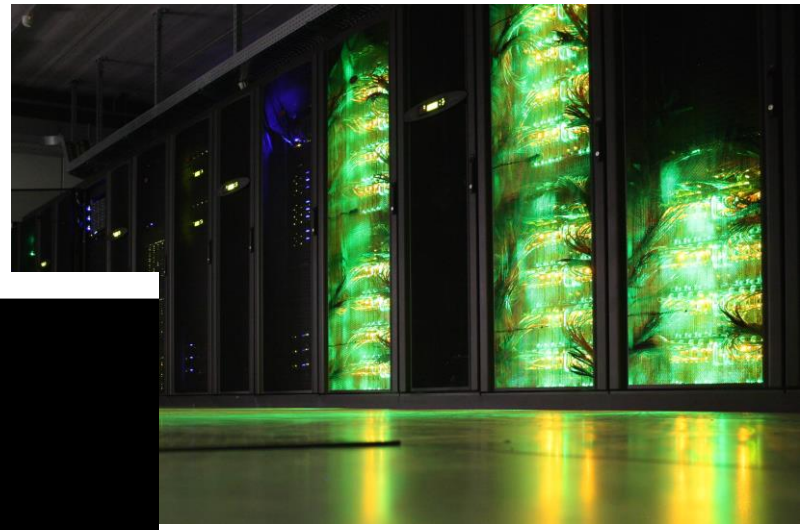
HPC-getorganelle

```
#!/bin/bash

#PBS -S /bin/bash
#PBS -l walltime=06:00:00
#PBS -N JOBNAME
#PBS -o stdout.$PBS_JOBID
#PBS -e stderr.$PBS_JOBID
#PBS -l nodes=1:ppn=all:gpus=2
#PBS -m abe
#PBS -M pieter.asselman@ugent.be

########
##INFO##
########

#DATArun on 2GPU/ 11Gbases - ~3h

##BASECALL##
guppy_basecaller -i /scratch/gent/vo/000/gvo00058/vs

##DEMULTIPLEX##
#guppy_barcoder --i /scratch/gent/vo/000/gvo00058/vs

##BASECALL and DEMULTIPLEX##
#guppy_basecaller -i /scratch/gent/vo/000/gvo00058/v

```

```
vsc43352@gligar04: ~
Using username "vsc43352".
Authenticating with public key "rsa-key-HPC"
Passphrase for key "rsa-key-HPC":
Last login: Tue Aug  3 15:14:06 2021 from gligarha01.gastly.os

STEVIN HPC-UGent infrastructure status on Wed, 04 Aug 2021 07:25:04

   cluster - full - free -  part - total - running - queued
             nodes  nodes   free   nodes    jobs      jobs
-----------------------------------------------------------------------
    swalot    90      0     20     125     N/A       N/A
    skitty    19      0     52      72     N/A       N/A
   victini     8      1     86      96     N/A       N/A
    joltik     9      0      0      10     N/A       N/A
    kirlia    15      0      1      16     N/A       N/A
     doduo   120      0      5     128     N/A       N/A

For a full view of the current loads and queues see:
https://hpc.ugent.be/clusterstate/
Updates on current system status and planned maintenance can be found on
https://www.ugent.be/hpc/en/infrastructure/status

We switched to new job command wrappers for all HPC-UGent Tier-2 clusters on Wed
 June 9th 2021 at 17:22.
This switch should be transparant: you don't need to change your workflow or job
 scripts.

If you notice any problems, or if any of these changes affect your work, please
contact hpc@ugent.be .
vsc43352@gligar04:~$
```



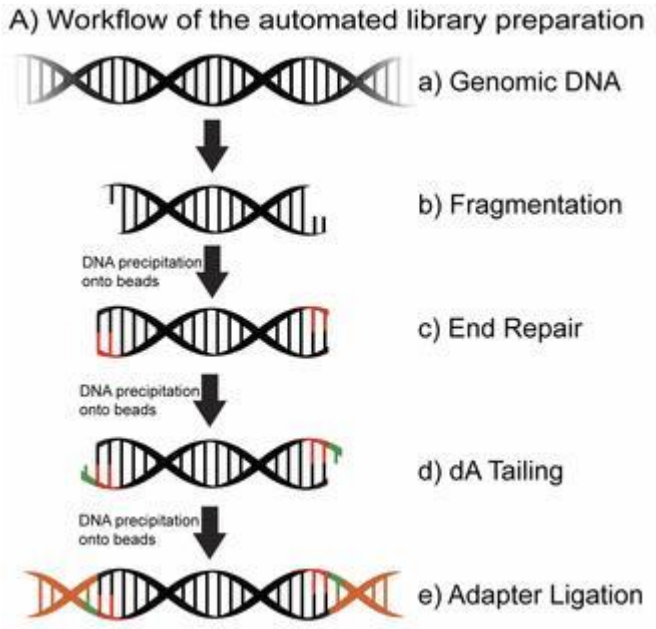
https://ugent.be/hpc
hpc@ugent.be

Pipeline: Cp assemblies

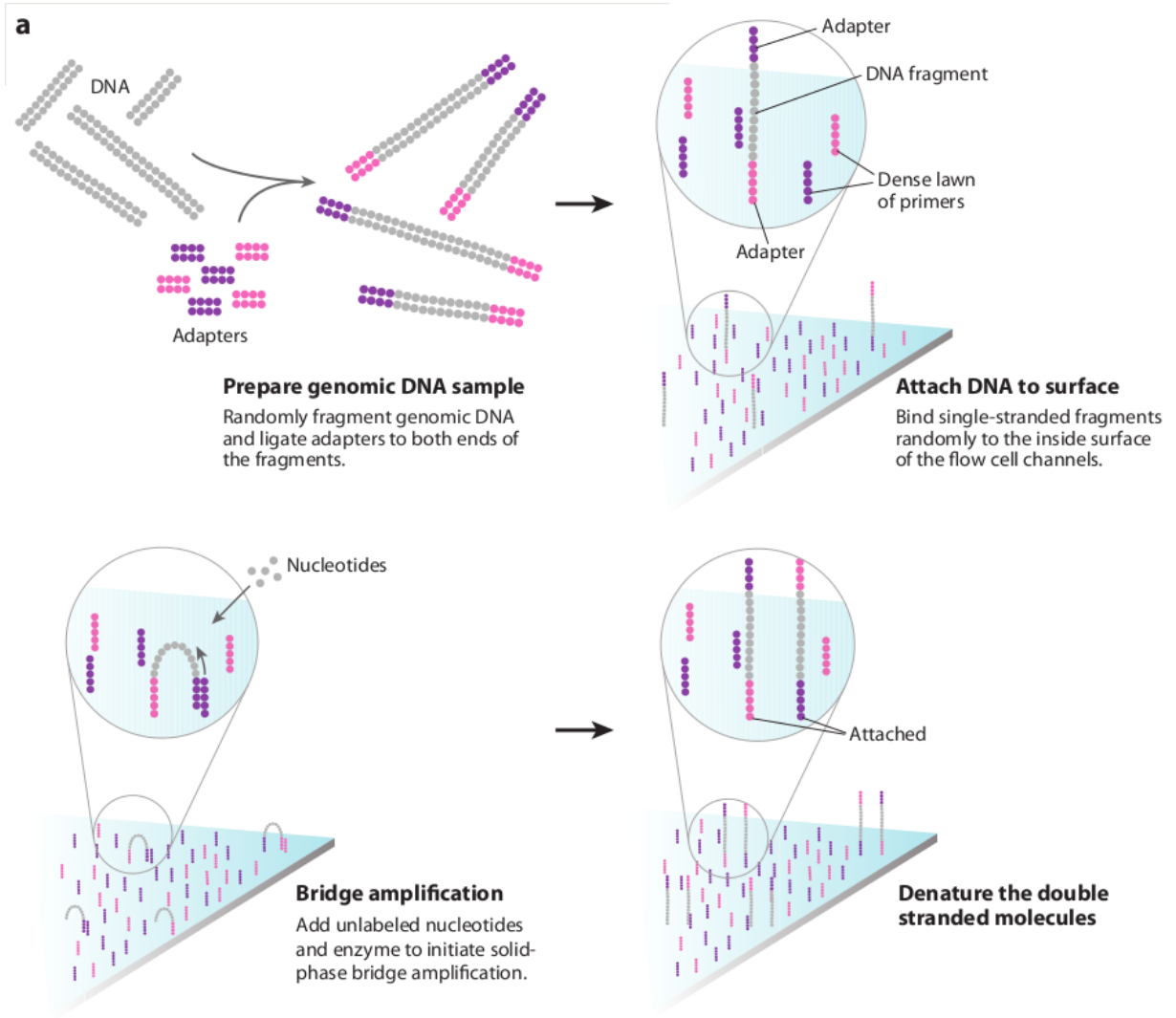
I. Lib prep
II. Sequencing - Demultiplexing
III. QC
  a) Sequencing Report
    - ✓ QC10: 1-10 error (90% accurate)
    - ✓ QC20: 1-100 error (99% accurate)
    - ✓ QC30: 1-1000 error (99,9% accurate)
  b) Fastqc: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/
    - ✓ Quick overview of all reads
    - ✓ Summary graphs
    - ✓ HTML-based report
    - ✓ Multiqc! Overview of multiple HTMP-reports from different analyses
  c) Trim
    - ✓ Trimmomatic: http://www.usadellab.org/cms/?page=trimmomatic
    - ✓ Cutadapt: https://cutadapt.readthedocs.io/en/stable/

