Mapping, SNP & SV

Christine Tranchant-Dubreuil & Francois Sabot

October, 2018

IRD - UMR DIADE

TP goal

- Quality control of NGS data

TP goal

- Quality control of NGS data
- Learn to manipulate NGS data

TP goal

- Quality control of NGS data
- Learn to manipulate NGS data
- Having a critical look on *Mapping*

TP goal

- Quality control of NGS data
- Learn to manipulate NGS data
- Having a critical look on *Mapping*
- Learn to launch a *Calling* and having a critical look

TP goal

- Quality control of NGS data
- Learn to manipulate NGS data
- Having a critical look on *Mapping*
- Learn to launch a *Calling* and having a critical look
- Learn the basic of Structural Variations

You are working on a Cluster...

What is a Cluster ?

You are working on a Cluster...

What is a Cluster ?

- A logical unit composed of multiple servers

You are working on a Cluster...

What is a Cluster ?

- A logical unit composed of multiple servers
- Will work as a unique powerful machine

You are working on a Cluster...

What is a Cluster ?

- A logical unit composed of multiple servers
- Will work as a unique powerful machine
- HPC: High-Performance Computing

You are working on a Cluster...

What is a Cluster ?

- A logical unit composed of multiple servers
- Will work as a unique powerful machine
- HPC: High-Performance Computing
- Higher storage capacity

You are working on a Cluster...

What is a Cluster ?

- A logical unit composed of multiple servers
- Will work as a unique powerful machine
- HPC: High-Performance Computing
- Higher storage capacity
- Higher reliability

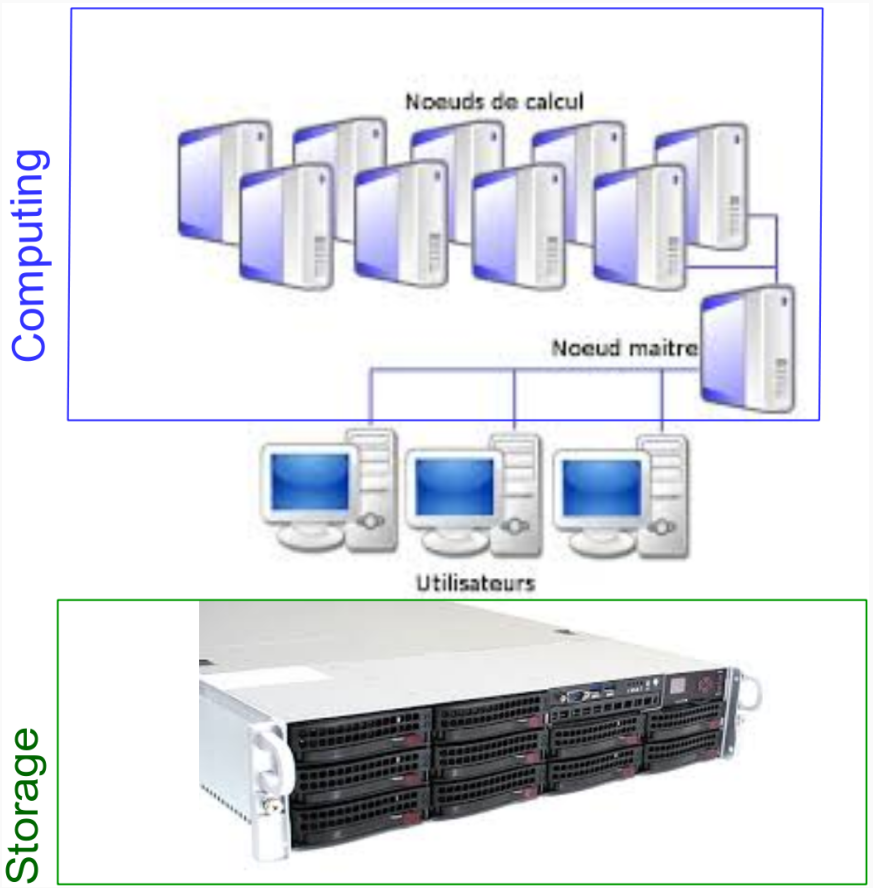
You are working on a Cluster...

What is a Cluster ?

- A logical unit composed of multiple servers
- Will work as a unique powerful machine
- HPC: High-Performance Computing
- Higher storage capacity
- Higher reliability
- Higher resources availability

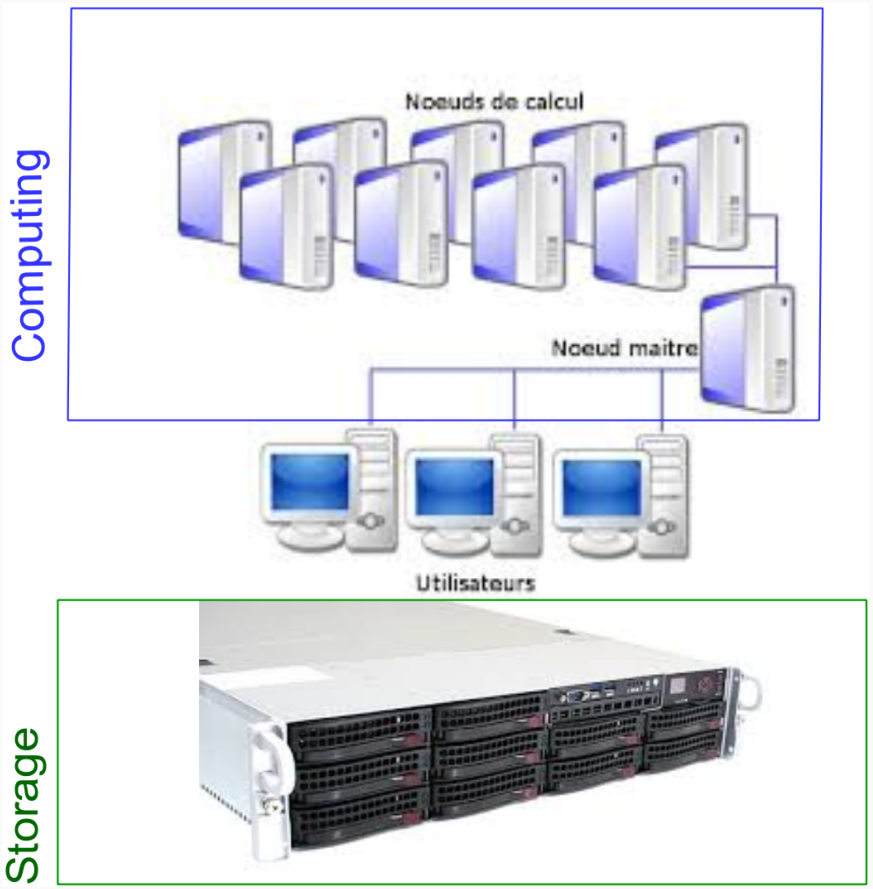
Master node and slave nodes

- The Master node:



