Variant calling in targeted sequencing

Eva Budinská
autumn 2023

Aim of variant calling

- - To define genomic positions with single nucleotide polymorphisms, deletions, insertions or long repetitive sequence insertions specific for the sample

  - and if of interest, consequently **call a genotype** (identify whether the sample is heterozygote or homozygote) for the allele at the particular position or **haplotype.**

Variant, genotype and haplotype calling

- **Variant calling –** identifies variable sites in genome
- **Genotype calling** – determines the genotype for each individual at each site (0/0, 1/1 – homozygote, 0/1 - heterozygote)
- **Haplotype calling** – determines the haplotype for each individual at each site

Site 1 **Reference: A** **Variants: G, C**	**Site 2** **Reference: C** **Variants: T, TT**	**Site 3** **Reference: T** **Variants: A**	**Site 4** **Reference: A** **Variants: T, G**	
A/G	C/T	T/T	A/T	**Genotype of an individual**
A	T	T	A	**Haplotype 1**
G	C	T	T	**Haplotype 2**

Typical applications

- **Mendelian disorders** – identification of causative genes in single gene disorders (germline mutations)

  Complex diseases – identification of candidate genes in complex diseases for further functional studies

  Somatic mutations – identification of constitutional mutations as well as driver and passenger genes in cancer

…

Whole genome/exome sequencing, targeted sequencing

Results of variant calling

Results of variant calling algorithms are presented in standardized

VCF file (**v**ariant **c**alling **f**ormat).

Variant call format (.vcf) and its binary form (.bcf)

Standard format file for results of variant calling

Developed in 2010 by 1000 genomes project (1000 Genomes | A Deep Catalog of Human Genetic Variation (internationalgenome.org))

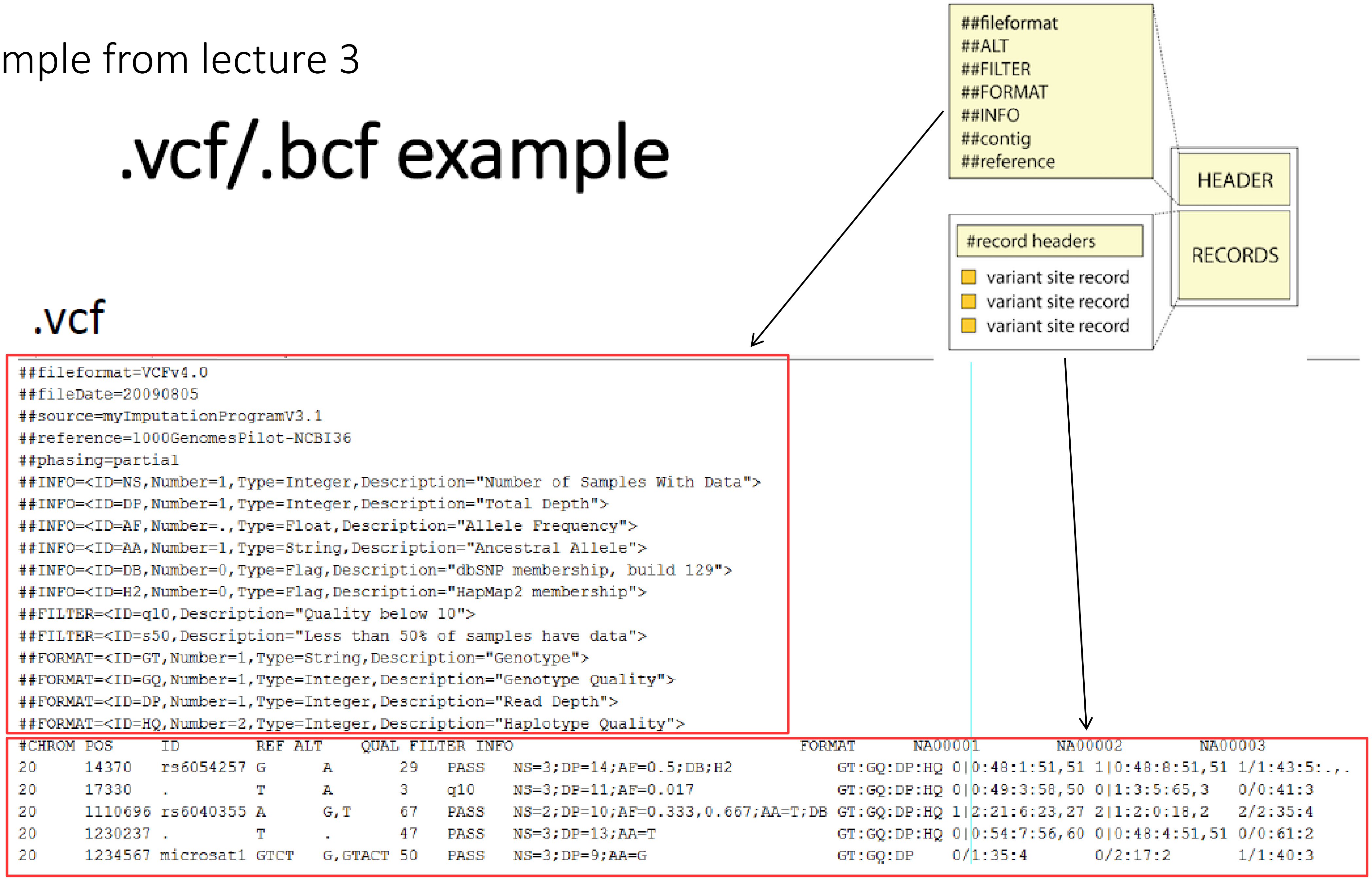
Current release: v4.3. (November 2022) - VCFv4.3.pdf (samtools.github.io)

.vcf/.bcf

...format example from lecture 3

.vcf/.bcf example

.vcf



##fileformat
##ALT
##FILTER
##FORMAT
##INFO
##contig
##reference

HEADER

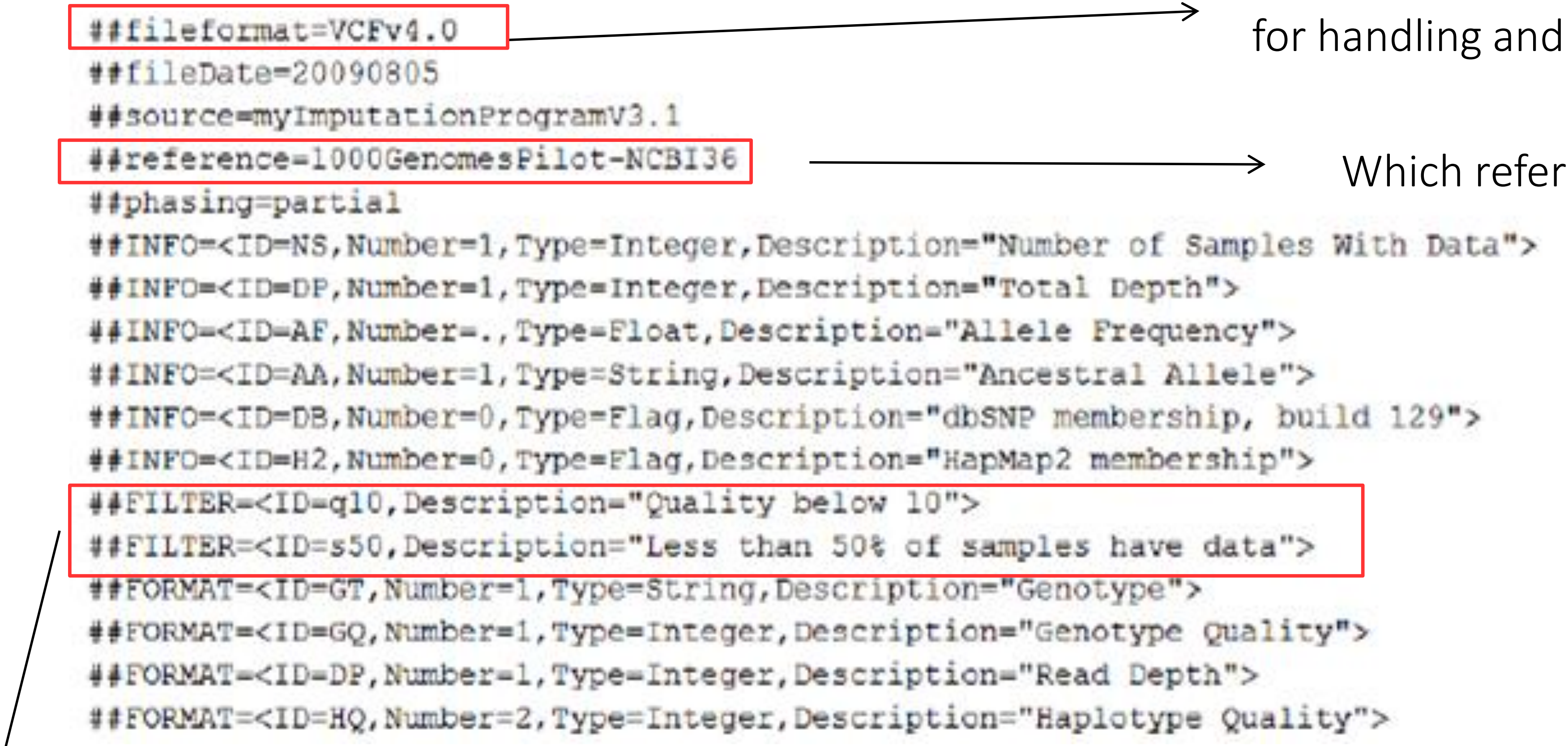
#record headers

□ variant site record
□ variant site record
□ variant site record

RECORDS

```
##fileformat=VCFv4.0
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=1000GenomesPilot-NCBI36
##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=.,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 GTCT   G,GTACT 50   PASS   NS=3;DP=9;AA=G                    GT:GQ:DP    0/1:35:4       0/2:17:2       1/1:40:3
```

Interpreting the file header



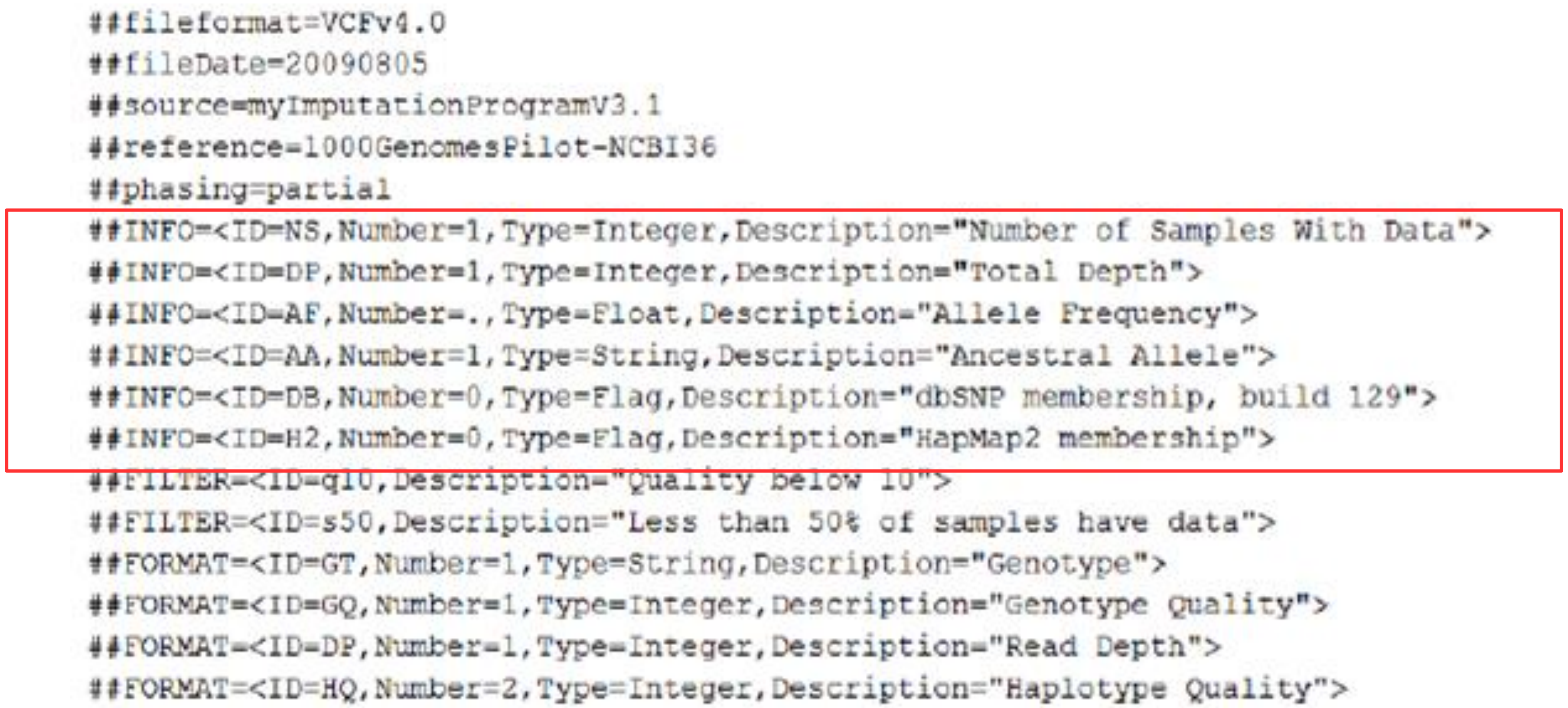
```
##fileformat=VCFv4.0
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=1000GenomesPilot-NCBI36
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=.,Type=Float,Description="Allele 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">
```

Version of the VCF specification to which the file conforms (important for handling and interpreting files!)

Which reference file was used

The filter lines – what type of filters have been applied to the data.

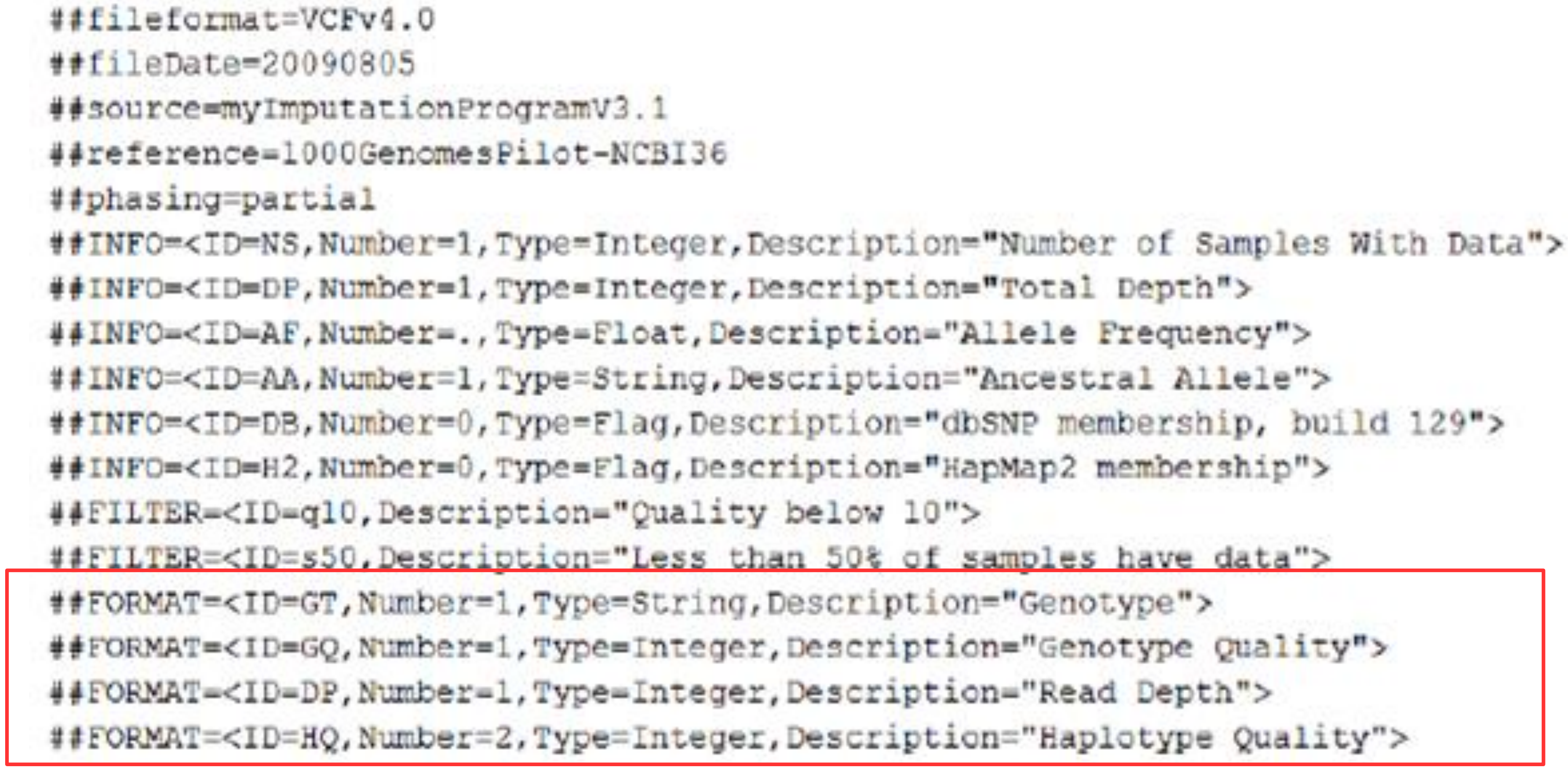
Interpreting the file header - INFO



```
##fileformat=VCFv4.0
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=1000GenomesPilot-NCBI36
##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=.,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">
```

INFO lines define **shortcuts for various site-level annotations**.
These are then later used in the INFO field in the variant specification.
Beware, **the definitions may differ** between tools generating vcf files...

