INF5380/INF9380

Variant calling

A commonly used pipeline and how to use parallelisation to make the *most* of it

Arvind Sundaram
NSC, Ullevål
March 2020

Adapted from slides by
Merete Molton Worren
Abdulrahman Azab (2016)

Parallelisation in bioinformatics

- (Most of the) Bioinformatic tools are ad-hoc tools designed to do one specific thing.
- Pipelines generally use several of these tools to form a chain of commands
- Many tools are designed to use multiple threads but this is not always possible and seldom uses multiple cores
- Parallelisation is a night-mare the building pipelines

nextflow A DSL for parallel and scalable computational pipelines

Variant calling

- A variant call is a conclusion that there is a nucleotide difference vs. some reference at a given position in an individual genome or transcriptome
- Usually accompanied by an estimate of variant frequency and some measure of confidence

Variant calling

- The goal is to find differences between a reference and your data
  - Somatic and germline
    - Disease/tumor vs normal
    - Different strains of bacteria
- Differences can include SNVs, Indels, Copy number variations
- This lecture focuses on SNVs

VCF/BCF

- Variant call format
- BCF - same as/similar to VCF but in binary format
  - Most softwares and tools prefer BCF or VCF in zip format
  - header and records

header

```
##fileformat=VCFv4.2
##source=myImputationProgramV3.1
##reference=file:///seq/references/1000GenomesPilot-NCBI36.fasta
##contig=<ID=20,length=62435964,assembly=B36,md5=xxxx,species="Homo sapiens",taxonomy=x>
...
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
...
##FILTER=<ID=q10,Description="Quality below 10">
...
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
…
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT NA00001 NA00002 NA00003
```

https://samtools.github.io/hts-specs/VCFv4.2.pdf

VCF/BCF: records

```
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT NA00001
20 14370 rs6054257 G A 29 PASS NS=3;DP=14;AF=0.5;DB;H2 GT:GQ:DP:HQ 0|0:48:1:51,51

20 17330 . T A 3 q10 NS=3;DP=11;AF=0.017 GT:GQ:DP:HQ 0|0:49:3:58,50

20 1110696 rs6040355 A G,T 67 PASS NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ 1|2:21:6:23,27
```

	#CHRO	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	SAMPLEs
SNP	20	3	.	C	G	.	PASS	DP=10		
Deletion	20	2	.	TC	C	.	PASS	DP=10		
Insertion	20	2	.	TC	TCA	.	PASS	DP=10		
Alleles	20	2	.	TC	TG,T	.	PASS	DP=10		
Alleles	20	2	.	TCG	TG,T,TCA	.	PASS	DP=10		