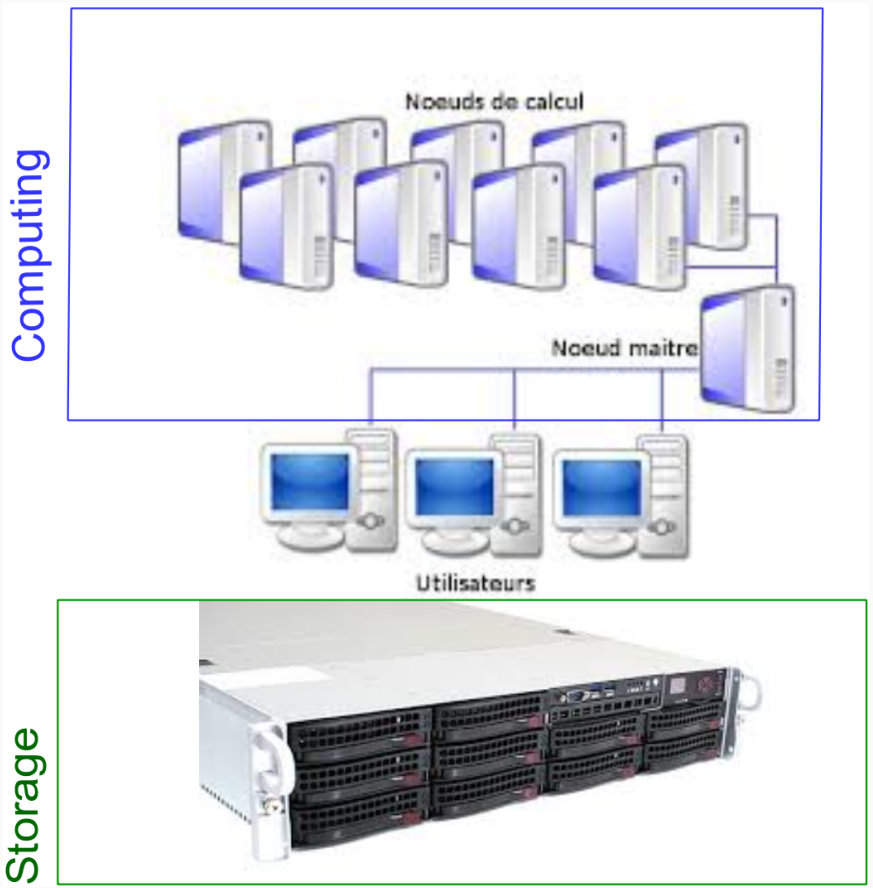
Master node and slave nodes

- The Master node:
  - Users connect on it



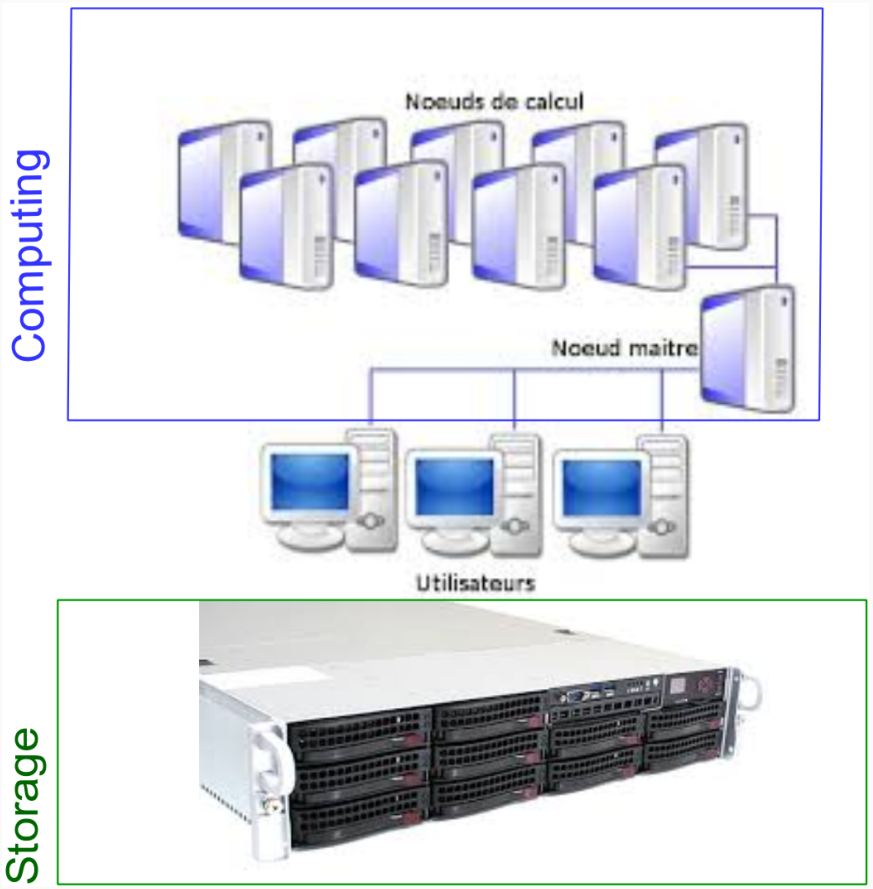
Master node and slave nodes

- The Master node:
  - Users connect on it
  - Schedules jobs



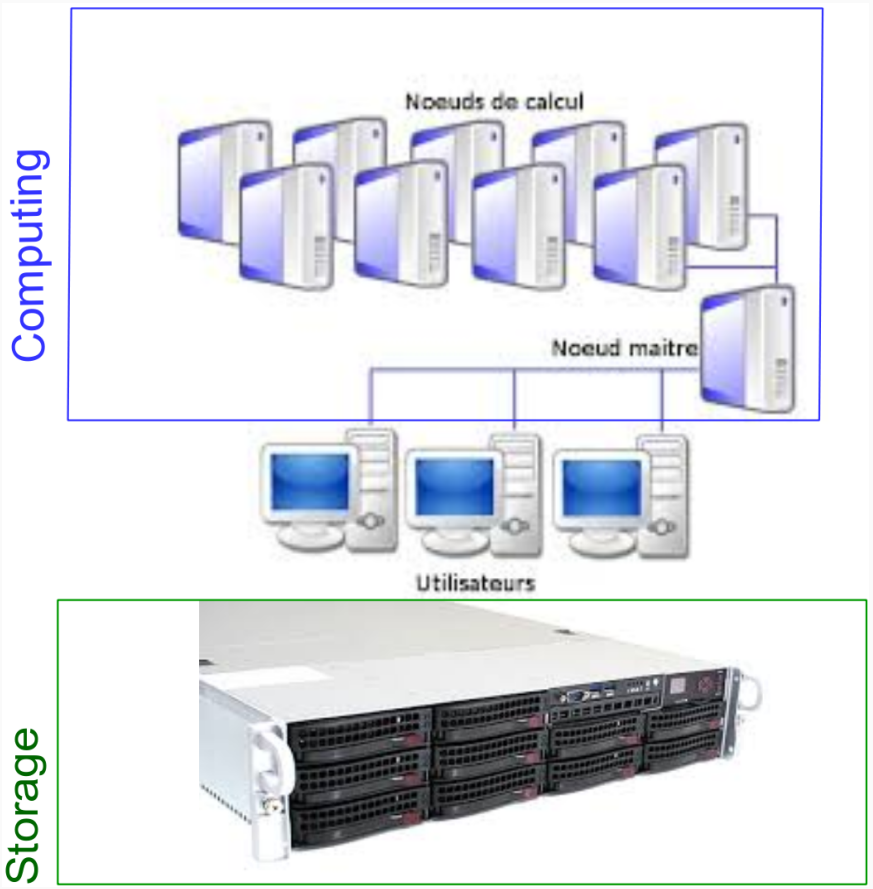
Master node and slave nodes

- The Master node:
  - Users connect on it
  - Schedules jobs
  - Manages resources and priorities



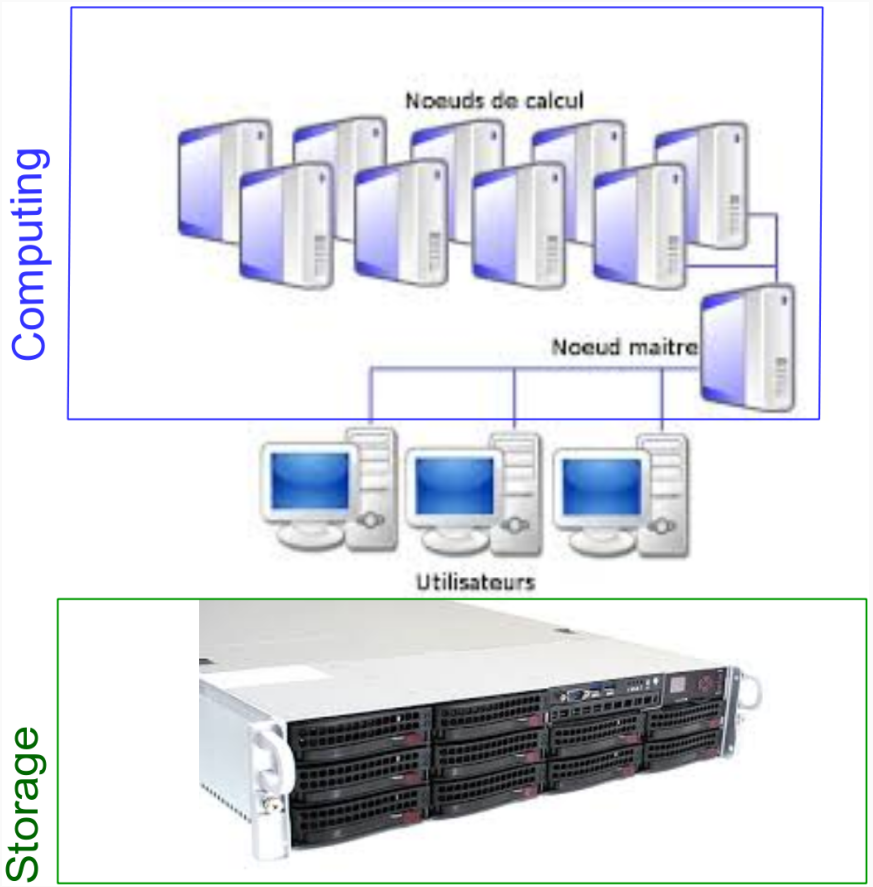
Master node and slave nodes

- The Master node:
  - Users connect on it
  - Schedules jobs
  - Manages resources and priorities
- The Computing Nodes:



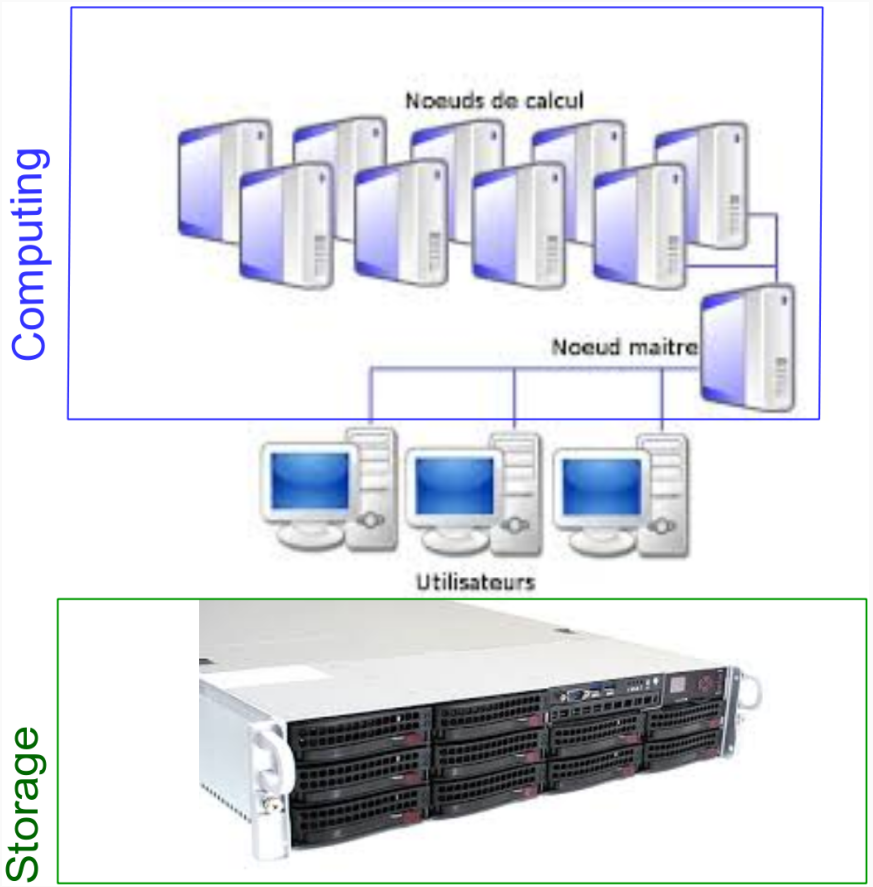
Master node and slave nodes

- The Master node:
  - Users connect on it
  - Schedules jobs
  - Manages resources and priorities
- The Computing Nodes:
  - Receive job instructions



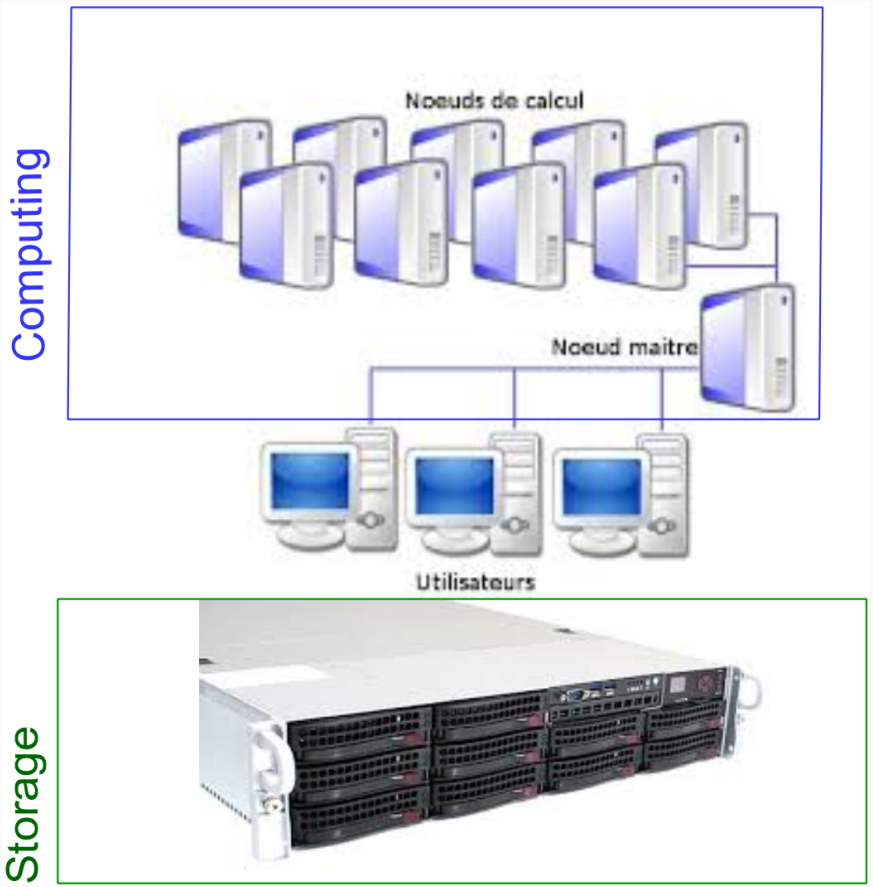
Master node and slave nodes

- The Master node:
  - Users connect on it
  - Schedules jobs
  - Manages resources and priorities
- The Computing Nodes:
  - Receive job instructions
  - Perform jobs



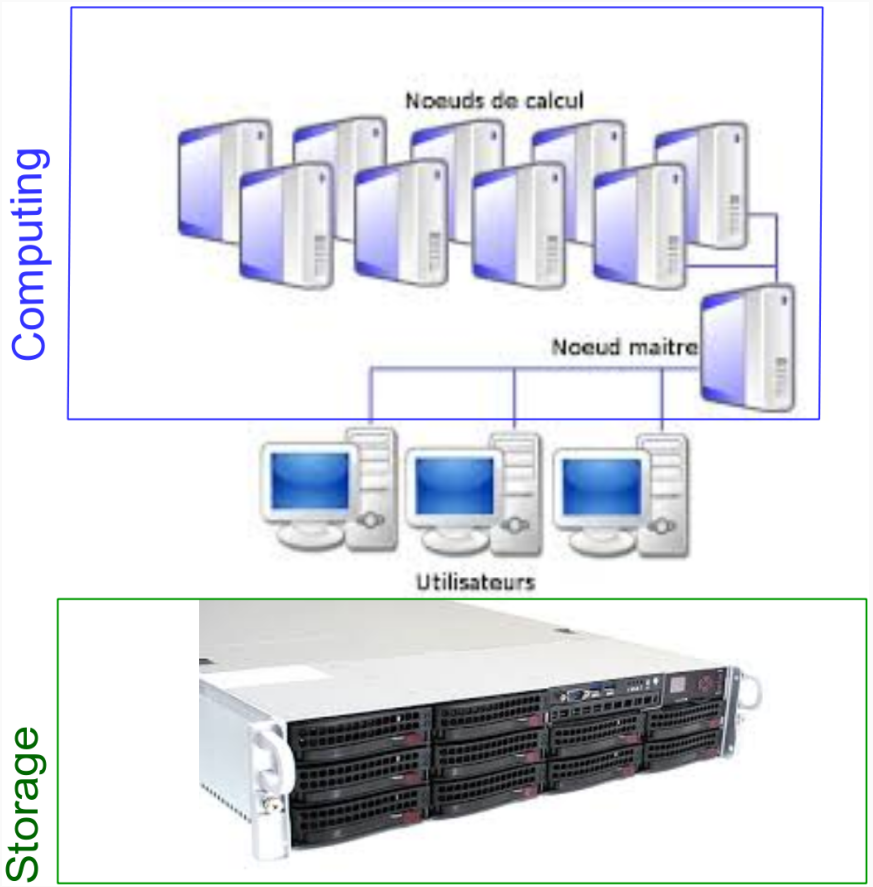
Master node and slave nodes

- The Master node:
  - Users connect on it
  - Schedules jobs
  - Manages resources and priorities
- The Computing Nodes:
  - Receive job instructions
  - Perform jobs
  - Send to Master the results
- The NAS:

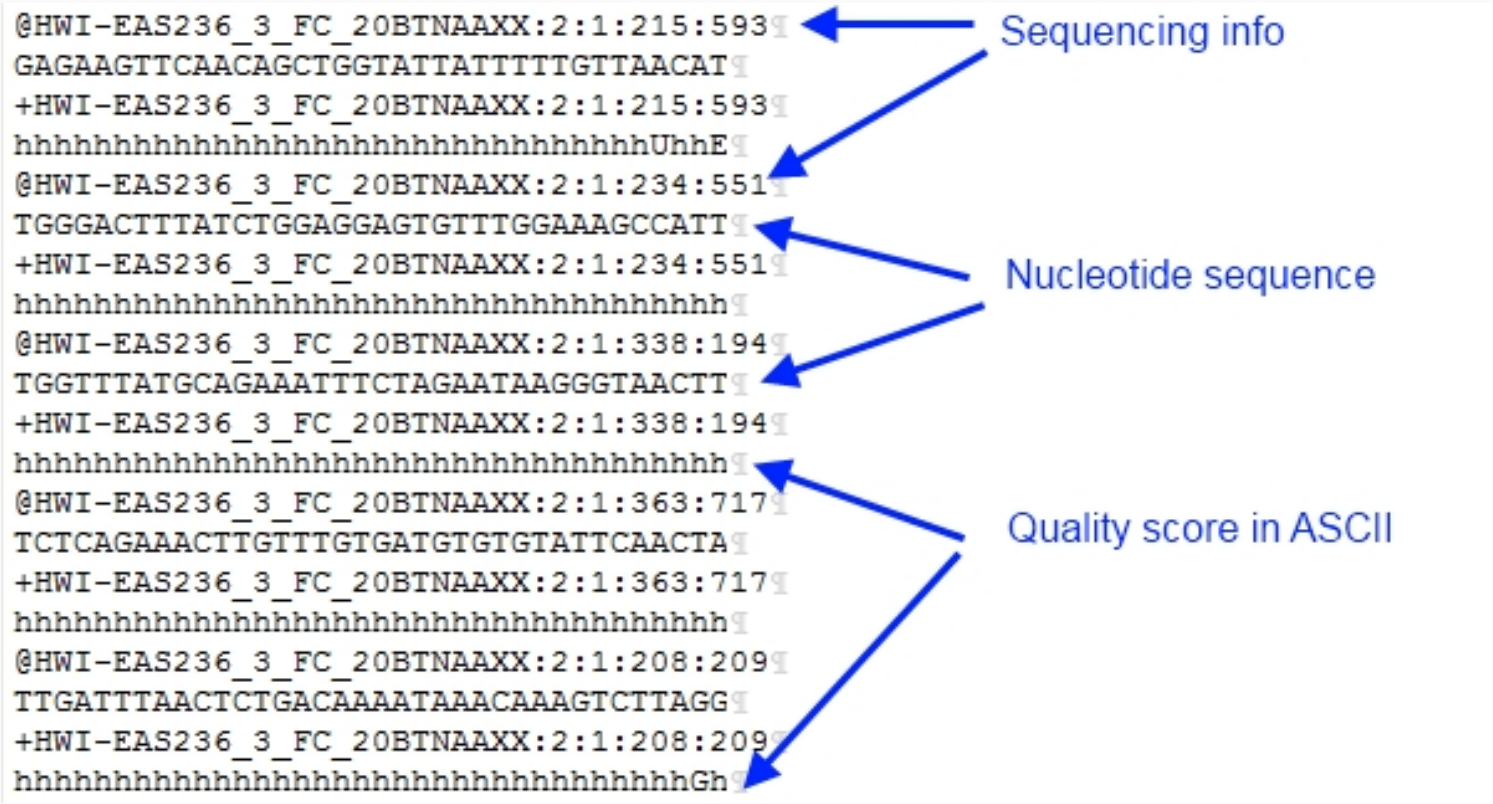


Master node and slave nodes

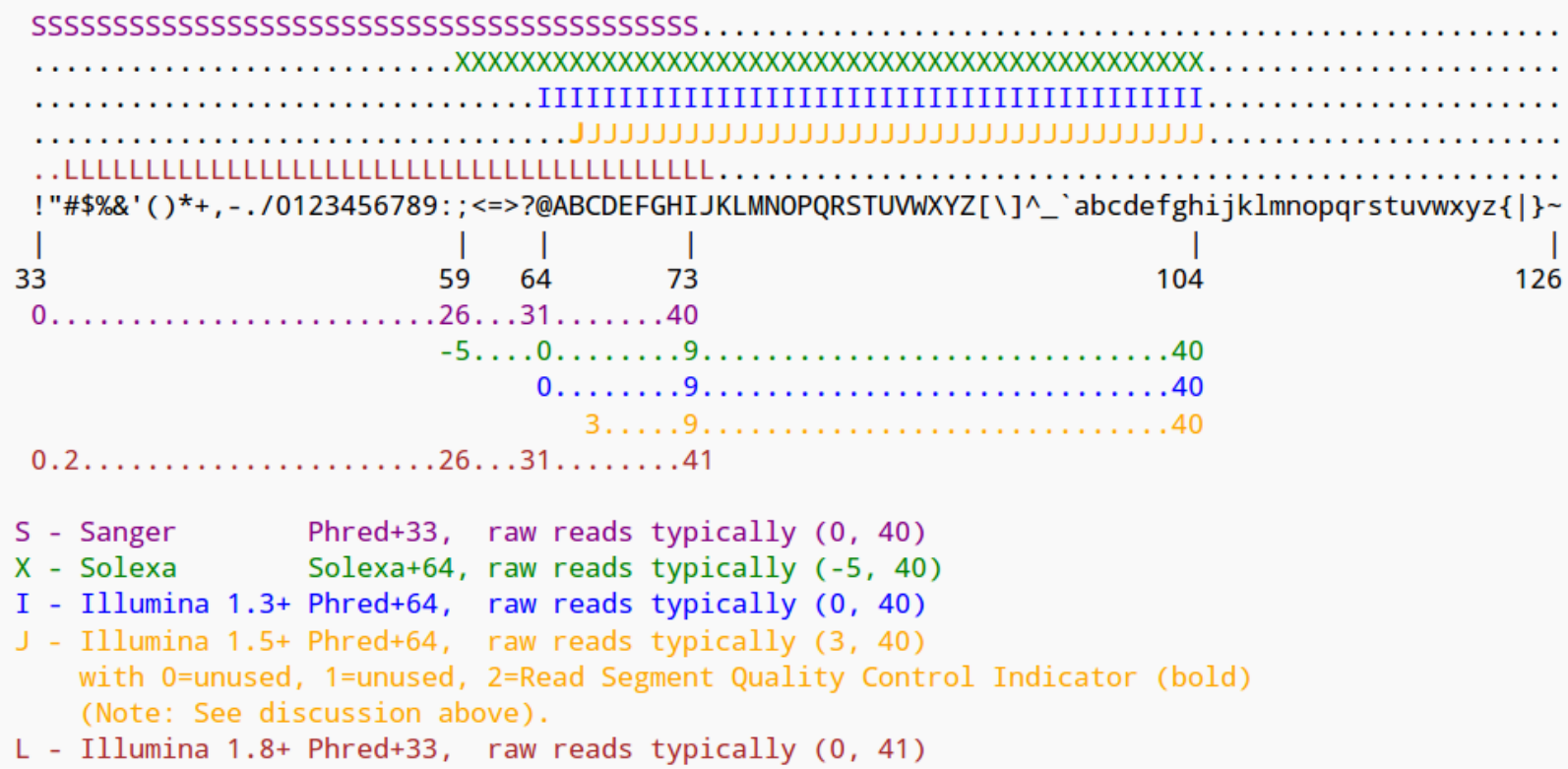
- The Master node:
  - Users connect on it
  - Schedules jobs
  - Manages resources and priorities
- The Computing Nodes:
  - Receive job instructions
  - Perform jobs
  - Send to Master the results
- The NAS:
  - Store data for computing



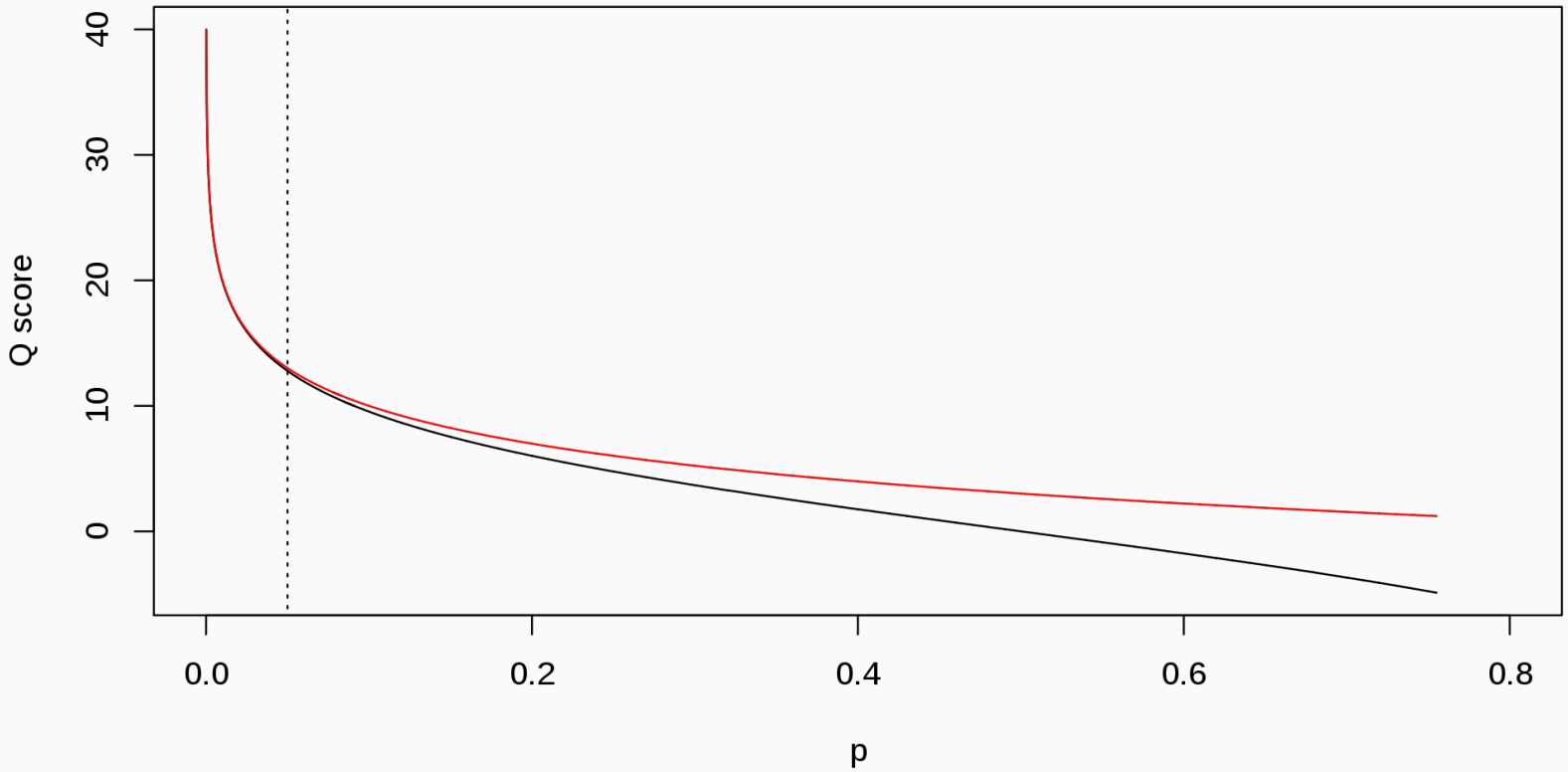
The FASTQ Format



The QPHRED Scale



The QPHRED Value



Preparing...

Data are on the CERAAS cluster, located on the storage at **/data/FORMATION/2018/TPsnpSV**

1. Connect to the cluster:
   ssh -X formationX@41.82.52.216

Preparing...

Data are on the CERAAS cluster, located on the storage at **/data/FORMATION/2018/TPsnpSV**

1. Connect to the cluster:
   ssh -X formationX@41.82.52.216
2. Launch a QRSH command:
   qrsh

Preparing...