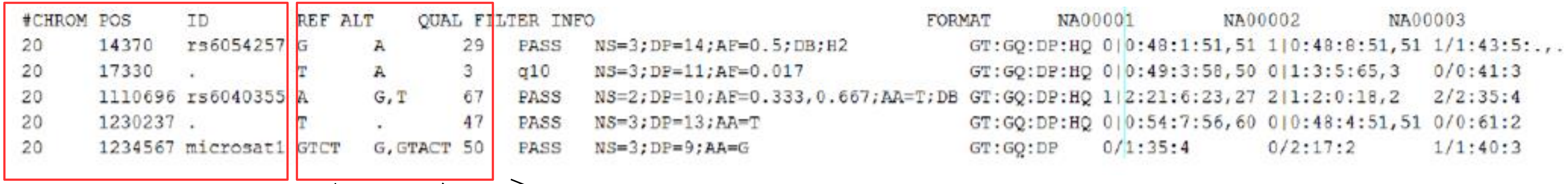
Interpreting the file header - FORMAT



```
##fileformat=VCFv4.0
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=1000GenomesPilot-NCBI36
##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=.,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">
```

FORMAT lines define **how** the genotype and other sample-level **information** is **represented**.
Beware, the definitions may differ between tools generating vcf files...

Interpreting the records:



```
#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 GTCT   G,GTACT 50   PASS   NS=3;DP=9;AA=G                    GT:GQ:DP    0/1:35:4       0/2:17:2       1/1:40:3
```

Reference sequence at the position

Alternative sequence(s) at the position

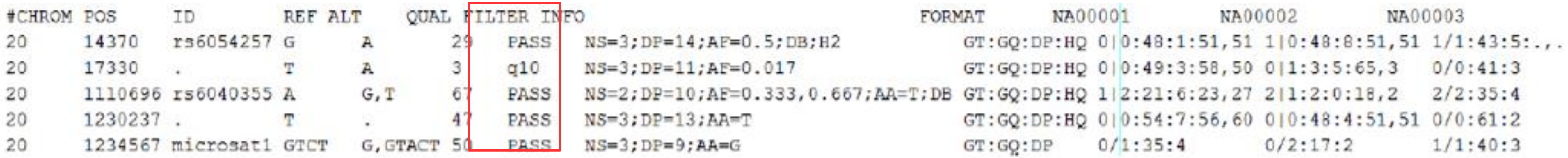
Quality: The Phred-scaled probability that REF/ALT polymorphism exists at this site given sequencing data.

Because the Phred scale is -10 * log(1-p), a value of 10 indicates a 1 in 10 chance of error, while a 100 indicates a 1 in 10^10 chance of error.

This number can grow very large when a large amount of data is used for variant calling => not often a very useful property for evaluating the quality of a variant call.

Identifiers of the position and gene at which the variant was found

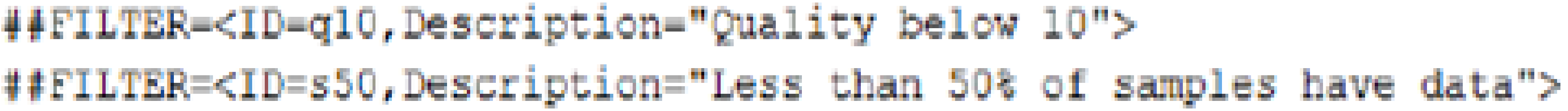
Interpreting the records: FILTER



```
#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 GTCT G,GTACT 50   PASS   NS=3;DP=9;AA=G                    GT:GQ:DP    0/1:35:4       0/2:17:2       1/1:40:3
```

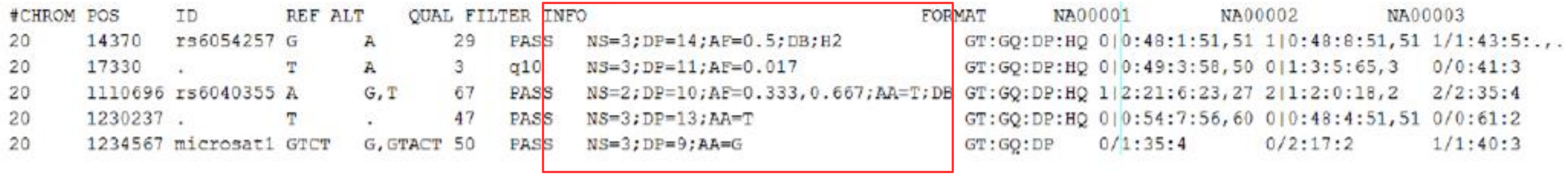
↓

Result of quality filters applied for filtering out low quality variant calls.
The values are defined in the header:



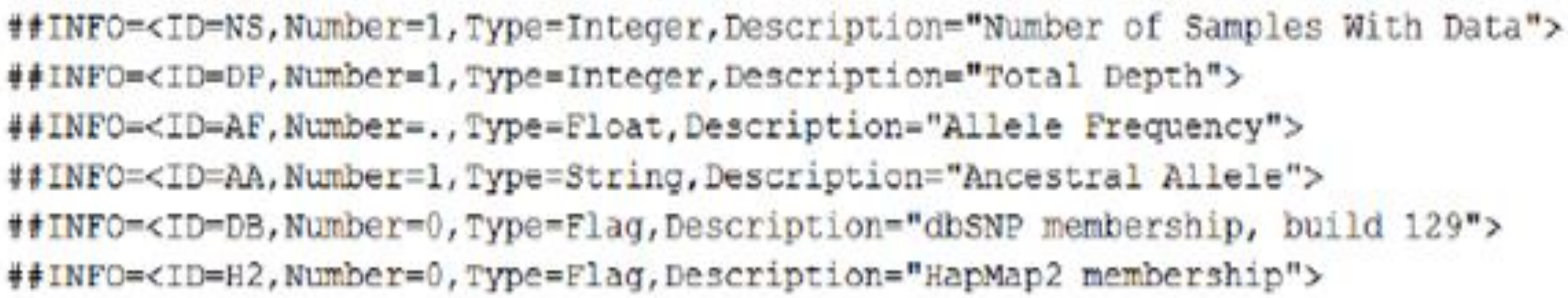
```
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
```

Let's get more in detail: INFO



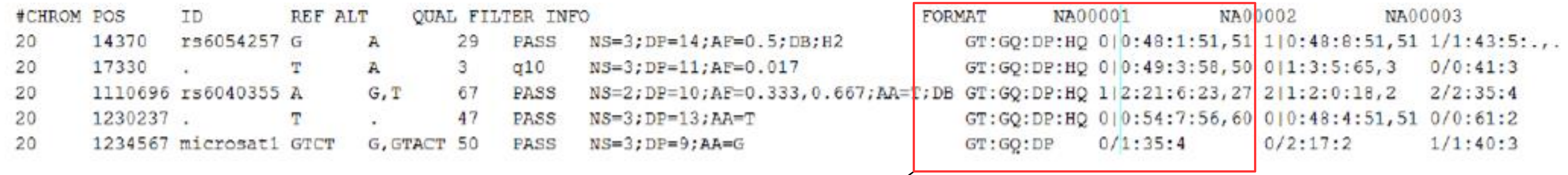
```
#CHROM POS     ID         REF ALT    QUAL FILTER INFO                              FORMAT      NA00001        NA00002        NA00003
20     14370   rs6054257 G      A       29   PASS   NS=3;DP=14;AF=0.5;DB;H2           GT:GQ:DP:HQ 0|0:48:1:51,51 1|0:48:8:51,51 1/1:43:5:.,.
20     17330   .         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 GTCT   G,GTACT 50   PASS   NS=3;DP=9;AA=G                    GT:GQ:DP    0/1:35:4       0/2:17:2       1/1:40:3
```

Various site-level annotations and their values, as defined in the INFO lines in the header.



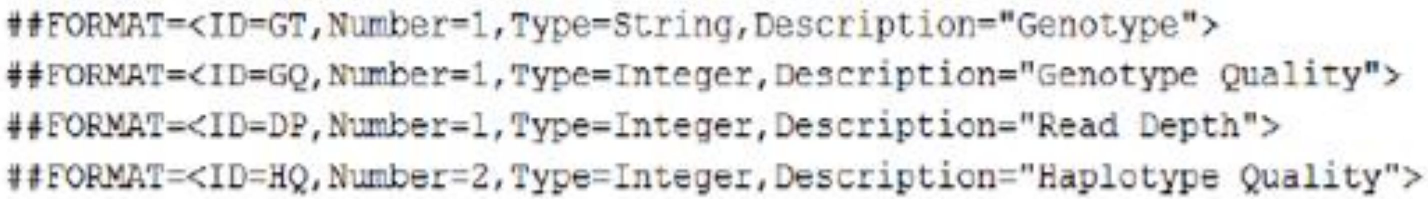
```
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=.,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
```

Let's get more in detail: FORMAT

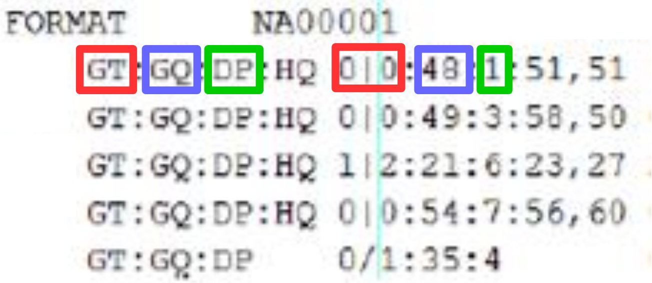


```
#CHROM POS     ID         REF  ALT     QUAL FILTER INFO                              FORMAT      NA00001        NA00002        NA00003
20     14370   rs6054257  G    A       29   PASS   NS=3;DP=14;AF=0.5;DB;H2           GT:GQ:DP:HQ 0|0:48:1:51,51 1|0:48:8:51,51 1/1:43:5:.,.
20     17330   .          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  GTCT G,GTACT 50   PASS   NS=3;DP=9;AA=G                    GT:GQ:DP    0/1:35:4       0/2:17:2       1/1:40:3
```

- How the genotype and other sample-level information is represented in the sample columns!



```
##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">
```



```
FORMAT      NA00001
GT:GQ:DP:HQ 0|0:48:1:51,51
GT:GQ:DP:HQ 0|0:49:3:58,50
GT:GQ:DP:HQ 1|2:21:6:23,27
GT:GQ:DP:HQ 0|0:54:7:56,60
GT:GQ:DP    0/1:35:4
```

Sample NA0001, first line variant:

- Genotype: 0|0 (homozygote for reference allele)
- Genotype quality: 48
- Read depth: 1
- Haplotype quality: 51 and 51

More info about vcf files

https://gatk.broadinstitute.org/hc/en-us/articles/360035531692-VCF-Variant-Call-Format

http://www.1000genomes.org/node/101

VCFv4.3.pdf (samtools.github.io)

Help files of the tools generating vcf files!

Variant identification

Tools can be (very roughly) divided based on what they can best do:

- **Germline callers** – central part of finding causes of rare inherited diseases
- **Somatic callers** – mainly cancer studies
- **CNV identification tools** – large structural modifications
- **SV (structural variants) identification tools** – inversions, translocations or large indels (insertions-deletions)

Variant calling – the basic steps after alignment

1. **The pileup step –** use SAMtools to generate counts of insertions, deletions and substitutions at each covered position in the BAM/SAM file (efficient in terms of time and memory)
2. **Estimation of background error rates (whatever the source)**
3. **Calling variants**
4. **Filtering variants**
5. **Calling genotypes**
6. **Calling haplotypes**

The variants

- Are **real mismatches** (from the perspective of alignment)

- We want to **keep these reads**

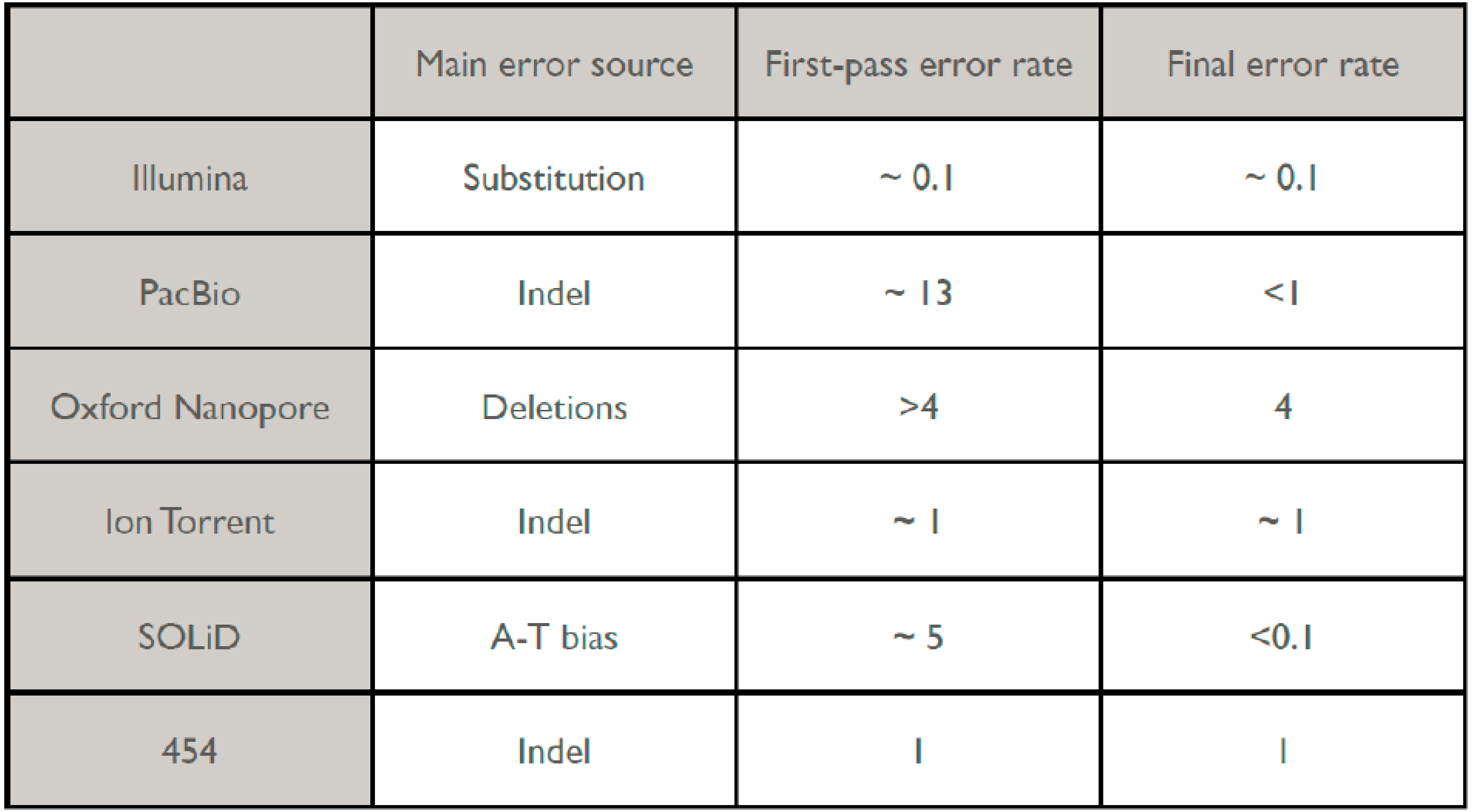
- Depending on goal of the study, the aligner must be able to handle them very properly (e.g. if we want to detect SNPs or SNVs)

Source of errors in variant calling

- **NGS data suffer** from **high error rates** that are due to multiple factors, including **base-calling** and **alignment errors**

