	1926
decontaminated GRCh38_PhiX pair1	29310
decontaminated GRCh38_PhiX pair2	29310
decontaminated GRCh38_PhiX orphan1	8
decontaminated GRCh38_PhiX orphan2	35922
final pair1	29310
final pair2	29310
final orphan1	8
final orphan2	35922

Check number of reads With grep we can identify the non-contaminated high-quality files

Script 1.3.6 (bash)

```bash
%%bash -s "$FILE_ID" "$FASTQ_STR" "$MIN_LEN"
ssh microbioinf@192.168.56.101 env FILE_ID=$1 FASTQ_STR=$2 MIN_LEN=$3 2>/dev/null /bin/bash <<"EOT"
export PATH=$PATH:/home/microbioinf/miniconda3/bin
cd Documentos/Tema_3
cd kneaddata_out_${FILE_ID}/
grep -c $FASTQ_STR ${FILE_ID}*fastq
```

Output

```
VIR_R1__kneaddata_GRCh38_PhiX_bowtie2_paired_contam_1.fastq:183
VIR_R1__kneaddata_GRCh38_PhiX_bowtie2_paired_contam_2.fastq:183
VIR_R1__kneaddata_GRCh38_PhiX_bowtie2_unmatched_1_contam.fastq:20
VIR_R1__kneaddata_GRCh38_PhiX_bowtie2_unmatched_2_contam.fastq:113
VIR_R1__kneaddata_paired_1.fastq:29310
VIR_R1__kneaddata_paired_2.fastq:29310
VIR_R1__kneaddata_unmatched_1.fastq:8
VIR_R1__kneaddata_unmatched_2.fastq:35922
```

Check quality

Script 1.3.7 (bash)

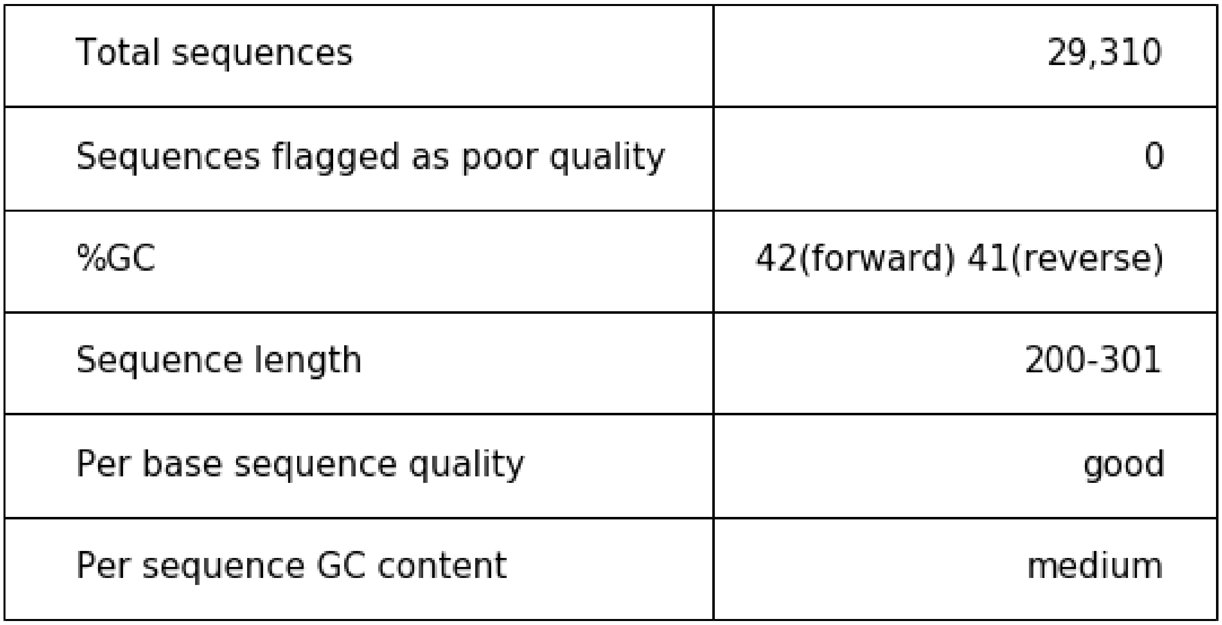
```bash
%%bash -s "$FILE_ID" "$FASTQ_STR" "$MIN_LEN"
ssh microbioinf@192.168.56.101 env FILE_ID=$1 FASTQ_STR=$2 MIN_LEN=$3 2>/dev/null /bin/bash <<"EOT"
export PATH=$PATH:/home/microbioinf/miniconda3/bin
cd Documentos/Tema_3
cd kneaddata_out_${FILE_ID}/
echo "#### Compute quality"
mkdir ${FILE_ID}_HighQuality
fastqc ${FILE_ID}_R1__kneaddata_paired_1.fastq -o ${FILE_ID}_HighQuality/
fastqc ${FILE_ID}_R1__kneaddata_paired_2.fastq -o ${FILE_ID}_HighQuality/
```

Output

```
#### Compute quality
Analysis complete for VIR_R1__kneaddata_paired_1.fastq
Analysis complete for VIR_R1__kneaddata_paired_2.fastq
```

Script 1.3.8 (python)

```python
df_high_qual = pd.DataFrame(data=['29,310', '0', '42(forward) 41(reverse)', '200-301', 'good', 'medium'],
                            index = ['Total sequences', 'Sequences flagged as poor quality', '%GC',
                        'Sequence length', 'Per base sequence quality','Per sequence GC content'])
plt.figure(figsize=(4,5))
plt.axis('off')
table = plt.table(cellText=df_high_qual.values, rowLabels=df_high_qual.index, colWidths = [0.3]*len(df_high_qual.columns),
           loc='top', cellLoc = 'right', rowLoc = 'left', bbox=[0,0,1,1]);
table.auto_set_font_size(False)
table.set_fontsize(15)
```



Total sequences	29,310
Sequences flagged as poor quality	0
%GC	42(forward) 41(reverse)
Sequence length	200-301
Per base sequence quality	good
Per sequence GC content	medium

1.3.3 Assembly (*spades*)

We are going to use a Refseq database of viral proteins (around 100Mb) from ncbi (ftp://ftp.ncbi.nlm.nih.gov/refseq/release/viral/), and you have to download it in two separated files that can be joined into one with cat.

In this step we run command **spades** with the paired high-quality and free of known contaminants reads.

Process for different K_MER

Script 1.3.9 (python)

```
K_MERS_LIST = ["25", "35", "45"]
K_MERS =  ",".join(K_MERS_LIST)
print(K_MERS)
```

Output

```
25,35,45
```

Script 1.3.10 (bash)

```bash
%%bash -s "$FILE_ID" "$FASTQ_STR" "$MIN_LEN" "$K_MERS"
ssh microbioinf@192.168.56.101 env FILE_ID=$1 FASTQ_STR=$2 MIN_LEN=$3 K_MERS=$4 2>/dev/null /bin/bash <<"EOT"

export PATH=$PATH:/home/microbioinf/miniconda3/bin
cd Documentos/Tema_3
cd kneaddata_out_${FILE_ID}/
echo "#### Compute assembly with no specified K_MER"
spades.py -1 ${FILE_ID}_R1__kneaddata_paired_1.fastq -2 ${FILE_ID}_R1__kneaddata_paired_2.fastq \
--sc -o ${FILE_ID}-Assembly${K_MER} 1>/dev/null
IFS=","
for K_MER in ${K_MERS}
do
echo "#### Compute assembly K_MER=${K_MER}"
spades.py -1 ${FILE_ID}_R1__kneaddata_paired_1.fastq -2 ${FILE_ID}_R1__kneaddata_paired_2.fastq \
--sc -k ${K_MER} -o ${FILE_ID}-Assembly${K_MER} 1>/dev/null
done
EOT
```