Data are on the CERAAS cluster, located on the storage at **/data/FORMATION/2018/TPsnpSV**

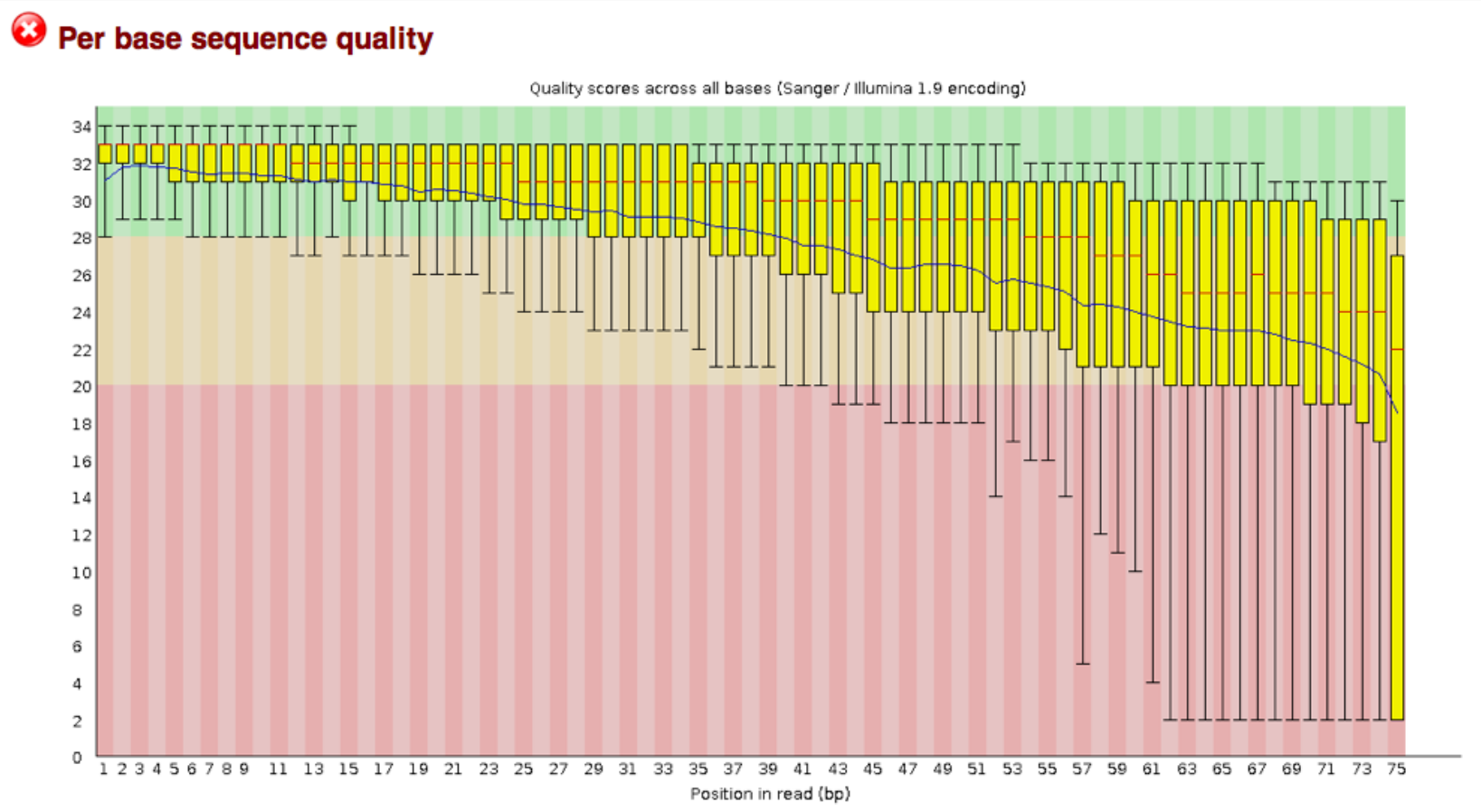
1. Connect to the cluster:
   ssh -X formationX@41.82.52.216
2. Launch a QRSH command:
   qrsh
3. Create your folder in scratch and go in it:
   mkdir /scratch/formationX
   cd /scratch/formationX

Preparing...

Data are on the CERAAS cluster, located on the storage at **/data/FORMATION/2018/TPsnpSV**

1. Connect to the cluster:
   ssh -X formationX@41.82.52.216
2. Launch a QRSH command:
   qrsh
3. Create your folder in scratch and go in it:
   mkdir /scratch/formationX
   cd /scratch/formationX
4. Transfer the data from nas using SCP:
   scp -r master:/data/FORMATION/2018/TPsnpSV .

Checking Quality



Quality Control

- Use FastQC on each data

  fastqc FILE.fastq

Quality Control

- Use FastQC on each data

  fastqc FILE.fastq

- What is the global quality of data ?

Quality Control

- Use FastQC on each data
  fastqc FILE.fastq
- What is the global quality of data ?
- For that you have to download it from your scratch to master then your own computer...

Cleaning Data

Why Cleaning Data ?

- Removing Adapters and Tags

Cleaning Data

Why Cleaning Data ?

- Removing Adapters and Tags
- Removing low quality bases (only conserving 20+ QPHRED value bases)

Cleaning Data

Why Cleaning Data ?

- Removing Adapters and Tags
- Removing low quality bases (only conserving 20+ QPHRED value bases)
- Removing contaminants (rRNA genes, organite data,...)

Cleaning Data

Tools for Cleaning Data

- FASTX-tools toolbox

Cleaning Data

Tools for Cleaning Data

- FASTX-tools toolbox
- CutAdapt

Cleaning Data

Tools for Cleaning Data

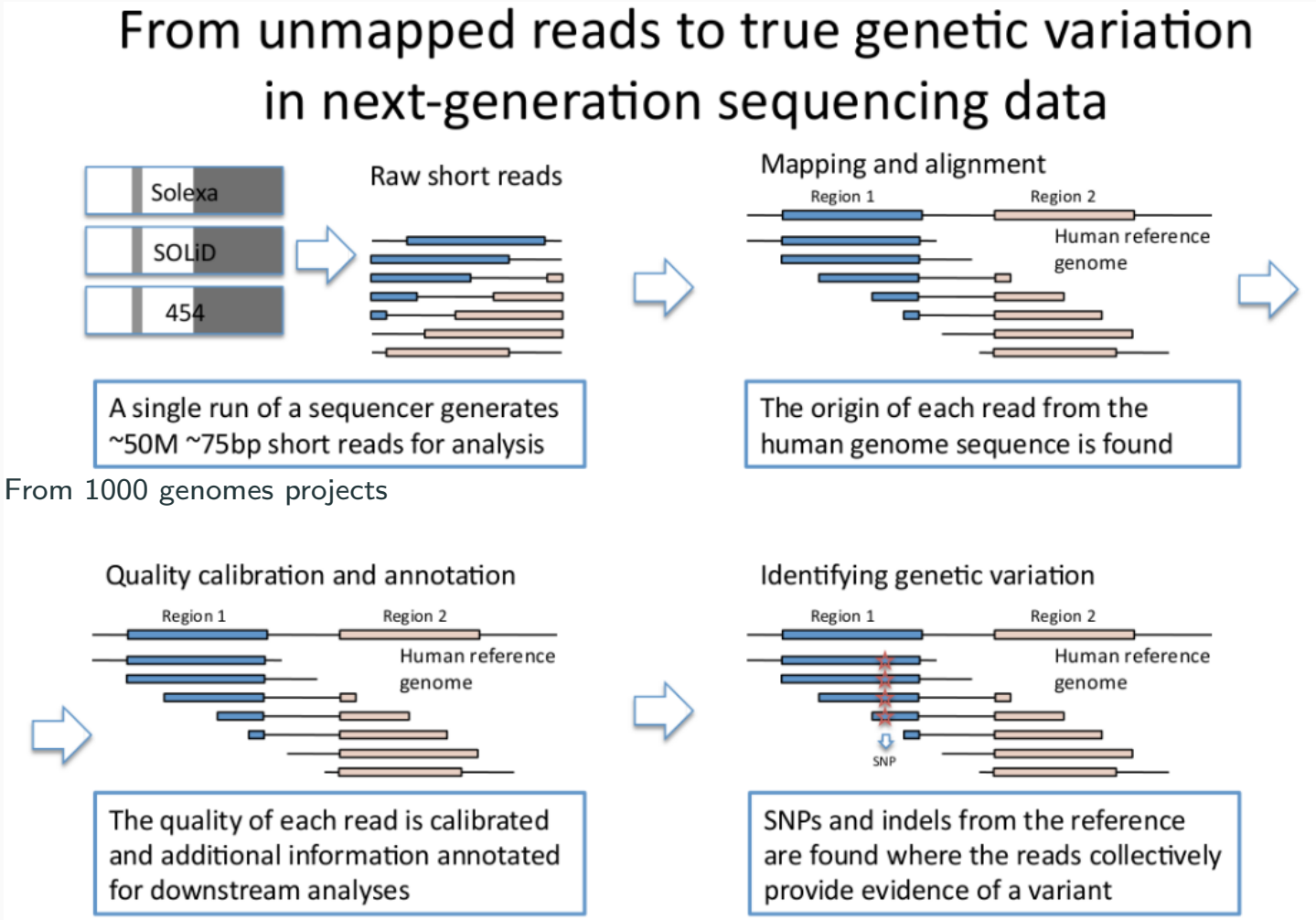
- FASTX-tools toolbox
- CutAdapt
- Trimmomatic

Cleaning Data

Tools for Cleaning Data

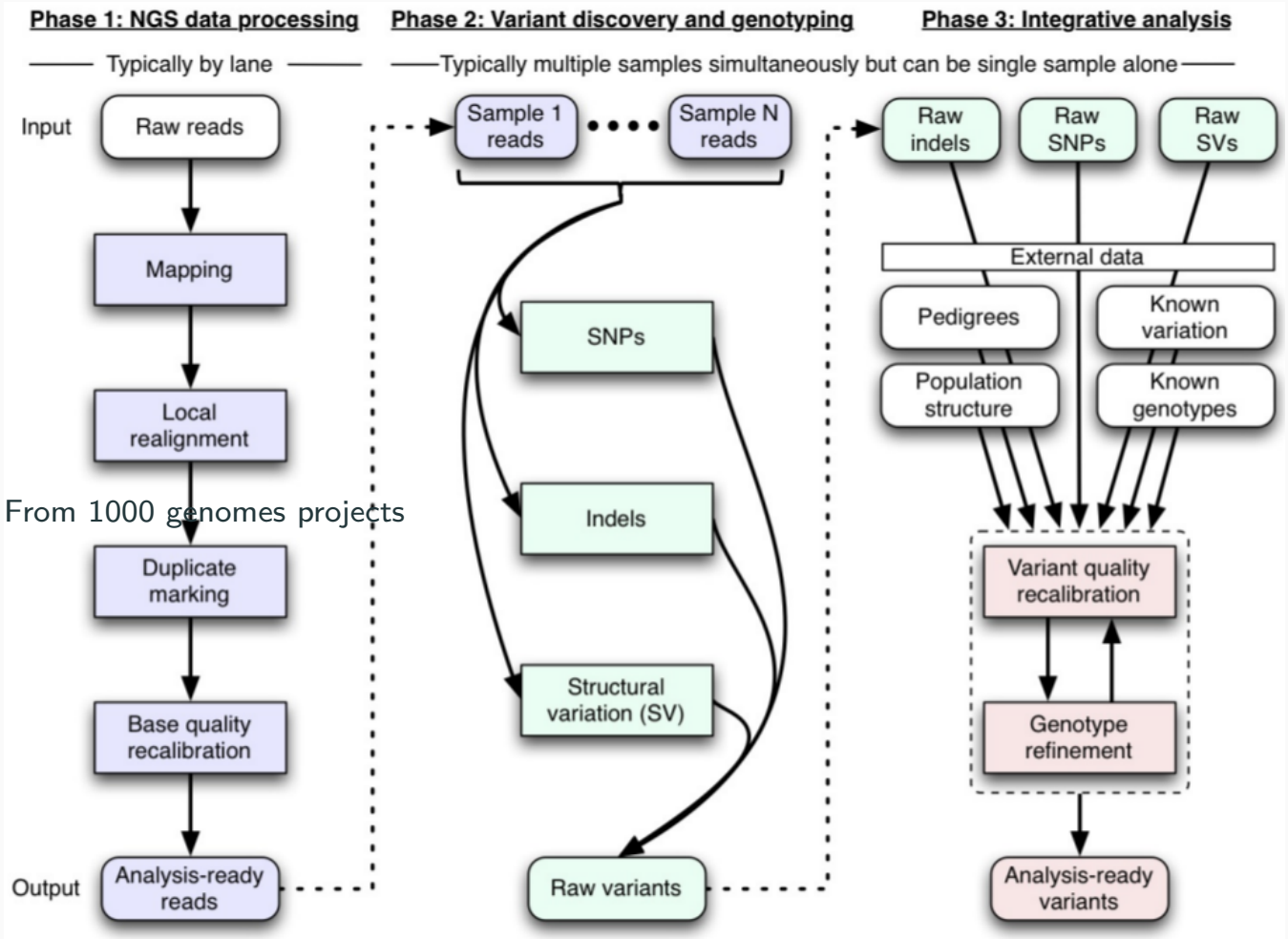
- FASTX-tools toolbox
- CutAdapt
- Trimmomatic
- Home-made Scripts based on QPHRED scale encoding

Principe of SNP calling



From 1000 genomes projects

Principe of SNP calling



From 1000 genomes projects

List of analyses

"Classic" launch

1. *Mapping*: bwa aln/sampe, bwa mem, bowtie2, ...