IV. Cp – nr Assembly: Getorganelle
    - ✓ Toolkit for assembly of organelle genome: https://github.com/Kinggerm/GetOrganelle

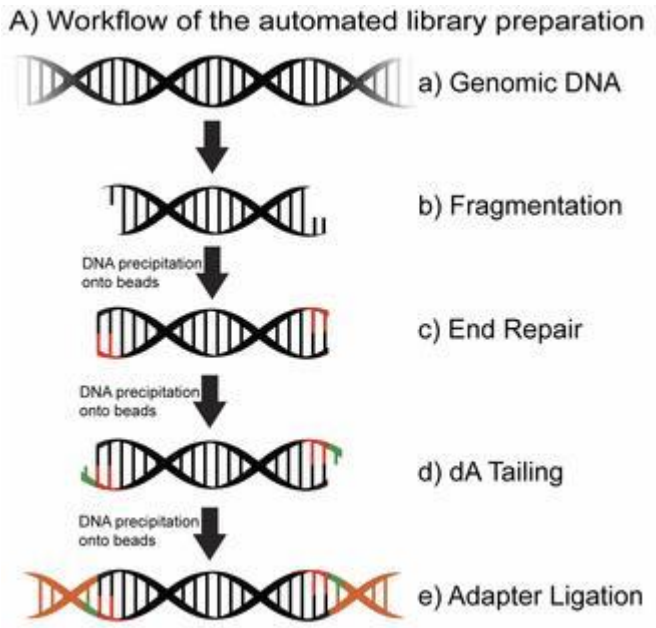
Pipeline: Library prep & Illumina Sequencing



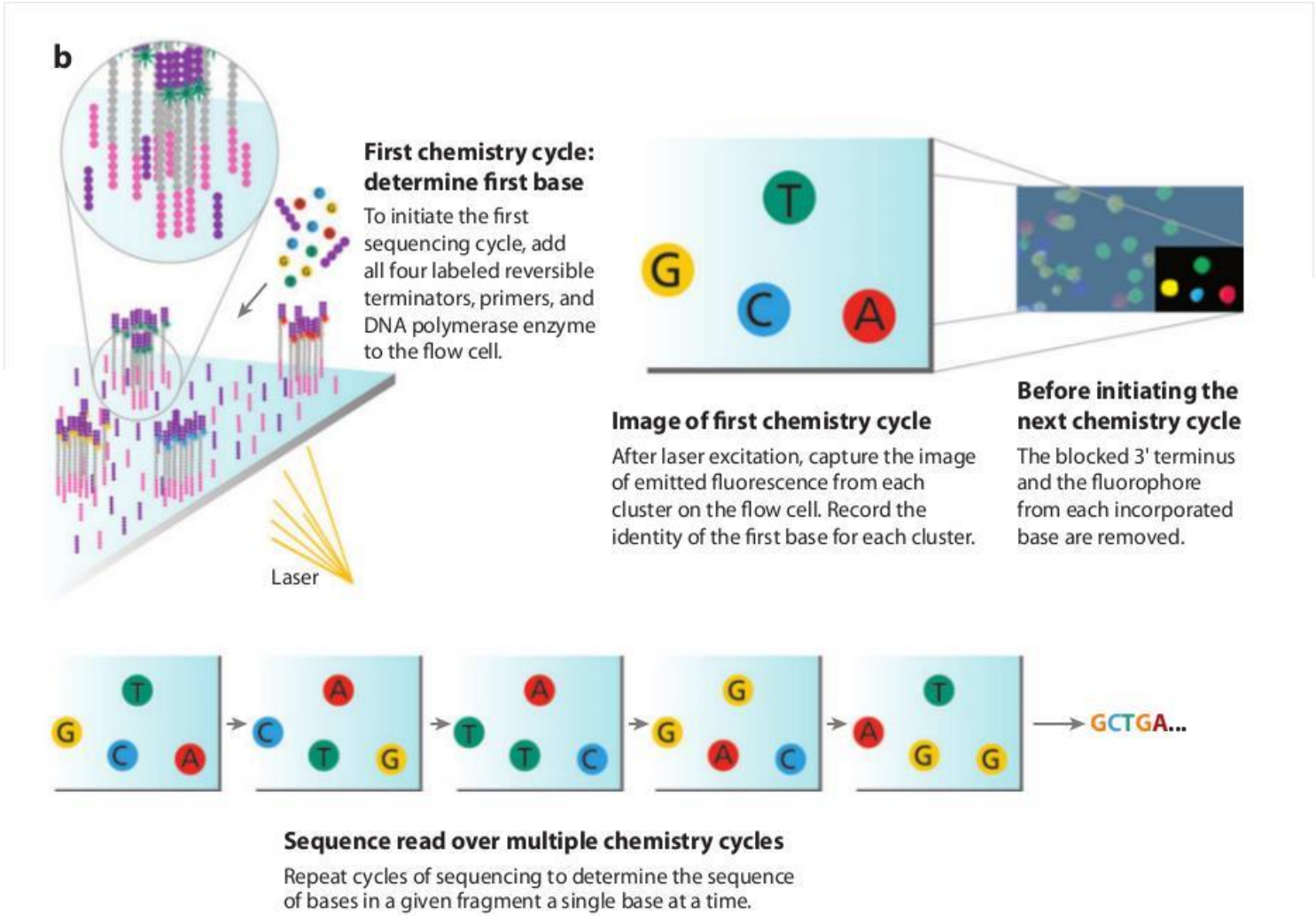
*Borgstrom E et.al2011 Large Scale Library Generation for High Throughput Sequencing. PLoS ONE 6:e19119. doi:10.1371/journal.pone.0019119 *



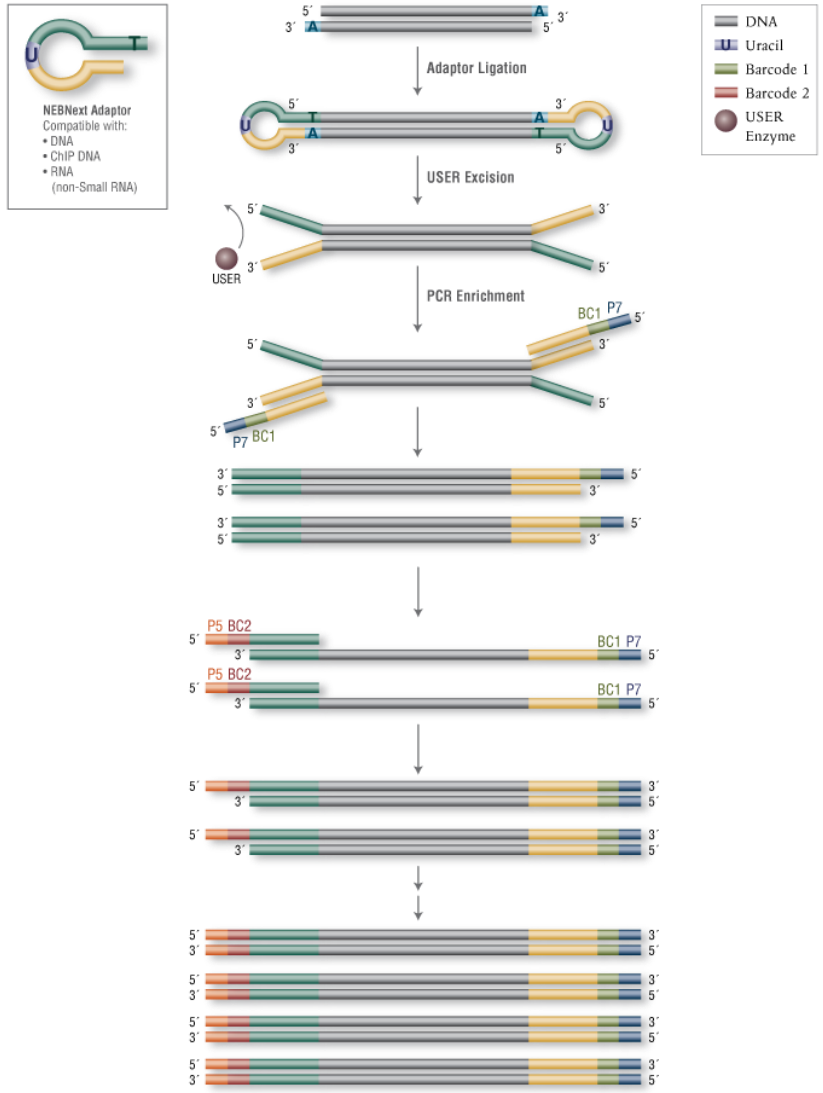
Pipeline: Library prep & Illumina Sequencing