Output

```
#### Compute assembly with no specified K_MER
```

Script 1.3.11 (bash)

```bash
%%bash -s "$FILE_ID" "$FASTQ_STR" "$MIN_LEN" "$K_MERS"
ssh microbioinf@192.168.56.101 env FILE_ID=$1 FASTQ_STR=$2 MIN_LEN=$3 K_MERS=$4 2>/dev/null /bin/bash <<"EOT"
export PATH=$PATH:/home/microbioinf/miniconda3/bin
cd Documentos/Tema_3
cd kneaddata_out_${FILE_ID}/
echo "#### Spades log with no specified K_MER"
tail -15 ${FILE_ID}-Assembly${K_MER}/spades.log | head -n 11
echo " "
IFS=","
for K_MER in ${K_MERS}
do
echo "#### Spades log with K_MER=${K_MER}"
tail -15 ${FILE_ID}-Assembly${K_MER}/spades.log | head -n 11
done
EOT
```

Output

```
#### Spades log with no specified K_MER
===== Assembling finished. Used k-mer sizes: 21, 33, 55

 * Corrected reads are in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly/corrected/
 * Assembled contigs are in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly/contigs.fasta
 * Assembled scaffolds are in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly/scaffolds.fasta
 * Assembly graph is in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly/assembly_graph.fastg
 * Assembly graph in GFA format is in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/V ⌋
  ↪  IR-Assembly/assembly_graph_with_scaffolds.gfa
 * Paths in the assembly graph corresponding to the contigs are in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly/contigs.paths
 * Paths in the assembly graph corresponding to the scaffolds are in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly/scaffolds.paths

======= SPAdes pipeline finished.

#### Spades log with K_MER=25
===== Assembling finished. Used k-mer sizes: 25

 * Corrected reads are in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly25/corrected/
 * Assembled contigs are in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly25/contigs.fasta
 * Assembled scaffolds are in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly25/scaffolds.fasta
 * Assembly graph is in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly25/assembly_graph.fastg
 * Assembly graph in GFA format is in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/V ⌋
  ↪  IR-Assembly25/assembly_graph_with_scaffolds.gfa
 * Paths in the assembly graph corresponding to the contigs are in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly25/contigs.paths
 * Paths in the assembly graph corresponding to the scaffolds are in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly25/scaffolds.paths

======= SPAdes pipeline finished.
#### Spades log with K_MER=35
===== Assembling finished. Used k-mer sizes: 35

 * Corrected reads are in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly35/corrected/
 * Assembled contigs are in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly35/contigs.fasta
 * Assembled scaffolds are in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly35/scaffolds.fasta
 * Assembly graph is in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly35/assembly_graph.fastg
 * Assembly graph in GFA format is in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/V⌋
  ↪  IR-Assembly35/assembly_graph_with_scaffolds.gfa
 * Paths in the assembly graph corresponding to the contigs are in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly35/contigs.paths
 * Paths in the assembly graph corresponding to the scaffolds are in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly35/scaffolds.paths

======= SPAdes pipeline finished.
#### Spades log with K_MER=45
===== Assembling finished. Used k-mer sizes: 45

 * Corrected reads are in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly45/corrected/
 * Assembled contigs are in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly45/contigs.fasta
 * Assembled scaffolds are in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly45/scaffolds.fasta
 * Assembly graph is in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly45/assembly_graph.fastg
 * Assembly graph in GFA format is in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/V⌋
  ↪  IR-Assembly45/assembly_graph_with_scaffolds.gfa
 * Paths in the assembly graph corresponding to the contigs are in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly45/contigs.paths
 * Paths in the assembly graph corresponding to the scaffolds are in
  ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/VIR-Assembly45/scaffolds.paths

======= SPAdes pipeline finished.
```

Script 1.3.12 (python)

```python
K_MERS_LIST = ["","25", "35", "45"]
K_MERS =  ",".join(K_MERS_LIST)
print(K_MERS)
```

Output

```
,25,35,45
```

Script 1.3.13 (bash)

```bash
%%bash -s "$FILE_ID" "$FASTQ_STR" "$MIN_LEN" "$K_MERS"
ssh microbioinf@192.168.56.101 env FILE_ID=$1 FASTQ_STR=$2 MIN_LEN=$3 K_MERS=$4 2>/dev/null
 ↪  /bin/bash <<"EOT"
export PATH=$PATH:/home/microbioinf/miniconda3/bin
cd Documentos/Tema_3
cd kneaddata_out_${FILE_ID}/
IFS=","
for K_MER in ${K_MERS}
do
    echo
    echo "#### Check output K_MER=${K_MER}"
    cd ${FILE_ID}-Assembly${K_MER}
    rep -c ">" *fasta
    grep ">" -m 8 contigs.fasta
    grep ">" -m 8 scaffolds.fasta
    grep "NN" *fasta
done
```