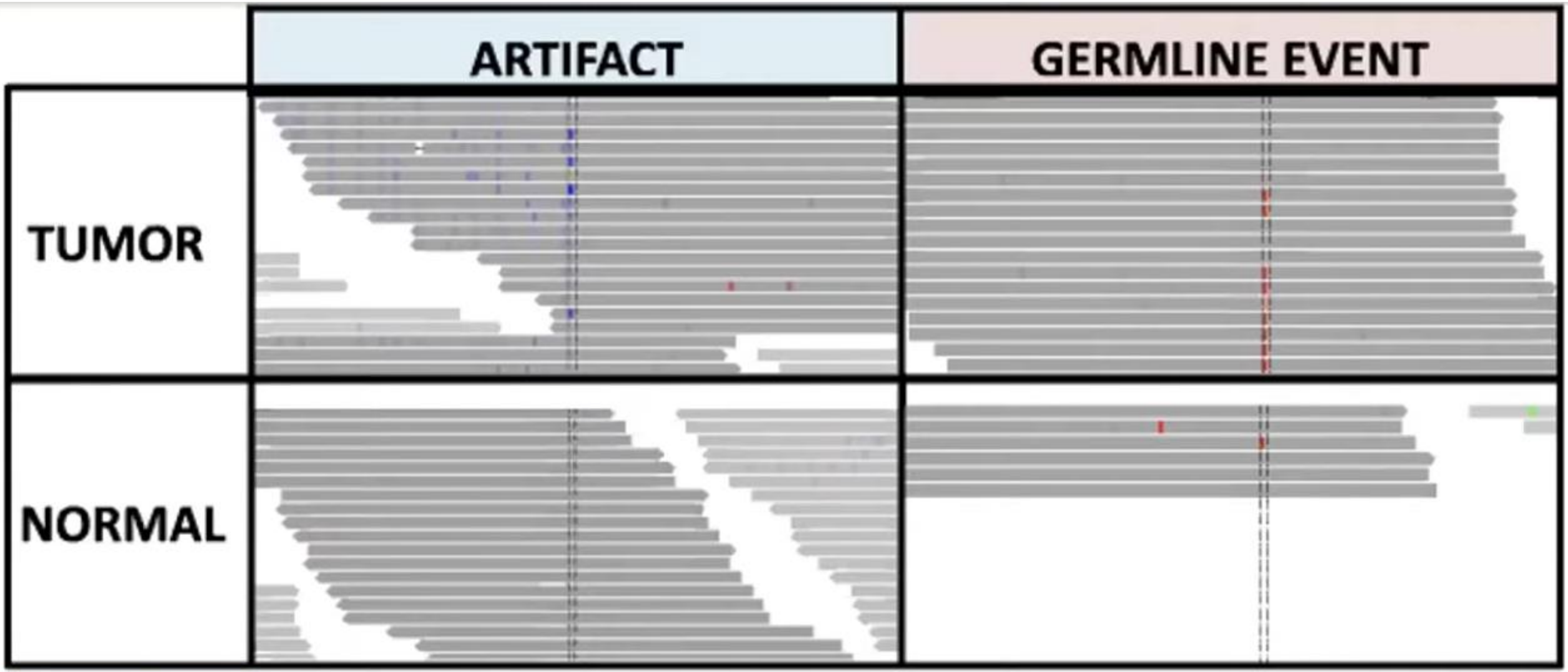
- **Low-coverage** sequencing (<5X per site per individual) – **high probability** that only **one of the two** chromosomes of a diploid individual has been sampled at specific site
- Accurate variant and genotype calling are **difficult**
- It is crucial to **quantify uncertainty** of the results

Instrument specific errors



	Main error source	First-pass error rate	Final error rate
Illumina	Substitution	~ 0.1	~ 0.1
PacBio	Indel	~ 13	<1
Oxford Nanopore	Deletions	>4	4
Ion Torrent	Indel	~ 1	~ 1
SOLiD	A-T bias	~ 5	<0.1
454	Indel	1	1

http://www.molecularecologist.com/next-gen-table-3c-2013/



1000 Genomes Project Consortium, *Nature* 2015.

At risk	Every base	~1667 germline variants / Mbp
Source	- Misread bases - Sequencing artifacts - Misaligned reads	- Low coverage in NORMAL
Solutions	*filters, Panel of Normals (PoN)*	*gnomAD*

VCF tools

- Different methods of calling vcf, often designed on specific NGS tools/data types – not necessarily compatible!

- To distinguish between **real variants** and **technical artifacts** (DNA polymerase and sequencing errors), the **background error rate must be estimated**

- The background error **rate is not constant** and can **vary at different positions => each position has a specific error rate**

- Variant callers **differ also based on the method estimating the background error rate** and whether they use specific methods **to LOCALLY REALIGN** the reads to perfect the precision

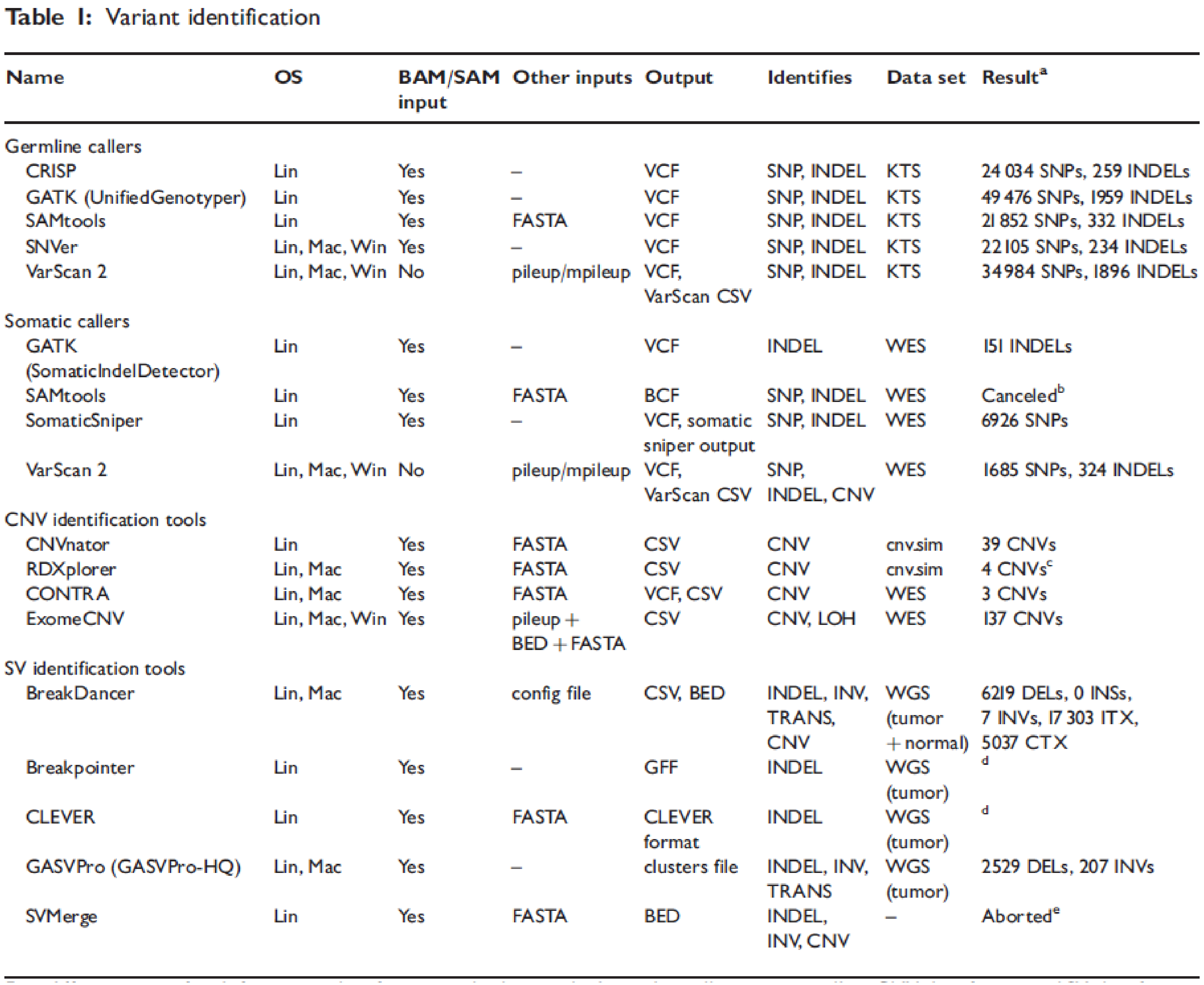


Table 1: Variant identification

Name	OS	BAM/SAM input	Other inputs	Output	Identifies	Data set	Result[a]
Germline callers							
CRISP	Lin	Yes	–	VCF	SNP, INDEL	KTS	24 034 SNPs, 259 INDELs
GATK (UnifiedGenotyper)	Lin	Yes	–	VCF	SNP, INDEL	KTS	49 476 SNPs, 1959 INDELs
SAMtools	Lin	Yes	FASTA	VCF	SNP, INDEL	KTS	21 852 SNPs, 332 INDELs
SNVer	Lin, Mac, Win	Yes	–	VCF	SNP, INDEL	KTS	22 105 SNPs, 234 INDELs
VarScan 2	Lin, Mac, Win	No	pileup/mpileup	VCF, VarScan CSV	SNP, INDEL	KTS	34 984 SNPs, 1896 INDELs
Somatic callers							
GATK (SomaticIndelDetector)	Lin	Yes	–	VCF	INDEL	WES	151 INDELs
SAMtools	Lin	Yes	FASTA	BCF	SNP, INDEL	WES	Canceled[b]
SomaticSniper	Lin	Yes	–	VCF, somatic sniper output	SNP, INDEL	WES	6926 SNPs
VarScan 2	Lin, Mac, Win	No	pileup/mpileup	VCF, VarScan CSV	SNP, INDEL, CNV	WES	1685 SNPs, 324 INDELs
CNV identification tools							
CNVnator	Lin	Yes	FASTA	CSV	CNV	cnv.sim	39 CNVs
RDXplorer	Lin, Mac	Yes	FASTA	CSV	CNV	cnv.sim	4 CNVs[c]
CONTRA	Lin, Mac	Yes	FASTA	VCF, CSV	CNV	WES	3 CNVs
ExomeCNV	Lin, Mac, Win	Yes	pileup + BED + FASTA	CSV	CNV, LOH	WES	137 CNVs
SV identification tools							
BreakDancer	Lin, Mac	Yes	config file	CSV, BED	INDEL, INV, TRANS, CNV	WGS (tumor + normal)	6219 DELs, 0 INSs, 7 INVs, 17 303 ITX, 5037 CTX
Breakpointer	Lin	Yes	–	GFF	INDEL	WGS (tumor)	[d]
CLEVER	Lin	Yes	FASTA	CLEVER format	INDEL	WGS (tumor)	[d]
GASVPro (GASVPro-HQ)	Lin, Mac	Yes	–	clusters file	INDEL, INV, TRANS	WGS (tumor)	2529 DELs, 207 INVs
SVMerge	Lin	Yes	FASTA	BED	INDEL, INV, CNV	–	Aborted[e]

Pabinger et al. (2013) Survey of tools for variant analysis of next-generation genome sequencing data. Brief Bioinform

Somatic variant callers – for tumours

- Background error estimation, tools use:
  - **matched normal / tumour samples**
  - **control samples to model the error noise** to provide the variant calling tool with the built-in model
  - **use information** from databases of **germline** SNPs

Somatic callers:

GATK – Mutect2

Strelka2

smCounter2

UMI-VarCal

UMI based (somatic) variant calling

Assumption: **reads that have the same UMI should be identical**

- Main approach:

1. Perform a majority vote within a UMI family
2. Build a consensus read for each UMI family
3. Apply a statistical method (like Beta distribution) to model background error rates at each position and apply standard filters to call final variants

DeepSNVMiner

MAGERI

smCounter2

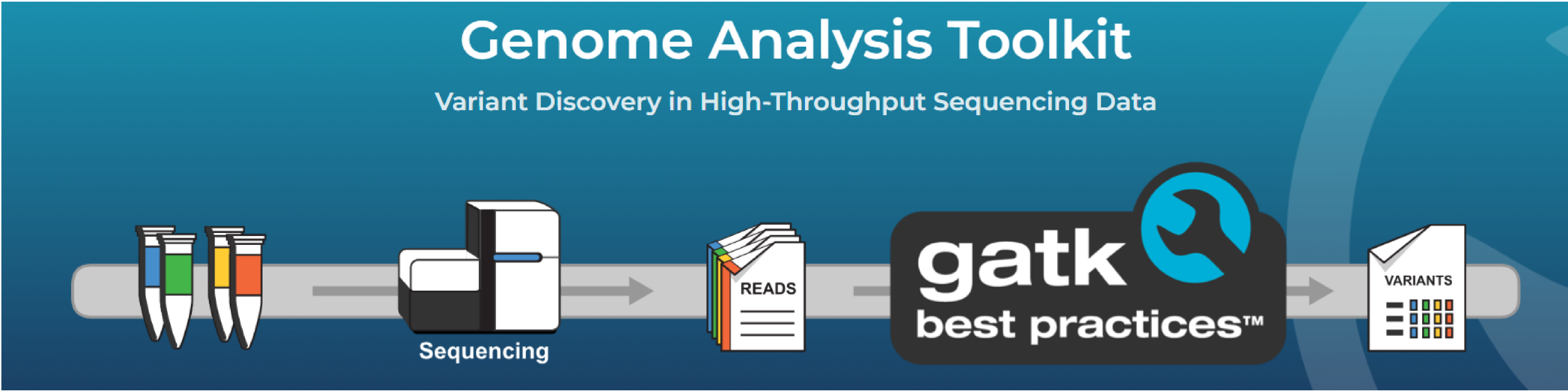
UMI-VarCal

UMI-VarCal

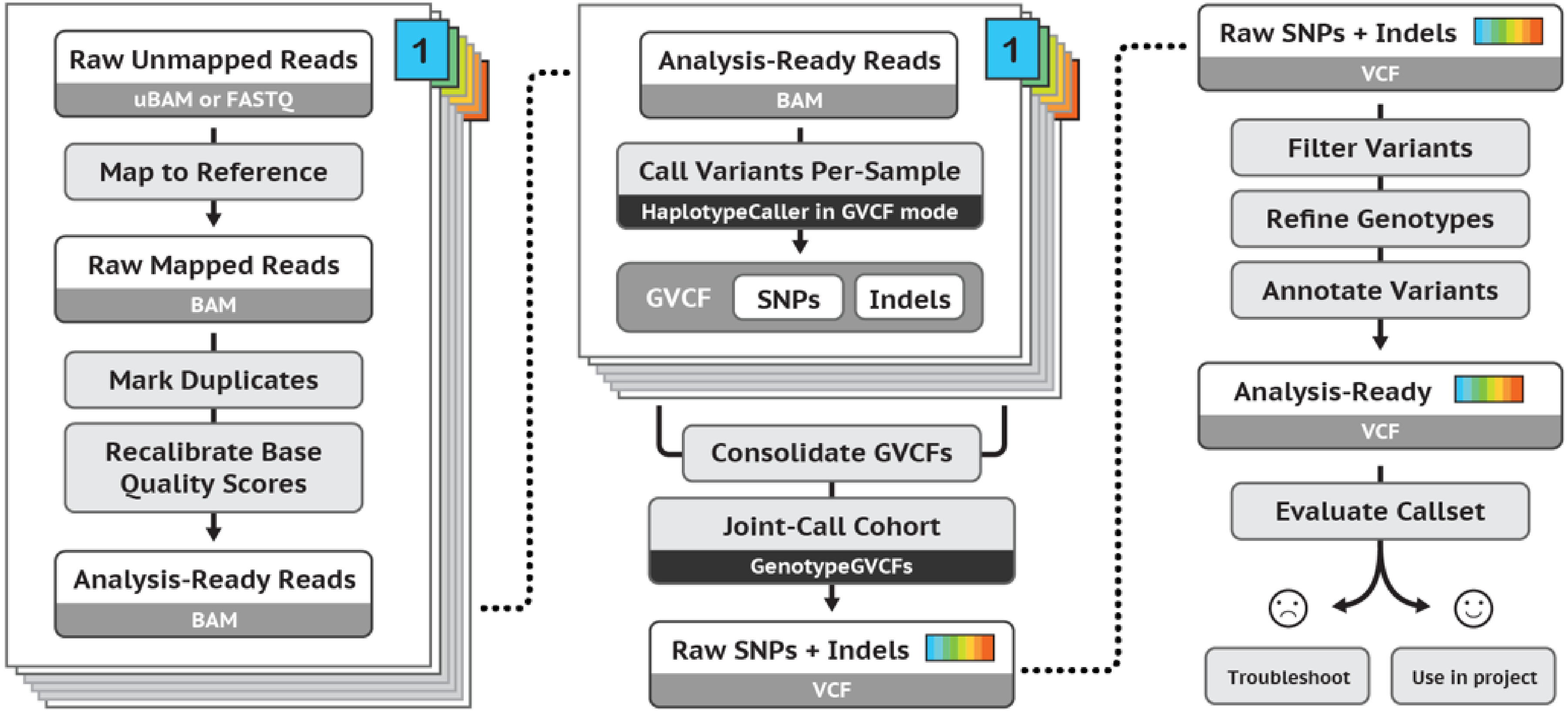
- **UMI-VarCal:**
  - **Does not rely on SAMtools (like MAGERI)**, uses innovative homemade pileup algorithm specifically designed to treat the UMI tags in the reads
  - After the pileup, a Poisson statistical test is applied at every position to determine if the frequency of the variant is significantly higher than the background error noise.
  - Finally, an analysis of UMI tags is performed, a strand bias and a homopolymer length filter are applied to achieve better accuracy
  - UMI-VarCal is faster than both raw-reads-based and UMI-based variant callers

GATK – Genome Analysis Toolkit

- GATK (broadinstitute.org)



GATK – Genome Analysis Toolkit



Best Practices for SNP and Indel discovery in germline DNA - leveraging groundbreaking methods for combined power and scalability.