List of analyses

"Classic" launch

1. *Mapping*: bwa aln/sampe, bwa mem, bowtie2, ...
2. *Cleaning mapping*: samtools, picard-tools,...

List of analyses

"Classic" launch

1. *Mapping*: bwa aln/sampe, bwa mem, bowtie2, ...
2. *Cleaning mapping*: samtools, picard-tools,...
3. *Realigning and Duplicates*: GATK, picard-tools,...

List of analyses

"Classic" launch

1. *Mapping*: bwa aln/sampe, bwa mem, bowtie2, ...
2. *Cleaning mapping*: samtools, picard-tools,...
3. *Realigning and Duplicates*: GATK, picard-tools,...
4. *SNP calling and Cleaning*: GATK,...

List of analyses

"Classic" launch

1. *Mapping*: bwa aln/sampe, bwa mem, bowtie2, ...
2. *Cleaning mapping*: samtools, picard-tools,...
3. *Realigning and Duplicates*: GATK, picard-tools,...
4. *SNP calling and Cleaning*: GATK,...

Between 8 and 15 different commands...

Be “lazy” but clever

Problems with manual launches

- Long
- Fastidious
- Error prone
- Tracability and reproducibilty not ensured

Be “lazy” but clever

Problems with manual launches

- Long
- Fastidious
- Error prone
- Tracability and reproducibilty not ensured

Solution $\Longrightarrow$ Workflow Manager

SouthGreen bioinformatics platform

TOGGLe

Tools for Generic NGS analysis

A framework to quickly build pipelines and to perform large-scale NGS analysis

Christine Tranchant-Dubreuil
christine.tranchant@ird.fr

Bioversity International · cirad · INRA SCIENCE & IMPACT · IRD Institut de Recherche pour le Développement FRANCE · Montpellier SupAgro

TOGGLe

- A toolbox to perform large-scale NGS analyses

19 modules, 120 functions
40 open-source tools

SouthGreen bioinformatics platform

TOGGLe

- A toolbox to perform large-scale NGS analyses

19 modules, 120 functions
40 open-source tools

- **Version 2 published in BMC bioinformatics**

RESEARCH

TOGGLE: Toolbox for generic NGS analyses

Cecile Monat[1*†], Christine Tranchant-Dubreuil[1†^], Ayité Kougbeadjo[2], Cédric Farcy[2], Mawussé Agbessi[1], Maryline Summo[2] and Francois Sabot[1^]

SouthGreen bioinformatics platform

Data preprocessing

Fastqc, Cutadapt
FastxTrimmer
Stack process_radstats

Structural Variations

MindTheGap,
BreakDancer, Pindel

RNA-seq Assembly

Trinity
TGI-CL

TOGGLE

ReadCount

Htseq-count

Mapping

Bwa aln, sampe/ samse
Bwa mem
Tophat2

SNP calling/ filtering

SAMtools, GATK,VarScan,SNPEff

SAM/BAM management

picardTools, SAMtools, GATK

SouthGreen bioinformatics platform

Data preprocessing

Fastqc, Cutadapt
FastxTrimmer
Stack process_radstats

Structural Variations

MindTheGap,
BreakDancer, Pindel

RNA-seq Assembly

TransAbyss, Trinity
TGI-CL

TOGGLE

ReadCount

Htseq-count

SNP calling/ filtering

SAMtools, GATK,VarScan,SNPEff

Mapping

Bwa aln, sampe/ samse
Bwa mem
Tophat2

SAM/BAM management

picardTools, SAMtools, GATK

https://github.com/SouthGreenPlatform/TOGGLE

TOGGLe

Version 2 → **Version 3**

From **hard-coded pipelines**
To **a bioinformatic pipeline framework**

SouthGreen bioinformatics platform

TOGGLe

Version 2 → **Version 3**

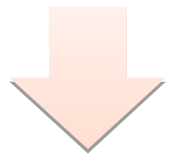
From **hard-coded pipelines**
To **a bioinformatic pipeline framework**

Biologists create their own pipeline through an easy and user-friendly approach

SouthGreen bioinformatics platform

How to perform an analysis with TOGGLE ?

A command-line based pipeline framework



A single command line

toggleGenerator.pl -d DIR-c FILE -o DIR

SouthGreen bioinformatics platform

What does TOGGLE need to run ?

- An input directory (with fastq, sam/bam, vcf files)
- The name of output directory used to store the data generated by the analyses
- A unique and simple configuration file to design the pipeline and define software parameters.
- Optional arguments : reference file, annotation…

SouthGreen bioinformatics platform

A simple configuration file

$order
1=fastqc
2=cutadapt
3=bwa mem
4=picardToolsSortSam
5=samToolsView
1000=gatkHaplotypeCaller
1001=gatkVariantFiltration

$cutadapt
-q 30
-m 35

$bwa mem
-n 5
…

$sge
-q bioinfo.q
-b Y

A simple configuration file

$order
1=fastqc
2=cutadapt
3=bwa mem
4=picardToolsSortSam
5=samToolsView
1000=gatkHaplotypeCaller
1001=gatkVariantFiltration

Create your own workflow

- The workflow order
- The list of softwares to run

One line = the step followed by the software's name

A simple configuration file

Create your own workflow

```
$order
1=fastqc
2=cutadapt
3=bwa mem
4=picardToolsSortSam
5=samToolsView
1000=gatkHaplotypeCaller
1001=gatkVariantFiltration
```

Step number < 1000

Step number >= 1000

A simple configuration file

$order
1=fastqc
2=cutadapt
3=bwa mem
4=picardToolsSortSam
5=samToolsView
1000=gatkHaplotypeCaller
1001=gatkVariantFiltration

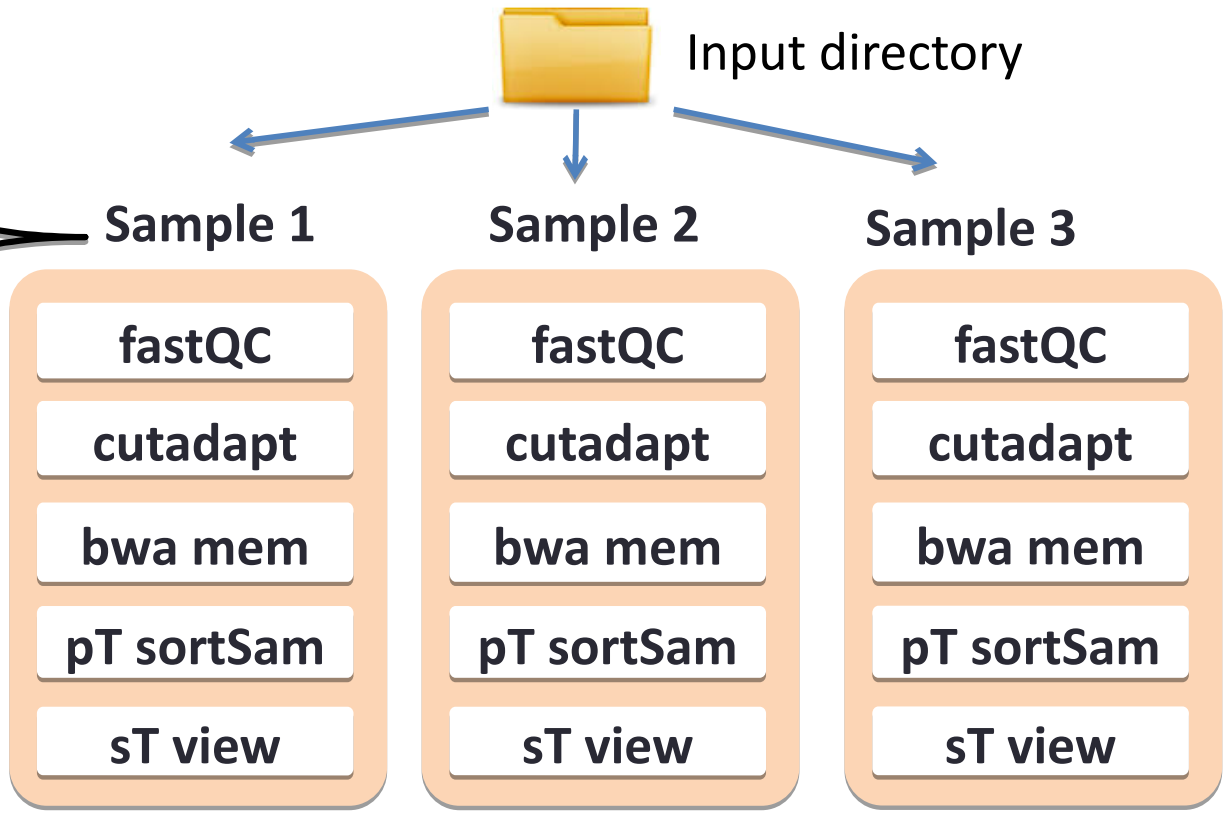
Create your own workflow

Step number < 1000

Parallel analysis by sample

Create your workflow

$order
1=fastqc
2=cutadapt
3=bwa mem
4=picardToolsSortSam
5=samToolsView
1000=gatkHaplotypeCaller
1001=gatkVariantFiltration

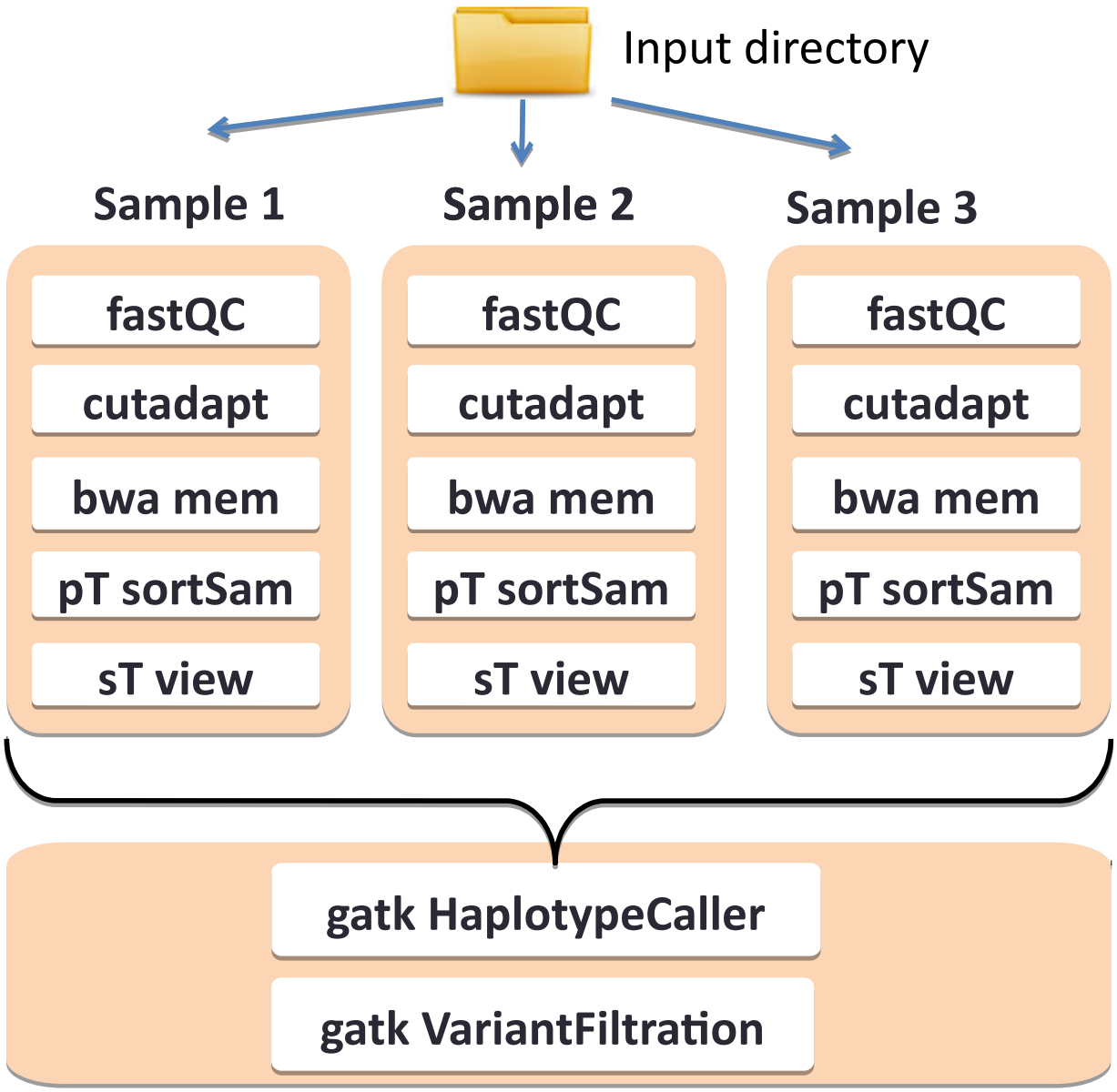


Create your workflow

$order
1=fastqc
2=cutadapt
3=bwa mem
4=picardToolsSortSam
5=samToolsView
1000=gatkHaplotypeCaller
1001=gatkVariantFiltration