Output

```
#### Check output K_MER=
>NODE_1_length_87312_cov_12.682593
>NODE_2_length_50238_cov_8.439731
>NODE_3_length_41290_cov_6.676319
>NODE_4_length_31040_cov_9.557496
>NODE_5_length_21751_cov_24.857531
>NODE_6_length_15576_cov_5.415630
>NODE_7_length_15551_cov_5.658363
>NODE_8_length_14173_cov_11.314563
>NODE_1_length_87312_cov_12.682593
>NODE_2_length_50238_cov_8.439731
>NODE_3_length_41290_cov_6.676319
>NODE_4_length_31040_cov_9.557496
>NODE_5_length_21751_cov_24.857531
>NODE_6_length_15576_cov_5.415630
>NODE_7_length_15551_cov_5.658363
>NODE_8_length_14173_cov_11.314563
scaffolds.fasta:TNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
scaffolds.fasta:NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNGGTGCTGGTCAGCAAGGTA
scaffolds.fasta:TGCAGTAGTGCAGCTGGAATCAATCACAATCTTTGNNNNNNNNNNNNNNNNNNNNNNNNN
scaffolds.fasta:NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
scaffolds.fasta:NNNNNNNNNNNNNNNTAATTGCTGAACTCTCGTGCTGCAGTAGTGCAGCTGGAATCAATC
scaffolds.fasta:GAGCGNNNNNNNNNNAATCAATATATCCTGATCCTATCCAGCTGTGTGTGCTGGGATGCC
scaffolds.fasta:ATAGCACATGAAAAAATCTAAACTTTTTGGTTCCAGGTTAGTTGAAAANNNNNNNNNNTG
scaffolds.fasta:CTGTAGATACGCCTGGTGTATAGACNNNNNNNNNNAAAATGATAATCCTGGAGTTCAAAC
scaffolds.fasta:AAAACAACTAATATGAATTCNNNNNNNNNNTTTATTTAAGTTAATCCGCGGGGCACACCT
scaffolds.fasta:CCTGCGCTGGAATTGGAGGGAGCGCAGCTTGATCCACGTTCTTCCCCACTNNNNNNNNNN
scaffolds.fasta:CTGGAGTACGTGCGGCAGTATTTTAAGCCGTTCGCATTTANNNNNNNNNNAGATCCGTTC
scaffolds.fasta:CAACTATGTCTTTAAACATCATTTCCTTTNNNNNNNNNNATTGTTCTTAATATAAATTAA
scaffolds.fasta:CATCACCTACGCGACTGGAATCAACGGGCGCACCAAGCTGTACGGCGGGCNNNNNNNNNN
scaffolds.fasta:TGGTAAATCGTTCAAATACNNNNNNNNNNGTAATGCCCATTATGTATGCAACTAAGCTAC
scaffolds.fasta:TTTACNNNNNNNNNNCAACAATTCCAGCATAGCTAGATAATTCTGGTATAACTGAACTTT
scaffolds.fasta:CTNNNNNNNNNNCTCGAAGCGGCGCTGTTCATCCATCGCGTTGGCAACGTCGAGGCCTTG
scaffolds.fasta:GGGTTACGAAATCCACTGCTTCTCCTAGATAATATACTTCTCCTGACCGCACCACATNNN
scaffolds.fasta:NNNNNNNCCAAACCGGTCGTAATTCCGCTCTACAATATCGCGAACCGTAGTGCTAATGGT
scaffolds.fasta:NNNNNNNNNGCTTCACATATGTTGAAGAGAGTTAAGTTAAATCCTACAGCTCTAATAACA
scaffolds.fasta:GATTACACGGACCATTAACAACAACGATAGTTAATTATATACCAGTTGAGACTNNNNNNN
scaffolds.fasta:NNNGATTCAAGAAGAGCCTGGTAGAATATTAAATCACCCAGCTCAAGATAAACCTCAATT
scaffolds.fasta:AANNNNNNNNNNCAAGACTTGCGATGGCAAGATATTTAGCGGTTCTTGATGCAGAACGAA
scaffolds.fasta:TTCGAGCCAGGTATGTTTCGAAACCGAACCTGTCAATGACGANNNNNNNNNNCTCTAACC
scaffolds.fasta:GAACTCCTCGTACTTCTGACGAGTGGCTCCCGGAAGACGCAGCATGGTGCGNNNNNNNNN
scaffolds.fasta:TAGNNNNNNNNNNCCAGCAAAGTAATGCGTAACCGCCGTACGGAAAACCGCACTGTCGAA
scaffolds.fasta:TGCCTTTAGTAATGATNNNNNNNNNNTATTAGACTTACTATCAAGATCTAATTGATCTAC
scaffolds.fasta:TTATCAAACAATTAGTAAAACGGTACAAAACAATTGAANNNNNNNNNNCAATTGAAGAAC
scaffolds.fasta:GTGCACTTTACCCCTTCCTGATTNNNNNNNNNNATCCCAATATTCTATATCGTCTAACTG
scaffolds.fasta:CGATTTCAGCAATAGTTTCTACGACCTCATNNNNNNNNNNCTTATATCGAATTGAAAACC
scaffolds.fasta:GGGCTTCGATATTATGTNNNNNNNNNNCTGAGTTTGTAGGGCTGTACTATACAGCTTACG
scaffolds.fasta:CGGACGGGGTGTAGCGCCTGGCCTNNNNNNNNNNACCGCGCGGCGGTTGATGACGTACTC
scaffolds.fasta:ACTGTATCTTTAGAGGGAGAAAACTCTTCTAAATATATGCTTTCATTAANNNNNNNNNNT
scaffolds.fasta:ATTCAATTTAGAATCATAAAAGNNNNNNNNNNTTAGGAATCTCAATTGTAGTTGGCTCAG
scaffolds.fasta:TATAGCACCAGCAGCGATGCCCTGANNNNNNNNNNGCACTGAAACCATACCTGCCGATTG

#### Check output K_MER=25
>NODE_1_length_87312_cov_12.682593
>NODE_2_length_50238_cov_8.439731
>NODE_3_length_41290_cov_6.676319
>NODE_4_length_31040_cov_9.557496
>NODE_5_length_21751_cov_24.857531
>NODE_6_length_15576_cov_5.415630
>NODE_7_length_15551_cov_5.658363
>NODE_8_length_14173_cov_11.314563
>NODE_1_length_87312_cov_12.682593
>NODE_2_length_50238_cov_8.439731
>NODE_3_length_41290_cov_6.676319
>NODE_4_length_31040_cov_9.557496
>NODE_5_length_21751_cov_24.857531
>NODE_6_length_15576_cov_5.415630
>NODE_7_length_15551_cov_5.658363
>NODE_8_length_14173_cov_11.314563
scaffolds.fasta:TNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
scaffolds.fasta:NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNGGTGCTGGTCAGCAAGGTA
scaffolds.fasta:TGCAGTAGTGCAGCTGGAATCAATCACAATCTTTGNNNNNNNNNNNNNNNNNNNNNNNNN
scaffolds.fasta:NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
scaffolds.fasta:NNNNNNNNNNNNNNNTAATTGCTGAACTCTCGTGCTGCAGTAGTGCAGCTGGAATCAATC
scaffolds.fasta:GAGCGNNNNNNNNNNAATCAATATATCCTGATCCTATCCAGCTGTGTGTGCTGGGATGCC