Genotype calling methods

- **Cutoff-based** methods (early methods)
  - simple: based on SNP and genotype calling on separate analyses of data from each individual

- **Probabilistic** methods
  - based usually on multiple samples and assigning probabilities for a given genotype

Cutoff-based genotype calling methods

Simple: based on SNP and genotype calling on separate analyses of data from each individual

Steps:

1. **filtering step** in which only high-confidence bases are kept (Qphred>20)
2. **genotype calling** – counting number of times each allele is observed, using fixed cutoffs (e.g. if the proportion of the alternative allele is between 20-80%, heterozygote is called)

Used mainly for high sequencing depths (>20X), so that the probability of a heterozygous individual falling outside the selected range(20-80%) is small

- Empirically determined cutoffs can be used

- Usually used in **targeted sequencing**

- Does not provide measures of uncertainty in the genotype inference

Probabilistic genotype calling methods

- For **moderate to low sequencing depths** (mainly WGS, WES) – genotype calling based on fixed cutoffs will typically lead to under-calling of heterozygous phenotypes
- Several probabilistic methods were developed that use **the quality score to provide a <u>posterior probability</u> for each genotype:**

<u>P (X | G) – genotype likelihood</u> for genotype G can be computed (X represents all of the read data for a particular individual and site)

P(G) – genotype prior

- From these two, Bayes' formula is used to calculate *P (G | X) – the posterior probability* of genotype G
- The genotype with the highest posterior probability is generally chosen and this probability is used as a measure of confidence.
- **Advantages:**
  - higher accuracy
  - natural framework for incorporating information regarding allelle frequencies and patterns of LD (linkage disequilibrium)

Calculating genotype likelihood *$P(X|G)$*

- Can be computed from quality scores for each read
- X_i – data in read I for a particular individual and particular site with genotype G
- $P(X_i\,|G)$ is given by rescaling of the quality core of X_i and the genotype likelihood $P(X|G)$ can be calculated directly from the data by taking the product of $P(X_i\,|G)$ over all *i*
- See for instance:
  - https://www.broadinstitute.org/gatk/media/docs/Samtools.pdf

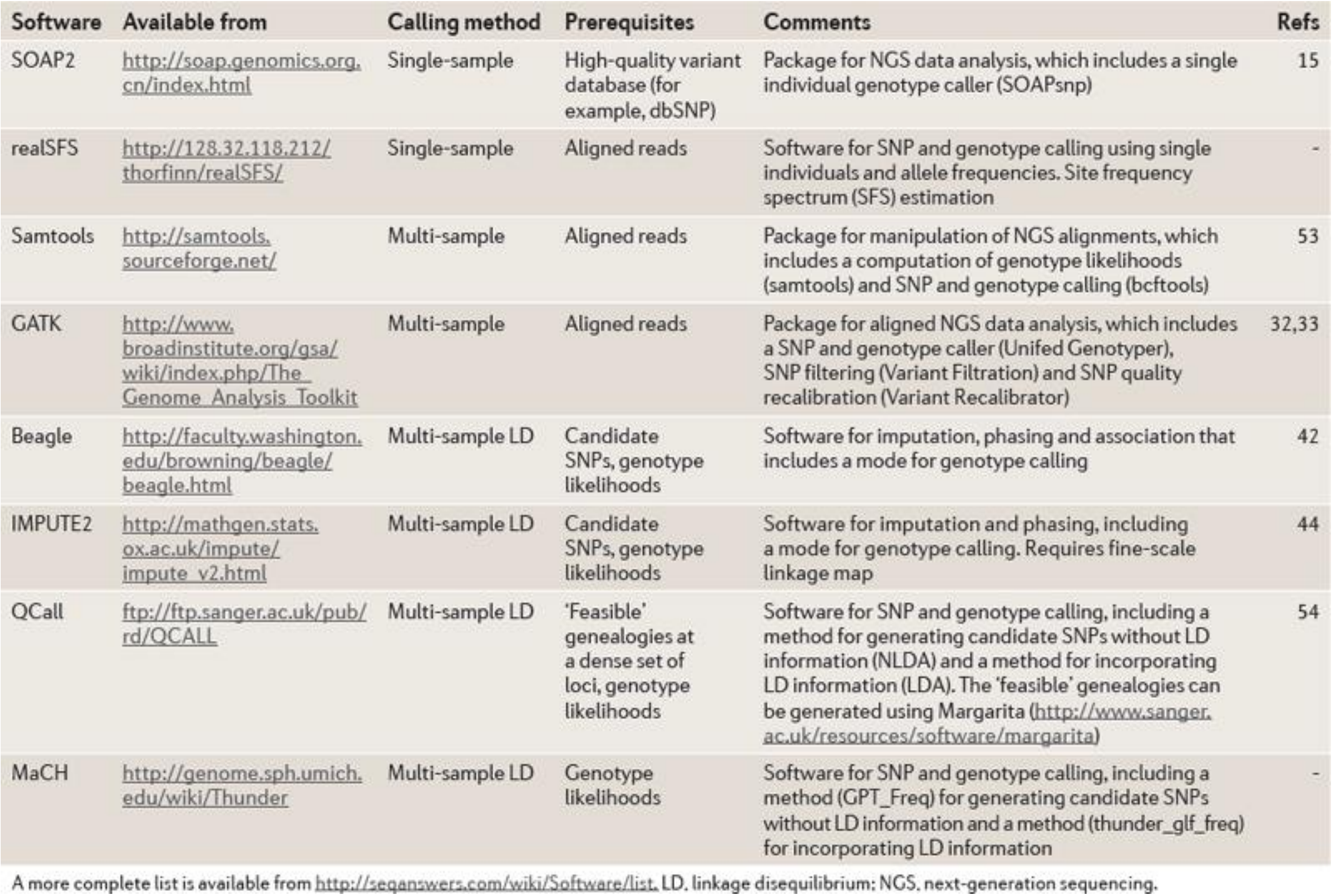
Calculating genotype priors

- Can be performed using single or multiple samples

- **Single-sample:** Prior-genotype probability may be chosen to assign equal probability to all genotypes, or it can be based on external information (e.g. the reference sequence, SNP databases or an available population sample)
  - In **SOAPsnp** tool, a prior is chosen by the use of **dbSNP**
  - Example: if a G/T polymorphism is reported in dbSNP, the prior probabilities are set to be 0.454 for each of the genotypes GG and TT; 0.0909 for GT and less than 10e-4 for all other genotypes

- **Multiple-samples:** Priors derived by joint analysis of multiple individuals – by analysis of allele or genotype frequencies estimated from larger data sets e.g. using *maximum likelihood*

Tools for genotype calling



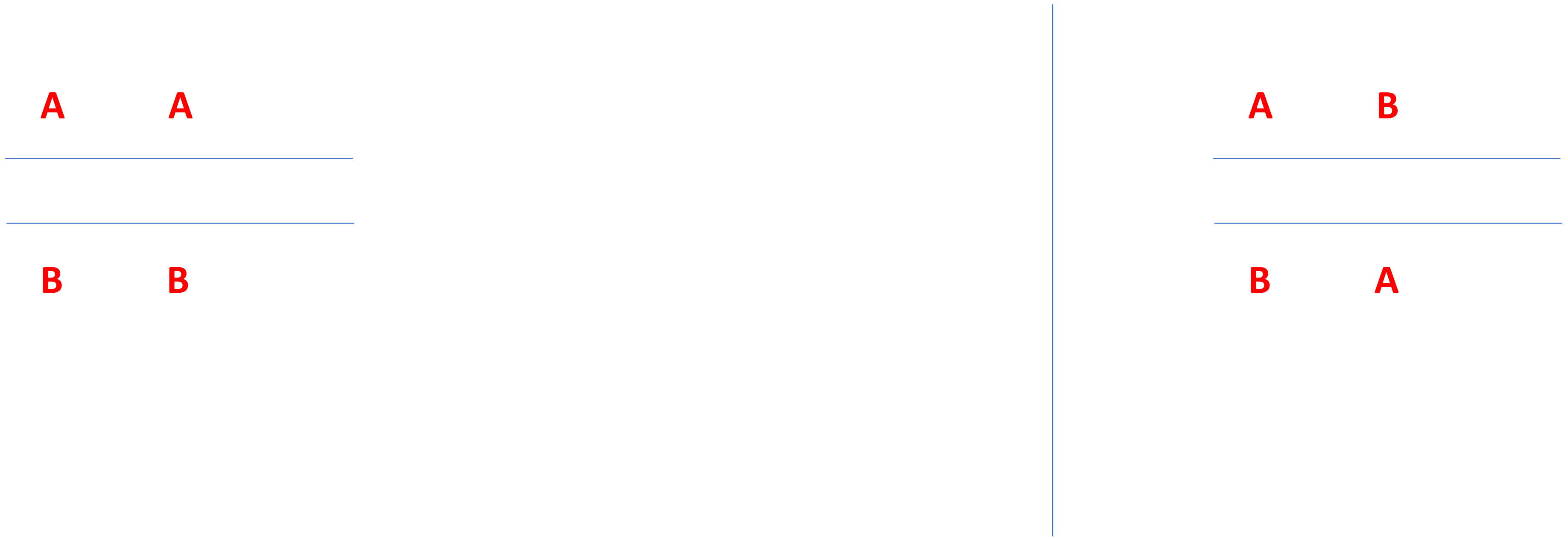
Software	Available from	Calling method	Prerequisites	Comments	Refs
SOAP2	http://soap.genomics.org.cn/index.html	Single-sample	High-quality variant database (for example, dbSNP)	Package for NGS data analysis, which includes a single individual genotype caller (SOAPsnp)	15
realSFS	http://128.32.118.212/thorfinn/realSFS/	Single-sample	Aligned reads	Software for SNP and genotype calling using single individuals and allele frequencies. Site frequency spectrum (SFS) estimation	-
Samtools	http://samtools.sourceforge.net/	Multi-sample	Aligned reads	Package for manipulation of NGS alignments, which includes a computation of genotype likelihoods (samtools) and SNP and genotype calling (bcftools)	53
GATK	http://www.broadinstitute.org/gsa/wiki/index.php/The_Genome_Analysis_Toolkit	Multi-sample	Aligned reads	Package for aligned NGS data analysis, which includes a SNP and genotype caller (Unifed Genotyper), SNP filtering (Variant Filtration) and SNP quality recalibration (Variant Recalibrator)	32,33
Beagle	http://faculty.washington.edu/browning/beagle/beagle.html	Multi-sample LD	Candidate SNPs, genotype likelihoods	Software for imputation, phasing and association that includes a mode for genotype calling	42
IMPUTE2	http://mathgen.stats.ox.ac.uk/impute/impute_v2.html	Multi-sample LD	Candidate SNPs, genotype likelihoods	Software for imputation and phasing, including a mode for genotype calling. Requires fine-scale linkage map	44
QCall	ftp://ftp.sanger.ac.uk/pub/rd/QCALL	Multi-sample LD	'Feasible' genealogies at a dense set of loci, genotype likelihoods	Software for SNP and genotype calling, including a method for generating candidate SNPs without LD information (NLDA) and a method for incorporating LD information (LDA). The 'feasible' genealogies can be generated using Margarita (http://www.sanger.ac.uk/resources/software/margarita)	54
MaCH	http://genome.sph.umich.edu/wiki/Thunder	Multi-sample LD	Genotype likelihoods	Software for SNP and genotype calling, including a method (GPT_Freq) for generating candidate SNPs without LD information and a method (thunder_glf_freq) for incorporating LD information	-

A more complete list is available from http://seqanswers.com/wiki/Software/list. LD, linkage disequilibrium; NGS, next-generation sequencing.

Nielsen et al. (2011): Genotype and SNP calling from next-generation sequencing data. Nature Reviews Genetics 12, 443-451

Haplotype calling / haplotype phasing

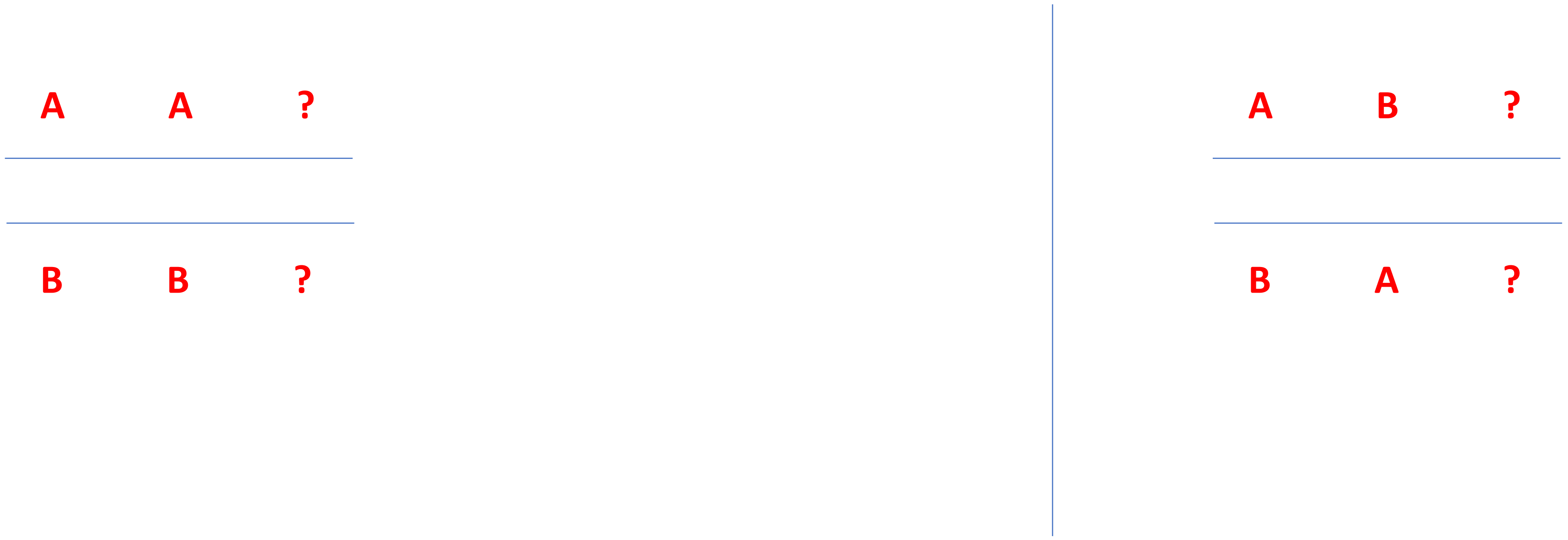
- The exponential growth problem….
- Example – **two** heterozygous genotypes (A/B)



[Brian Browning | Haplotype phasing: methods and accuracy - YouTube]

Haplotype calling / haplotype phasing

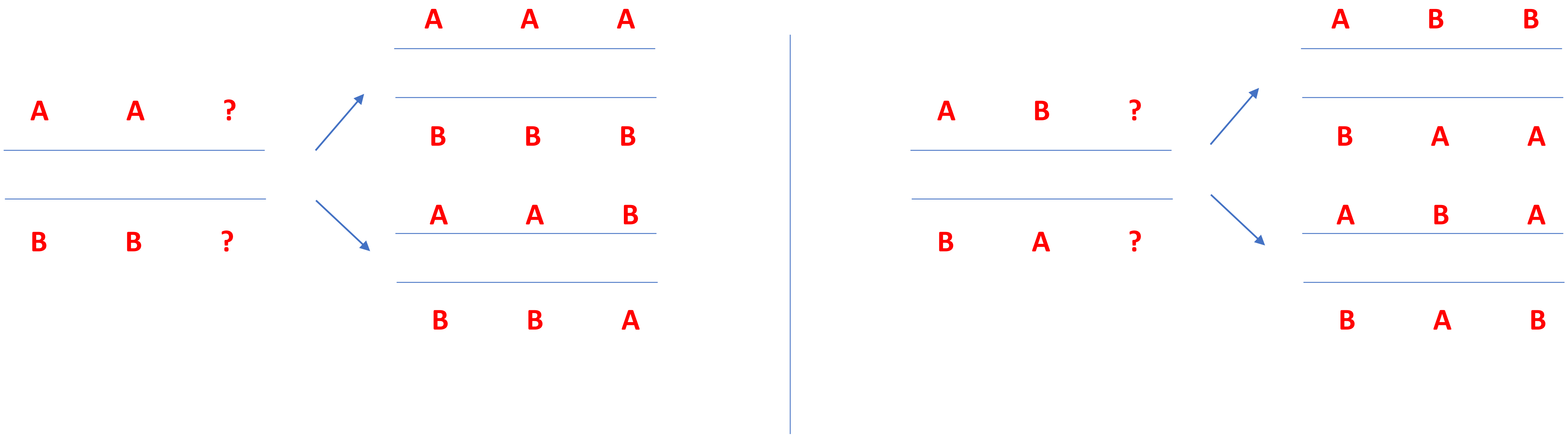
- The exponential growth problem….
- Example – **three** heterozygous genotypes (A/B)?



Brian Browning | Haplotype phasing: methods and accuracy - YouTube

Haplotype calling / haplotype phasing

- The exponential growth problem….
- Example – three heterozygous genotypes (A/B)?



A A ?

B B ?

A A A

B B B

A A B

B B A

A B ?

B A ?