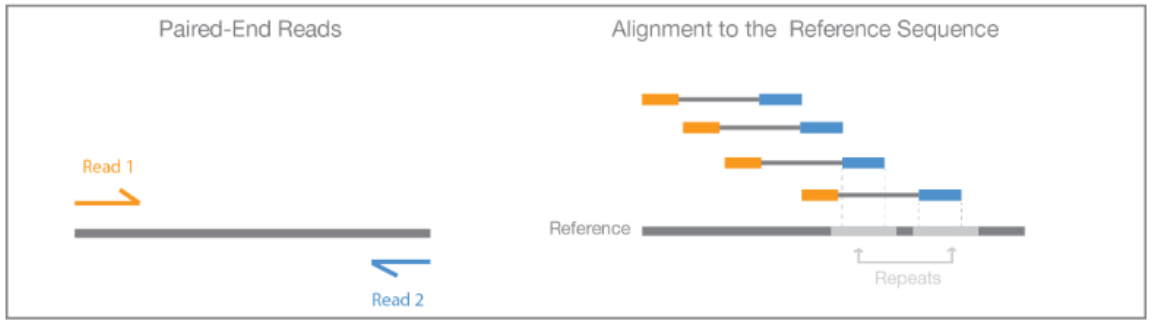
*Borgstrom E et.al2011 Large Scale Library Generation for High Throughput Sequencing. PLoS ONE 6:e19119. doi:10.1371/journal.pone.0019119 *



Pipeline: Library prep



Dual index barcoding P5/BC2 – BC1/P7

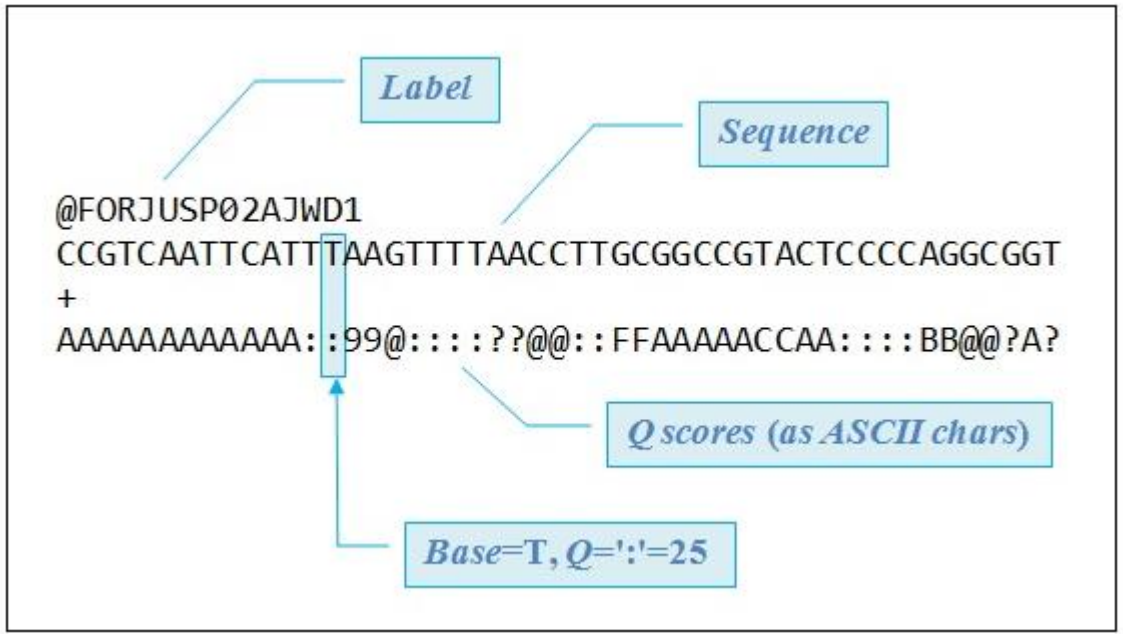


Pipeline: Data format

fastq

```
@de4d97b8-c1c5-4d48-83dd-cf35c8b1b262 runid=86431727e3e8469fcfb154f8fc6e278e2ca5217d read=4 ch=443
start_time=2017-10-30T10:11:28Z
TGATGCTTCGTTCAGTTGCGTATTGCTCTGCCAGATACGCCCGCGCAGATGATTACGGCTCACCTGCTGGCTGCCAGCTGCTCAGGTCCACCTGCGCCGAC
CGGTGCCGTGTTCCGGGTAGCCTGTTTCATCAGTAGTTTCAGTGTATGACGTGGGCGTTTATGCGAAACTGGTCTGTGCTGTCAGCGTCATCCTGCGCATC
CGGATGTAAACGGGCGCGGCGGCGGGTTACCCATCACCACCGGGGCCCGGATGCGGGTGGTTTGCTATAATGGGTGATGATGTCTTTTCCGTCACCCAGCC
ACCGTTACGTTGTATCTAGACAGCAGGCGGACTACGACGGTTCTATCACCCTTTGGGGTGGTTTCCTGTACGAACGGAAGGTAAAGCGCAGACGGTCGATG
TTTGCAGATTGTAATGGGTGCGGGTGATCGGCGTGTCGTATTTCGCTTGTACTTGGCGCGCACCGTCTCAGGCCAGCGGAGTTCAAATCCCTCCAGCGGAT
CTATCCTCTCACCAGCCCGGAACACCTGTGGTACGGATATATTTAGTATTCCCCTCAGTATCCAGCACCAGCGTACTGTGTTCAGCCGCGCTTTTCCGTAT
CCACCGGACCTTCAATCCAGCCTTCGCTGGTGGCATCGATCCACACTCCAGCAGCCGTGGACTTGGAGTTGTCAGCCAGCTGCGCAGGGGTATGCCCTTAC
TGCTTCCTTTACCCATTTCTCCGCTCCACCATTAGCCGAAACCGCAGGCCAGTTCCACATAAAACATTTTGCATCTTGACCGAAATCACCTGGCAACCACC
GTCCCCTTCGTCTGCCGTGCCAGTCTCGCAGAGCCGCGTATTTGCGCATTCCCCATGCAGACAGGCCAGAACATTTGCCTGGGCAGCCATTATCAGTGGGG
AGAAATAGGTGTTCTGCTTACCGTTATCCGTTGTCTGTATACGGGGTTCGGCACCGGTGCCAGCATCCGCCACTACCGGGCGCCATACGGCACCGGAAAAC
AGGATGCCAGTCATACCACCGGCCCCAATGAAAACTGCCCCCATGCTGCAAGGGTGGCTCCGGCGAGTAAAGAATGATCCGGCGTGGCGGCAGCCCAGGAC
AATCTGGAATCGCCACCGGCTGACTGGCGACTCTGGGAACAATGAATTACAGCGCCATCAGGCCAGAGTCTCATGCATAACACCGCCGTTAACCGGACATT
TGGCGTCCGGCAATCCGTACCTGATACCAGCCAATCGCTCAGTTTCACAGCCGACGCCAGGGCTATGTGGCCAGTGCCCGGATGGCTTCAGCCCA
+
##$$$(')-,7A83<<D+5.)+,731++)),('))-$./6<8+2*3,:::744*322(+(1*210265.7,,)5,'+::8789*/-,+/+17026289<::
4'2?+0<<>;>4/,)6*2$$$'.*56/.3:<%(4',1;@3(C@:3*+.17-,%-05162($(&'%+))).3852-),*,/-19-32=8?<=AC?45264?<
4=?>:*2(3,2-6AAAA<</-50*4269%/(-)-..<@,14/-?36:'*??,+,&.:2>>230((,/30+6.3/-,-.1387<>8ED>0:169-298,+.3
(,('-(#&*,0--++*'&--.'(((*+/*4188/3:898=.@0,)8:5>C79D2+-.(0/?4A(*'$$%&,+%&57)6*9,.%')+/2E8A>@/CB96887
8C=7<120-*$(',-4:.((,&*&'(.02.83+3*++&/(5C0+&'&%#%%&'%%*.,)'**296--,,+7:8?*+&3-*++2*&&(*:=-+:.%.43:,0
10798*((,0(('*$+))/:352*/./0.+,+-4,;+0'.+6C6(//4<D5CD.*53'6&,11669%6+02334;8,*-%5>)))*(+/,+**(&(118;:
9839@4C25++-%3,1?(%-/1-1.-./.03,.4),0*-;>((1.1+2'*&,.&,.(7-A/677+)%)/5*//0&.-'%',,+%'&45>6=45';D7)2+-
))(/21))8((-)($+88:32--22;C-39(3-+)+.(+,);E256754-&17C0-*.)/+20+---A2,/2,('&(*15/&1052<9<90-)*(&(+*55
9=2?:+09:;=C;66:/3/4*$/<<78$)&'3'(,(***%,-(),+-(,--)+*&&"%&+65/409(%-,50/1?105A:;2*(/+:433:024'87/+9;
--(1A-,3-</5?61*))''-0+6501=29:D223.*-,-':(346-7.,&&#&%//@<@,*,,'54(,/+)+&'..+**5&*.,($'*5-(.;/6/3439
&87>86.*&)1*&,-='&,'//+-.*0-:857/&*(,4BFD><:65,111/-8314<938,-/6-)&*8B(')116:CCB8=?,>0@2'62+-94*$4620
<40;;+9$%'327:'5$,&)#($)&-*,..=884,889996
+)/-+')'),5<A==8'<+0>*((,,:*3%&%.&,16;<?3
@4ea9b632-cbca-455d-9061-88117b879114 run
start_time=2017-10-30T10:11:30Z
TGTTACTTTAGTTCAGACCATTGCACCATCAGATTATGTTT
ACACGTTTTGCCCGTACTCGGTCACGAACAAATCCTGATTA
CCCTTATTGAGGGTAAAGACGCCCAGAATTGAAAACGCGGC
```