Step number >= 1000

Global analysis (all samples)



A simple configuration file

$order
1=fastqc
2=cutadapt
3=bwa mem
4=picardToolsSortSam
5=samToolsView
1000=gatkHaplotypeCaller
1001=gatkVariantFiltration

$cutadapt
-q 30
-m 35

$bwa mem
-n 5
…

$sge
-q bioinfo.q
-b Y

Software parameters

One tag per software ($softwareName) followed by the list of options

Analyses *Mapping*

- Create a **TOGGLe** configuration file with as order:
  1. bwa aln
  2. bwa sampe
  3. picardtools sortsam
  4. samtools view
  5. samtools index
  6. gatkindelrealigner
  7. picardtools markduplicates

Analyses *Mapping*

- Create a **TOGGLe** configuration file with as order:
  1. bwa aln
  2. bwa sampe
  3. picardtools sortsam
  4. samtools view
  5. samtools index
  6. gatkindelrealigner
  7. picardtools markduplicates
- Add the options...

Analyses *Mapping*

- Create a **TOGGLe** configuration file with as order:
  1. bwa aln
  2. bwa sampe
  3. picardtools sortsam
  4. samtools view
  5. samtools index
  6. gatkindelrealigner
  7. picardtools markduplicates
- Add the options...
- Ok, we help... Look to configuration files in */data/FORMATION/2018/TPsnpSV*

Observing mapping

- Download Tablet (use Google and Tablet+NGS)

Observing mapping

- Download Tablet (use Google and Tablet+NGS)
- Transfer the BAMs and the reference from the node to the master then to your local computer (use scp at each point)

Observing mapping

- Download Tablet (use Google and Tablet+NGS)
- Transfer the BAMs and the reference from the node to the master then to your local computer (use scp at each point)
- Open Tablet, look at the mapping and try to find SNPs

Analyses *calling*

- Pick up all last BAM file (*MarkDuplicates* steps)

Analyses *calling*

- Pick up all last BAM file (*MarkDuplicates* steps)
- Launch a **TOGGLe** with *1000=gatkUnifiedGenotyper*

Analyses *calling*

- Pick up all last BAM file (*MarkDuplicates* steps)
- Launch a **TOGGLe** with *1000=gatkUnifiedGenotyper*
- Try to launch a **TOGGLe** from *FASTQ* to *VCF*

SAM file format for Sequence Alignment Map

SAM format : http://samtools.sourceforge.net/samtools.shtml

Type	Tag	Description
HD - header	VN*	File format version.
	SO	Sort order. Valid values are: *unsorted*, *queryname* or *coordinate*.
	GO	Group order (full sorting is not imposed in a group). Valid values are: *none*, *query* or *reference*.
SQ - Sequence dictionary	SN*	Sequence name. Unique among all sequence records in the file. The value of this field is used in alignment records.
	LN*	Sequence length.
	AS	Genome assembly identifier. Refers to the reference genome assembly in an unambiguous form. Example: HG18.
	M5	MD5 checksum of the sequence in the uppercase (gaps and space are removed)
	UR	URI of the sequence
	SP	Species.
RG - read group	ID*	Unique read group identifier. The value of the ID field is used in the RG tags of alignment records.
	SM*	Sample (use pool name where a pool is being sequenced)
	LB	Library
	DS	Description
	PU	Platform unit (e.g. lane for Illumina or slide for SOLiD); should be a full, unambiguous identifier
	PI	Predicted median insert size (maybe different from the actual median insert size)
	CN	Name of sequencing center producing the read.
	DT	Date the run was produced (ISO 8601 date or date/time).
	PL	Platform/technology used to produce the read.
PG - Program	ID*	Program name
	VN	Program version
	CL	Command line
CO - comment		One-line text comments

SAM file format for Sequence Alignment Map

SAM format : http://samtools.sourceforge.net/samtools.shtml

Type	Tag	Description
HD - header	VN*	File format version.
	SO	Sort order. Valid values are: *unsorted*, *queryname* or *coordinate*.
	GO	Group order (full sorting is not imposed in a group). Valid values are: *none*, *query* or *reference*.
SQ - Sequence dictionary	SN*	Sequence name. Unique among all sequence records in the file. The value of this field is used in alignment records.
	LN*	Sequence length.
	AS	Genome assembly identifier. Refers to the reference genome assembly in an unambiguous form. Example: HG18.
	M5	MD5 checksum of the sequence in the uppercase (gaps and space are removed)
	UR	URI of the sequence
	SP	Species.
RG - read group	ID*	Unique read group identifier. The value of the ID field is used in the RG tags of alignment records.
	SM*	Sample (use pool name where a pool is being sequenced)
	LB	Library
	DS	Description
	PU	Platform unit (e.g. lane for Illumina or slide for SOLiD); should be a full, unambiguous identifier
	PI	Predicted me
	CN	Name of seq
	DT	Date the run
	PL	Platform/tech
PG - Program	ID*	Program nar
	VN	Program ver
	CL	Command li
CO - comment		One-line text

```
@HD VN:1.3 SO:coordinate
@SQ SN:ref LN:45
r001 163 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002   0 ref  9 30 3S6M1P1I4M * 0   0 AAAAGATAAGGATA     *
r003   0 ref  9 30 5H6M       * 0   0 AGCTAA    *   NM:i:1
r004   0 ref 16 30 6M14N5M    * 0   0 ATAGCTTCAGC       *
r003  16 ref 29 30 6H5M       * 0   0 TAGGC     *   NM:i:0
r001  83 ref 37 30 9M         = 7 -39 CAGCGCCAT         *
```

SAM file format for Sequence Alignment Map

SAM format : http://samtools.sourceforge.net/samtools.shtml

Col	Name	Description
1	QNAME	Query NAME of the read or the read pair
2	FLAG	bitwise FLAG (pairing, strand, mate strand, etc.)
3	RNAME	Reference sequence NAME
4	POS	1-based leftmost POSition of clipped alignment
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIGAR	extended CIGAR string (operations: MIDNSHP)
7	NRNM	Mate Reference NaMe (`=' if same as RNAME)
8	MPOS	1-based leftmost Mate POSition
9	ISIZE	inferred Insert SIZE
10	SEQ	query SEQuence on the reference
11	QUAL	query QUALity (ASCII-33

```
@HD VN:1.3 SO:coordinate
@SQ SN:ref LN:45
r001 163 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002   0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003   0 ref  9 30 5H6M       *  0   0 AGCTAA   *   NM:i:1
r004   0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003  16 ref 29 30 6H5M       *  0   0 TAGGC    *   NM:i:0
r001  83 ref 37 30 9M         =  7 -39 CAGCGCCAT         *
```

SAM file format for Sequence Alignment Map

SAM format : http://samtools.sourceforge.net/samtools.shtml

Suppose we have the following alignment with bases in lower cases clipped from the alignment. Read r001/1 and r001/2 constitute a read pair; r003 is a chimeric read; r004 represents a split alignment.

```
Coor     12345678901234  56789012345678901234567890123456789012345
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

+r001/1        TTAGATAAAGGATA*CTG
+r002         aaaAGATAA*GGATA
+r003       gcctaAGCTAA
+r004                     ATAGCT..............TCAGC
-r003                           ttagctTAGGC
-r001/2                                        CAGCGCCAT
```

SAM file format for Sequence Alignment Map

SAM format : http://samtools.sourceforge.net/samtools.shtml

Suppose we have the following alignment with bases in lower cases clipped from the alignment. Read r001/1 and r001/2 constitute a read pair; r003 is a chimeric read; r004 represents a split alignment.

```
Coor     12345678901234  56789012345678901234567890123456
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

+r001/1        TTAGATAAAGGATA*CTG
+r002         aaaAGATAA*GGATA
+r003       gcctaAGCTAA
+r004                     ATAGCT..............TCAGC
-r003                           ttagctTAGGC
-r001/2                                        CAGCGCCAT
```

The corresponding SAM format is:

```
@HD VN:1.3 SO:coordinate
@SQ SN:ref LN:45
r001 163 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002   0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003   0 ref  9 30 5H6M       *  0   0 AGCTAA   *   NM:i:1
r004   0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003  16 ref 29 30 6H5M       *  0   0 TAGGC    *   NM:i:0
r001  83 ref 37 30 9M         =  7 -39 CAGCGCCAT         *
```

the SAM bitwise FLAG

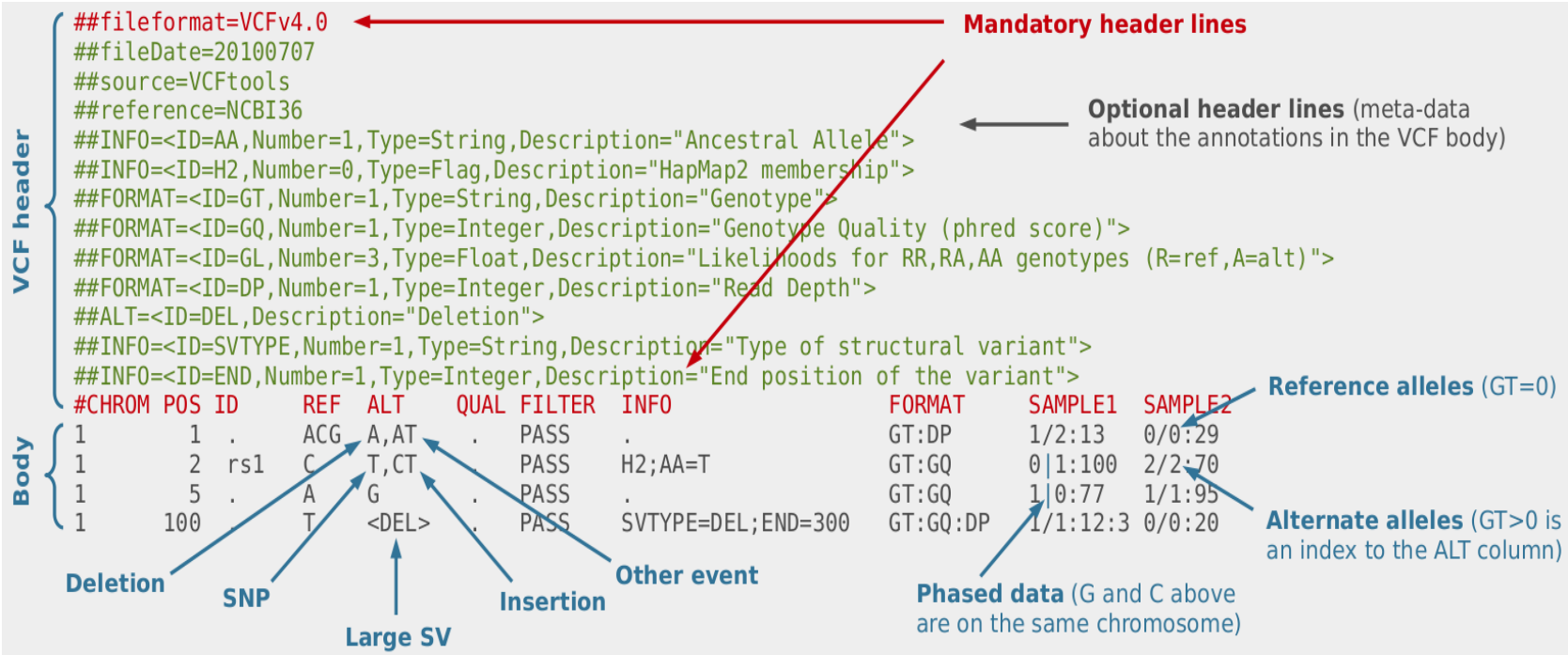
SAM format: FLAG field

numeric	binary	description
1	00000001	template has multiple fragments in sequencing
2	00000010	each fragment properly mapped according to aligner
4	00000100	fragment is unmapped
8	00001000	mate is unmapped
16	00010000	sequence is reverse complemented
32	00100000	sequence of mate is reversed
64	01000000	is first fragment in template
128	10000000	is second fragment in template