+ Structural variants

http://vcftools.sourceforge.net/VCF-poster.pdf

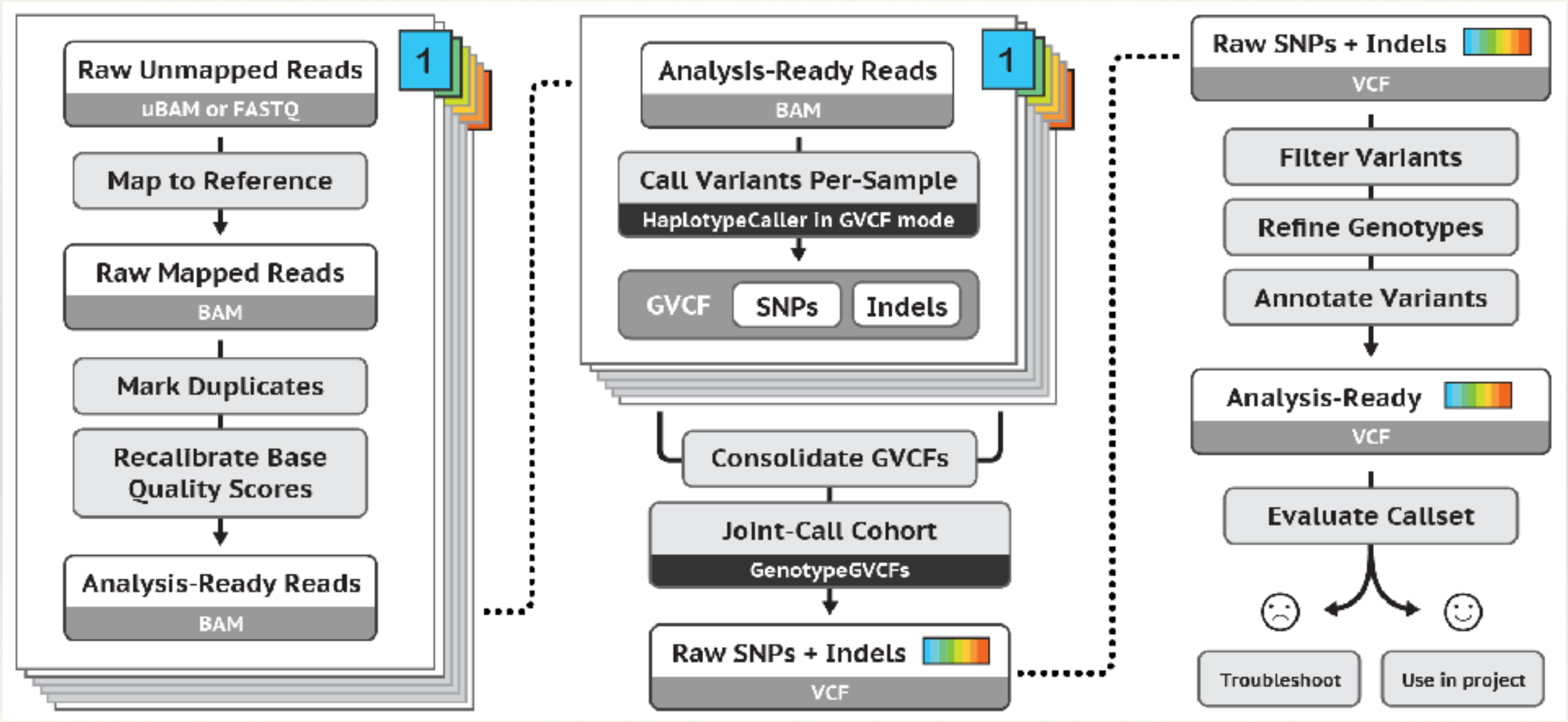
SNV calling

- Several ways of doing things
- Different experiments may require different analysis steps
- In todays lecture you will learn a very basic SNV calling pipeline. The focus will also be on parallelisation.

SNV calling

https://www.baseclear.com/blog/bioinformatics/baseclears-variant-detection-pipeline-renewed/

Typical workflow



