

Lesson 05

1. Secondary DNA analysis
2. Variant calling
3. Cancer/virus analysis

Exercise 1

The screenshot shows a web-based application interface for managing tasks and projects. At the top, there is a navigation bar with links for Projects, Data, Public Apps, Public projects, Automations, Developer, Staff, a notification bell, and a user account named vladimirk. Below the navigation bar, there is a secondary menu with links for Dashboard, Files, Apps, and Tasks, where Tasks is currently selected. The main content area displays a completed task titled "FastQC run - 02-18-20 15:08:07". The task status is "COMPLETED" and it was executed on Feb. 18, 2020 at 16:08 by tamaram. It includes details about spot instances, memoization, price (\$0.01), and duration (2 minutes). The task details section shows the app used was FastQC (Revision: 2). The "Inputs" section lists two FASTQ files: G26234.HCC1187.2.converted.pe_1_1Mreads.fastq and G26234.HCC1187.2.converted.pe_2_1Mreads.fastq. The "App Settings" section shows the number of threads set to 1. The "Outputs" section lists one Report HTML file (G26234.HCC1187.2.converted.pe_1_1Mreads_fastqc.html) and one Report ZIP file (G26234.HCC1187.2.converted.pe_1_1Mreads_fastqc.zip).

project ⓘ

Interactive Analysis Notes

COMPLETED **FastQC run - 02-18-20 15:08:07**

Executed on Feb. 18, 2020 16:08 by tamaram

Spot Instances: On ⓘ | Memoization: On ⓘ | Price: \$0.01 ⓘ Refund View refunds | Duration: 2 minutes ⓘ

App: FastQC - Revision: 2

Inputs ⚙️

Input FASTQ Files ⓘ ⚙️

G26234.HCC1187.2.converted.pe_1_1Mreads.fastq
G26234.HCC1187.2.converted.pe_2_1Mreads.fastq