From J. Aerts, in SlideShare

VCF file format for Variant Call Format

The Variant Call Format (VCF) used in bioinformatics for storing gene sequence variations



VCF file format for Variant Call Format

```
##fileformat=VCFv4.1
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=file:///seq/references/1000GenomesPilot-NCBI36.fasta
##contig=<ID=20,length=62435964,assembly=B36,md5=f126cdf8a6e0c7f379d618ff66beb2da,species="Homo sapiens",taxonomy=x>
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS     ID        REF    ALT     QUAL FILTER INFO                              FORMAT      NA00001        NA00002        NA00003
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 1|0:48:8:51,51 1/1:43:5:.,.
20     17330   .         T      A       3    q10    NS=3;DP=11;AF=0.017               GT:GQ:DP:HQ 0|0:49:3:58,50 0|1:3:5:65,3   0/0:41:3
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 2|1:2:0:18,2   2/2:35:4
20     1230237 .         T      .       47   PASS   NS=3;DP=13;AA=T                   GT:GQ:DP:HQ 0|0:54:7:56,60 0|0:48:4:51,51 0/0:61:2
20     1234567 microsat1 GTC    G,GTCT  50   PASS   NS=3;DP=9;AA=G                    GT:GQ:DP    0/1:35:4       0/2:17:2       1/1:40:3''
```

- **Variation 1 :** a good SNP
- **Variation 2 :** a possible SNP that has been filtered out because its quality is below 10
- **Variation 3 :** a site at which two alternate alleles are called, with one of them (T) being ancestral (possibly a reference sequencing error)
- **Variation 4 :** a site that is called monomorphic reference (i.e. with no alternate alleles)
- **Variation 5 :** a microsatellite with two alternative alleles, one a deletion of 2 bases (TC), and the other an insertion of one base (T).

Observing calling

- Open Tablet, load BAM, reference and VCF (pick it on your local computer)

Observing calling

- Open Tablet, load BAM, reference and VCF (pick it on your local computer)
- Look for SNP and their mapping

Observing calling

- Open Tablet, load BAM, reference and VCF (pick it on your local computer)
- Look for SNP and their mapping
- Relaunch the same analyses removing one step or changing parameters...

Basics filtering

- Using GATK Variant Filtration, a flag per filter

Basics filtering

- Using GATK Variant Filtration, a flag per filter
- Depth filter:
  DP<10 or DP>20000

Basics filtering

- Using GATK Variant Filtration, a flag per filter
- Depth filter:
  DP<10 or DP>20000
- MQ0 filter:
  MQ0 < 4 or MQ0 < 0.1 DP

Basics filtering

- Using GATK Variant Filtration, a flag per filter
- Depth filter:
  DP<10 or DP>20000
- MQ0 filter:
  MQ0 < 4 or MQ0 < 0.1 DP
- QUAL filter:
  QUAL < 200

Basics filtering

- Using GATK Variant Filtration, a flag per filter
- Depth filter:
  DP<10 or DP>20000
- MQ0 filter:
  MQ0 < 4 or MQ0 < 0.1 DP
- QUAL filter:
  QUAL < 200
- SNPcluster filter:
  more than 3 SNP per 10b

Basics filtering

The command is then...

```
java -jar GenomeAnalysisTK.jar -T VariantFiltration
-R REFERENCE.FASTA -V INPUT.VCF --filterExpression
'QUAL<200' --filterName 'LOW-QUAL' --filterExpression
'MQ0>=4 && ((MQ0/(1.0DP)) > 0.1)' --filterName
'LOW-MQ0' --filterExpression 'DP<10' --filterName
'LOWDP' --clusterSize 3 --clusterWindowSize 10 --
filterExpression 'DP>20000' --filterName 'HIGH-DP' -o
OUTPUT.VCF
```

More filtering

- Recoding with SNP PASS:

```
vcftools --vcf FILEIN --remove-filtered-all
--recode --recode-INFO-all --out FILEOUT
```

More filtering

- Recoding with SNP PASS:

```
vcftools --vcf FILEIN --remove-filtered-all
--recode --recode-INFO-all --out FILEOUT
```

- Missing data filtering:

```
vcftools --vcf FILEIN --max-missing-count 12
--recode --recode-INFO-all --out FILEOUT
```

More filtering

- Recoding with SNP PASS:

```
vcftools --vcf FILEIN --remove-filtered-all
--recode --recode-INFO-all --out FILEOUT
```

- Missing data filtering:

```
vcftools --vcf FILEIN --max-missing-count 12
--recode --recode-INFO-all --out FILEOUT
```

- MAF:

```
vcftools --vcf FILEIN --output FILEOUT --maf 0.1
```

More filtering

- Recoding with SNP PASS:

```
vcftools --vcf FILEIN --remove-filtered-all
--recode --recode-INFO-all --out FILEOUT
```

- Missing data filtering:

```
vcftools --vcf FILEIN --max-missing-count 12
--recode --recode-INFO-all --out FILEOUT
```

- MAF:

```
vcftools --vcf FILEIN --output FILEOUT --maf 0.1
```

- Biallelic SNPs:

```
vcftools --vcf FILEIN --min-alleles 2
--max-alleles 2 --recode --recode-INFO-all --out
FILEOUT
```

- *sNMF*: toll to estimate ancestry coefficients

- *sNMF*: toll to estimate ancestry coefficients
- Developed by E. Frichot and O. Francois, TIMC-IMAG

- *sNMF*: toll to estimate ancestry coefficients
- Developed by E. Frichot and O. Francois, TIMC-IMAG
- R and Command-line version

Basics Population Genomics using *sNMF*

- *sNMF*: toll to estimate ancestry coefficients
- Developed by E. Frichot and O. Francois, TIMC-IMAG
- R and Command-line version
- Much faster than ADMIXTURE or STRUCTURE, as efficient

In your QRSH session, copy the files from /data/FORMATION/2018/TpPop in your /scratch folder, and follow these commands:

1. `scp master:/data/FORMATION/2018/TpPop/sample.vcf .`

In your QRSH session, copy the files from /data/FORMATION/2018/TpPop in your /scratch folder, and follow these commands:

1. `scp master:/data/FORMATION/2018/TpPop/sample.vcf .`
2. `grep "#" sample.vcf > subsamples.vcf`

In your QRSH session, copy the files from /data/FORMATION/2018/TpPop in your /scratch folder, and follow these commands:

1. `scp master:/data/FORMATION/2018/TpPop/sample.vcf .`
2. `grep "#" sample.vcf > subsamples.vcf`
3. `shuf -n 5000 sample.vcf | grep -v "#" > tmp.vcf`

In your QRSH session, copy the files from /data/FORMATION/2018/TpPop in your /scratch folder, and follow these commands:

1. `scp master:/data/FORMATION/2018/TpPop/sample.vcf .`
2. `grep "#" sample.vcf > subsamples.vcf`
3. `shuf -n 5000 sample.vcf | grep -v "#" > tmp.vcf`
4. `cat tmp.vcf » subsamples.vcf`

The VCF is now subsmapled and ready for pop analysis

1. `vcf2geno subsamples.vcf test.geno`

The VCF is now subsmapled and ready for pop analysis

1. `vcf2geno subsamples.vcf test.geno`
2. `for K in {1..10}; do echo K=$K; sNMF -x test.geno -K $K -c > test.$K.log; done`

The VCF is now subsmapled and ready for pop analysis

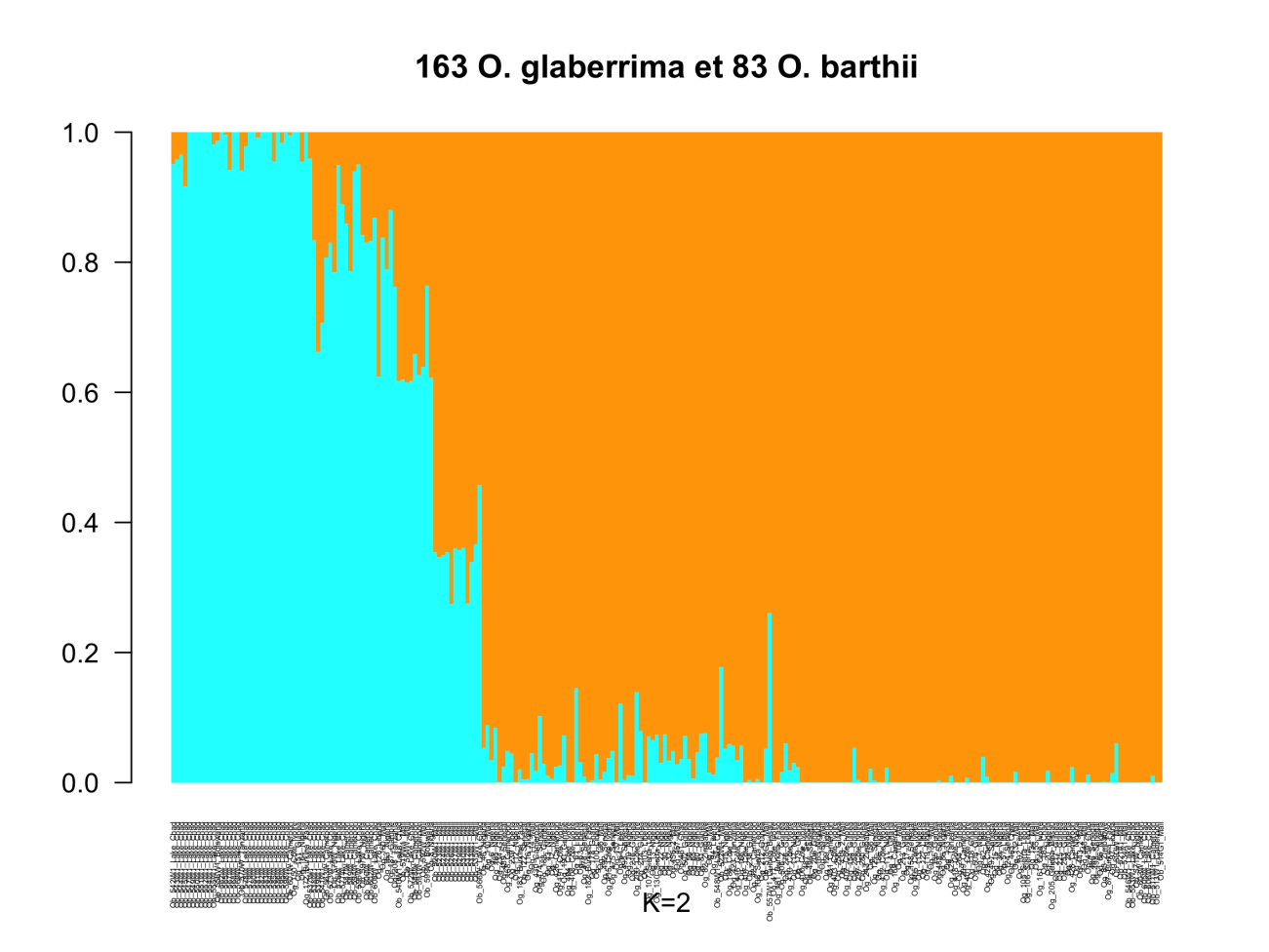
1. `vcf2geno subsamples.vcf test.geno`
2. `for K in {1..10}; do echo K=$K; sNMF -x test.geno -K $K -c > test.$K.log; done`
3. `grep "Cross-Entropy (masked data):" test.*.log`

The VCF is now subsmapled and ready for pop analysis

1. `vcf2geno subsamples.vcf test.geno`
2. `for K in {1..10}; do echo K=$K; sNMF -x test.geno -K $K -c > test.$K.log; done`
3. `grep "Cross-Entropy (masked data):" test.*.log`
4. Look at all files for the best Cross-Entropy value