$
ASCII : 36
P: 0,5 (50% chance the basecall is wrong)
Q: 3



ASCII_BASE=33 Illumina, Ion Torrent, PacBio and Sanger

Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII
0	1.00000	33 !	11	0.07943	44 ,	22	0.00631	55 7	33	0.00050	66 B
1	0.79433	34 "	12	0.06310	45 -	23	0.00501	56 8	34	0.00040	67 C
2	0.63096	35 #	13	0.05012	46 .	24	0.00398	57 9	35	0.00032	68 D
3	0.50119	36 $	14	0.03981	47 /	25	0.00316	58 :	36	0.00025	69 E
4	0.39811	37 %	15	0.03162	48 0	26	0.00251	59 ;	37	0.00020	70 F
5	0.31623	38 &	16	0.02512	49 1	27	0.00200	60 <	38	0.00016	71 G
6	0.25119	39 '	17	0.01995	50 2	28	0.00158	61 =	39	0.00013	72 H
7	0.19953	40 (	18	0.01585	51 3	29	0.00126	62 >	40	0.00010	73 I
8	0.15849	41)	19	0.01259	52 4	30	0.00100	63 ?	41	0.00008	74 J
9	0.12589	42 *	20	0.01000	53 5	31	0.00079	64 @	42	0.00006	75 K
10	0.10000	43 +	21	0.00794	54 6	32	0.00063	65 A			

ASCII_BASE=64 Old Illumina

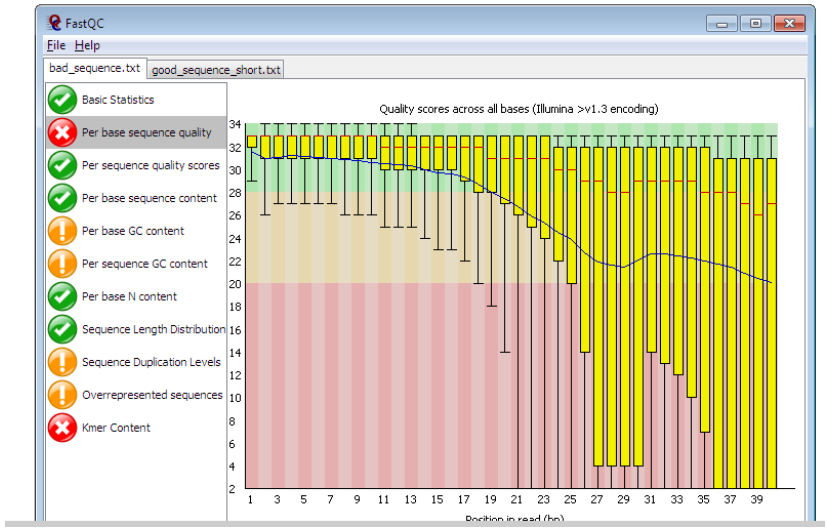
Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII
0	1.00000	64 @	11	0.07943	75 K	22	0.00631	86 V	33	0.00050	97 a
1	0.79433	65 A	12	0.06310	76 L	23	0.00501	87 W	34	0.00040	98 b
2	0.63096	66 B	13	0.05012	77 M	24	0.00398	88 X	35	0.00032	99 c
3	0.50119	67 C	14	0.03981	78 N	25	0.00316	89 Y	36	0.00025	100 d
4	0.39811	68 D	15	0.03162	79 O	26	0.00251	90 Z	37	0.00020	101 e
5	0.31623	69 E	16	0.02512	80 P	27	0.00200	91 [	38	0.00016	102 f
6	0.25119	70 F	17	0.01995	81 Q	28	0.00158	92 \	39	0.00013	103 g
7	0.19953	71 G	18	0.01585	82 R	29	0.00126	93]	40	0.00010	104 h
8	0.15849	72 H	19	0.01259	83 S	30	0.00100	94 ^	41	0.00008	105 i
9	0.12589	73 I	20	0.01000	84 T	31	0.00079	95 _	42	0.00006	106 j
10	0.10000	74 J	21	0.00794	85 U	32	0.00063	96 `			

Pipeline: Cp assemblies

I. Lib prep
II. Sequencing - Demultiplexing
III. QC
  a) Sequencing Report
    - ✓ QC10: 1-10 error (90% accurate)
    - ✓ QC20: 1-100 error (99% accurate)
    - ✓ QC30: 1-1000 error (99,9% accurate)
  b) Fastqc: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/
    - ✓ Quick overview of all reads
    - ✓ Summary graphs
    - ✓ HTML-based report
    - ✓ Multiqc! Overview of multiple HTMP-reports from different analyses
  c) Trim
    - ✓ Trimmomatic: http://www.usadellab.org/cms/?page=trimmomatic
    - ✓ Cutadapt: https://cutadapt.readthedocs.io/en/stable/

IV. Hybpiper
  - ✓ Toolkit designed for targeted sequence capture: https://github.com/mossmatters/HybPiper

Quality controle – Fastqc



https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Get QC info through fastqc:

- Download test data and scripts

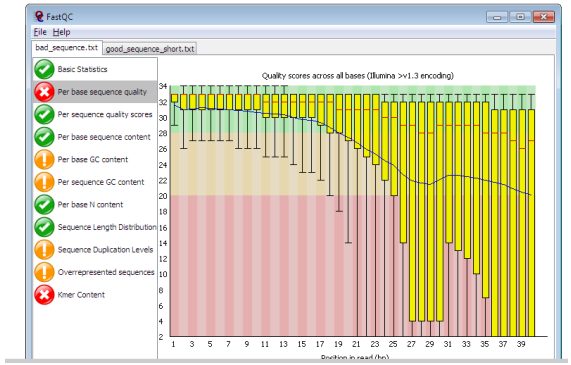
```
$ git clone https://github.com/MycoMatics/intro-cpgenomes.git
$ cd intro-cpgenomes
$ kinit yourusername@UGENT.be
$ cp /UGent/yourusername/shares/data_hub_cemofe/Courses/illumina-QC/Illumina_data/OX1.tar.gz
```

- Take a look at the fastq.sh script

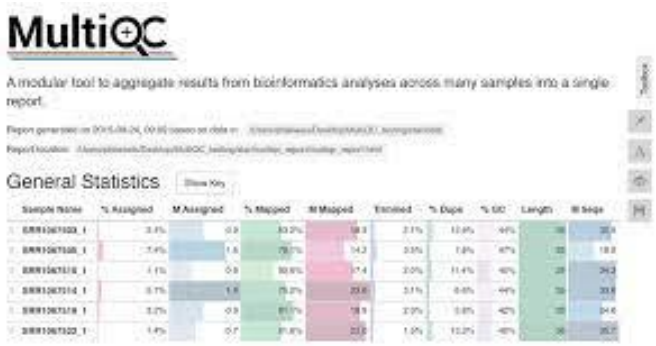
Quality controle – Fastqc

```
#!/usr/bin/bash
# Line 1 is a she-bang that indicates that this is a Bash script.

#PBS -N fastqc.$PBS_JOBID               #  Line 3-7 inform the scheduler about the resources required by this job:
#PBS -l nodes=1:ppn=all                             # singe node (nodes=1) all core (ppn=all)
#PBS -o stdout.$PBS_JOBID                   # redirect sterr stdout to separate files
#PBS -e stderr.$PBS_JOBID
#PBS -l walltime=01:00:00                           # run for at most 2 minutes (walltime=00:02:00 max is 72hours)
#PBS -m abe                                         # send mail when job (a)bort (b)egin (e)nd
#PBS -M <youremailaddresshere>                      # specify your email address here


#Request software

ml FastQC/0.11.9-Java-11
ml MultiQC/1.14-foss-2022a

#Stage in data: Go to your current working directory and make sure both your data and scrit is there

cd $PBS_O_WORKDIR

#Make output directories
mkdir fastqc-reports

#Software commands
fastqc ./OX0001/*.fq.gz -o ./fastqc-reports &&  # run fastqc on all present fq.gz files
multiqc .              # run multiqc on all fastqc output && in previous command prevents from starting multiqc early