App Settings

Number of Threads ⓘ 1

Show all ⓘ

Outputs ⚙️

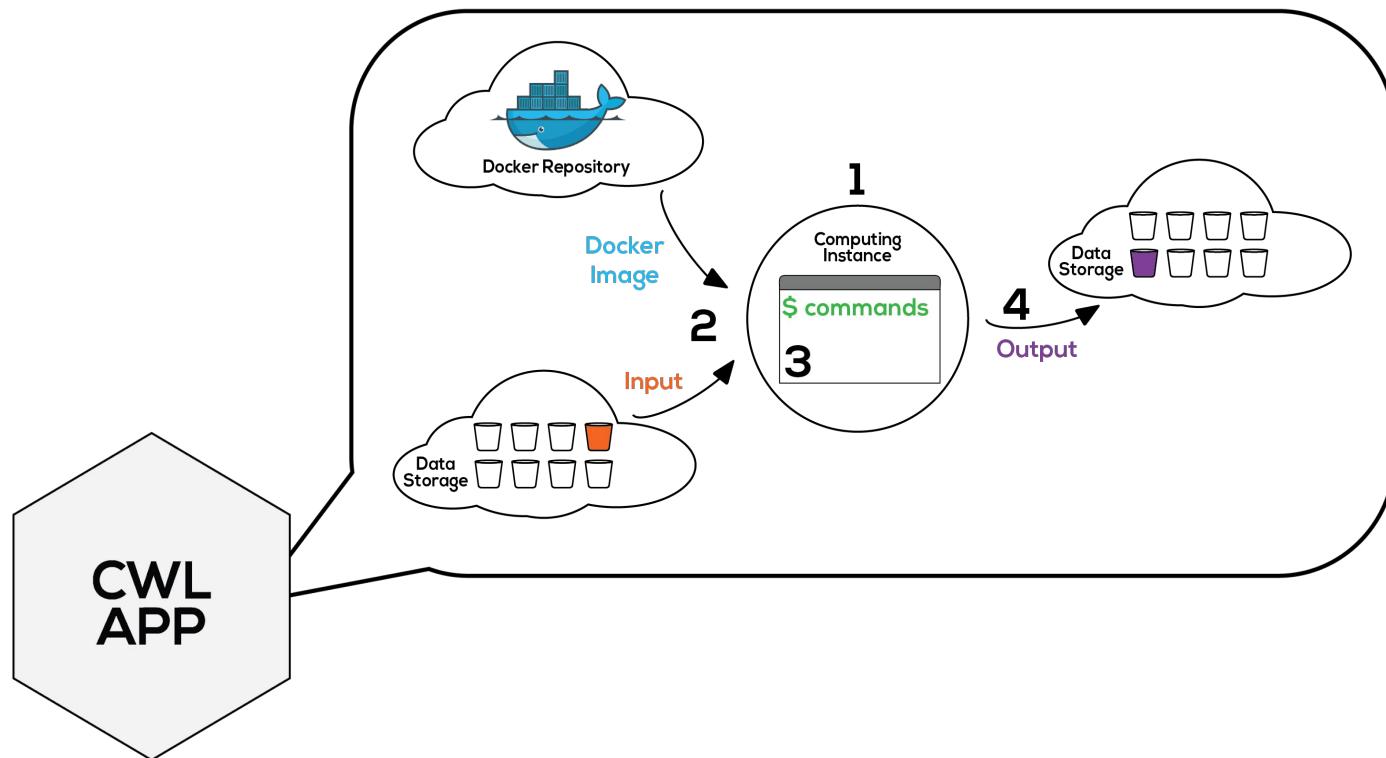
Report HTML ⓘ ⚙️

G26234.HCC1187.2.converted.pe_1_1Mreads_fastqc.html
G26234.HCC1187.2.converted.pe_2_1Mreads_fastqc.html

Report ZIP ⓘ ⚙️

G26234.HCC1187.2.converted.pe_1_1Mreads_fastqc.zip
G26234.HCC1187.2.converted.pe_2_1Mreads_fastqc.zip

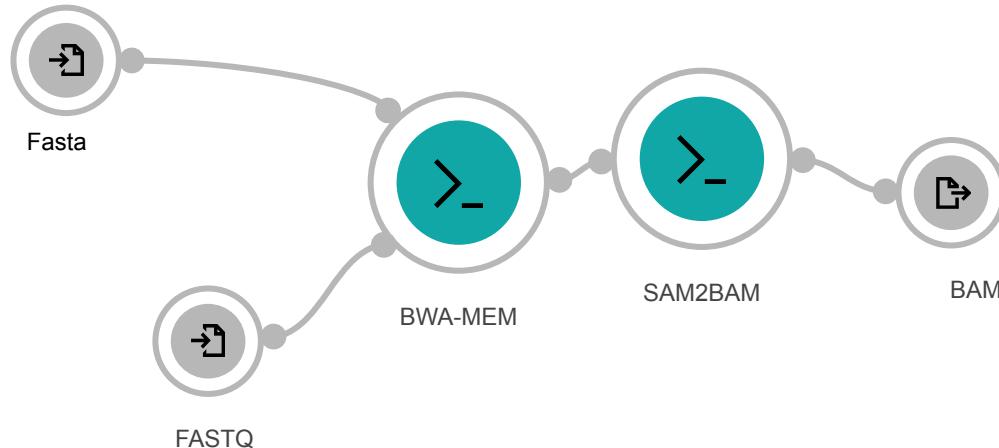
CWL - Recap



What is a workflow? - Recap

- Acyclic graph of tools connected to perform some analysis
- Workflow's nodes are:
 - Inputs (file or parameter)
 - Tools
 - Outputs
 - Workflow

```
bwa mem ref.fa read1.fq read2.fq >  
aln.sam  
sam2bam aln.sam > aln.bam
```