https://github.com/roonysgalbi/variantcalling

A couple of resources

- GATK Broad Institute
  - https://software.broadinstitute.org/gatk
- BCFtools
  - https://samtools.github.io/bcftools
- Teaching material
  - https://datacarpentry.org/wrangling-genomics

Hands-on session

Background and Metadata

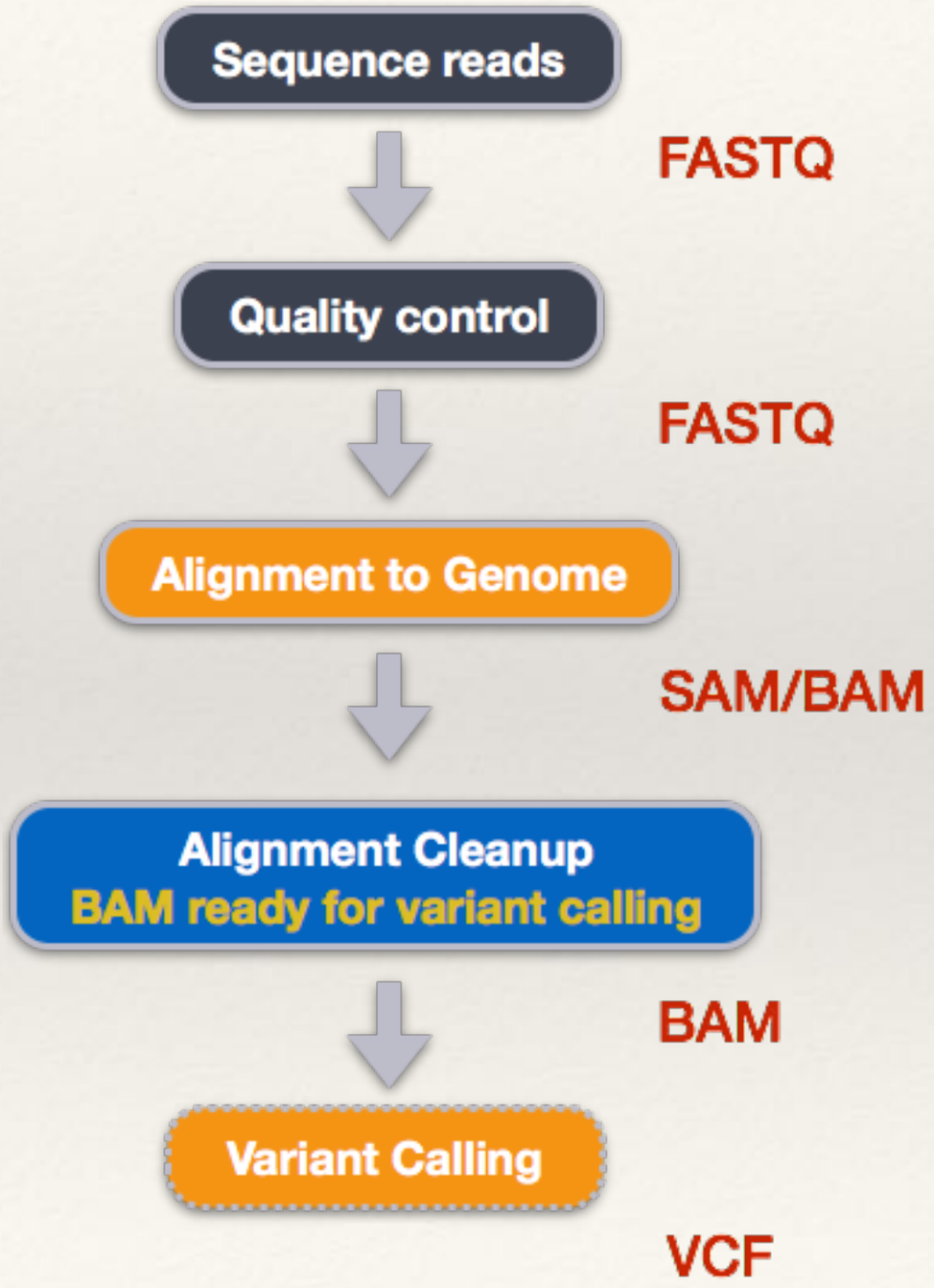
- *E.coli* long-term evolution experiment led by Richard Lenski.
  - https://datacarpentry.org/wrangling-genomics/01-background/index.html