A B B

B A A

A B A

B A B

Brian Browning | Haplotype phasing: methods and accuracy - YouTube

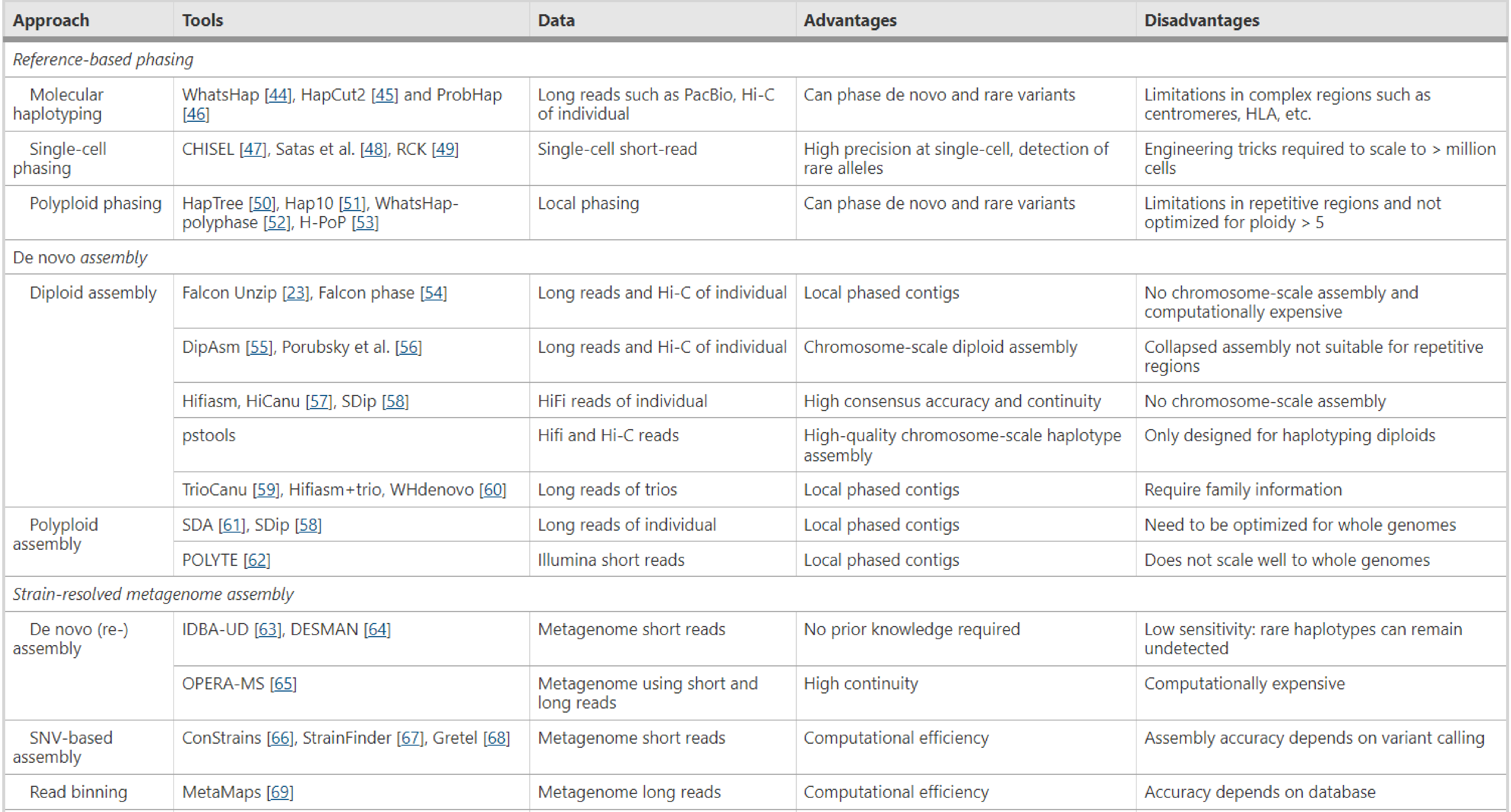
Haplotype phasing algorithms

- Main approaches:

- Clark’s algorithm
- EM algorithm (Arlequin, PL-EM)
- Coalescent-based methods and hidden Markov models (MACH, IMPUTE2, PHASE, BEAGLE)

- Haplotype phasing: Existing methods and new developments - PMC (nih.gov)

From: Computational methods for chromosome-scale haplotype reconstruction



Approach	Tools	Data	Advantages	Disadvantages
Reference-based phasing				
Molecular haplotyping	WhatsHap [44], HapCut2 [45] and ProbHap [46]	Long reads such as PacBio, Hi-C of individual	Can phase de novo and rare variants	Limitations in complex regions such as centromeres, HLA, etc.
Single-cell phasing	CHISEL [47], Satas et al. [48], RCK [49]	Single-cell short-read	High precision at single-cell, detection of rare alleles	Engineering tricks required to scale to > million cells
Polyploid phasing	HapTree [50], Hap10 [51], WhatsHap-polyphase [52], H-PoP [53]	Local phasing	Can phase de novo and rare variants	Limitations in repetitive regions and not optimized for ploidy > 5
De novo *assembly*				
Diploid assembly	Falcon Unzip [23], Falcon phase [54]	Long reads and Hi-C of individual	Local phased contigs	No chromosome-scale assembly and computationally expensive
	DipAsm [55], Porubsky et al. [56]	Long reads and Hi-C of individual	Chromosome-scale diploid assembly	Collapsed assembly not suitable for repetitive regions
	Hifiasm, HiCanu [57], SDip [58]	HiFi reads of individual	High consensus accuracy and continuity	No chromosome-scale assembly
	pstools	Hifi and Hi-C reads	High-quality chromosome-scale haplotype assembly	Only designed for haplotyping diploids
	TrioCanu [59], Hifiasm+trio, WHdenovo [60]	Long reads of trios	Local phased contigs	Require family information
Polyploid assembly	SDA [61], SDip [58]	Long reads of individual	Local phased contigs	Need to be optimized for whole genomes
	POLYTE [62]	Illumina short reads	Local phased contigs	Does not scale well to whole genomes
Strain-resolved metagenome assembly				
De novo (re-) assembly	IDBA-UD [63], DESMAN [64]	Metagenome short reads	No prior knowledge required	Low sensitivity: rare haplotypes can remain undetected
	OPERA-MS [65]	Metagenome using short and long reads	High continuity	Computationally expensive
SNV-based assembly	ConStrains [66], StrainFinder [67], Gretel [68]	Metagenome short reads	Computational efficiency	Assembly accuracy depends on variant calling
Read binning	MetaMaps [69]	Metagenome long reads	Computational efficiency	Accuracy depends on database

Is genotype/haplotype calling necessary?

- Attention – sometimes, it is not of interest to call a genotype – e.g. if heterogenous tissues are sampled and we can expect different clones!
  - Example: cancer cells – one clone from thousands can regrow into metastasis!

Variant annotation

- We have the possibility of predicting the functional impact of variants in an automated fashion
- This is very important for biological interpretation of the results!
- Not all the tools are able to annotate all types of variants
- Output: usually vcf file, with information on annotation in the INPUT field

Example of annotated .vcf file

<table>
<tr><th>#CHROM</th><th>POS</th><th>ID</th><th>REF</th><th>ALT</th><th>QUAL</th><th>FILTER</th><th>INFO</th><th>FORMAT</th></tr>
<tr><td>chr1</td><td>868329</td><td>.</td><td>A</td><td>C</td><td>25.56</td><td>LowQual</td><td>AC=2;AF=1.00;AN=2;DP=2;FS=0.000;MLEAC=2;MLEAF=1.00;MQ=99.00;QD=12.78;SOR=0.693;ANN=C|intron_variant|MODIFIER|SAMD11|SAMD11|transcript|NM_152486.2|Coding|4/13|c.305+1860A>C||||||</td><td>GT:AD:DP:FT:GQ:PL</td></tr>
<tr><td>chr1</td><td>1665702</td><td>.</td><td>T</td><td>C</td><td>62.55</td><td>PASS</td><td>AC=2;AF=1.00;AN=2;DP=2;FS=0.000;MLEAC=2;MLEAF=1.00;MQ=60.00;QD=31.27;SOR=2.303;ANN=C|intron_variant|MODIFIER|SLC35E2|SLC35E2|transcript|NM_182838.2|Coding|5/5|c.732+427A>G||||||,C|intron_variant|MODIFIER|SLC35E2|SLC35E2|transcript|NM_001199787.1|Coding|6/6|c.732+427A>G||||||</td><td>GT:AD:DP:FT:GQ:PL</td></tr>
<tr><td>chr1</td><td>1665740</td><td>.</td><td>T</td><td>C</td><td>66.55</td><td>PASS</td><td>AC=2;AF=1.00;AN=2;DP=2;FS=0.000;MLEAC=2;MLEAF=1.00;MQ=60.00;QD=33.27;SOR=2.303;ANN=C|intron_variant|MODIFIER|SLC35E2|SLC35E2|transcript|NM_182838.2|Coding|5/5|c.732+389A>G||||||,C|intron_variant|MODIFIER|SLC35E2|SLC35E2|transcript|NM_001199787.1|Coding|6/6|c.732+389A>G||||||</td><td>GT:AD:DP:FT:GQ:PL</td></tr>
</table>

Variant annotation tools

- Pabinger et al. (2013) Survey of tools for variant analysis of next-generation genome sequencing data. Brief Bioinform

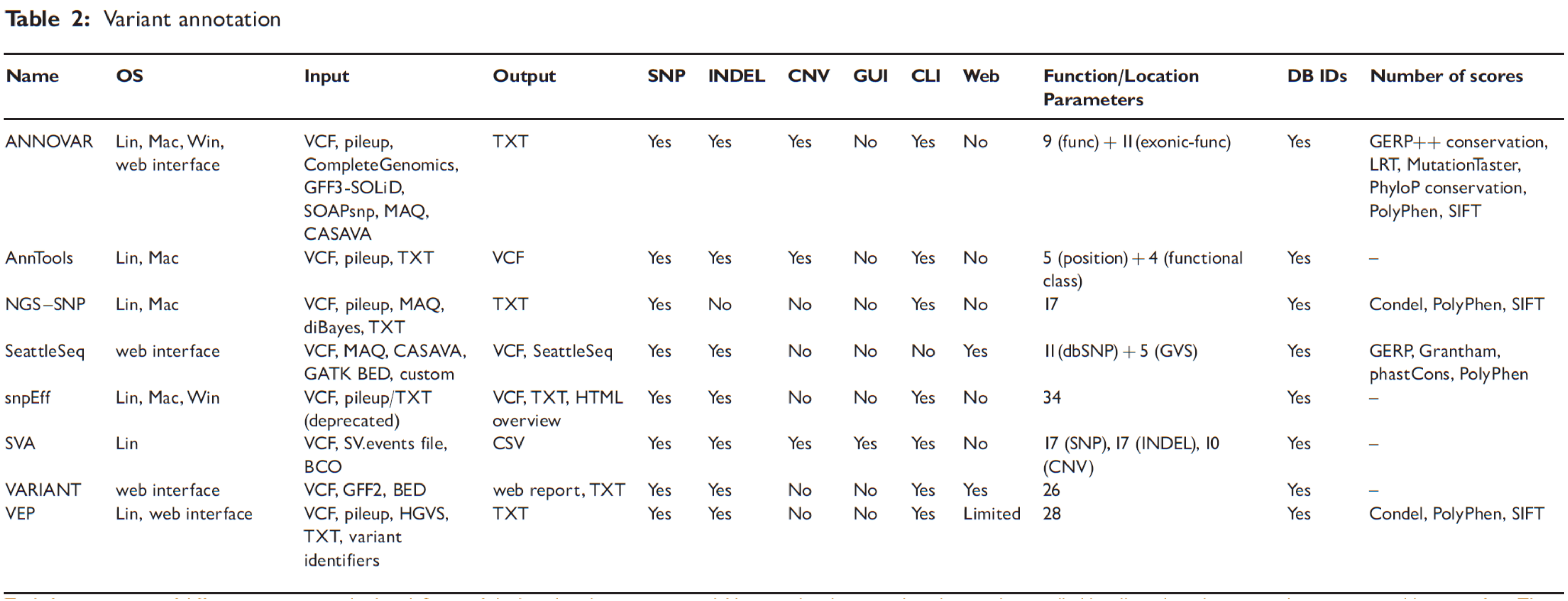


Table 2: Variant annotation

Name	OS	Input	Output	SNP	INDEL	CNV	GUI	CLI	Web	Function/Location Parameters	DB IDs	Number of scores
ANNOVAR	Lin, Mac, Win, web interface	VCF, pileup, CompleteGenomics, GFF3-SOLiD, SOAPsnp, MAQ, CASAVA	TXT	Yes	Yes	Yes	No	Yes	No	9 (func) + 11 (exonic-func)	Yes	GERP++ conservation, LRT, MutationTaster, PhyloP conservation, PolyPhen, SIFT
AnnTools	Lin, Mac	VCF, pileup, TXT	VCF	Yes	Yes	Yes	No	Yes	No	5 (position) + 4 (functional class)	Yes	–
NGS–SNP	Lin, Mac	VCF, pileup, MAQ, diBayes, TXT	TXT	Yes	No	No	No	Yes	No	17	Yes	Condel, PolyPhen, SIFT
SeattleSeq	web interface	VCF, MAQ, CASAVA, GATK BED, custom	VCF, SeattleSeq	Yes	Yes	No	No	No	Yes	11 (dbSNP) + 5 (GVS)	Yes	GERP, Grantham, phastCons, PolyPhen
snpEff	Lin, Mac, Win	VCF, pileup/TXT (deprecated)	VCF, TXT, HTML overview	Yes	Yes	No	No	Yes	No	34	Yes	–
SVA	Lin	VCF, SV.events file, BCO	CSV	Yes	Yes	Yes	Yes	Yes	No	17 (SNP), 17 (INDEL), 10 (CNV)	Yes	–
VARIANT	web interface	VCF, GFF2, BED	web report, TXT	Yes	Yes	No	No	Yes	Yes	26	Yes	–
VEP	Lin, web interface	VCF, pileup, HGVS, TXT, variant identifiers	TXT	Yes	Yes	No	No	Yes	Limited	28	Yes	Condel, PolyPhen, SIFT

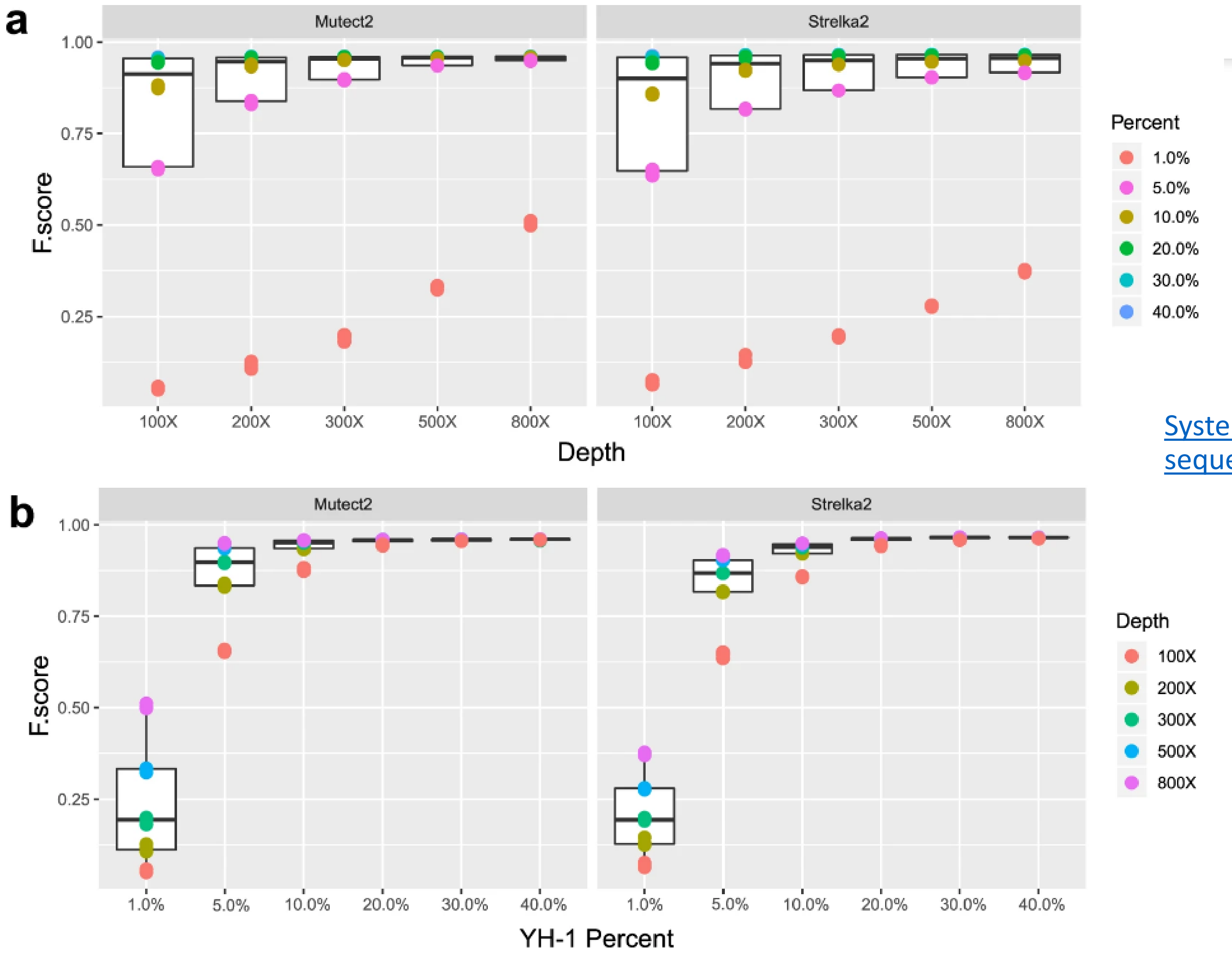
Additional resources

- Best practices for variant calling in clinical sequencing | Genome Medicine | Full Text (biomedcentral.com)

Specific issues in **variant calling**

The effect of sequencing depth and mutation frequency on performance of variant callers

From: Systematic comparison of somatic variant calling performance among different sequencing depth and mutation frequency



Systematic comparison of somatic variant calling performance among different sequencing depth and mutation frequency | Scientific Reports (nature.com)

F-score box-scatter plot. The box-scatter plot of F-score, the colors represent different mutation frequency (**a**) and sequencing depths (**b**).

Base calling algorithms

- **Reducing the error rate of base calls** and **improving the accuracy** of the per-base quality score (Phred) have i**mportant implications** for assembly, **variant detection** and downstream population-genomic analyses.