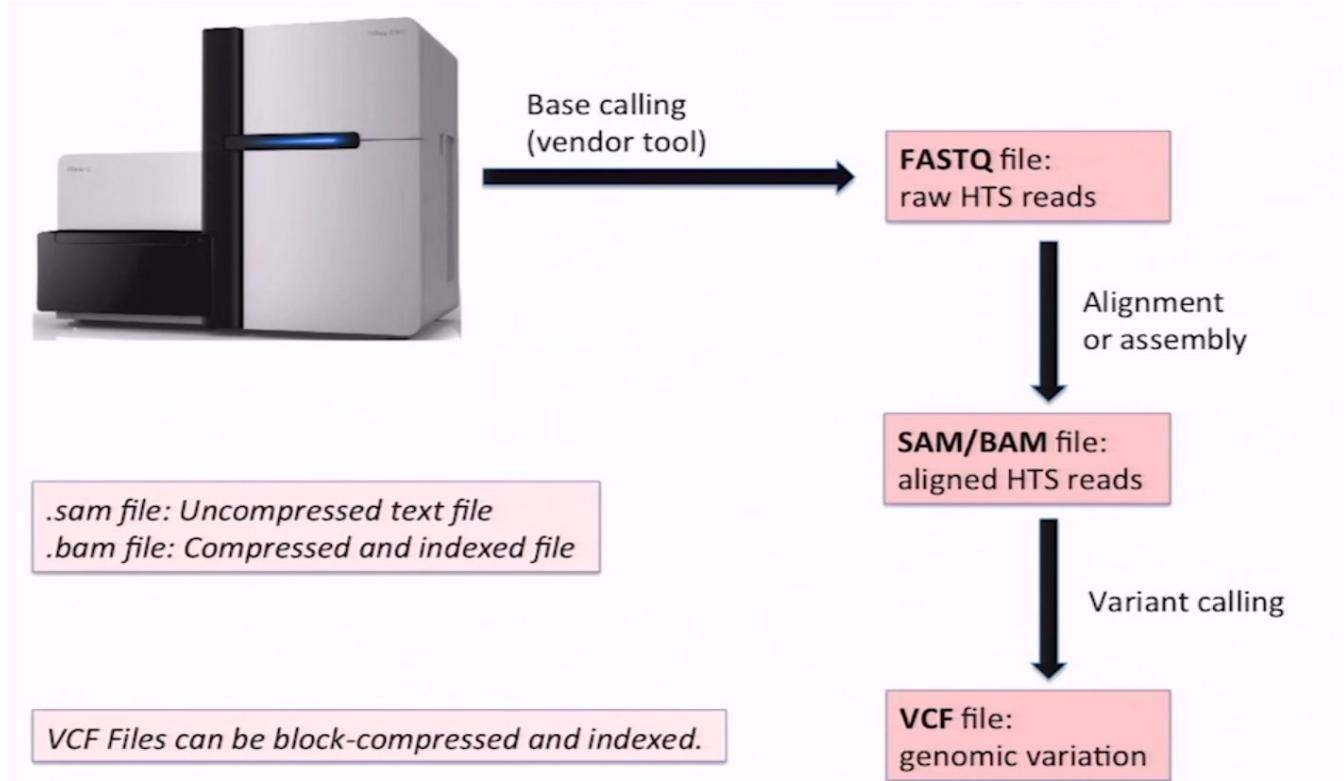


Serbian bioinformatics landscape

- [Velsera](#) (Seven Bridges)
- [BGI-Research](#) (MGI) - Omics for all
- [Absci](#) (Designing antibodies from scratch using AI)
- [Institut za veštačku inteligenciju](#) Srbije
- [APIS Assay Technologies](#) (Breast Cancer Subtyping Kit)
- [Persida Bio](#) (Diagnostic grade genome visualization, annotation and interpretation tool)
- [Centogene](#) (rare diseases)
- [Natera](#) (cell-free DNA tests help inform more personalized health care decisions)
- [Genial](#) - genetic tests

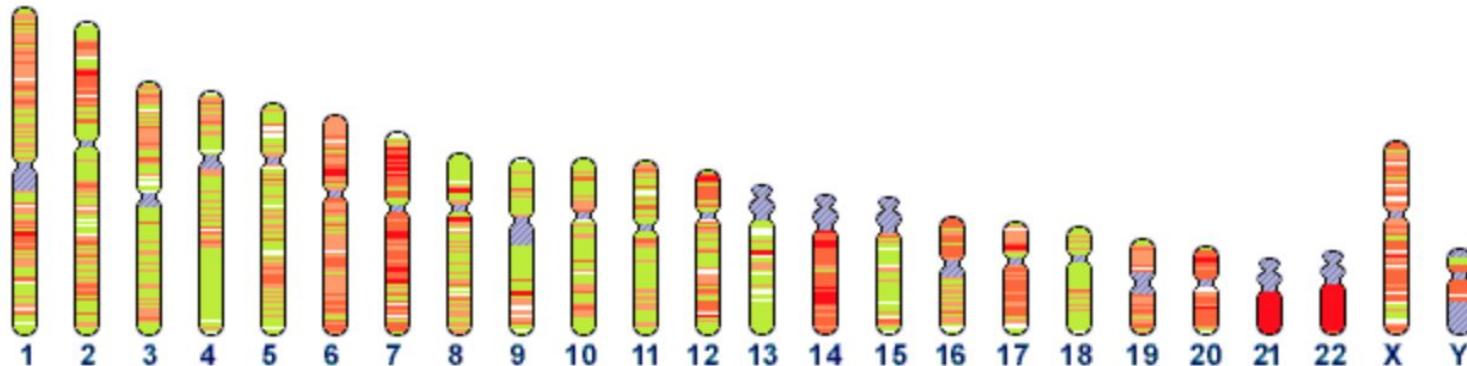
1. Secondary DNA analysis.