scaffolds.fasta:ATAGCACATGAAAAAATCTAAACTTTTTGGTTCCAGGTTAGTTGAAAANNNNNNNNNNTG
scaffolds.fasta:CTGTAGATACGCCTGGTGTATAGACNNNNNNNNNNAAAATGATAATCCTGGAGTTCAAAC
scaffolds.fasta:AAAACAACTAATATGAATTCNNNNNNNNNNTTTATTTAAGTTAATCCGCGGGGCACACCT
scaffolds.fasta:CCTGCGCTGGAATTGGAGGGAGCGCAGCTTGATCCACGTTCTTCCCCACTNNNNNNNNNN
scaffolds.fasta:CTGGAGTACGTGCGGCAGTATTTTAAGCCGTTCGCATTTANNNNNNNNNNAGATCCGTTC
scaffolds.fasta:CAACTATGTCTTTAAACATCATTTCCTTTNNNNNNNNNNATTGTTCTTAATATAAATTAA
scaffolds.fasta:CATCACCTACGCGACTGGAATCAACGGGCGCACCAAGCTGTACGGCGGGCNNNNNNNNNN
scaffolds.fasta:TGGTAAATCGTTCAAATACNNNNNNNNNNGTAATGCCCATTATGTATGCAACTAAGCTAC
scaffolds.fasta:TTTACNNNNNNNNNNCAACAATTCCAGCATAGCTAGATAATTCTGGTATAACTGAACTTT
scaffolds.fasta:CTNNNNNNNNNNCTCGAAGCGGCGCTGTTCATCCATCGCGTTGGCAACGTCGAGGCCTTG
scaffolds.fasta:GGGTTACGAAATCCACTGCTTCTCCTAGATAATATACTTCTCCTGACCGCACCACATNNN
scaffolds.fasta:NNNNNNNCCAAACCGGTCGTAATTCCGCTCTACAATATCGCGAACCGTAGTGCTAATGGT
scaffolds.fasta:NNNNNNNNNGCTTCACATATGTTGAAGAGAGTTAAGTTAAATCCTACAGCTCTAATAACA
scaffolds.fasta:GATTACACGGACCATTAACAACAACGATAGTTAATTATATACCAGTTGAGACTNNNNNNN
scaffolds.fasta:NNNGATTCAAGAAGAGCCTGGTAGAATATTAAATCACCCAGCTCAAGATAAACCTCAATT
scaffolds.fasta:AANNNNNNNNNNCAAGACTTGCGATGGCAAGATATTTAGCGGTTCTTGATGCAGAACGAA
scaffolds.fasta:TTCGAGCCAGGTATGTTTCGAAACCGAACCTGTCAATGACGANNNNNNNNNNCTCTAACC
scaffolds.fasta:GAACTCCTCGTACTTCTGACGAGTGGCTCCCGGAAGACGCAGCATGGTGCGNNNNNNNNN
scaffolds.fasta:TAGNNNNNNNNNNCCAGCAAAGTAATGCGTAACCGCCGTACGGAAAACCGCACTGTCGAA
scaffolds.fasta:TGCCTTTAGTAATGATNNNNNNNNNNTATTAGACTTACTATCAAGATCTAATTGATCTAC
scaffolds.fasta:TTATCAAACAATTAGTAAAACGGTACAAAACAATTGAANNNNNNNNNNCAATTGAAGAAC
scaffolds.fasta:GTGCACTTTACCCCTTCCTGATTNNNNNNNNNNATCCCAATATTCTATATCGTCTAACTG
scaffolds.fasta:CGATTTCAGCAATAGTTTCTACGACCTCATNNNNNNNNNNCTTATATCGAATTGAAAACC
scaffolds.fasta:GGGCTTCGATATTATGTNNNNNNNNNNCTGAGTTTGTAGGGCTGTACTATACAGCTTACG
scaffolds.fasta:CGGACGGGGTGTAGCGCCTGGCCTNNNNNNNNNNACCGCGCGGCGGTTGATGACGTACTC
scaffolds.fasta:ACTGTATCTTTAGAGGGAGAAAACTCTTCTAAATATATGCTTTCATTAANNNNNNNNNNT
scaffolds.fasta:ATTCAATTTAGAATCATAAAAGNNNNNNNNNNTTAGGAATCTCAATTGTAGTTGGCTCAG
scaffolds.fasta:TATAGCACCAGCAGCGATGCCCTGANNNNNNNNNNGCACTGAAACCATACCTGCCGATTG

#### Check output K_MER=35
>NODE_1_length_87312_cov_12.682593
>NODE_2_length_50238_cov_8.439731
>NODE_3_length_41290_cov_6.676319
>NODE_4_length_31040_cov_9.557496
>NODE_5_length_21751_cov_24.857531
>NODE_6_length_15576_cov_5.415630
>NODE_7_length_15551_cov_5.658363
>NODE_8_length_14173_cov_11.314563
>NODE_1_length_87312_cov_12.682593
>NODE_2_length_50238_cov_8.439731
>NODE_3_length_41290_cov_6.676319
>NODE_4_length_31040_cov_9.557496
>NODE_5_length_21751_cov_24.857531
>NODE_6_length_15576_cov_5.415630
>NODE_7_length_15551_cov_5.658363
>NODE_8_length_14173_cov_11.314563
scaffolds.fasta:TNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
scaffolds.fasta:NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNGGTGCTGGTCAGCAAGGTA
scaffolds.fasta:TGCAGTAGTGCAGCTGGAATCAATCACAATCTTTGNNNNNNNNNNNNNNNNNNNNNNNNN
scaffolds.fasta:NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
scaffolds.fasta:NNNNNNNNNNNNNNNTAATTGCTGAACTCTCGTGCTGCAGTAGTGCAGCTGGAATCAATC
scaffolds.fasta:GAGCGNNNNNNNNNNAATCAATATATCCTGATCCTATCCAGCTGTGTGTGCTGGGATGCC
scaffolds.fasta:ATAGCACATGAAAAAATCTAAACTTTTTGGTTCCAGGTTAGTTGAAAANNNNNNNNNNTG
scaffolds.fasta:CTGTAGATACGCCTGGTGTATAGACNNNNNNNNNNAAAATGATAATCCTGGAGTTCAAAC
scaffolds.fasta:AAAACAACTAATATGAATTCNNNNNNNNNNTTTATTTAAGTTAATCCGCGGGGCACACCT
scaffolds.fasta:CCTGCGCTGGAATTGGAGGGAGCGCAGCTTGATCCACGTTCTTCCCCACTNNNNNNNNNN
scaffolds.fasta:CTGGAGTACGTGCGGCAGTATTTTAAGCCGTTCGCATTTANNNNNNNNNNAGATCCGTTC
scaffolds.fasta:CAACTATGTCTTTAAACATCATTTCCTTTNNNNNNNNNNATTGTTCTTAATATAAATTAA
scaffolds.fasta:CATCACCTACGCGACTGGAATCAACGGGCGCACCAAGCTGTACGGCGGGCNNNNNNNNNN
scaffolds.fasta:TGGTAAATCGTTCAAATACNNNNNNNNNNGTAATGCCCATTATGTATGCAACTAAGCTAC
scaffolds.fasta:TTTACNNNNNNNNNNCAACAATTCCAGCATAGCTAGATAATTCTGGTATAACTGAACTTT
scaffolds.fasta:CTNNNNNNNNNNCTCGAAGCGGCGCTGTTCATCCATCGCGTTGGCAACGTCGAGGCCTTG
scaffolds.fasta:GGGTTACGAAATCCACTGCTTCTCCTAGATAATATACTTCTCCTGACCGCACCACATNNN
scaffolds.fasta:NNNNNNNCCAAACCGGTCGTAATTCCGCTCTACAATATCGCGAACCGTAGTGCTAATGGT
scaffolds.fasta:NNNNNNNNNGCTTCACATATGTTGAAGAGAGTTAAGTTAAATCCTACAGCTCTAATAACA
scaffolds.fasta:GATTACACGGACCATTAACAACAACGATAGTTAATTATATACCAGTTGAGACTNNNNNNN