```

Quality controle – Fastqc + MultiQC



https://www.bioinformatics.babraham.ac.uk/projects/fastqc/



https://multiqc.info/

Get QC info through fastqc:

- Download test data and scripts

```
$ git clone https://github.com/MycoMatics/intro-cpgenomes.git
$ cd intro-cpgenomes
$ kinit yourusername@UGENT.be
$ cp /UGent/yourusername/shares/data_hub_cemofe/Courses/illumina-QC/Illumina_data/OX1.tar.gz
```

- Run the fastq.sh script

```
$ qsub fastq.sh
```

Quality controle – Fastqc

- **Basic statistics**
- **Per base sequence quality**
  => Range of quality values across all bases in each position
- **Per sequence quality scores**
  => Quality distribution over all sequences
- **Per base sequence content**
  => IF GC content ~50% => 25% chance of either A,T,G or C
  => Bias at start: PCR duplicates, adapter/primer contamination
- **Per sequence GC content**
  => GC distribution over al sequences => contamination indication other organisms
- **Per base N content**
- **Sequence length distribution**
- **Sequence duplication levels**
  => Most sequences occur once (WGS)
    - Low level duplication: high level coverage througout whole genome
    - High level duplication: enrichment bias
- **Overrepresented sequences**
- **Adapter content**

Pipeline: Cp assemblies

I. Lib prep
II. Sequencing - Demultiplexing
III. QC
  a) Sequencing Report
    - ✓ QC10: 1-10 error (90% accurate)
    - ✓ QC20: 1-100 error (99% accurate)
    - ✓ QC30: 1-1000 error (99,9% accurate)
  b) Fastqc: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/
    - ✓ Quick overview of all reads
    - ✓ Summary graphs
    - ✓ HTML-based report
    - ✓ Multiqc! Overview of multiple HTMP-reports from different analyses
  c) Trim
    - ✓ Trimmomatic: http://www.usadellab.org/cms/?page=trimmomatic
    - ✓ Cutadapt: https://cutadapt.readthedocs.io/en/stable/

IV. Cp – nr Assembly: Getorganelle
  - ✓ Toolkit for assemblie of organelle genome: https://github.com/Kinggerm/GetOrganelle

Quality control: Trimming sequences

Simple Mode

Figure modified from Bolger et al. 2014 (see link below). Caption from Bolger et al. 2014.

The dark blue and red are the initial raw sequence that goes into Trimmomatic

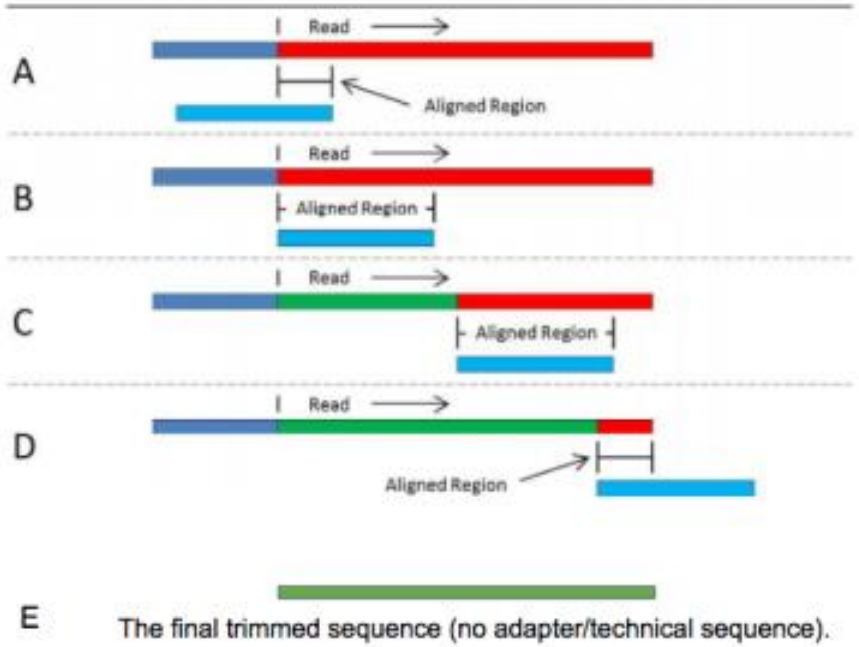


Fig. 1. Putative sequence alignments as tested in Simple Mode. The alignment process begins with partial overlap at the 5' end of the read (A), increasing to a full length 5' overlap (B), followed by full overlaps at all positions (C) and finishes with partial overlap at the 3' end of the read (D). Note that the upstream 'adapter' sequence is for illustration only, and is not part of the read or the aligned region.

Here the program "missed" removing the little bits that aligned well in A and D.

Quality control: Trimming sequences

Figure modified from Bolger et al. 2014 (see link below). Caption from Bolger et al. 2014.

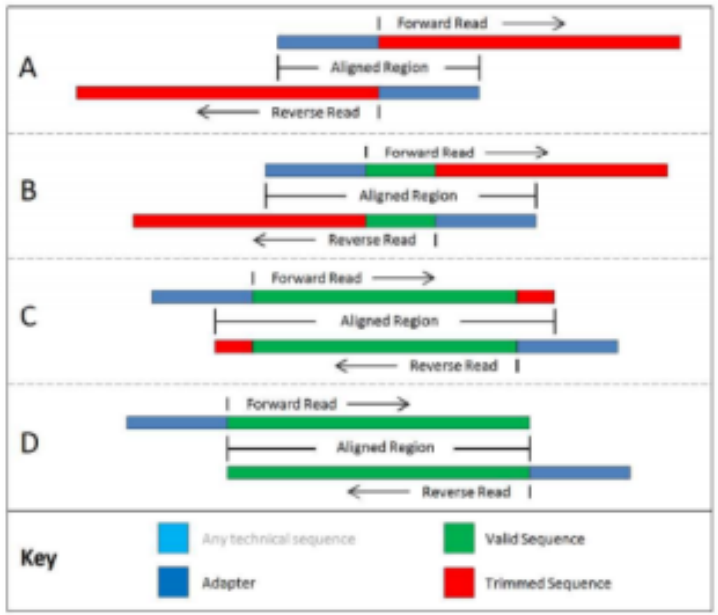


Fig. 2. Putative sequence alignments as tested in Palindrome Mode. The alignment process begins with the adapters completely overlapping the reads (A) testing for immediate 'read-though', then proceeds by checking for later overlap (B), including partial adapter read-though (C), finishing when the overlap indicates no read-through into the adapters (D).

https://academic.oup.com/bioinformatics/article/30/15/2114/2390096/Trimmomatic-a-flexible-trimmer-for-Illumina

Quality controle – Trimmomatic

http://www.usadellab.org/cms/?page=trimmomatic

Use trimmomatic to clip adapter sequences from your reads:

- Data structure folder intro-cp genomes
  - `adapters`
    - `alladapterstrimmomatic.fa`
  - `OX0001`
    - `_1.fq.gz AND _2.fq.gz`
  - `trimmomatic_v04.sh`
- take a look the trimomatic_v04.sh script

```bash
#!/bin/bash
#PBS -N trimmomatic
#PBS -l nodes=1:ppn=8
#PBS -o stdout.$PBS_JOBID
#PBS -e stderr.$PBS_JOBID
#PBS -l walltime=0:30:00
#PBS -m abe
#PBS -M pieter.asselman@ugent.be

# Data Paths
ILLUMINA_RAWDATA=/yourpathto/OX0001 #take notice of the file extension names different options '.fq.gz' '.fastq.gz'
ILLUMINA_ADAPTERS=/yourpathto/adapters

# make trimmed data directory
mkdir $ILLUMINA_RAWDATA/trimmed-data

# Create sampleslist to itterate, serves as input for itteration process in trimmomatic
for file  in $ILLUMINA_RAWDATA/*_1.fq.gz
 do
  sample_name=$(basename "$file" _1.fq.gz) # extract sample name
  echo "${sample_name}" >> $ILLUMINA_RAWDATA/samples.txt
  done &&


# Load modules
module load Trimmomatic/0.39-Java-11

### Run trimmomatic SE OR PE reads #-out what you don't need!

while read p
do

### ACTIVATE FOR SE data
#java -jar $EBROOTTRIMMOMATIC/trimmomatic-0.39.jar SE -phred33 \
#-trimlog $TRIMMOMATIC_OUT/$p".log" \
#$ILLUMINA_RAWDATA/$p"_R1.fastq.gz" \
#$TRIMMOMATIC_OUT/$p"_R1_trimmed.fastq" \
#ILLUMINACLIP:$ILLUMINA_ADAPERS/alladapterstrimmomatic.fa:2:30:10:1:TRUE SLIDINGWINDOW:5:20

### ACTIVATE FOR PE data
java -jar $EBROOTTRIMMOMATIC/trimmomatic-0.39.jar PE -phred33 -trimlog $ILLUMINA_RAWDATA/trimmed-data/$p".log" \
        $ILLUMINA_RAWDATA/$p"_1.fq.gz" \
        $ILLUMINA_RAWDATA/$p"_2.fq.gz" \
        $ILLUMINA_RAWDATA/trimmed-data/$p"_1_trimmed_paired.fastq" \
        $ILLUMINA_RAWDATA/trimmed-data/$p"_1_trimmed_unpaired.fastq" \
        $ILLUMINA_RAWDATA/trimmed-data/$p"_2_trimmed_paired.fastq" \
        $ILLUMINA_RAWDATA/trimmed-data/$p"_2_trimmed_unpaired.fastq" \
        ILLUMINACLIP:$ILLUMINA_ADAPTERS/alladapterstrimmomatic.fa:2:30:10:1:TRUE SLIDINGWINDOW:5:20

done < $ILLUMINA_RAWDATA/samples.txt
```