Secondary DNA analysis



Human genome stats

- The human genome has 23 pairs of chromosomes:
 - Males have 22 autosomes, one X and one Y chromosome
 - Females have 22 autosomes and two X chromosomes
- There is also a separate Mitochondrial DNA contig
- Total length of reference human genome is ~3 Gigabases (3B basepairs)
- About 100 000 genes covering about 30 Megabases



Human genome HG38

- Current version of the chromosomes is *HG38* (*well HG39 is almost out*) (though 37 is still widely used)
- Released December 24th 2013
- Additional sequences are added to the genome:
 - Unplaced sequences (some genomes contain them, somewhere)
 - Unlocalized sequences (chromosome known, but coordinates are not)
 - Alternate sequences (some genomes contain them, instead of something parts)
 - Human Herpesvirus 4 type 1
- Patches are commonly added
- <http://www.ncbi.nlm.nih.gov/projects/genome/assembly/grc/human/>

FASTA file format

- A simple format for storing reference files
- Two “types” of lines:
 - Description lines, start with ‘>’, contain sequence name and optional description
 - Sequence lines follow a description line and contain the actual nucleotide sequence
- Sequences are represented by a single description line,
followed by one or more sequence lines (usually 70 bases or less in a line)

```
>chr 1
ACTGCCCTCCGTACCTACTGCCCTCCGTACCTACTGCCCTCCGTACCT
ACTGCCCTCCGTACCTACTGCCCTCCGTACCTACTGCCCTCCGTACCT
ACTGCCCTCCGTACCTACTGCCCTCCGTACCTACTGCCCTCCGTACCT
ACTGCCCTCCGTACCTACTGCCCTCCGTACCTACTGCCCTCCG
```

FASTA File IUPAC Codes (and contig names)

- A, C, T, G, U - Nucleotides (Adenine, Cytosine, Guanine, Thymine, Uracil)
- Ambiguous bases:
 - R - A or G
 - Y - C, T, or U
 - N - Any base
 - K, M, S, W, B, D, H, V, X, - https://en.wikipedia.org/wiki/FASTA_format

- Often two versions
 - Chromosome 1, 2, 3, ...
 - Chromosome chr1, chr2, chr3, ...
- Incompatibility between reference genomes - MOST common issue in bioinformatics: 1,2,3,... vs. chr1, chr2, chr3,...
- some are found
Consortium style)
style)

FASTA index (fai)

- Some tools build or require indices of the fasta file
 - Needs to be in the same folder, and have the same name as the fasta + .fai extension
- Fai file structure:
 1. The name of the sequence
 2. The length of the sequence
 3. The offset of the first base in the file
 4. The number of bases in each fasta line
 5. The number of bytes in each fasta line

Genes in HG38 (UCSC Genome Browser)

- A FASTA file contains raw sequences of nucleotide from a genome
- Some sections of these sequences have biological meanings
- Examples:
 - Chromosome 17, 43.04 Mb - 43.13 Mb is the BRCA1 Gene, implicated in breast cancer
 - Chromosome X, 73.82 Mb - 73.85 Mb is the XIST lncRNA, implicated in X inactivation
 - Chromosome 1, 121.1 Mb - 124.3 Mb is the Chromosome 1 centromere
- UCSC Genome browser is a place where different annotation tracks can be explored in depth
- UCSC Genome browser is located at <https://genome.ucsc.edu/>

Annotations file formats

- Annotations are stored in few (similar) file types
 - **BED**
 - GTF
 - GFF
 - ...
- **BED file format** (tab-separated):

CHROM START END NAME score ticks blocks

- Chrom, start and end are required
- Start is 0-based
- End is open interval (not included)

Short Read Alignment

- For Reference genome and set of reads: Report at least one “good” alignment for each read
 - Where in the reference genome did read originate?
- What is “good”? For now, we concentrate on:
 - Fewer mismatches are better
 - Failing to align a low-quality base is better than failing to align a high-quality base

...TGATCATA... better than ...TGATCATA...
...TTTT... GATCAA ...TTT... GAGAAT

...TGATATTA... better than ...TGATcATA...
...TTT... GATcaT ...TTT... GTACAT

Penalty (score) - based alignment

- Some of the reads will match perfectly to the reference
- Many reads will not:
 - Genomic variations ('mutations', SNP, Indel, etc.)
 - Sequencing errors
- Aligners commonly take a score-based approach:
 - Calculate a score, related to the distance between the read and the local reference sequence
 - Place the read so that the score is maximised
- Many algorithms exist, there is a speed/precision tradeoff