scaffolds.fasta:NNNGATTCAAGAAGAGCCTGGTAGAATATTAAATCACCCAGCTCAAGATAAACCTCAATT
scaffolds.fasta:AANNNNNNNNNNCAAGACTTGCGATGGCAAGATATTTAGCGGTTCTTGATGCAGAACGAA
scaffolds.fasta:TTCGAGCCAGGTATGTTTCGAAACCGAACCTGTCAATGACGANNNNNNNNNNCTCTAACC
scaffolds.fasta:GAACTCCTCGTACTTCTGACGAGTGGCTCCCGGAAGACGCAGCATGGTGCGNNNNNNNNN
scaffolds.fasta:TAGNNNNNNNNNNCCAGCAAAGTAATGCGTAACCGCCGTACGGAAAACCGCACTGTCGAA
scaffolds.fasta:TGCCTTTAGTAATGATNNNNNNNNNNTATTAGACTTACTATCAAGATCTAATTGATCTAC
scaffolds.fasta:TTATCAAACAATTAGTAAAACGGTACAAAACAATTGAANNNNNNNNNNCAATTGAAGAAC
scaffolds.fasta:GTGCACTTTACCCCTTCCTGATTNNNNNNNNNNATCCCAATATTCTATATCGTCTAACTG
scaffolds.fasta:CGATTTCAGCAATAGTTTCTACGACCTCATNNNNNNNNNNCTTATATCGAATTGAAAACC
scaffolds.fasta:GGGCTTCGATATTATGTNNNNNNNNNNCTGAGTTTGTAGGGCTGTACTATACAGCTTACG
scaffolds.fasta:CGGACGGGGTGTAGCGCCTGGCCTNNNNNNNNNNACCGCGCGGCGGTTGATGACGTACTC
scaffolds.fasta:ACTGTATCTTTAGAGGGAGAAAACTCTTCTAAATATATGCTTTCATTAANNNNNNNNNNT
scaffolds.fasta:ATTCAATTTAGAATCATAAAAGNNNNNNNNNNTTAGGAATCTCAATTGTAGTTGGCTCAG
scaffolds.fasta:TATAGCACCAGCAGCGATGCCCTGANNNNNNNNNNGCACTGAAACCATACCTGCCGATTG

#### Check output K_MER=45
>NODE_1_length_87312_cov_12.682593
>NODE_2_length_50238_cov_8.439731
>NODE_3_length_41290_cov_6.676319
>NODE_4_length_31040_cov_9.557496
>NODE_5_length_21751_cov_24.857531
>NODE_6_length_15576_cov_5.415630
>NODE_7_length_15551_cov_5.658363
>NODE_8_length_14173_cov_11.314563
>NODE_1_length_87312_cov_12.682593
>NODE_2_length_50238_cov_8.439731
>NODE_3_length_41290_cov_6.676319
>NODE_4_length_31040_cov_9.557496
>NODE_5_length_21751_cov_24.857531
>NODE_6_length_15576_cov_5.415630
>NODE_7_length_15551_cov_5.658363
>NODE_8_length_14173_cov_11.314563
scaffolds.fasta:TNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
scaffolds.fasta:NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNGGTGCTGGTCAGCAAGGTA
scaffolds.fasta:TGCAGTAGTGCAGCTGGAATCAATCACAATCTTTGNNNNNNNNNNNNNNNNNNNNNNNNN
scaffolds.fasta:NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
scaffolds.fasta:NNNNNNNNNNNNNNNTAATTGCTGAACTCTCGTGCTGCAGTAGTGCAGCTGGAATCAATC
scaffolds.fasta:GAGCGNNNNNNNNNNAATCAATATATCCTGATCCTATCCAGCTGTGTGTGCTGGGATGCC
scaffolds.fasta:ATAGCACATGAAAAAATCTAAACTTTTTGGTTCCAGGTTAGTTGAAAANNNNNNNNNNTG
scaffolds.fasta:CTGTAGATACGCCTGGTGTATAGACNNNNNNNNNNAAAATGATAATCCTGGAGTTCAAAC
scaffolds.fasta:AAAACAACTAATATGAATTCNNNNNNNNNNTTTATTTAAGTTAATCCGCGGGGCACACCT
scaffolds.fasta:CCTGCGCTGGAATTGGAGGGAGCGCAGCTTGATCCACGTTCTTCCCCACTNNNNNNNNNN
scaffolds.fasta:CTGGAGTACGTGCGGCAGTATTTTAAGCCGTTCGCATTTANNNNNNNNNNAGATCCGTTC
scaffolds.fasta:CAACTATGTCTTTAAACATCATTTCCTTTNNNNNNNNNNATTGTTCTTAATATAAATTAA
scaffolds.fasta:CATCACCTACGCGACTGGAATCAACGGGCGCACCAAGCTGTACGGCGGGCNNNNNNNNNN
scaffolds.fasta:TGGTAAATCGTTCAAATACNNNNNNNNNNGTAATGCCCATTATGTATGCAACTAAGCTAC
scaffolds.fasta:TTTACNNNNNNNNNNCAACAATTCCAGCATAGCTAGATAATTCTGGTATAACTGAACTTT
scaffolds.fasta:CTNNNNNNNNNNCTCGAAGCGGCGCTGTTCATCCATCGCGTTGGCAACGTCGAGGCCTTG
scaffolds.fasta:GGGTTACGAAATCCACTGCTTCTCCTAGATAATATACTTCTCCTGACCGCACCACATNNN
scaffolds.fasta:NNNNNNNCCAAACCGGTCGTAATTCCGCTCTACAATATCGCGAACCGTAGTGCTAATGGT
scaffolds.fasta:NNNNNNNNNGCTTCACATATGTTGAAGAGAGTTAAGTTAAATCCTACAGCTCTAATAACA
scaffolds.fasta:GATTACACGGACCATTAACAACAACGATAGTTAATTATATACCAGTTGAGACTNNNNNNN
scaffolds.fasta:NNNGATTCAAGAAGAGCCTGGTAGAATATTAAATCACCCAGCTCAAGATAAACCTCAATT
scaffolds.fasta:AANNNNNNNNNNCAAGACTTGCGATGGCAAGATATTTAGCGGTTCTTGATGCAGAACGAA
scaffolds.fasta:TTCGAGCCAGGTATGTTTCGAAACCGAACCTGTCAATGACGANNNNNNNNNNCTCTAACC
scaffolds.fasta:GAACTCCTCGTACTTCTGACGAGTGGCTCCCGGAAGACGCAGCATGGTGCGNNNNNNNNN
scaffolds.fasta:TAGNNNNNNNNNNCCAGCAAAGTAATGCGTAACCGCCGTACGGAAAACCGCACTGTCGAA
scaffolds.fasta:TGCCTTTAGTAATGATNNNNNNNNNNTATTAGACTTACTATCAAGATCTAATTGATCTAC
scaffolds.fasta:TTATCAAACAATTAGTAAAACGGTACAAAACAATTGAANNNNNNNNNNCAATTGAAGAAC
scaffolds.fasta:GTGCACTTTACCCCTTCCTGATTNNNNNNNNNNATCCCAATATTCTATATCGTCTAACTG
scaffolds.fasta:CGATTTCAGCAATAGTTTCTACGACCTCATNNNNNNNNNNCTTATATCGAATTGAAAACC
scaffolds.fasta:GGGCTTCGATATTATGTNNNNNNNNNNCTGAGTTTGTAGGGCTGTACTATACAGCTTACG
scaffolds.fasta:CGGACGGGGTGTAGCGCCTGGCCTNNNNNNNNNNACCGCGCGGCGGTTGATGACGTACTC
scaffolds.fasta:ACTGTATCTTTAGAGGGAGAAAACTCTTCTAAATATATGCTTTCATTAANNNNNNNNNNT
scaffolds.fasta:ATTCAATTTAGAATCATAAAAGNNNNNNNNNNTTAGGAATCTCAATTGTAGTTGGCTCAG
scaffolds.fasta:TATAGCACCAGCAGCGATGCCCTGANNNNNNNNNNGCACTGAAACCATACCTGCCGATTG
```