https://en.wikipedia.org/wiki/E._coli_long-term_evolution_experiment

Today's workflow

- Quality control with FastQC
- Quality filtering with Trimmomatic
- Mapping/Alignment with BWA - MEM
- ~~Preprocessing BAM file~~
  - ~~Mark duplicates, Realign indels (GATK v3), Base calibration~~
- BCLtools - calculate genotype likelihood followed by variant calling

Today's workflow



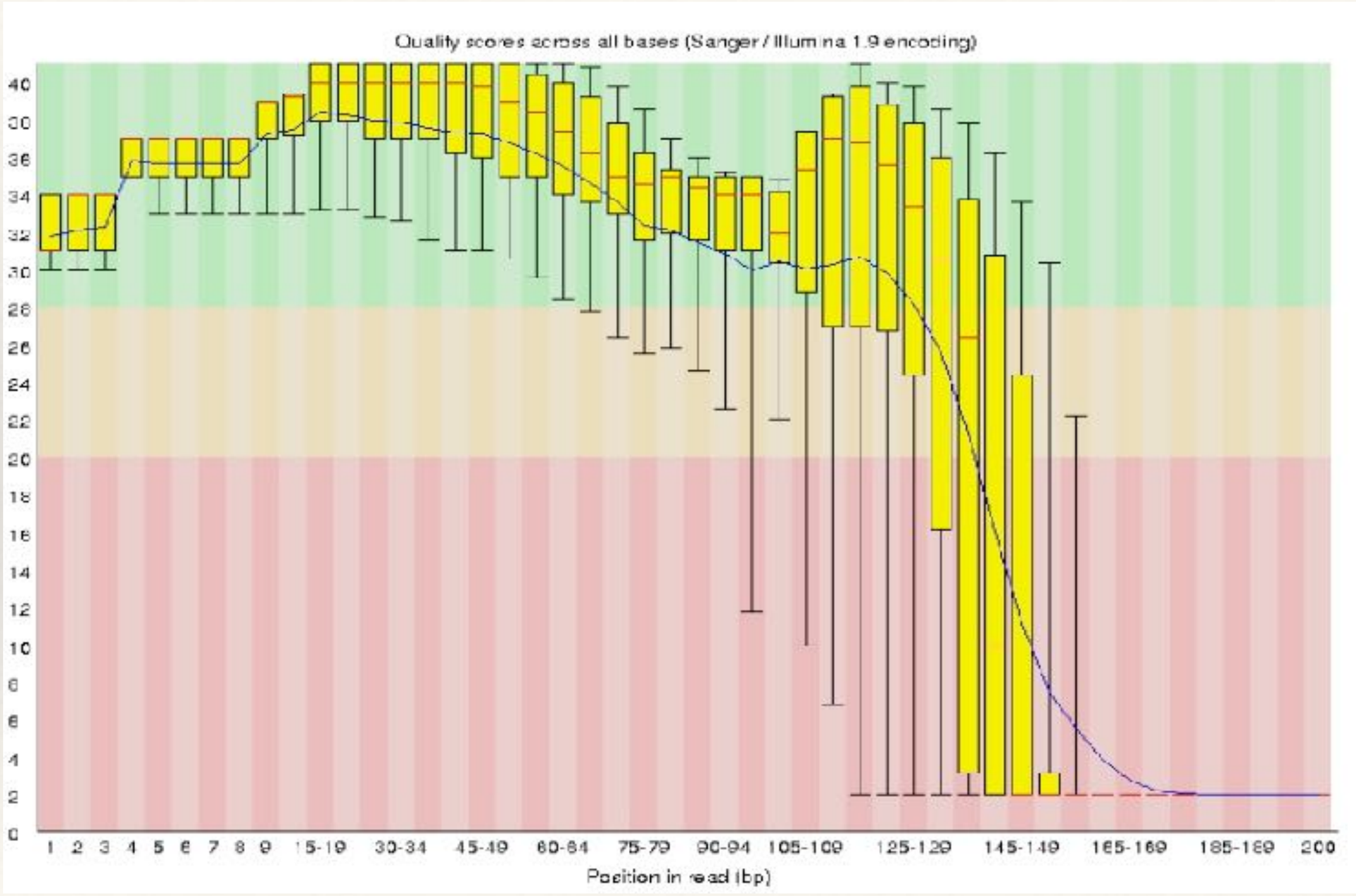
Quality control - FastQC

- Fastq files straight from the sequencer often have bad quality and/or contamination
- We need to know, and we need to fix it before doing variant calling
- Running FastQC will give us an overview of what our reads look like

How does FastQC work?

- Series of analysis modules
- For all these modules, you get results from all the reads.
- A link to the FastQC documentation can be found at their website
  - http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

FastQC per base quality plot



Parallelising FastQC

- No option for parallelising with only one file
- If you work with several files from different samples, you should run them in parallel to save time
- From the FastQC manual:

> -t --threads
>
> Specifies the number of files which can be processed simultaneously. Each thread will be allocated 250MB of memory so you shouldn't run more threads than your available memory will cope with, and not more than 6 threads on a 32 bit machine

Time for practical exercises!

1. Get Ready

2. Quality Control

Discussion

- Why does it not help to add more threads than the number of files?
- Why is the real time the only one that changes when we add more threads?
- Can you think of a way to parallelise FastQC for only one file?

Data preprocessing

- Remove/Trim sequences with bad quality
- Remove/Trim sequences that matches the adapter sequences

- Trimmomatic is one of several programs that can be used to do this
  - http://www.usadellab.org/cms/?page=trimmomatic

Trimmomatic

- Quick start:
  - Paired End:
    - `java -jar trimmomatic-0.XX.jar PE -phred33 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:TruSeq3-PE.fa:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36`
  - Single End:
    - `java -jar trimmomatic-0.XX.jar SE -phred33 input.fq.gz output.fq.gz ILLUMINACLIP:TruSeq3-SE:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36`

Trimmomatic

- ILLUMINACLIP
  - Cut adapter and other Illumina-specific sequences from the read
  - Adapter file location
- SLIDINGWINDOW
  - Perform a sliding window trimming, cutting once the average quality within the window falls below a threshold.
- LEADING
  - Cut bases off the start of a read, if below a threshold quality
- TRAILING
  - Cut bases off the end of a read, if below a threshold quality
- CROP
  - Cut the read to a specified length
- HEADCROP
  - Cut the specified number of bases from the start of the read
- MINLEN
  - Drop the read if it is below a specified length

Parallelising Trimmomatic

- Try -threads option

Time for practical exercises!

3. Quality filtering

Discussion

- Why did we need the line number in this exercise to be dividable by 4
- What's your opinion on the overhead in this parallelisation exercise?
- Are there unused possibilities for parallelising Prinseq lite?
- Would it be useful?

Alignment/Mapping

- Mapping – where in the genome does the read come from?
- Alignment – what is the exact placement of each base in the read?
- A number of different mappers/aligners are available
- We will use BWA - MEM

Burrows-Wheeler alignment tool

- Indexing based on Burrows - Wheeler transform
- Several different alignment algorithms
- We will use the MEM algorithm (maximal exact matches)
  - http://bio-bwa.sourceforge.net/bwa.shtml

Parallelising BWA-MEM

- Index - no command line option for parallelising
- MEM - option for threads

Time for practical exercises!

4. Alignment

Discussion

- Is BWA- MEM a mapper or an aligner?
- How do you think the BWA - MEM algorithm is parallelised ?

Variant calling

- The data is now ready, so we can do the main task, variant calling
- We do this with the BCFtools mpileup and vcfutils.pl
- It can find both SNPs and indels

Time for practical exercises!

5. Variant calling

Discussion

- Check ‘threads’ option in bcftools!!