Two-step aligners

- Finding precise alignments can be expensive (for a human genome ~1 billion reads)
- Most modern aligners take a two-step approach (also called “**seed and extend**”)
- First find “coarse” alignments or seeds
- Then do fine grain alignments in the vicinity of the seeds
- Choose the best scoring fine grain alignment

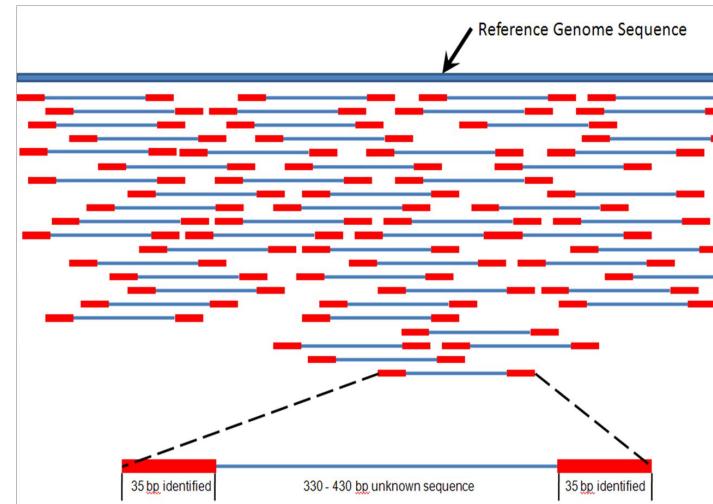
A simple coarse aligner

- Example: a simple hash-based scheme
- Create a hash-table which holds the positions, where each kmer in the genome occurs
- For each kmer in the read find the list of locations
- Regions with hits from many different kmers are hits
- Some kmers will have no hits, some many
- Indices quickly grow too large for whole genomes

BWA-MEM usage

```
bwa mem ref.fa read1.fq read2.fq > aln.sam
```

- Reference genome index must exist
- Paired-end reads
- Primary and secondary alignment (random)



SAM file format - BWA-MEM output

Line from SAM file:

```
SRR035022.2621862 163 16 59999 37 22S54M = 60102 179 CCAACCCAACCTAACCTAACCTAACCTAACCTAACCTAACCTAACCGACCCTCACCC  
>AAA=>?AA>@B@B?AABAB?AABAB?AAC@B?@AB@A?A>A@A?AAAAB??ABAB?79A?AAB;B?@?@<=8:8 XT:A:M XN:i:2 SM:i:37 AM:i:37 XM:i:0 XO:i:0 XG:i:0 RG:Z:SRR035022 NM:i:2  
MD:Z:ON0N52 OQ:Z:CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCCCCCBCC@CCCCCCCCCCCCCCCC;CCC BBC?CCC ACCACA@
```

QNAME	SRR035022.2621862
FLAG	163
RNAME	16
POS	59999
MAQ	37
CIGAR	22S54M
MRNM	=
MPOS	60102
ISIZE	179
SEQ	CCAACCCAACCTAACCTAACCTAACCTAACCTAACCTAACCTAACCTAACCGACCCTCACCC
QUAL	>AAA=>?AA>@B@B?AABAB?AABAB?AAC@B?@AB@A?A>A@A?AAAAB??ABAB?79A?AAB;B?@?@<=8:8
TAG	XT:A:M
TAG	XN:i:2
TAG	SM:i:37
TAG	AM:i:37
TAG	XM:i:0
TAG	XO:i:0
TAG	XG:i:0
TAG	RG:Z:SRR035022
TAG	NM:i:2
TAG	MD:Z:ON0N52
TAG	OQ:Z:CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCCCCCBCC@CCCCCCCCCCCCCCCC;CCC BBC?CCC ACCACA@

- 1 the read is paired in sequencing, no matter whether it is mapped in a pair
- 1 the read is mapped in a proper pair
- 0 not unmapped
- 0 mate is not unmapped
- 0 forward strand
- 1 mate strand is negative
- 0 the read is not the first read in a pair
- 1 the read is the second read in a pair