Comparison of assemblies (*quast*)

Script 1.3.14 (bash)

```bash
%%bash -s "$FILE_ID" "$FASTQ_STR" "$MIN_LEN" "$K_MER"
ssh microbioinf@192.168.56.101 env FILE_ID=$1 FASTQ_STR=$2 MIN_LEN=$3 K_MER=$4 2>/dev/null
↪  /bin/bash <<"EOT"
export PATH=$PATH:/home/microbioinf/miniconda3/bin
cd Documentos/Tema_3
cd kneaddata_out_${FILE_ID}/
echo "#### Compare assemblies FILE_ID=${FILE_ID}"
for assembly in ${FILE_ID}-Assembly*;
    do echo "Processing $assembly file...";
    cp ${assembly}/contigs.fasta contigs-${assembly}.fasta
    cp ${assembly}/scaffolds.fasta scaffolds-${assembly}.fasta
done
quast.py contigs* scaffolds* -R ../ECTV-MoscowGenome.fasta 1>/dev/null
EOT
```

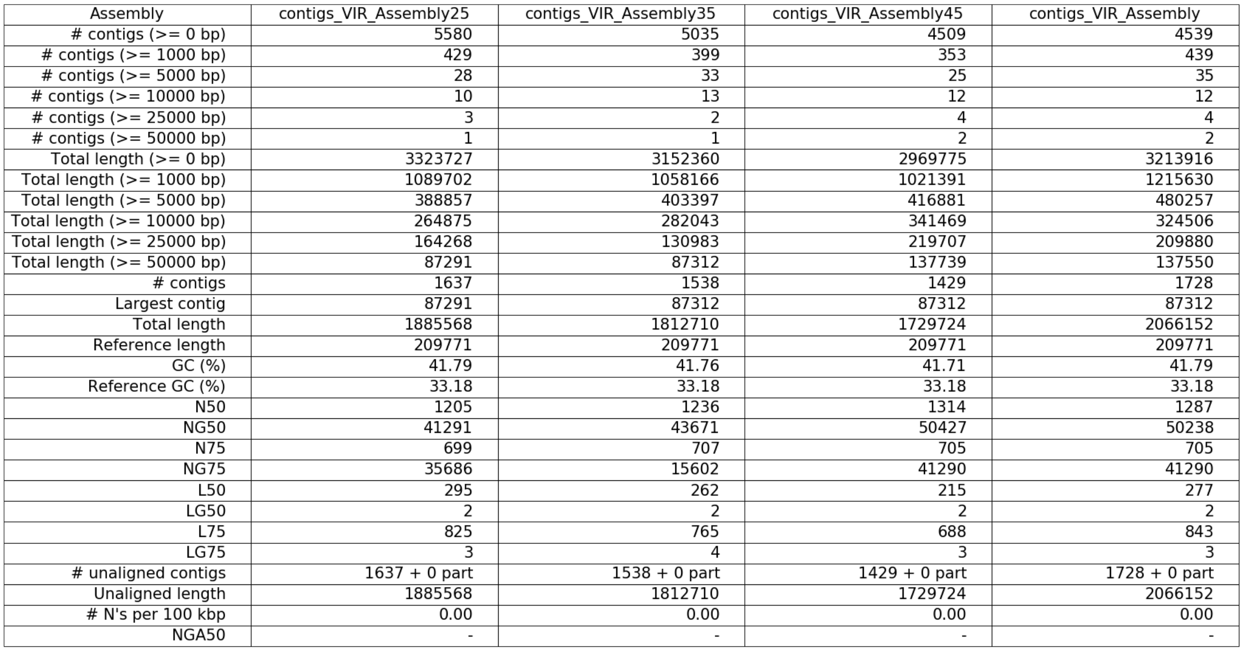
Output

```
#### Compare assemblies FILE_ID=VIR
Processing VIR-Assembly file...
Processing VIR-Assembly25 file...
Processing VIR-Assembly35 file...
Processing VIR-Assembly45 file...
```

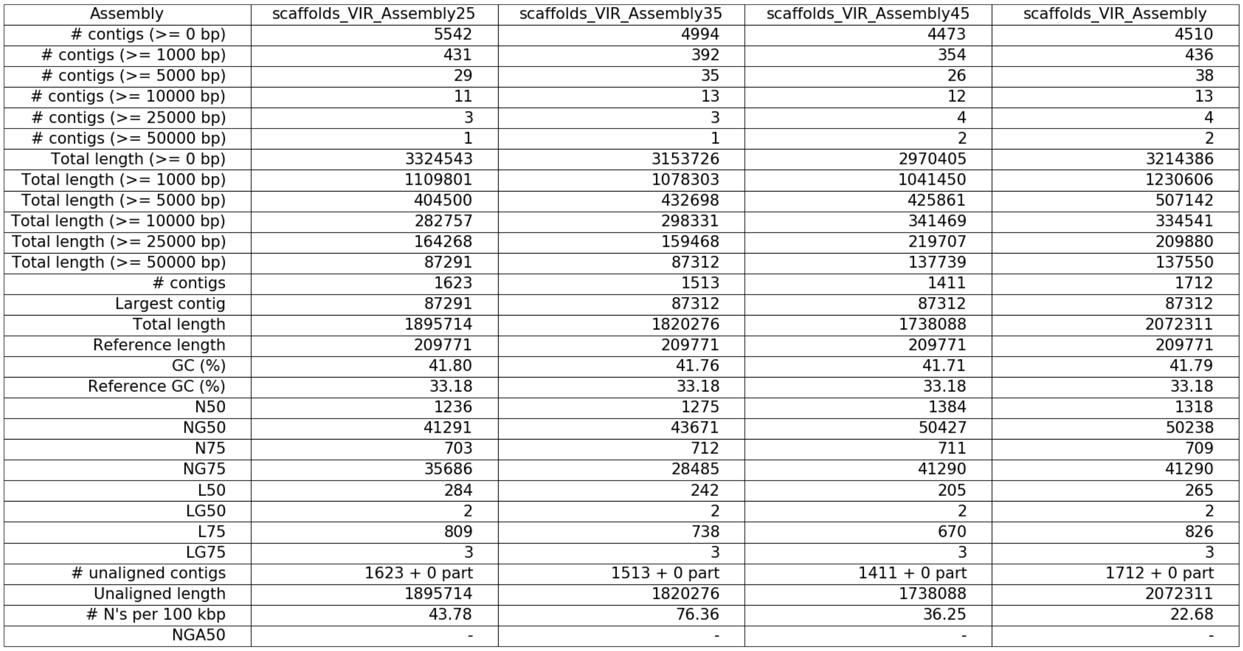
Script 1.3.15 (python)

```python
data = """
cat Documentos/Tema_3/kneaddata_out_%s/quast*/latest/report.tsv
EOT
""" % FILE_ID
output = !ssh microbioinf@192.168.56.101 /bin/bash <<'EOT' {data}
data = []
# To list of lists
for row in output:
    data.append(row.split('\t'))
# To dataframe
```

```
df_quast = pd.DataFrame(data[1:], columns=data[0])

df_quast_contigs = df_quast.iloc[:,0:5]
fig = plt.figure(figsize=(15,8))
ax = plt.subplot(111)
ax.axis('off')
table = plt.table(cellText=df_quast_contigs.values, colLabels=df_quast_contigs.columns,
          colWidths = [2]*len(df_quast_contigs.columns),
          loc='top',
          cellLoc = 'right', rowLoc = 'left',
          bbox=[0,0,2,2]);

table.auto_set_font_size(False)
table.set_fontsize(21)

df_quast_scaffolds = df_quast.iloc[:,[0,5,6,7,8]]
fig = plt.figure(figsize=(15,8))
ax = plt.subplot(111)
ax.axis('off')
table = plt.table(cellText=df_quast_scaffolds.values, colLabels=df_quast_scaffolds.columns,
          colWidths = [2]*len(df_quast_scaffolds.columns),
          loc='top',
          cellLoc = 'right', rowLoc = 'left',
          bbox=[0,0,2,2]);

table.auto_set_font_size(False)
table.set_fontsize(21)
```



Assembly	contigs_VIR_Assembly25	contigs_VIR_Assembly35	contigs_VIR_Assembly45	contigs_VIR_Assembly
# contigs (>= 0 bp)	5580	5035	4509	4539
# contigs (>= 1000 bp)	429	399	353	439
# contigs (>= 5000 bp)	28	33	25	35
# contigs (>= 10000 bp)	10	13	12	12
# contigs (>= 25000 bp)	3	2	4	4
# contigs (>= 50000 bp)	1	1	2	2
Total length (>= 0 bp)	3323727	3152360	2969775	3213916
Total length (>= 1000 bp)	1089702	1058166	1021391	1215630
Total length (>= 5000 bp)	388857	403397	416881	480257
Total length (>= 10000 bp)	264875	282043	341469	324506
Total length (>= 25000 bp)	164268	130983	219707	209880
Total length (>= 50000 bp)	87291	87312	137739	137550
# contigs	1637	1538	1429	1728
Largest contig	87291	87312	87312	87312
Total length	1885568	1812710	1729724	2066152
Reference length	209771	209771	209771	209771
GC (%)	41.79	41.76	41.71	41.79
Reference GC (%)	33.18	33.18	33.18	33.18
N50	1205	1236	1314	1287
NG50	41291	43671	50427	50238
N75	699	707	705	705
NG75	35686	15602	41290	41290
L50	295	262	215	277
LG50	2	2	2	2
L75	825	765	688	843
LG75	3	4	3	3
# unaligned contigs	1637 + 0 part	1538 + 0 part	1429 + 0 part	1728 + 0 part
Unaligned length	1885568	1812710	1729724	2066152
# N's per 100 kbp	0.00	0.00	0.00	0.00
NGA50	-	-	-	-