- Aside of NGS platform provided base calling algorithms, several other have been developed to **optimize data acquisition, improving by 5-30%** the error rates:
- Pyrobayes (454 Roche)
- Rsolid (SOLiD)
- BayesCall, Ibis (Illumina)

Alignment

- The **accuracy of alignment** has a **crucial** role in variant detection.

- Incorrectly aligned reads may lead to errors in SNP and genotype calling

- Alignment algorithms – need to be:

- able to cope with **sequencing errors** and potentially **real differences** (point mutations and indels) between the reference genome and the sequenced genome

- able to produce **well calibrated** alignment quality values (variant calls and their posterior probabilities depend on these scores)

Alignment - mismatches

- The amount of **sequence identity** required between each read and the reference sequence is determined by a **trade-off** between **accuracy** and **length**

- The **optimal** choice of tolerable number of **mismatches** depends on the **organism** and also on the **genome part!**

- e.g. *Drosophila melanogaster* is more variable than human – using mapping criteria for human on *Drosophila* may lead to a severe loss of sequencing depth for *Drosophila*. Vice-versa, using *Drosophila melanogaster* mapping criteria on human would lead to large amount of incorrectly aligned reads

- MHC complex – very variable between individuals – consider combining alignment and assembly!

Steps increasing the precision

- **Pair-end reads** are highly recommended, **if not even a requirement** for **WES** and **WGS** to overcome the problem of ambiguity

- **Reads** that can **only** be mapped with **many mismatches** should be **discarded** from the analysis => **variants** backed **only** by such reads should not be considered

- Multiple reads originating from only one template might be sequenced, interfering with variant calling statistics => remove the PCR duplicates after alignment in **WGS** and **WES (not in targeted sequencing!!)**

Which aligners?

Bowtie – cannot perform gapped alignment => cannot find short indels => think of Bowtie2!

BWA

BWT vs hash-based: BWT are faster, BUT – not as sensitive as hash-based => can introduce **mapping biases in regions with high variability***

According to a comparative study*, **Novoalign** and **Stampy** currently produce the most accurate overall results, being at the same time fast enough

However, Stampy not applicable to SOLiD colour reads

SHRiMP2 and BFAST – hash based, capable of dealing with SOLiD colour reads

*Lunter, G. & Goodson, M. (2011) Stampy: a statistical algorithm for sensitive and fast mapping of Illumina sequence reads. Genome Res. 19: 936-939

Stampy*

- Mapping hash-based with optional speedup using BWA
- No support of color reads
- Combines the speed-up of BWT and the sensitivity of hash-based aligners
- http://www.well.ox.ac.uk/project-stampy
- for download you need to register

- *Lunter, G. & Goodson, M. (2011) Stampy: a statistical algorithm for sensitive and fast mapping of Illumina sequence reads. Genome Res. 19: 936-939

Stampy + BWA

- Steps:
- 1. Create BWA index for the same reference genome as used by Stampy
- bwa index

- 2. Map the reads using BWA in the usual way and use samtools to convert the resulting SAM file into a BAM file
- bwa aln
- bwa samse or sampe
- samtools view

- 3. Remap the BAM file using Stampy, but keep the well-mapped reads
- ./stampy.py -g hg18 \
- -h hg18 \
- -- bamkeepgoodreads -M bwa.bam

After alignment

- Step 1: Post-alignment **quality control (assessment of target coverage, removal of duplicates** and **non-unique alignments, ...)**

- Step 2: **Realignment** around **indels** - variant calling specific step

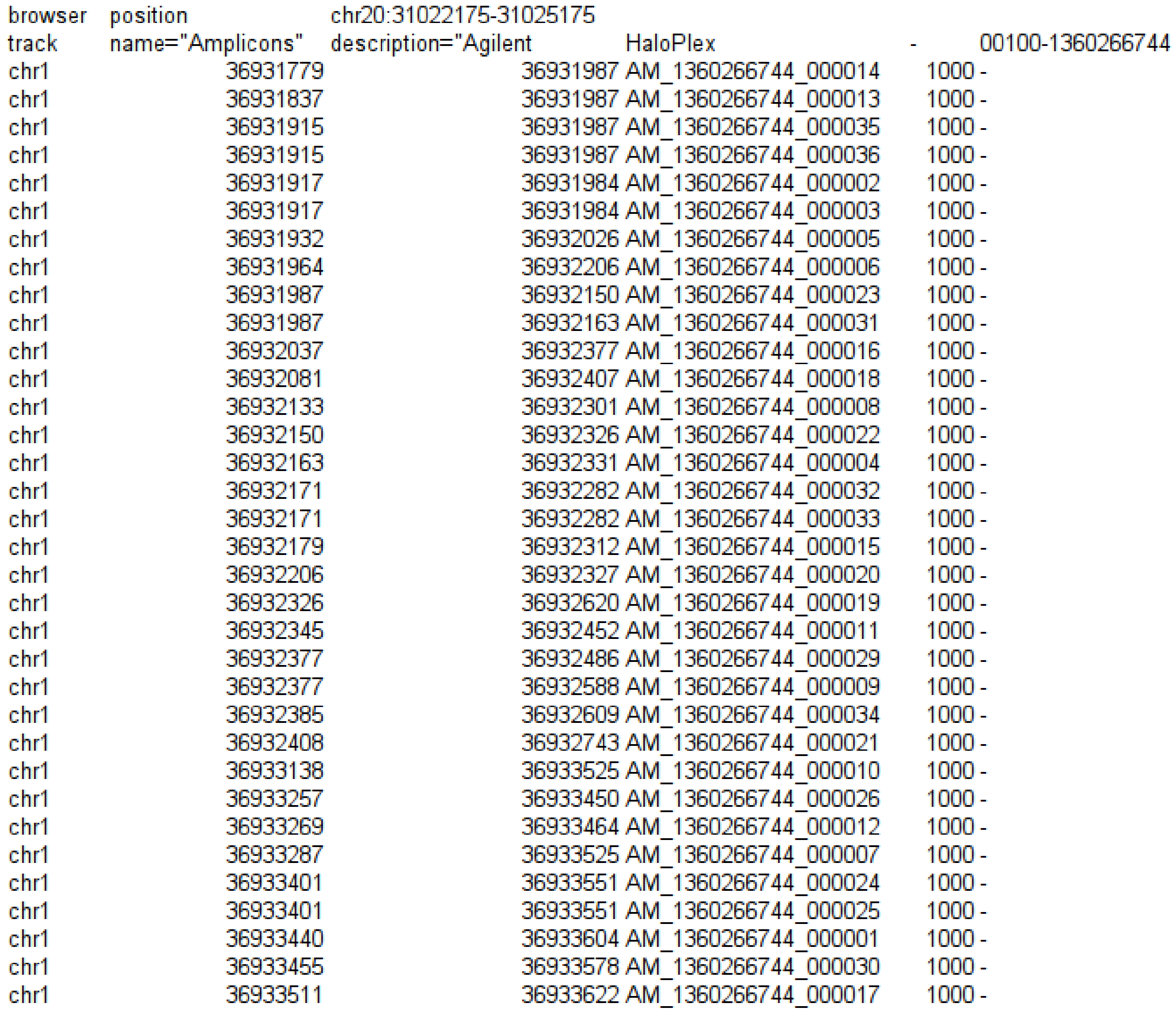
- Step 3: Quality score **recalibration** - variant calling specific step

Assessment of target coverage

- QC step very specific for **targeted** sequencing
- Target coverage – is the selection of targets working?
- Specific files used: “**bait** list” and **“target** list**”** (both are **.bed** files, provided by company supplying the enrichment kit)

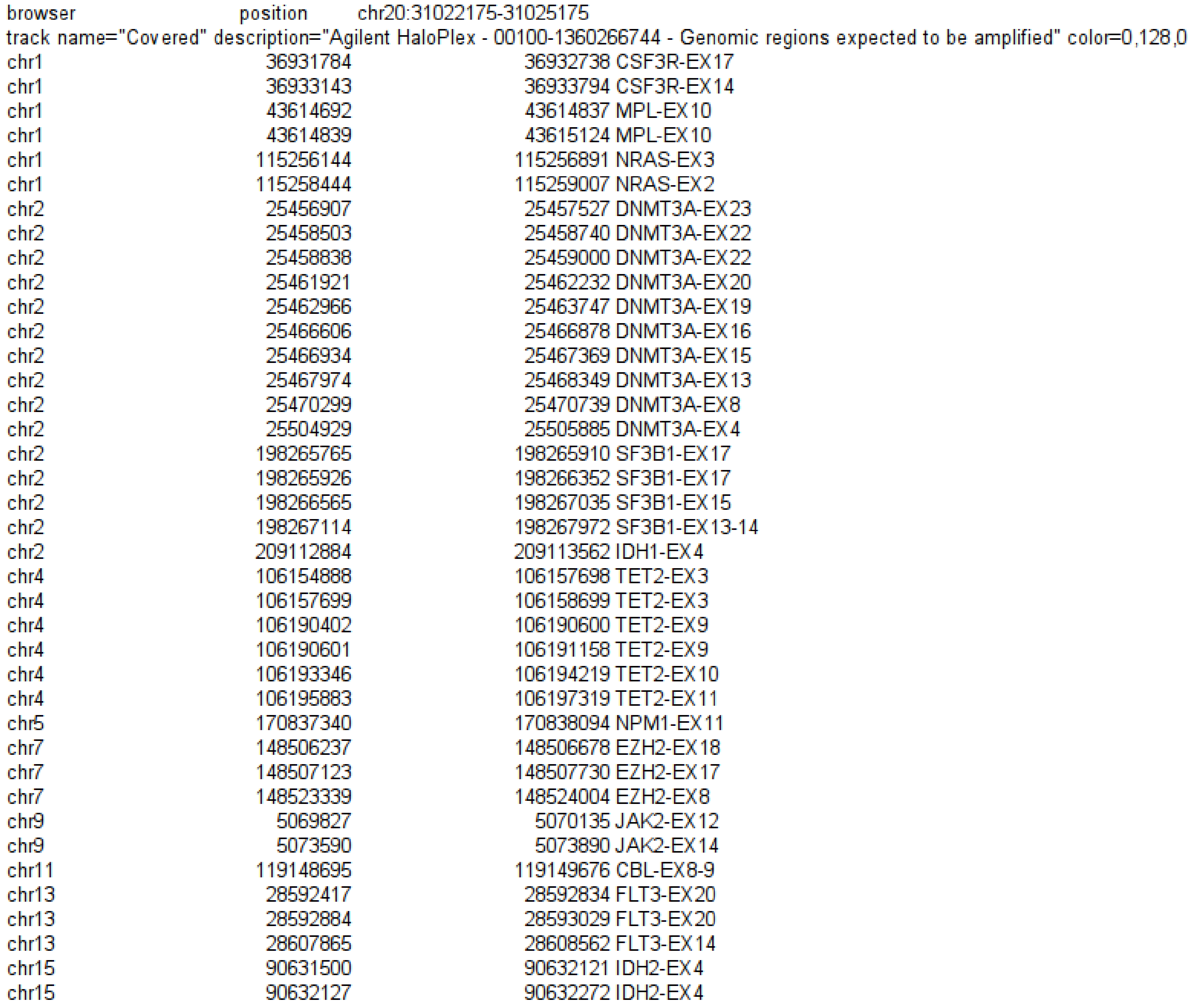
- ***Baits*** are **oligonucleotides** that retrieve specific RNA or genomic DNA used for selection/enrichment of target regions. The desired DNA or RNA molecules hybridize with the baits, and others do not
- ***Targets*** are sequences of interest

Bait file



```
browser   position            chr20:31022175-31025175
track     name="Amplicons"    description="Agilent          HaloPlex                  -        00100-1360266744
chr1                36931779                        36931987 AM_1360266744_000014      1000 -
chr1                36931837                        36931987 AM_1360266744_000013      1000 -
chr1                36931915                        36931987 AM_1360266744_000035      1000 -
chr1                36931915                        36931987 AM_1360266744_000036      1000 -
chr1                36931917                        36931984 AM_1360266744_000002      1000 -
chr1                36931917                        36931984 AM_1360266744_000003      1000 -
chr1                36931932                        36932026 AM_1360266744_000005      1000 -
chr1                36931964                        36932206 AM_1360266744_000006      1000 -
chr1                36931987                        36932150 AM_1360266744_000023      1000 -
chr1                36931987                        36932163 AM_1360266744_000031      1000 -
chr1                36932037                        36932377 AM_1360266744_000016      1000 -
chr1                36932081                        36932407 AM_1360266744_000018      1000 -
chr1                36932133                        36932301 AM_1360266744_000008      1000 -
chr1                36932150                        36932326 AM_1360266744_000022      1000 -
chr1                36932163                        36932331 AM_1360266744_000004      1000 -
chr1                36932171                        36932282 AM_1360266744_000032      1000 -
chr1                36932171                        36932282 AM_1360266744_000033      1000 -
chr1                36932179                        36932312 AM_1360266744_000015      1000 -
chr1                36932206                        36932327 AM_1360266744_000020      1000 -
chr1                36932326                        36932620 AM_1360266744_000019      1000 -
chr1                36932345                        36932452 AM_1360266744_000011      1000 -
chr1                36932377                        36932486 AM_1360266744_000029      1000 -
chr1                36932377                        36932588 AM_1360266744_000009      1000 -
chr1                36932385                        36932609 AM_1360266744_000034      1000 -
chr1                36932408                        36932743 AM_1360266744_000021      1000 -
chr1                36933138                        36933525 AM_1360266744_000010      1000 -
chr1                36933257                        36933450 AM_1360266744_000026      1000 -
chr1                36933269                        36933464 AM_1360266744_000012      1000 -
chr1                36933287                        36933525 AM_1360266744_000007      1000 -
chr1                36933401                        36933551 AM_1360266744_000024      1000 -
chr1                36933401                        36933551 AM_1360266744_000025      1000 -
chr1                36933440                        36933604 AM_1360266744_000001      1000 -
chr1                36933455                        36933578 AM_1360266744_000030      1000 -
chr1                36933511                        36933622 AM_1360266744_000017      1000 -
```

Target file



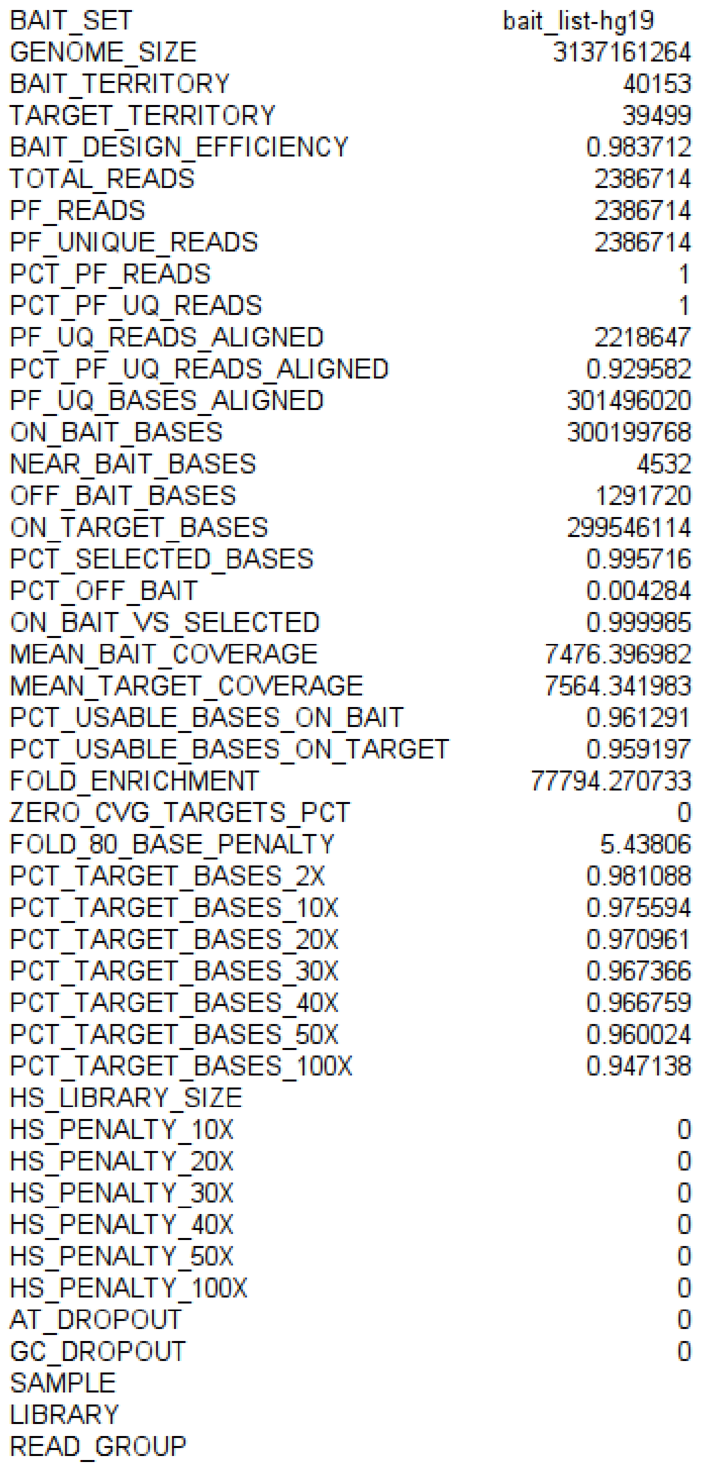
```
browser          position     chr20:31022175-31025175
track name="Covered" description="Agilent HaloPlex - 00100-1360266744 - Genomic regions expected to be amplified" color=0,128,0
chr1           36931784            36932738 CSF3R-EX17
chr1           36933143            36933794 CSF3R-EX14
chr1           43614692            43614837 MPL-EX10
chr1           43614839            43615124 MPL-EX10
chr1          115256144           115256891 NRAS-EX3
chr1          115258444           115259007 NRAS-EX2
chr2           25456907            25457527 DNMT3A-EX23
chr2           25458503            25458740 DNMT3A-EX22
chr2           25458838            25459000 DNMT3A-EX22
chr2           25461921            25462232 DNMT3A-EX20
chr2           25462966            25463747 DNMT3A-EX19
chr2           25466606            25466878 DNMT3A-EX16
chr2           25466934            25467369 DNMT3A-EX15
chr2           25467974            25468349 DNMT3A-EX13
chr2           25470299            25470739 DNMT3A-EX8
chr2           25504929            25505885 DNMT3A-EX4
chr2          198265765           198265910 SF3B1-EX17
chr2          198265926           198266352 SF3B1-EX17
chr2          198266565           198267035 SF3B1-EX15
chr2          198267114           198267972 SF3B1-EX13-14
chr2          209112884           209113562 IDH1-EX4
chr4          106154888           106157698 TET2-EX3
chr4          106157699           106158699 TET2-EX3
chr4          106190402           106190600 TET2-EX9
chr4          106190601           106191158 TET2-EX9
chr4          106193346           106194219 TET2-EX10
chr4          106195883           106197319 TET2-EX11
chr5          170837340           170838094 NPM1-EX11
chr7          148506237           148506678 EZH2-EX18
chr7          148507123           148507730 EZH2-EX17
chr7          148523339           148524004 EZH2-EX8
chr9            5069827             5070135 JAK2-EX12
chr9            5073590             5073890 JAK2-EX14
chr11         119148695           119149676 CBL-EX8-9
chr13          28592417            28592834 FLT3-EX20
chr13          28592884            28593029 FLT3-EX20
chr13          28607865            28608562 FLT3-EX14
chr15          90631500            90632121 IDH2-EX4
chr15          90632127            90632272 IDH2-EX4
```