Quality controle – Trimmomatic

http://www.usadellab.org/cms/?page=trimmomatic

Use trimmomatic to clip adaptersequences from your reads:

- Data structure folder intro-cpgenomes
  - `adapters`
    - `alladapterstrimmomatic.fa`
  - `OX0001`
    - `_1.fq.gz AND _2.fq.gz`
  - `trimmomatic_v04.sh`

- Run the trimomatic_v04.sh script (make necessary adjustments to fit to your environment)

```
$ qsub trimmomatic_v04.sh
```

Quality control: Trimming sequences

```
module load Trimmomatic/0.39-Java-11
java -jar $EBROOTTRIMMOMATIC/trimmomatic-0.39.jar PE \
    input_forward.fq.gz \
    input_reverse.fq.gz \
    output_forward_paired.fq.gz \
    output_forward_unpaired.fq.gz \
    output_reverse_paired.fq.gz \
    output_reverse_unpaired.fq.gz \
    ILLUMINACLIP:<pathto>\adapters.fa:2:30:10:2:keepBothReads LEADING:3 TRAILING:3 MINLEN:36 SLIDINGWINDOW:4:15
```

This will perform the following:

- **Remove adapters** (ILLUMINACLIP:TruSeq3-PE.fa:2:30:10); adapter sequences can be found here
  - Allow maximally 2mismatches.
  - extended seeds and clipped if score of 30 (about 50 bases) for PE (SE=10 ~17 bases)
  - Remove leading low quality or N bases (below quality 3) (LEADING:3)
- **Remove trailing low quality or N bases** (below quality 3) (TRAILING:3)
- **Scan the read with a 4-base** wide sliding window, cutting when the average quality per base drops below 15 (SLIDINGWINDOW:4:15)
- **Drop reads** below the 36 bases long (MINLEN:36)

Quality control: Trimming sequences

OUTPUT TRIMMOMATIC

Fastq files

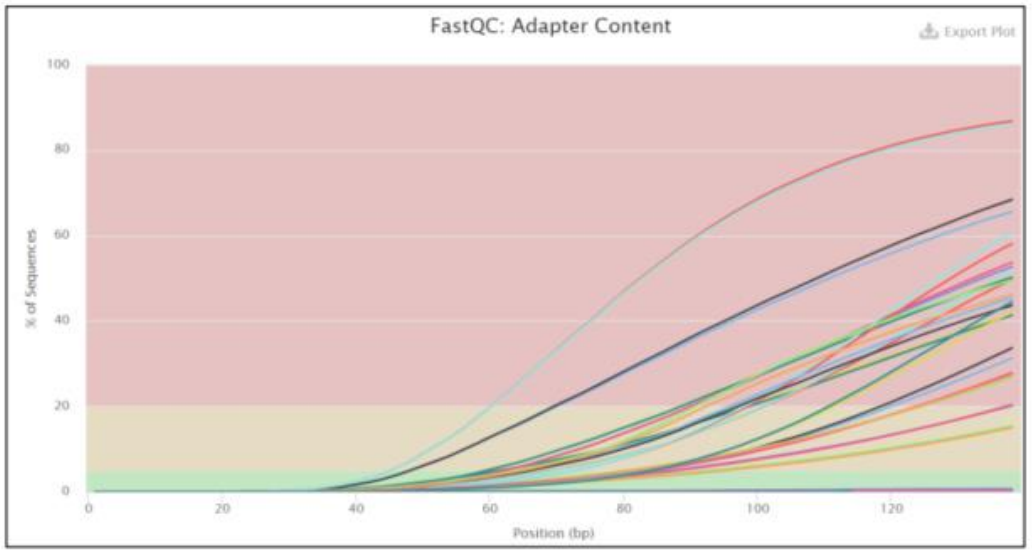
- ✓ output_forward_paired.fq.gz
- ✓ output_**forward_unpaired**.fq.gz
- ✓ output_reverse_paired.fq.gz
- ✓ output_**reverse_unpaired**.fq.gz

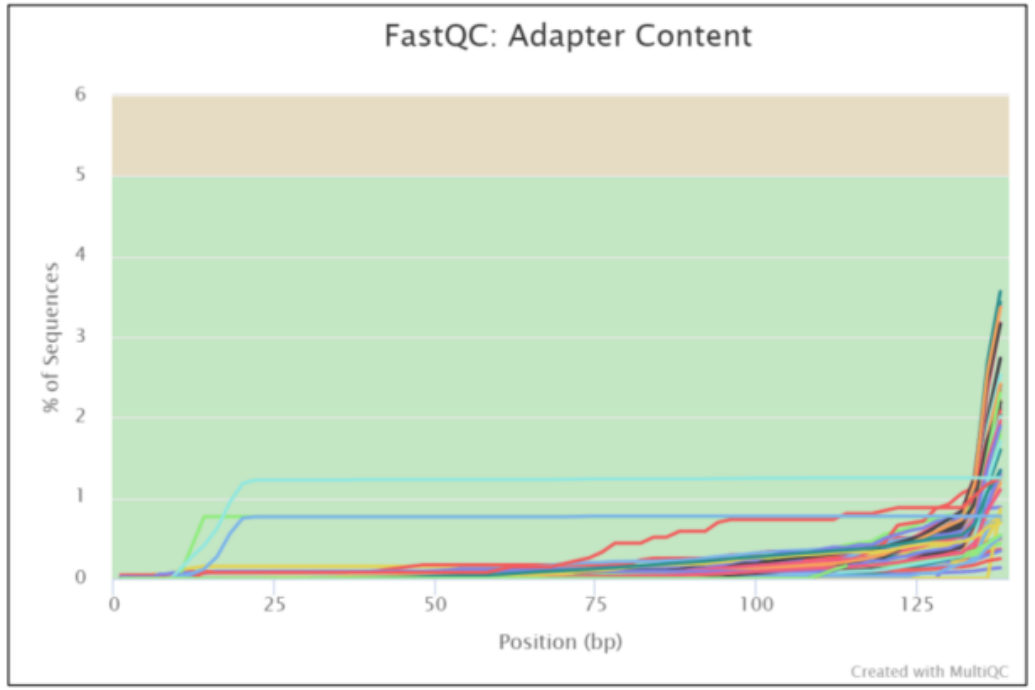
→ Unpaired reads can be mapped like SE reads if necessary

Output logs

- ✓ Input Read Pairs: 16923155
- ✓ Both Surviving Reads: 13396826
- ✓ Both Surviving Read Percent: **79.16**
- ✓ Forward Only Surviving Reads: 3199726
- ✓ Forward Only Surviving Read Percent: 18.91
- ✓ Reverse Only Surviving Reads: 126909
- ✓ Reverse Only Surviving Read Percent: 0.75
- ✓ Dropped Reads: 199694
- ✓ Dropped Read Percent: 1.18

Quality control: Trimmomatic (before and after)