Assembly	scaffolds_VIR_Assembly25	scaffolds_VIR_Assembly35	scaffolds_VIR_Assembly45	scaffolds_VIR_Assembly
# contigs (>= 0 bp)	5542	4994	4473	4510
# contigs (>= 1000 bp)	431	392	354	436
# contigs (>= 5000 bp)	29	35	26	38
# contigs (>= 10000 bp)	11	13	12	13
# contigs (>= 25000 bp)	3	3	4	4
# contigs (>= 50000 bp)	1	1	2	2
Total length (>= 0 bp)	3324543	3153726	2970405	3214386
Total length (>= 1000 bp)	1109801	1078303	1041450	1230606
Total length (>= 5000 bp)	404500	432698	425861	507142
Total length (>= 10000 bp)	282757	298331	341469	334541
Total length (>= 25000 bp)	164268	159468	219707	209880
Total length (>= 50000 bp)	87291	87312	137739	137550
# contigs	1623	1513	1411	1712
Largest contig	87291	87312	87312	87312
Total length	1895714	1820276	1738088	2072311
Reference length	209771	209771	209771	209771
GC (%)	41.80	41.76	41.71	41.79
Reference GC (%)	33.18	33.18	33.18	33.18
N50	1236	1275	1384	1318
NG50	41291	43671	50427	50238
N75	703	712	711	709
NG75	35686	28485	41290	41290
L50	284	242	205	265
LG50	2	2	2	2
L75	809	738	670	826
LG75	3	3	3	3
# unaligned contigs	1623 + 0 part	1513 + 0 part	1411 + 0 part	1712 + 0 part
Unaligned length	1895714	1820276	1738088	2072311
# N's per 100 kbp	43.78	76.36	36.25	22.68
NGA50	-	-	-	-

1.3.4 Assembly (*metaspades*)

Process for different K_MER

Script 1.3.16 (python)

```
K_MERS_LIST = ["25", "35", "45"]
K_MERS =  ",".join(K_MERS_LIST)
print(K_MERS)
```

Output

```
25,35,45
```

Script 1.3.17 (bash)

```
%%bash -s "$FILE_ID" "$FASTQ_STR" "$MIN_LEN" "$K_MERS"
ssh microbioinf@192.168.56.101 env FILE_ID=$1 FASTQ_STR=$2 MIN_LEN=$3 K_MERS=$4 2>/dev/null /bin/bash <<"EOT"
export PATH=$PATH:/home/microbioinf/miniconda3/bin
cd Documentos/Tema_3
cd kneaddata_out_${FILE_ID}/
echo "#### Compute assembly with no specified K_MER"
metaspades.py -1 ${FILE_ID}_R1__kneaddata_paired_1.fastq -2 ${FILE_ID}_R1__kneaddata_paired_2.fastq \
--meta -o meta-${FILE_ID}-Assembly${K_MER} 1>/dev/null
IFS=","
```

```bash
for K_MER in ${K_MERS}
do
echo "#### Compute assembly K_MER=${K_MER}"
metaspades.py -1 ${FILE_ID}_R1__kneaddata_paired_1.fastq -2 ${FILE_ID}_R1__kneaddata_paired_2.fastq \
--meta -k ${K_MER} -o meta-${FILE_ID}-Assembly${K_MER} 1>/dev/null
done
EOT
```

Output

```
#### Compute assembly with no specified K_MER
#### Compute assembly K_MER=25
#### Compute assembly K_MER=35
#### Compute assembly K_MER=45
```

Script 1.3.18 (bash)

```bash
%%bash -s "$FILE_ID" "$FASTQ_STR" "$MIN_LEN" "$K_MERS"
ssh microbioinf@192.168.56.101 env FILE_ID=$1 FASTQ_STR=$2 MIN_LEN=$3 K_MERS=$4 2>/dev/null /bin/bash <<"EOT"
export PATH=$PATH:/home/microbioinf/miniconda3/bin
cd Documentos/Tema_3
cd kneaddata_out_${FILE_ID}/
echo "#### Spades log with no specified K_MER"
tail -15 meta-${FILE_ID}-Assembly${K_MER}/spades.log | head -n 11
echo " "
IFS=","
for K_MER in ${K_MERS}
do
echo "#### Spades log with K_MER=${K_MER}"
tail -15 meta-${FILE_ID}-Assembly${K_MER}/spades.log | head -n 11
done
EOT
```

Output

```
#### Spades log with no specified K_MER
===== Assembling finished. Used k-mer sizes: 21, 33, 55

 * Corrected reads are in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly/corrected/
 * Assembled contigs are in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly/contigs.fasta
 * Assembled scaffolds are in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly/scaffolds.fasta
 * Assembly graph is in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly/assembly_graph.fastg
 * Assembly graph in GFA format is in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly/assembly_graph_with_scaffolds.gfa
 * Paths in the assembly graph corresponding to the contigs are in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly/contigs.paths
 * Paths in the assembly graph corresponding to the scaffolds are in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly/scaffolds.paths

======= SPAdes pipeline finished.

#### Spades log with K_MER=25
===== Assembling finished. Used k-mer sizes: 25

 * Corrected reads are in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly25/corrected/
 * Assembled contigs are in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly25/contigs.fasta
 * Assembled scaffolds are in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly25/scaffolds.fasta
 * Assembly graph is in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly25/assembly_graph.fastg
 * Assembly graph in GFA format is in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly25/assembly_graph_with_scaffolds.gfa
 * Paths in the assembly graph corresponding to the contigs are in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly25/contigs.paths
 * Paths in the assembly graph corresponding to the scaffolds are in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly25/scaffolds.paths

======= SPAdes pipeline finished.
#### Spades log with K_MER=35
===== Assembling finished. Used k-mer sizes: 35

 * Corrected reads are in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly35/corrected/
 * Assembled contigs are in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly35/contigs.fasta
 * Assembled scaffolds are in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly35/scaffolds.fasta
 * Assembly graph is in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly35/assembly_graph.fastg
 * Assembly graph in GFA format is in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly35/assembly_graph_with_scaffolds.gfa
 * Paths in the assembly graph corresponding to the contigs are in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly35/contigs.paths
 * Paths in the assembly graph corresponding to the scaffolds are in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly35/scaffolds.paths

======= SPAdes pipeline finished.
#### Spades log with K_MER=45
===== Assembling finished. Used k-mer sizes: 45

 * Corrected reads are in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly45/corrected/
 * Assembled contigs are in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly45/contigs.fasta
 * Assembled scaffolds are in
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly45/scaffolds.fasta
 * Assembly graph is in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assemb ⌋
 ↪  ly45/assembly_graph.fastg
 * Assembly graph in GFA format is in /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/m ⌋
 ↪  eta-VIR-Assembly45/assembly_graph_with_scaffolds.gfa
 * Paths in the assembly graph corresponding to the contigs are in
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly45/contigs.paths
 * Paths in the assembly graph corresponding to the scaffolds are in
 ↪  /home/microbioinf/Documentos/Tema_3/kneaddata_out_VIR/meta-VIR-Assembly45/scaffolds.paths

======= SPAdes pipeline finished.
```

Comparison of assemblies (*quast*)

Script 1.3.19 (bash)