Assessment of target coverage

- Tools: picard **CollectHsMetrics**

- java -jar picard.jar CollectHsMetrics \
- I=your.bam \
- O=result.txt \
- R=reference_sequence.fasta \
- BAIT_INTERVALS=bait_list.bed \ TARGET_INTERVALS=target_list.bed \

CollectHsMetrics result



BAIT_SET	bait_list-hg19
GENOME_SIZE	3137161264
BAIT_TERRITORY	40153
TARGET_TERRITORY	39499
BAIT_DESIGN_EFFICIENCY	0.983712
TOTAL_READS	2386714
PF_READS	2386714
PF_UNIQUE_READS	2386714
PCT_PF_READS	1
PCT_PF_UQ_READS	1
PF_UQ_READS_ALIGNED	2218647
PCT_PF_UQ_READS_ALIGNED	0.929582
PF_UQ_BASES_ALIGNED	301496020
ON_BAIT_BASES	300199768
NEAR_BAIT_BASES	4532
OFF_BAIT_BASES	1291720
ON_TARGET_BASES	299546114
PCT_SELECTED_BASES	0.995716
PCT_OFF_BAIT	0.004284
ON_BAIT_VS_SELECTED	0.999985
MEAN_BAIT_COVERAGE	7476.396982
MEAN_TARGET_COVERAGE	7564.341983
PCT_USABLE_BASES_ON_BAIT	0.961291
PCT_USABLE_BASES_ON_TARGET	0.959197
FOLD_ENRICHMENT	77794.270733
ZERO_CVG_TARGETS_PCT	0
FOLD_80_BASE_PENALTY	5.43806
PCT_TARGET_BASES_2X	0.981088
PCT_TARGET_BASES_10X	0.975594
PCT_TARGET_BASES_20X	0.970961
PCT_TARGET_BASES_30X	0.967366
PCT_TARGET_BASES_40X	0.966759
PCT_TARGET_BASES_50X	0.960024
PCT_TARGET_BASES_100X	0.947138
HS_LIBRARY_SIZE	
HS_PENALTY_10X	0
HS_PENALTY_20X	0
HS_PENALTY_30X	0
HS_PENALTY_40X	0
HS_PENALTY_50X	0
HS_PENALTY_100X	0
AT_DROPOUT	0
GC_DROPOUT	0
SAMPLE	
LIBRARY	
READ_GROUP	

http://broadinstitute.github.io/picard/picard-metric-definitions.html#HsMetrics

Coverage using bedtools

- coverageBed -b file.bam \
- -a targets.bed \
- > result.txt

Coverage using `bedtools`

Result:

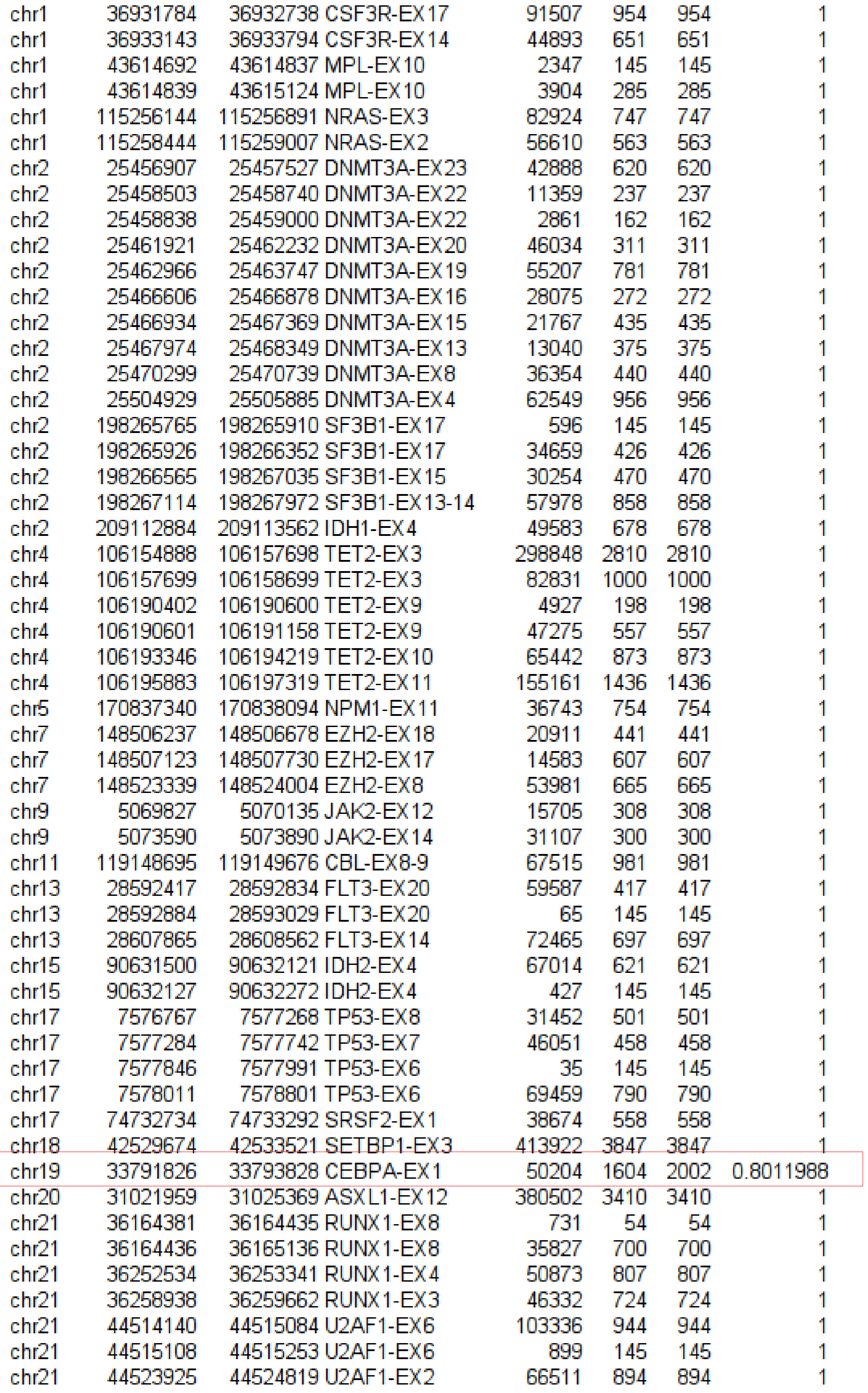
- chromosome
- target start position
- target end position
- name of target region

- **number of features** in bam file that **overlapped** (by at least one base pair) with target region

- **number of bases** in target region that had non-zero coverage by reads in bam file

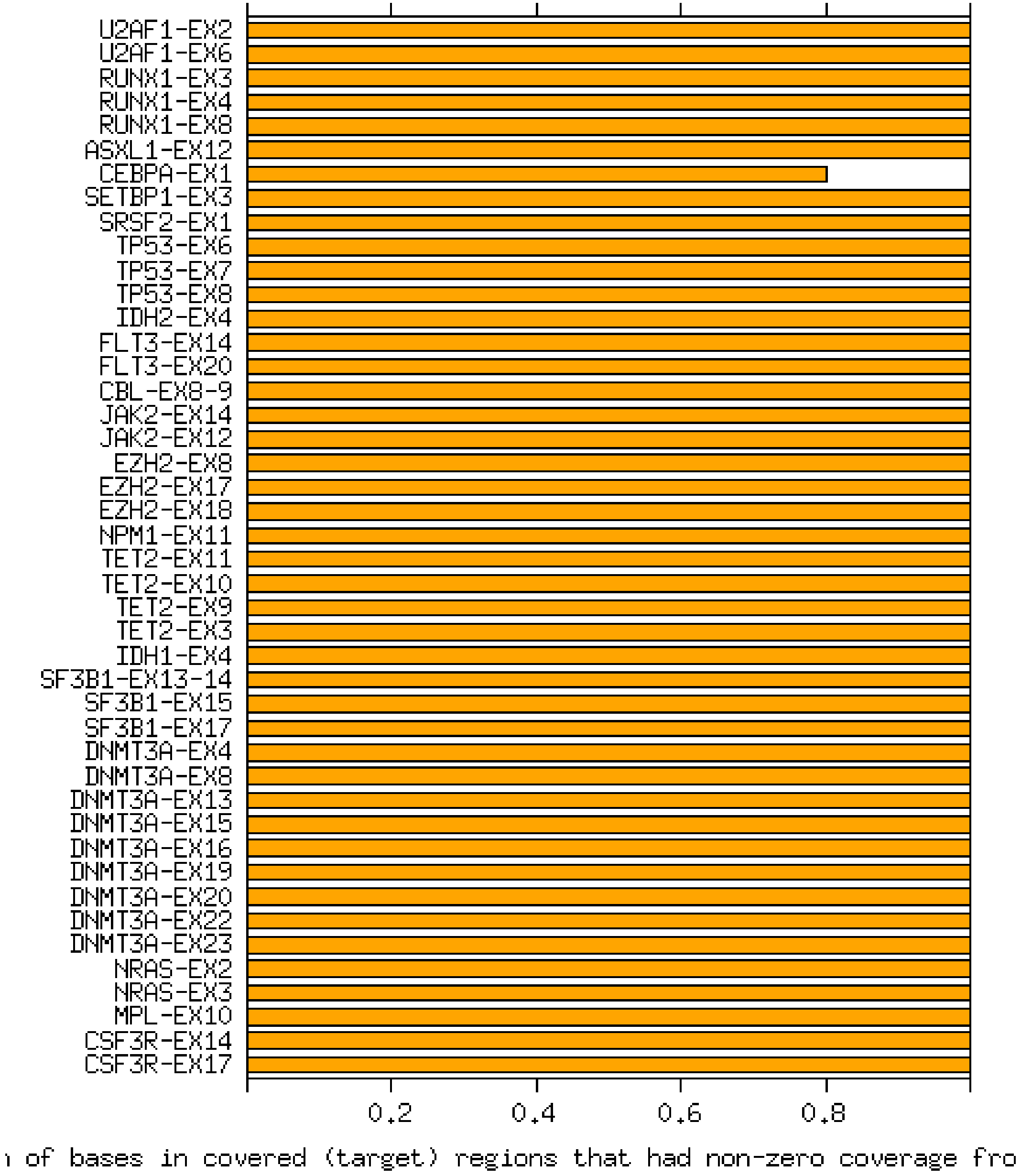
- the **length of target region**

- the **fraction of bases** in target region that had non-zero **coverage** by reads in bam file



```
chr1     36931784   36932738 CSF3R-EX17       91507   954   954          1
chr1     36933143   36933794 CSF3R-EX14       44893   651   651          1
chr1     43614692   43614837 MPL-EX10          2347   145   145          1
chr1     43614839   43615124 MPL-EX10          3904   285   285          1
chr1    115256144  115256891 NRAS-EX3         82924   747   747          1
chr1    115258444  115259007 NRAS-EX2         56610   563   563          1
chr2     25456907   25457527 DNMT3A-EX23      42888   620   620          1
chr2     25458503   25458740 DNMT3A-EX22      11359   237   237          1
chr2     25458838   25459000 DNMT3A-EX22       2861   162   162          1
chr2     25461921   25462232 DNMT3A-EX20      46034   311   311          1
chr2     25462966   25463747 DNMT3A-EX19      55207   781   781          1
chr2     25466606   25466878 DNMT3A-EX16      28075   272   272          1
chr2     25466934   25467369 DNMT3A-EX15      21767   435   435          1
chr2     25467974   25468349 DNMT3A-EX13      13040   375   375          1
chr2     25470299   25470739 DNMT3A-EX8       36354   440   440          1
chr2     25504929   25505885 DNMT3A-EX4       62549   956   956          1
chr2    198265765  198265910 SF3B1-EX17         596   145   145          1
chr2    198265926  198266352 SF3B1-EX17       34659   426   426          1
chr2    198266565  198267035 SF3B1-EX15       30254   470   470          1
chr2    198267114  198267972 SF3B1-EX13-14    57978   858   858          1
chr2    209112884  209113562 IDH1-EX4         49583   678   678          1
chr4    106154888  106157698 TET2-EX3        298848  2810  2810          1
chr4    106157699  106158699 TET2-EX3         82831  1000  1000          1
chr4    106190402  106190600 TET2-EX9          4927   198   198          1
chr4    106190601  106191158 TET2-EX9         47275   557   557          1
chr4    106193346  106194219 TET2-EX10        65442   873   873          1
chr4    106195883  106197319 TET2-EX11       155161  1436  1436          1
chr5    170837340  170838094 NPM1-EX11        36743   754   754          1
chr7    148506237  148506678 EZH2-EX18        20911   441   441          1
chr7    148507123  148507730 EZH2-EX17        14583   607   607          1
chr7    148523339  148524004 EZH2-EX8         53981   665   665          1
chr9      5069827    5070135 JAK2-EX12        15705   308   308          1
chr9      5073590    5073890 JAK2-EX14        31107   300   300          1
chr11   119148695  119149676 CBL-EX8-9        67515   981   981          1
chr13    28592417   28592834 FLT3-EX20        59587   417   417          1
chr13    28592884   28593029 FLT3-EX20           65   145   145          1
chr13    28607865   28608562 FLT3-EX14        72465   697   697          1
chr15    90631500   90632121 IDH2-EX4         67014   621   621          1
chr15    90632127   90632272 IDH2-EX4           427   145   145          1
chr17     7576767    7577268 TP53-EX8         31452   501   501          1
chr17     7577284    7577742 TP53-EX7         46051   458   458          1
chr17     7577846    7577991 TP53-EX6            35   145   145          1
chr17     7578011    7578801 TP53-EX6         69459   790   790          1
chr17    74732734   74733292 SRSF2-EX1        38674   558   558          1
chr18    42529674   42533521 SETBP1-EX3      413922  3847  3847          1
chr19    33791826   33793828 CEBPA-EX1        50204  1604  2002  0.8011988
chr20    31021959   31025369 ASXL1-EX12      380502  3410  3410          1
chr21    36164381   36164435 RUNX1-EX8          731    54    54          1
chr21    36164436   36165136 RUNX1-EX8        35827   700   700          1
chr21    36252534   36253341 RUNX1-EX4        50873   807   807          1
chr21    36258938   36259662 RUNX1-EX3        46332   724   724          1
chr21    44514140   44515084 U2AF1-EX6       103336   944   944          1
chr21    44515108   44515253 U2AF1-EX6          899   145   145          1
chr21    44523925   44524819 U2AF1-EX2        66511   894   894          1
```

Visualization of coverage



Removing duplicates?

- Multiple reads originating from only one template (PCR effect) might be sequenced

- This interferes with variant calling statistics => remove the PCR duplicates

- However a duplicate could be PCR effect or reading same fragment twice, there is no way to tell.

- The degree to which we expect duplicate fragments is highly dependent on the **depth of the library** and the **type of library** (whole genome, exome, transcriptome, ChIP-Seq, etc.).

- We **do not remove duplicates** in amplicon based targeted sequencing – here we expect high number of duplicate reads by design!

Defining a duplicate

- Different ways of definition of duplicates:

- **Reads with identical sequence:**

+ reduces the pool of reads before mapping
– accurate removal is dependent on low error rate (if an error, then the read is not considered a duplicate)

- **Reads with the same mapping position:**

+ not influenced by error rate
– reads at the same position may come from different DNA fragments (diploid genoma and polymorphism present...)