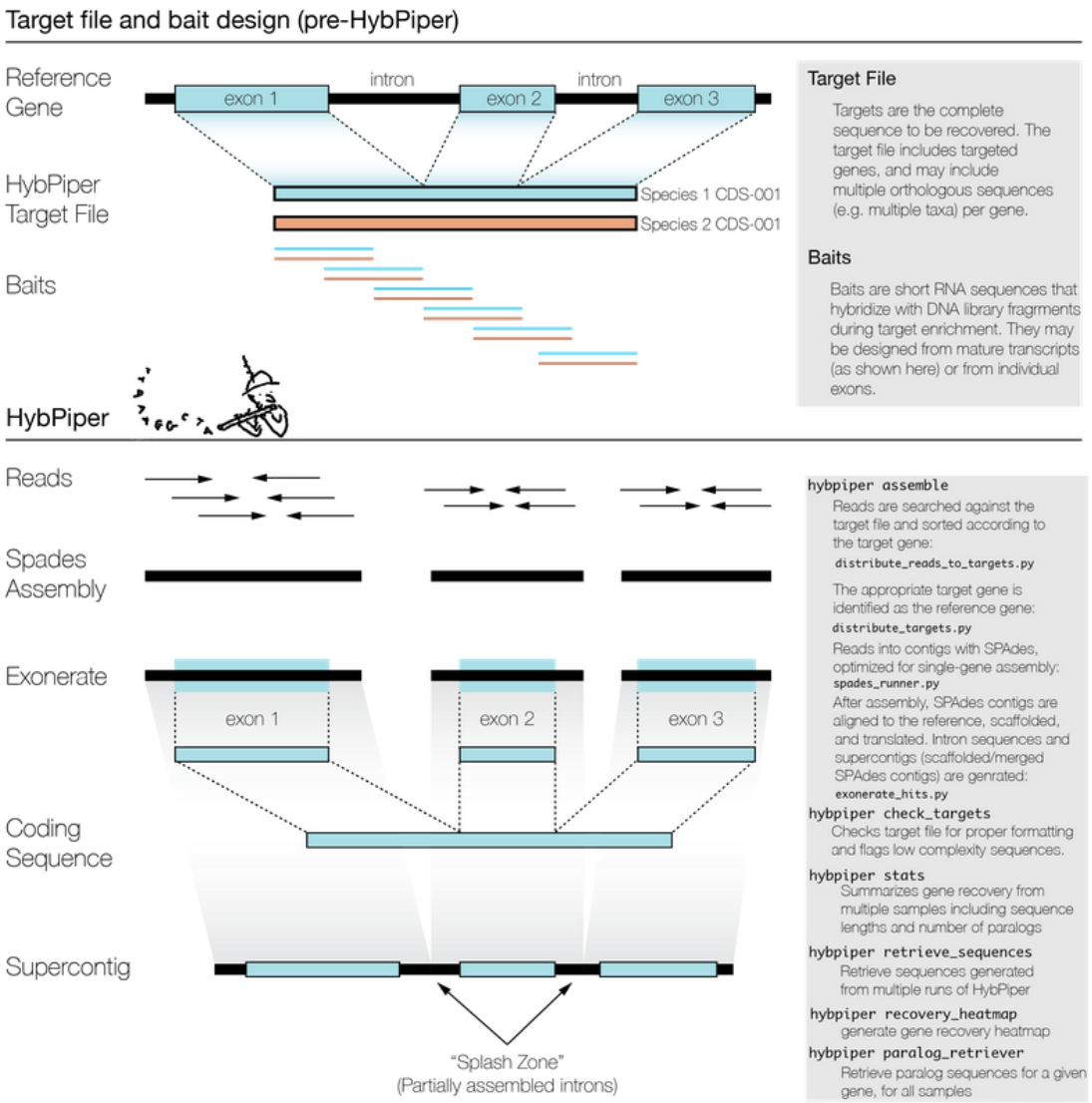


Pipeline: Cp assemblies

I. Lib prep
II. Sequencing - Demultiplexing
III. QC
  a) Sequencing Report
    - ✓ QC10: 1-10 error (90% accurate)
    - ✓ QC20: 1-100 error (99% accurate)
    - ✓ QC30: 1-1000 error (99,9% accurate)
  b) Fastqc: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/
    - ✓ Quick overview of all reads
    - ✓ Summary graphs
    - ✓ HTML-based report
    - ✓ Multiqc! Overview of multiple HTMP-reports from different analyses
  c) Trim
    - ✓ Trimmomatic: http://www.usadellab.org/cms/?page=trimmomatic
    - ✓ Cutadapt: https://cutadapt.readthedocs.io/en/stable/

IV. Cp – nr Assembly: Getorganelle
  - ✓ Toolkit for assemblie of organelle genome: https://github.com/Kinggerm/GetOrganelle

Hybpiper 2



Applications in Plant Sciences
Official publication of the Botanical Society of America

Appl Plant Sci. 2016 Jul; 4(7): apps.1600016.
Published online 2016 Jul 12. doi: 10.3732/apps.1600016
PMCID: PMC4948903
PMID: 27437175

HybPiper: Extracting coding sequence and introns for phylogenetics from high-throughput sequencing reads using target enrichment[1]

Matthew G. Johnson,[2,6] Elliot M. Gardner,[2,3] Yang Liu,[4] Rafael Medina,[4] Bernard Goffinet,[4] A. Jonathan Shaw,[5] Nyree J. C. Zerega,[2,3] and Norman J. Wickett[2,3]

Read the paper: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4948903/

Hybpiper

https://github.com/mossmatters/HybPiper

Run hybpiper on the test data.

The tool is extremely well documented. This is an exercise for home.

- https://github.com/mossmatters/HybPiper
- Download the test data
- Run on the HPC
- Let's take a look at the script

```
#!/usr/bin/bash
#PBS -N hybpiper_conda_tutorialdata
#PBS -l nodes=1:ppn=8
#PBS -o hybpiper.$PBS_JOBID.stdout
#PBS -e hybpiper.$PBS_JOBID.Stderr
#PBS -l walltime=01:00:00
#PBS -m abe
#PBS -M pieter.asselman@ugent.be

cd $PBS_O_WORKDIR

source ~/.bashrc
conda activate hybpiper &&

# Unpack the test dataset
tar -zxf test_reads.fastq.tar.gz

# Remove any previous runs
parallel rm -r {} :::: namelist.txt


# Run main HybPiper command with all available CPUs
while read sample_name
do
  hybpiper assemble -r ${sample_name}*.fastq -t_dna test_targets.fasta --prefix ${sample_name} --bwa  --run_intronerate
done < namelist.txt


# Get runs statistics
hybpiper stats -t_dna test_targets.fasta gene namelist.txt


# Get heatmap of length recovery
hybpiper recovery_heatmap seq_lengths.tsv

# Recover DNA and amino-acid sequences
hybpiper retrieve_sequences -t_dna test_targets.fasta dna --sample_names namelist.txt --fasta_dir 01_dna_seqs
hybpiper retrieve_sequences -t_dna test_targets.fasta aa --sample_names namelist.txt --fasta_dir 02_aa_seqs


# Recover paralog sequences
hybpiper paralog_retriever namelist.txt -t_dna test_targets.fasta


echo "DONE!"
```

Hybpiper is installed in a conda env. You first need to activate the env.

Download and prep data

Run hybpiper tools

Hybpiper

https://github.com/mossmatters/HybPiper

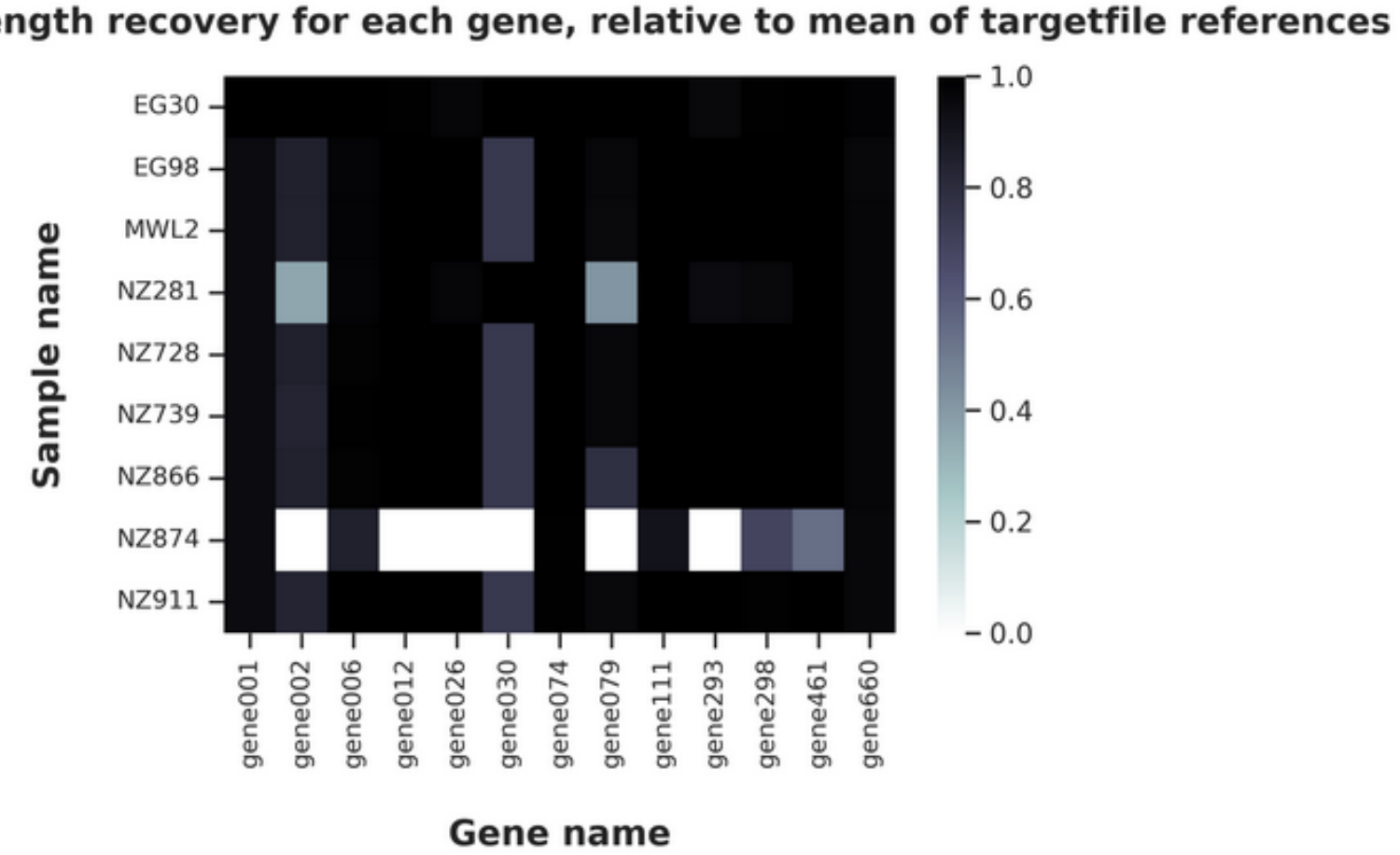
Run hybpiper on the test data.