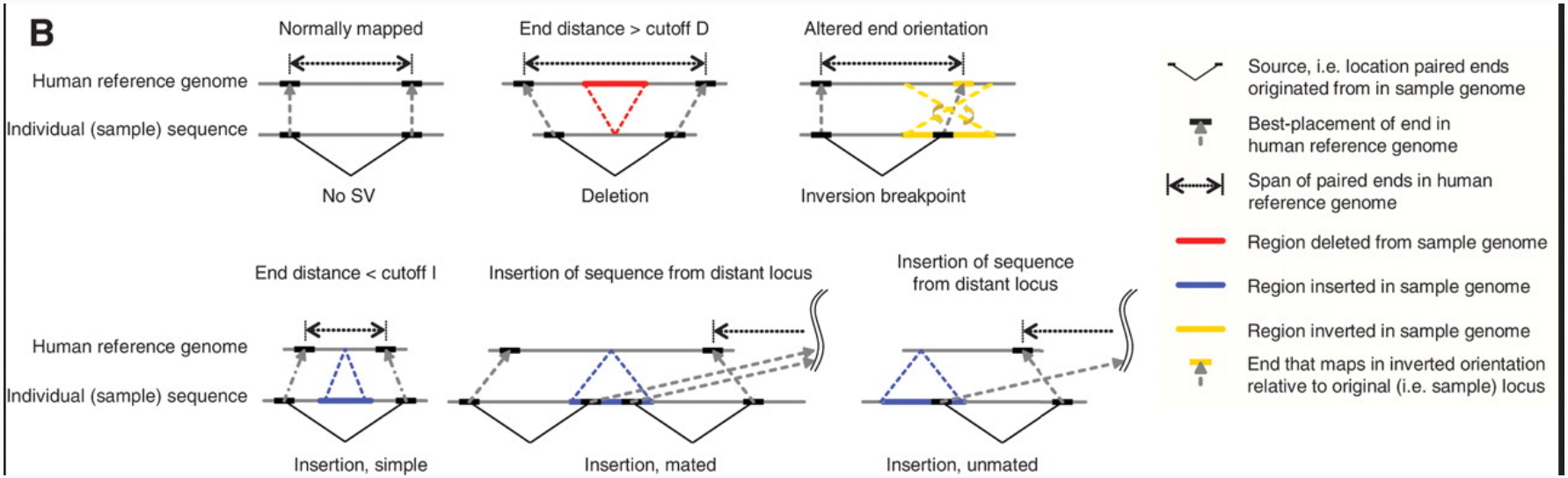
Use this with e.g. CLUMPP to extract the figure

Example in Global Genotyping & Population Genomics



From Cubry et al, 2018

Structural Variations



From Korbel et al, 2007

Structural Variations, a tool among others

We can use **BreakDancer** (*Chen et al, 2009, Nature Methods*)

BE CAREFUL: if you do it, do it through QRSH and scp copy on /scratch!!

- From the raw BAMs, launch:
  */usr/local/breakdancer-1.1.2/perl/bam2cfg.pl *.bam > bamConf.cfg*

Structural Variations, a tool among others

We can use **BreakDancer** (*Chen et al, 2009, Nature Methods*)

BE CAREFUL: if you do it, do it through QRSH and scp copy on /scratch!!

- From the raw BAMs, launch:
  */usr/local/breakdancer-1.1.2/perl/bam2cfg.pl *.bam > bamConf.cfg*
- Launch the **BreakDancer** command:
  /usr/local/breakdancer-1.1.2/cpp/breakdancer-max bamConf.cfg > breakDancer.out

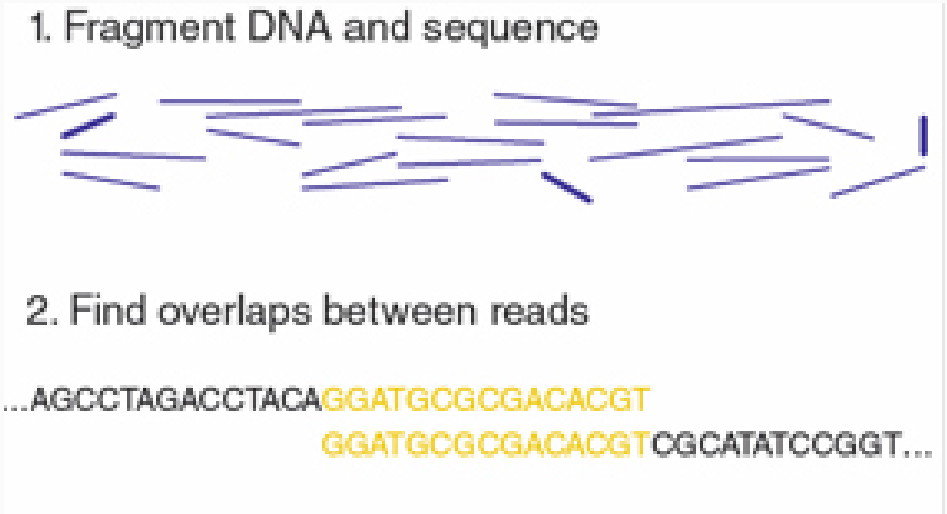
Structural Variations, a tool among others

We can use **BreakDancer** (*Chen et al, 2009, Nature Methods*)

BE CAREFUL: if you do it, do it through QRSH and scp copy on /scratch!!

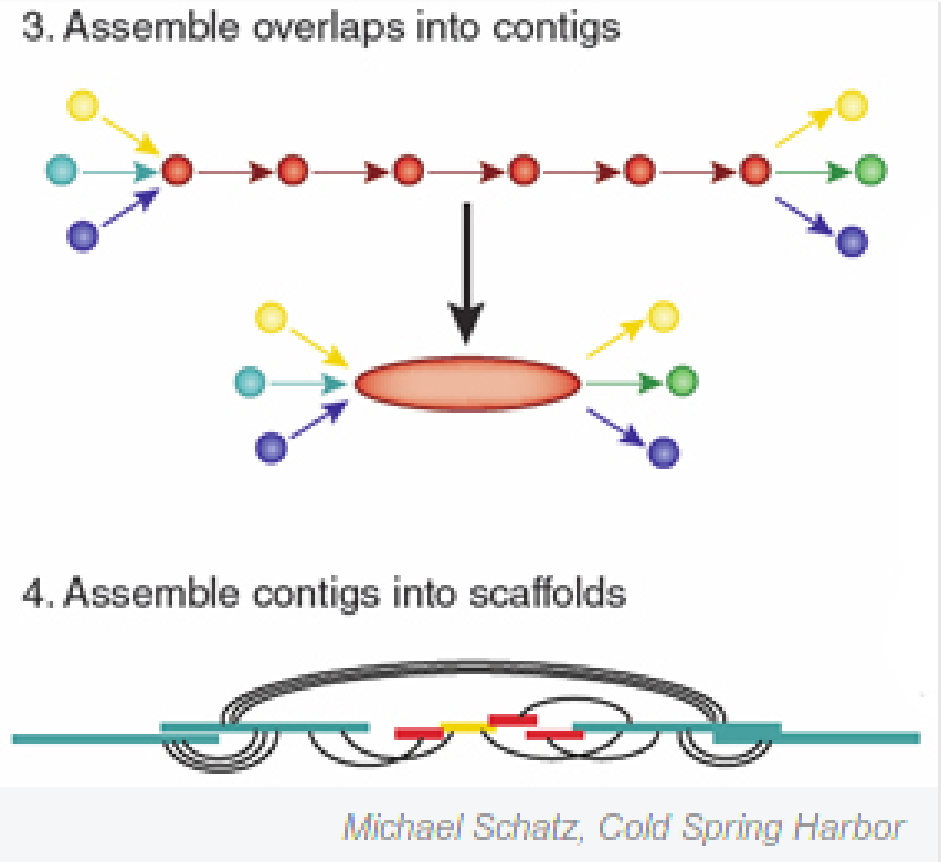
- From the raw BAMs, launch:
  */usr/local/breakdancer-1.1.2/perl/bam2cfg.pl *.bam > bamConf.cfg*
- Launch the **BreakDancer** command:
  /usr/local/breakdancer-1.1.2/cpp/breakdancer-max bamConf.cfg > breakDancer.out
- Check the results...

Assembly



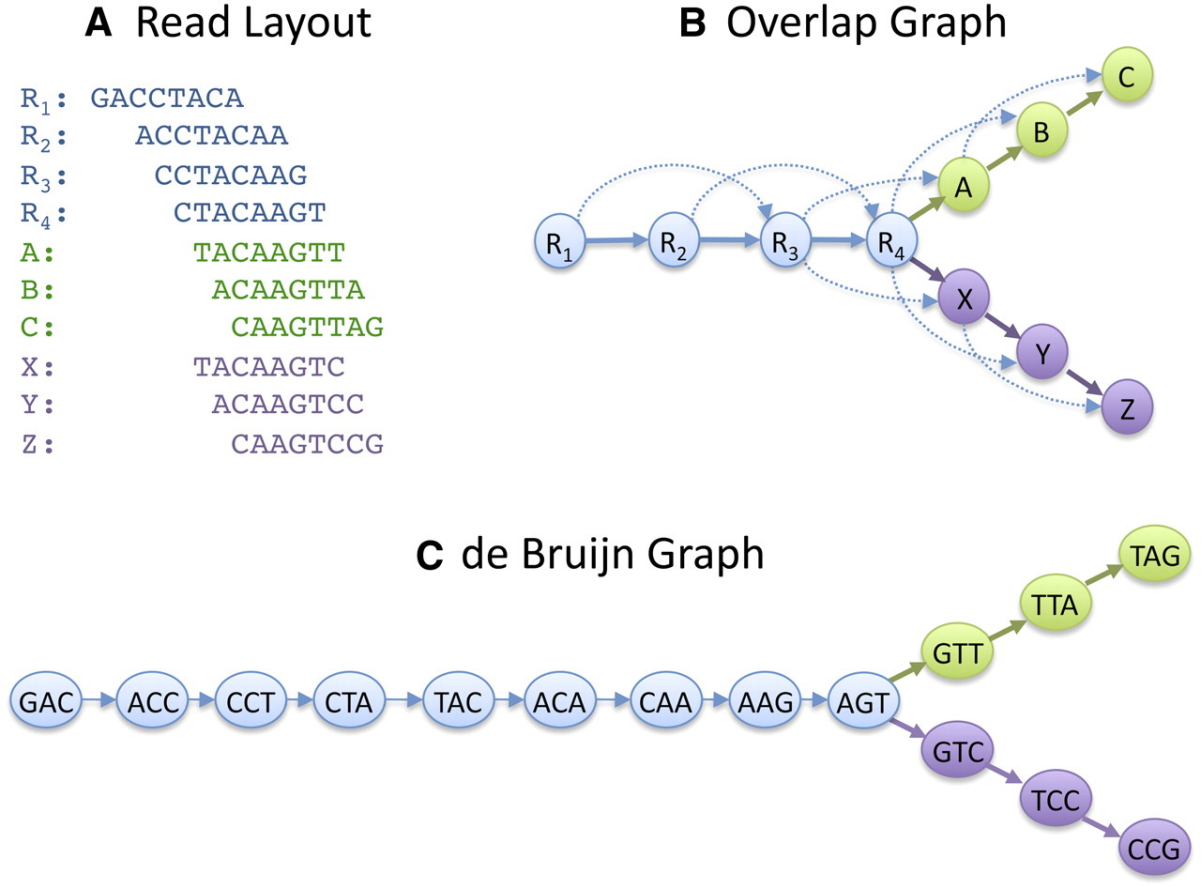
From Baker, 2012

Assembly



From Baker, 2012

Assembly



From Schatz, 2010

SV using Assembly, for small genomes

1. Download using ***scp*** the folder ***TPassembly*** in ***/data/FORMATION/2018/TPassembly***

SV using Assembly, for small genomes

1. Download using ***scp*** the folder ***TPassembly*** in ***/data/FORMATION/2018/TPassembly***
2. Go in this folder and check the data in the ***Ebola*** folder. What are they ?

SV using Assembly, for small genomes

1. Download using ***scp*** the folder ***TPassembly*** in ***/data/FORMATION/2018/TPassembly***
2. Go in this folder and check the data in the ***Ebola*** folder. What are they ?
3. Check them using ***FASTQC***

SV using Assembly, for small genomes

1. Download using ***scp*** the folder ***TPassembly*** in ***/data/FORMATION/2018/TPassembly***
2. Go in this folder and check the data in the ***Ebola*** folder. What are they ?
3. Check them using ***FASTQC***
4. Use the ***Abyss*** tool to assemble the pair-end data

SV using Assembly, for small genomes

1. Download using ***scp*** the folder ***TPassembly*** in ***/data/FORMATION/2018/TPassembly***
2. Go in this folder and check the data in the ***Ebola*** folder. What are they ?
3. Check them using ***FASTQC***
4. Use the ***Abyss*** tool to assemble the pair-end data
5. Use the ***BLAST*** tool to compare it to the ***ebola.fasta*** file as database

SV using Assembly, for small genomes

1. Download using ***scp*** the folder ***TPassembly*** in ***/data/FORMATION/2018/TPassembly***
2. Go in this folder and check the data in the ***Ebola*** folder. What are they ?
3. Check them using ***FASTQC***
4. Use the ***Abyss*** tool to assemble the pair-end data
5. Use the ***BLAST*** tool to compare it to the ***ebola.fasta*** file as database
6. Use the ***MUSCLE*** tool to align your assembly to the ***ebola.fasta***

Thanks for your attention