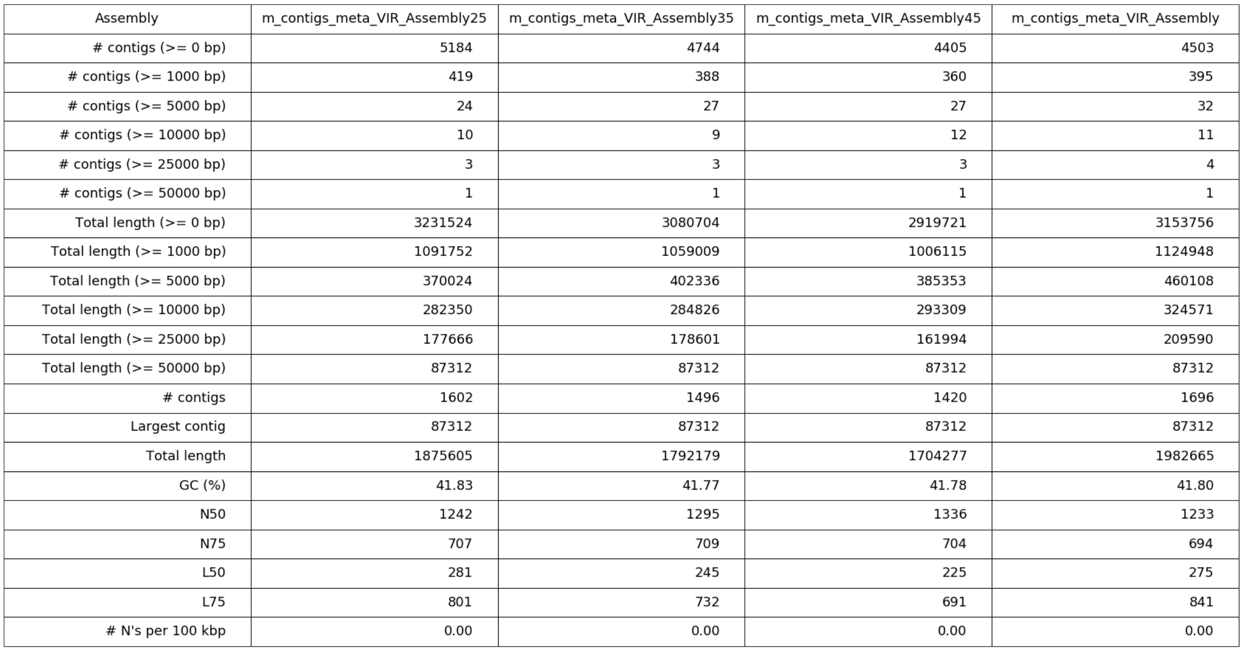
```bash
%%bash -s "$FILE_ID"
ssh microbioinf@192.168.56.101 env FILE_ID=$1 2>/dev/null /bin/bash <<"EOT"
export PATH=$PATH:/home/microbioinf/miniconda3/bin
cd Documentos/Tema_3
cd kneaddata_out_${FILE_ID}/
echo "#### Compare assemblies FILE_ID=${FILE_ID}"
for assembly in meta-${FILE_ID}-Assembly*;
    do echo "Processing $assembly file...";
    cp ${assembly}/contigs.fasta m-contigs-${assembly}.fasta
    cp ${assembly}/scaffolds.fasta m-scaffolds-${assembly}.fasta
done
quast.py m-contigs* m-scaffolds* 1>/dev/null
EOT
```

Output

```
#### Compare assemblies FILE_ID=VIR
Processing meta-VIR-Assembly file...
Processing meta-VIR-Assembly25 file...
Processing meta-VIR-Assembly35 file...
Processing meta-VIR-Assembly45 file...
```

Script 1.3.20 (python)

```python
data = """
cat Documentos/Tema_3/kneaddata_out_%s/quast*/latest/report.tsv
EOT
""" % FILE_ID
output = !ssh microbioinf@192.168.56.101 /bin/bash <<'EOT' {data}
data = []
# To list of lists
for row in output:
    data.append(row.split('\t'))
# To dataframe
df_quast_meta = pd.DataFrame(data[1:], columns=data[0])

df_quast_contigs = df_quast_meta.iloc[:,0:5]
fig = plt.figure(figsize=(15,8))
ax = plt.subplot(111)
ax.axis('off')
table = plt.table(cellText=df_quast_contigs.values, colLabels=df_quast_contigs.columns,
          colWidths = [2]*len(df_quast_contigs.columns),
          loc='top',
          cellLoc = 'right', rowLoc = 'left',
          bbox=[0,0,2,2]);

table.auto_set_font_size(False)
table.set_fontsize(18)

df_quast_scaffolds = df_quast_meta.iloc[:,[0,5,6,7,8]]
fig = plt.figure(figsize=(15,8))
ax = plt.subplot(111)
ax.axis('off')
table = plt.table(cellText=df_quast_scaffolds.values, colLabels=df_quast_scaffolds.columns,
          colWidths = [2]*len(df_quast_scaffolds.columns),
          loc='top',
          cellLoc = 'right', rowLoc = 'left',
          bbox=[0,0,2,2]);

table.auto_set_font_size(False)
table.set_fontsize(18)
```



Assembly	m_contigs_meta_VIR_Assembly25	m_contigs_meta_VIR_Assembly35	m_contigs_meta_VIR_Assembly45	m_contigs_meta_VIR_Assembly
# contigs (>= 0 bp)	5184	4744	4405	4503
# contigs (>= 1000 bp)	419	388	360	395
# contigs (>= 5000 bp)	24	27	27	32
# contigs (>= 10000 bp)	10	9	12	11
# contigs (>= 25000 bp)	3	3	3	4
# contigs (>= 50000 bp)	1	1	1	1
Total length (>= 0 bp)	3231524	3080704	2919721	3153756
Total length (>= 1000 bp)	1091752	1059009	1006115	1124948
Total length (>= 5000 bp)	370024	402336	385353	460108
Total length (>= 10000 bp)	282350	284826	293309	324571
Total length (>= 25000 bp)	177666	178601	161994	209590
Total length (>= 50000 bp)	87312	87312	87312	87312
# contigs	1602	1496	1420	1696
Largest contig	87312	87312	87312	87312
Total length	1875605	1792179	1704277	1982665
GC (%)	41.83	41.77	41.78	41.80
N50	1242	1295	1336	1233
N75	707	709	704	694
L50	281	245	225	275
L75	801	732	691	841
# N's per 100 kbp	0.00	0.00	0.00	0.00

Assembly	m_scaffolds_meta_VIR_Assembly25	m_scaffolds_meta_VIR_Assembly35	m_scaffolds_meta_VIR_Assembly45	m_scaffolds_meta_VIR_Assembly
# contigs (>= 0 bp)	5125	4696	4360	4466
# contigs (>= 1000 bp)	418	390	357	392
# contigs (>= 5000 bp)	27	28	30	33
# contigs (>= 10000 bp)	11	8	11	10
# contigs (>= 25000 bp)	3	4	4	5
# contigs (>= 50000 bp)	1	2	1	1
Total length (>= 0 bp)	3233154	3081960	2920711	3154396
Total length (>= 1000 bp)	1119353	1086350	1027274	1144212
Total length (>= 5000 bp)	400553	416267	415499	482429
Total length (>= 10000 bp)	298340	286198	293409	324671
Total length (>= 25000 bp)	177666	220055	190746	249756
Total length (>= 50000 bp)	87312	138582	87312	87312
# contigs	1584	1475	1398	1675
Largest contig	87312	87312	87312	87312
Total length	1893693	1804462	1714995	1990311
GC (%)	41.83	41.76	41.77	41.79
N50	1279	1384	1387	1255
N75	718	717	712	699
L50	265	231	210	260
L75	779	710	668	819
# N's per 100 kbp	86.08	71.49	57.73	32.16