The tool is extremely well documented. This is an exercise for home.

- https://github.com/mossmatters/HybPiper
- Download the test data
- Run on the HPC
- Let's take a look at the script
- Output (more to discover than just a fancy heatmap, but just to get the gist of it)

Hybpiper

https://github.com/mossmatters/HybPiper



Documentation

- Website: https://www.ugent.be/hpc/en
- HPC & linux documentation: https://www.ugent.be/hpc/en/support/documentation.htm
- Open Stack Dashboard: https://login.hpc.ugent.be
- Cluster state info: https://shieldon.ugent.be:8083/pbsmon-web-users/
- Software installation: https://www.ugent.be/hpc/en/support/software-installation-request

  Tip: DIY installations 'easybuild' see Chp 28

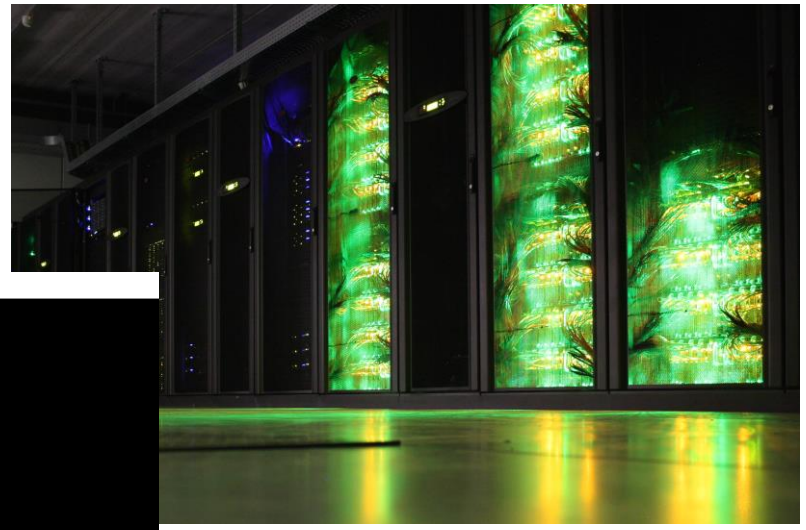
- Flemish compute center: https://www.vscentrum.be/

Need help with issues: hpc@ugent.be

Introduction HPC-UGent

```
#!/bin/bash

#PBS -S /bin/bash
#PBS -l walltime=06:00:00
#PBS -N JOBNAME
#PBS -o stdout.$PBS_JOBID
#PBS -e stderr.$PBS_JOBID
#PBS -l nodes=1:ppn=all:gpus=2
#PBS -m abe
#PBS -M pieter.asselman@ugent.be

########
##INFO##
########

#DATArun on 2GPU/ 11Gbases - ~3h

##BASECALL##
guppy_basecaller -i /scratch/gent/vo/000/gv

##DEMULTIPLEX##
#guppy_barcoder --i /scratch/gent/vo/000/gvo00058/vs

##BASECALL and DEMULTIPLEX##
#guppy_basecaller -i /scratch/gent/vo/000/gvo00058/v
```

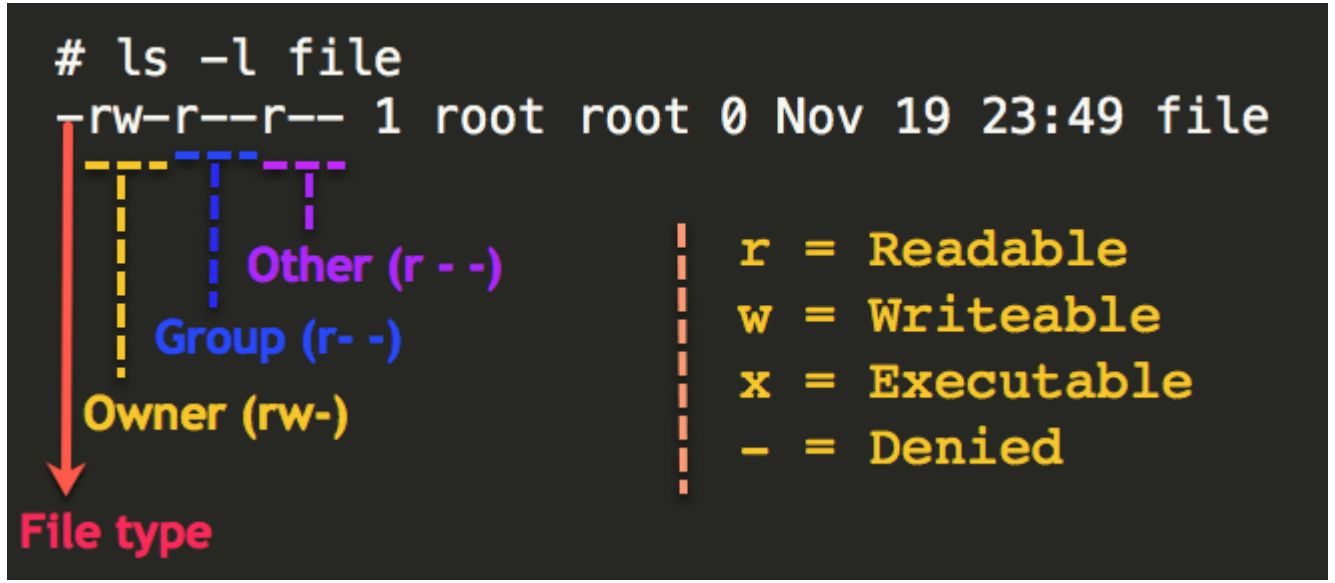
Questions?
Comments?
Suggestions?
Concerns?

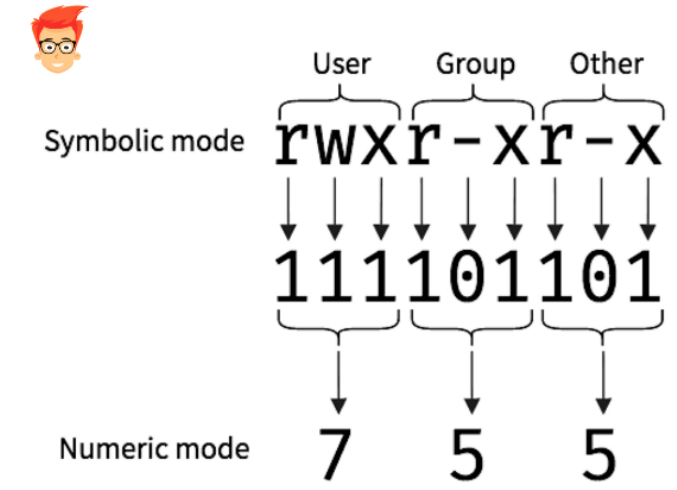


```
https://www.ugent.be/hpc/en/infrastructure/status

We switched to new job command wrappers for all HPC-UGent Tier-2 clusters on Wed
 June 9th 2021 at 17:22.
This switch should be transparant: you don't need to change your workflow or job
 scripts.

If you notice any problems, or if any of these changes affect your work, please
contact hpc@ugent.be .
vsc43352@gligar04:~$
```

https://ugent.be/hpc
hpc@ugent.be

DEMO – file permissions





/user/gent/433/vsc43352/data_gent_vo/data_share_group

UGO

Examples:

```
chmod u+rwx,g+rwx,o-rwx
chmod u=rwx,g=rwx
chgrp –R groupname file(or folder)
```

=> check for groups: `$ groups`

drwxrwxrwx

d = Directory
r = Read
w = Write
x = Execute

7	rwx	111
6	rw-	110
5	r-x	101
4	r--	100
3	-wx	011
2	-w-	010
1	--x	001
0	---	000

chmod 777

rwx | rwx | rwx
Owner | Group | Others