Tutorial: Spatial comparison of read qualities in a HISEQ/MISEQ run

Umer Zeeshan Ijaz

Problem Statement: Given a HISEQ/MISEQ run that comprises pooled samples, how do you visualize the read qualities on flow-cell tiles when you are given paired-end reads for each sample in FASTQ format?

Step 1: Get average quality score + flow cell information and store as dat files i.e redirect output > *_R2.dat in the following example (excluding "| head -10")

```
perl -ne 'chomp;$_=~m/@(.*?):(.*?):(.*?):(.*?):(.*?):(.*?):(.*?)\s/;print
$4.",",$5.",".$6.",".$7;<>;<>;$_=<>;chomp;map{$s+=ord($_)-33}
split("");print",".($s/length($_))."\n";$s=0;' < FILE.fastq > FILE.dat
```

This one-liner gives:

Column 1: flow cell lane
Column 2: tile number within the flowcell lane
Column 3: x-coordinate of the cluster within the tile
Column 4: y-coordinate of the cluster within the tile
Column 5: Average Quality Score

For example,

```
[uzi@quince-srv2 ~]$ perl -ne 'chomp;$_=~m/@(.*?):(.*?):(.*?):(.*?):(.*?):(.*?):(.*?)\s/;print $4.",",
$5.",".$6.",".$7;<>;<>;$_=<>;chomp;map{$s+=ord($_)-33}split("");print",".($s/length($_))."\n";$s=0;' <
5_TAAGGCGA-TAGATCGC_L001_R2_001.fastq | head -10
1,1101,9593,26322,26.4814814814815
1,1101,7952,34132,27.6470588235294
1,1101,8189,100645,26.3333333333333
1,1102,10633,88138,25.8048780487805
1,1103,5153,25465,36.7029702970297
1,1103,9533,58452,27.4752475247525
1,1103,21248,64336,28.1842105263158
1,1104,19815,7437,23.0769230769231
1,1105,14548,7890,30.4653465346535
1,1105,8795,73701,21.3076923076923
```

Step 2: Collate all the forward reads dat files and all the reverse reads dat files to generate R1.dat and R2.dat:

```
[uzi@quince-srv2 ~]$ cat 1_R1.dat 2_R1.dat 3_R1.dat 4_R1.dat 5_R1.dat 6_R1.dat
7_R1.dat … > R1.dat
[uzi@quince-srv2 ~]$ cat 1_R2.dat 2_R2.dat 3_R2.dat 4_R2.dat 5_R2.dat 6_R2.dat
7_R2.dat … > R2.dat
```

Step 3: Find unique flow-cell tile information for both R1 and R2:

```
[uzi@quince-srv2 ~]$ awk –F”,” ‘{print $2}’ R1.dat | sort | uniq > R1.uniq
[uzi@quince-srv2 ~]$ awk –F”,” ‘{print $2}’ R2.dat | sort | uniq > R2.uniq
```

Step 4: Extract data for flow-cell tiles in a separate folder and save them as space-separated (required by gnuplot) text file. This will create txt files 1101.txt, 1102.txt, and so on.

```
[uzi@quince-srv2 ~]$ for i in `cat R1.uniq`; do awk -F"," -v pattern=$i
'$2==pattern{gsub(","," ",$0);print $0}’ R1.dat > $i.txt; done

[uzi@quince-srv2 ~]$ for i in `cat R2.uniq`; do awk -F"," -v pattern=$i
'$2==pattern{gsub(","," ",$0);print $0}’ R2.dat > $i.txt; done
```

Step 5: make a file gnuscript.txt with the following contents. This script when called generates a scatterplot as *.txt.ps file

```
PATH=$0
DATAFILE=$1
set palette
plot PATH."/".DATAFILE using 3:4:5 with dots
palette notitle
set xlabel "X"
set ylabel "Y"
set term postscript color
set output DATAFILE.'.ps'
replot
set term x11
```