- Paired-end reads and longer reads help to correctly identify duplicates

- The best way to deal with duplicates that correspond to PCR amplification bias is to **reduce their generation in the first place**

- **UMIs! – unique molecular identifiers**

Marking and removing duplicates

- **picard MarkDuplicates – the most recommended tool**

- samtools rmdup/rmdupse – alternative

- *"Essentially what Picard does (for pairs; single-end data is also handled) is to find the 5' coordinates and mapping orientations of each read pair. When doing this it takes into account all clipping that has taking place as well as any gaps or jumps in the alignment. You can thus think of it as determining "if all the bases from the read were aligned, where would the 5' most base have been aligned". It then matches all read pairs that have identical 5' coordinates and orientations and marks as duplicates all but the "best" pair. "Best" is defined as the read pair having the highest sum of base qualities as bases with Q >= 15."*

Indels

- **Difficult** to handle
- **Complicated** mapping and statistics
- Often necessary to perform **realignment** around indels
- Before realignment

```
ATCGATCGCTAAAAC
   GATCGCAAAA-C
```

- After realignment

```
ATCGATCGCTAAAAC
   GATCGC-AAAAC
```

Insertions/deletions – a lots of troubles

```
GCGGAGagaccaacc    GCGGAGag-accaacc
||||||          => ||||||
GCGGAGggaaccacc    GCGGAGggaacc-acc
```

```
GCGGAGagaccaacc    GCGGAGaga-ccaacc
||||||          => ||||||
GCGGAGggaaccacc    GCGGAGggaacca-cc
```

Indels – a real example

AML data, NPM1 gene, patient heterozygous for short insertion CATG:

```
wt:   ATT CAA GAT CTC TGG CAG TGG
       |||  |||  |||  |||  ||
mut: ATT CAA GAT CTC TGC ATG GCA GTG G
```

The NGS identified insertion TGCA due to incorrect alignment:

```
wt:   ATT CAA GAT CTC TGG CAG TGG
       |||  |||  |||  |||
mut: ATT CAA GAT CTC TGC ATG GCA GTG G
```

pindel

- Common variant calling tools are not designed to find long repetitive inserts!
- **Pindel** is a very good alternative – mainly for targeted sequencing data.
- http://gmt.genome.wustl.edu/packages/pindel/

- Generally, it is recommended to combine multiple tools, based on the biological questions and data type!

Realignment around indels

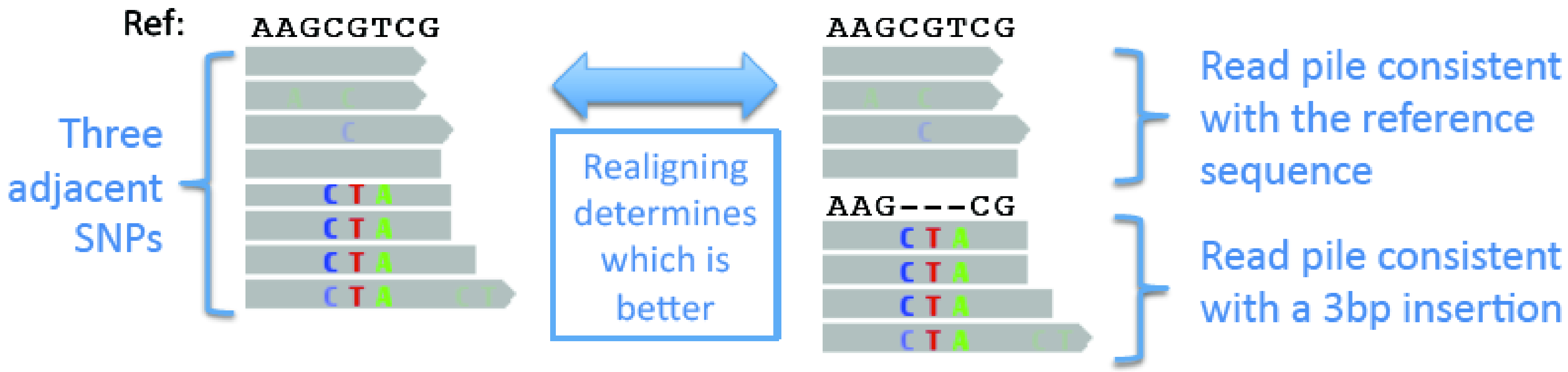
- These artifact mismatches can harm base quality recalibration and variant detection (unless a sophisticated caller like the HaplotypeCaller or MuTect2 is used)

Three types of realignment targets

- Known sites
- Indels seen in original alignments
- Sites where evidence suggests a hidden indel

Local realignment identifies most parsimonious alignment along all reads at a problematic locus

1. Find the best alternate consensus sequence that, together with the reference, best fits the reads in a pile (maximum of 1 indel)



Ref: AAGCGTCG

Three adjacent SNPs

Realigning determines which is better

AAGCGTCG

AAG---CG

Read pile consistent with the reference sequence

Read pile consistent with a 3bp insertion

2. The score for an alternate consensus is the total sum of the quality scores of mismatching bases

3. If the score of the best alternate consensus is sufficiently better than the original alignments (using a LOD score), then we accept the proposed realignment of the reads

https://www.broadinstitute.org/gatk/events/slides/1212/GATKwh0-BP-2-Realignment.pdf

Did the realignment work properly?

- Indel Realigner changes the CIGAR string of realigned reads but maintains the original CIGAR (with OC tag)
  - So it's very easy to check that realignment was performed and/or how many reads were adjusted
- BUT no formal measure to assess the accuracy or completeness of the realignment process

https://www.broadinstitute.org/gatk/events/slides/1212/GATKwh0-BP-2-Realignment.pdf

Realignment around indels

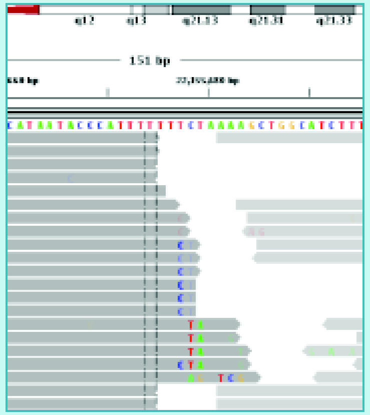
- **GATK**
- 2 step process:
- Determining (small) suspicious intervals which are likely in need of realignment (RealignerTargetCreator tool)
- Running the realigner over those intervals (IndelRealigner)

- **SMRA** – realigning reads in color space (SOLiD)

Indel Realignment steps/tools

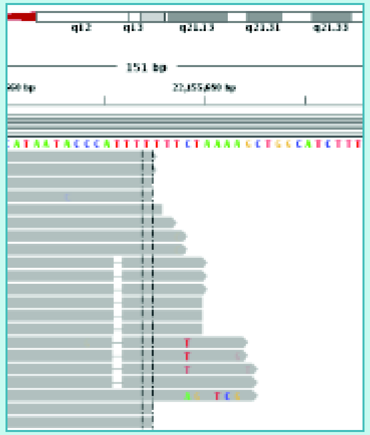
- Identify what regions need to be realigned

  ➔ **RealignerTargetCreator**



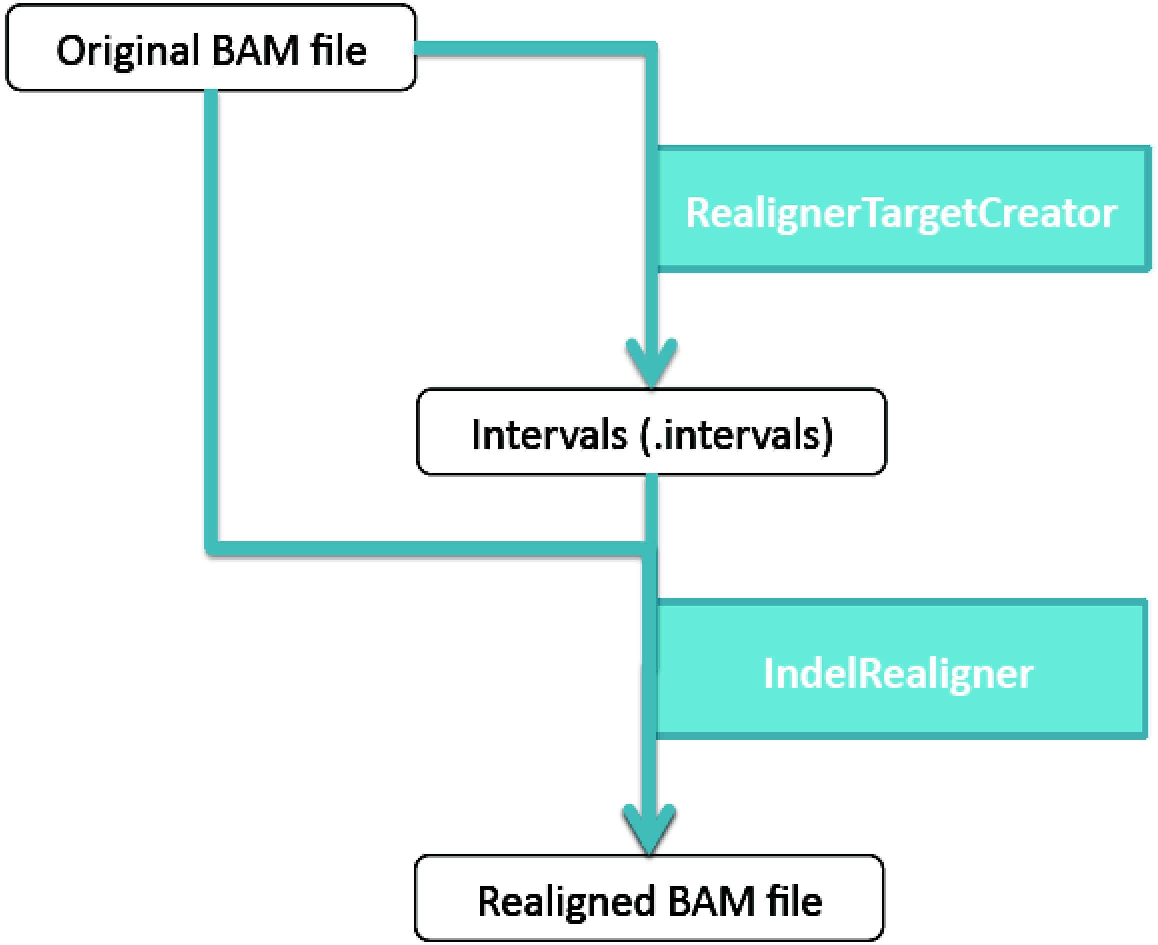
- Perform the actual realignment

  ➔ **IndelRealigner**



https://www.broadinstitute.org/gatk/events/slides/1212/GATKwh0-BP-2-Realignment.pdf

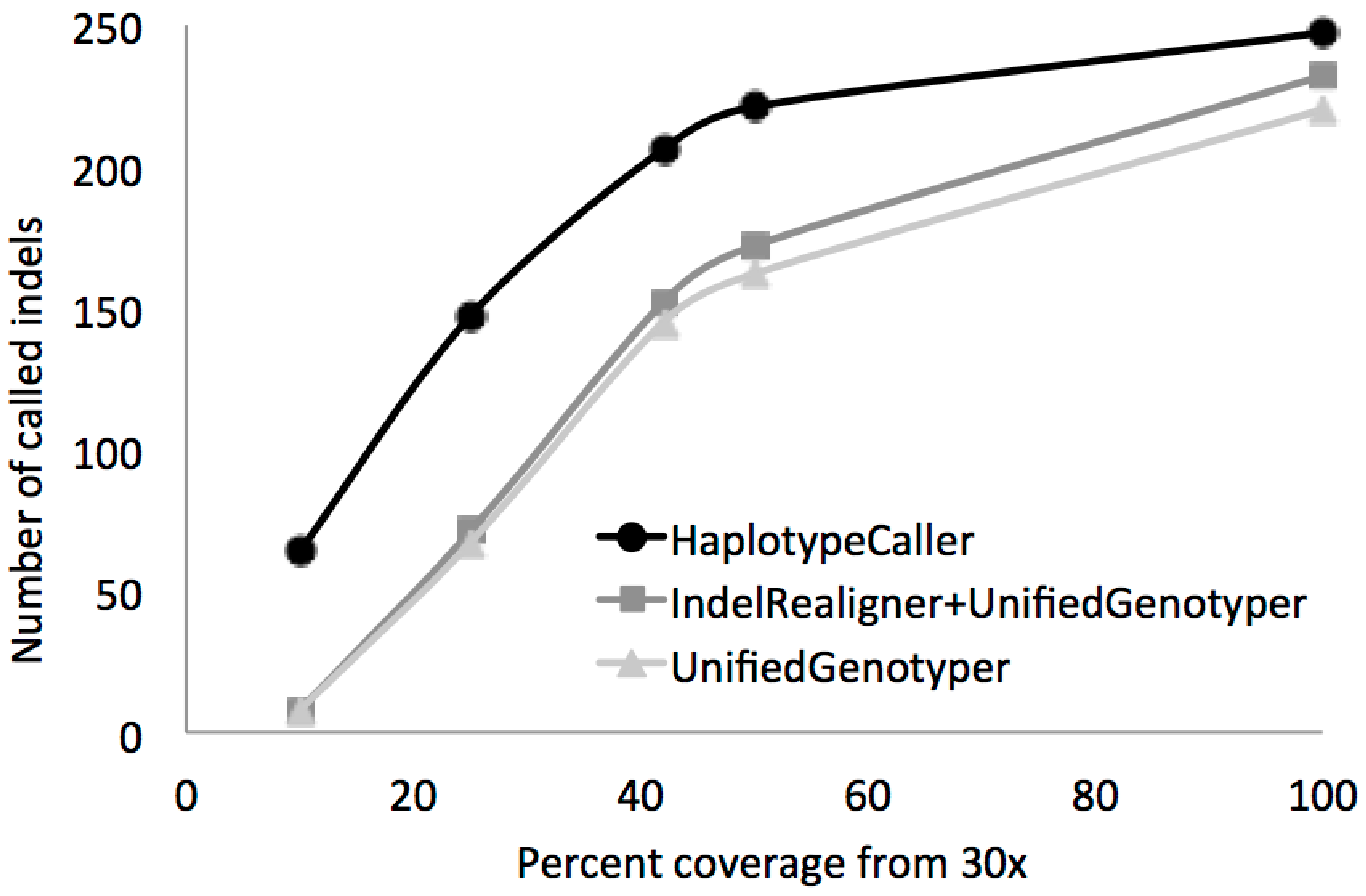
Indel Realignment workflow



https://www.broadinstitute.org/gatk/events/slides/1212/GATKwh0-BP-2-Realignment.pdf

Illustration B.2: HaplotypeCaller boosts indel detection.

Data is for the 1 Mbp region between 10:96,000,000-97,000,000, for the PCR-free WGS NA12878 sample described in Tutorial#7156 that has duplicates already marked. The 30x coverage depth BAM was downsampled using PrintReads and the –dfrac parameter set to 0.50, 0.42, 0.25 and 0.10. Downsampled data was used directly with HaplotypeCaller, UnifiedGenotyper or with indel realignment then UnifiedGenotyper. Number of indels called is according to the *n_indel* metric of VariantEval's IndelSummary module. @shlee May 2016



https://gatkforums.broadinstitute.org/gatk/discussion/7847

Long indels

- However – GATK does not call well longer indels! - these analyzed by **pindel**, which needs realignment!

Recalibration of base quality scores

- Phred-like quality scores issued by the sequencing platforms may often deviate from the true error rate
- Having accurate quality scores is essential for the modern SNP calling algorithms, as they **integrate the Phred scores** of the bases covering the site to be examined into their **scoring function**
- **Main idea: estimate true base quality based on known sites without SNPs**
- Recalibration tools/approaches:

- SOAPsnp (http://soap.genomics.org.cn/soapsnp.html)
- GATK, ...

Recalibration of quality scores

- All recalibrating algorithms use a **comprehensive database of known SNPs.**

- If no such database is available, one can first identify candidate polymorphic sites that are highly likely to be real and use the remaining sites for the recalibration procedure – in this case, **another round of SNP calling** should be performed with recalibrated quality scores.

- Also – in **targeted sequencing –** most of the targets is expected to have variants. In this case, the recalibration is not recommended (e.g. true SNPs would be evaluated as mismatch rate)

SOAPsnp

- Exploits sites in the **reference genome** without any reported SNPs. On these sites, it computes the empirical mismatch rate as an estimate for the true base quality.

- For a:
- given machine provided quality score
- sequencing cycle (position of base in the read) and
- substitution type (e.g. A->G: A in reference and G in read)
- it **calculates the average mismatch rate** with respect to the **reference genome**

- This mismatch rate is then used as the **recalibrated quality score** by adding to the raw quality scores the residual differences between empirical quality scores and the mismatch rates implied by the raw quality scores

GATK

- **Similar concept as SOAPsnp:**

- 1. bases are grouped with respect to several features (raw quality, dinucleotide content)
- 2. empirical mismatch rate is computed and used to correct the raw quality score

Additional resources

- Edge effects in calling variants from targeted amplicon sequencing | BMC Genomics (springer.com)
- Low concordance of multiple variant-calling pipelines: practical implications for exome and genome sequencing | Genome Medicine | Full Text (biomedcentral.com)
- Systematic benchmark of state-of-the-art variant calling pipelines identifies major factors affecting accuracy of coding sequence variant discovery | BMC Genomics (springer.com)
- Toward better understanding of artifacts in variant calling from high-coverage samples | Bioinformatics | Oxford Academic (oup.com)