Step 6: Call the above script and use imagemagick to convert the *.txt.ps into *.txt.jpg

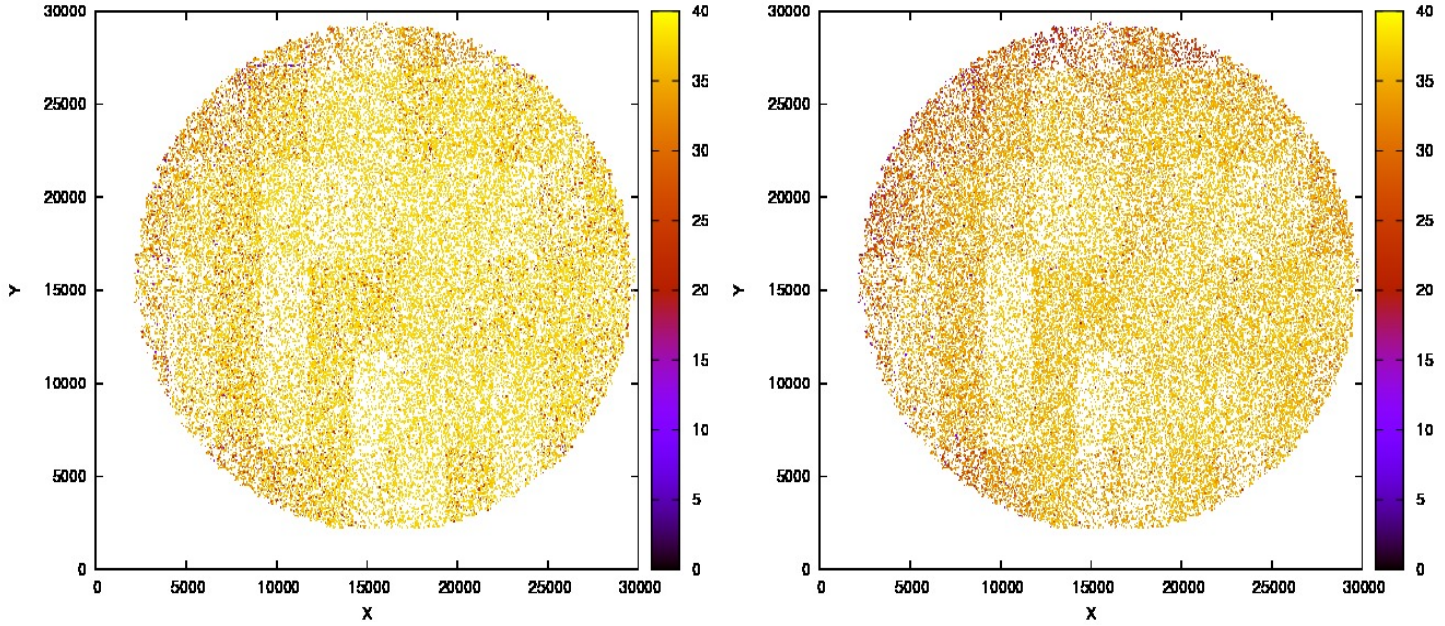
```
[uzi@quince-srv2 ~]$ for i in `ls ????.txt`; do echo processing $i
now;echo "call \"gnuscript.txt\" \"'.'\" \"'$i'\"" | gnuplot; convert
$i.ps $i.jpg; done
```

Step 7: To get the total reads corresponding to each tile, use

```
[uzi@quince-srv2 ~]$ awk –F”,” ‘{print $2}’ R1.dat | sort | uniq -c
[uzi@quince-srv2 ~]$ awk –F”,” ‘{print $2}’ R2.dat | sort | uniq -c
```

Here are the images for the flow-cell tiles 1101 and 1102. Left image corresponds to the forward reads and right image corresponds to the reverse reads. Lighter dots are higher quality reads.

1101



1102