Only Read group ID
is present in every
read

SAM file format

Flag	Description
0x0001	the read is paired in sequencing, no matter whether it is mapped in a pair
0x0002	the read is mapped in a proper pair (depends on the protocol, normally inferred during alignment) ¹
0x0004	the query sequence itself is unmapped
0x0008	the mate is unmapped ¹
0x0010	strand of the query (0 for forward; 1 for reverse strand)
0x0020	strand of the mate ¹
0x0040	the read is the first read in a pair ^{1,2}
0x0080	the read is the second read in a pair ^{1,2}
0x100	the alignment is not primary (a read having split hits may have multiple primary alignment records)
0x200	the read fails platform/vendor quality checks
0x400	the read is either a PCR duplicate or an optical duplicate

op	Description
M	Alignment match (can be a sequence match or mismatch)
I	Insertion to the reference
D	Deletion from the reference
N	Skipped region from the reference
S	Soft clip on the read (clipped sequence present in <seq>)
H	Hard clip on the read (clipped sequence NOT present in <seq>)
P	Padding (silent deletion from the padded reference sequence)

NM	i	Edit distance to the reference, including ambiguous bases but excluding clipping
OQ	Z	Original base quality (usually before recalibration). Same encoding as QUAL.
OP	i	Original mapping position (usually before realignment)
OC	Z	Original CIGAR (usually before realignment)
PG	Z	Program. Value matches the header PG-ID tag if @PG is present.
PQ	i	Phred likelihood of the template, conditional on both the mapping being correct
PU	Z	Platform unit. Value to be consistent with the header RG-PU tag if @RG is present.
Q2	Z	Phred quality of the mate/next fragment. Same encoding as QUAL.
R2	Z	Sequence of the mate/next fragment in the template.
RG	Z	Read group. Value matches the header RG-ID tag if @RG is present in the header.

Field	Regular expression	Range	Description
QNAME	[^ \t\n\r]+		Query pair NAME if paired; or Query NAME if unpaired ²
FLAG	[0-9]+	[0,2 ¹⁶ -1]	bitwise FLAG (Section 2.2.2)
RNAME	[^ \t\n\r@=]+		Reference sequence NAME ³
POS	[0-9]+	[0,2 ²⁹ -1]	1-based leftmost POSition/coordinate of the clipped sequence
MAPQ	[0-9]+	[0,2 ⁸ -1]	MAPping Quality (phred-scaled posterior probability that the mapping position of this read is incorrect) ⁴
CIGAR	([0-9]+[MIDNSHP])+ *		extended CIGAR string
MRNM	[^ \t\n\r@=]+		Mate Reference sequence NaMe; “=” if the same as <RNAME> ³
MPOS	[0-9]+	[0,2 ²⁹ -1]	1-based leftmost Mate POSition of the clipped sequence
ISIZE	-?[0-9]+	[-2 ²⁹ ,2 ²⁹]	inferred Insert SIZE ⁵
SEQ	[acgtnACGTN.=]+ *		query SEQuence; “=” for a match to the reference; n/N. for ambiguity; cases are not maintained ^{6,7}
QUAL	[!~-]+ *	[0,93]	query QUALity; ASCII-33 gives the Phred base quality ^{6,7}
TAG	[A-Z][A-Z0-9]		TAG
VTYPE	[AifZH]		Value TYPE
VALUE	[^\t\n\r]+		match <VTYPE> (space allowed)

Pysam - Python fasta interface

- **Pysam** - a Python toolkit for working with genomic files
- `pysam.Fastafile`:
 - Create a fasta file parser: `fasta = pysam.Fastafile(path_to_file)`
 - Get the sequence names in the file: `fasta.references`
 - Get the lengths of the sequences: `fasta.lengths`
 - Retrieve a (part of) sequence: `fasta.(sequence_name, [start], [stop])`
- Fasta coordinates in pysam are zero-based
 - Not true for all file types in pysam!
- Other fasta interfaces for Python exist
 - `pyfasta` - works only on fasta files
 - `biopython` - a much larger toolkit that complement sequences, etc.
 - We are describing pysam, as it covers all the file types covered in the course

Pysam

- Pysam can be used to process BAM files
- pysam.AlignmentFile
 - AlignmentFile(path_to_file)
 - Iteration through reads in BAM file:

```
for read in AlignmentFile(path_to_file):
```

- Reads are wrapped in AlignedSegment objects
 - AlignedSegment provides access to read fields and helpers
- pysam.AlignmentFile supports fetching regions
 - The BAM file needs to be sorted and BAI indexed!

Exercise 2: Pysam - CGC Interactive analysis

- Create an `AlignmentFile` object for “merged-tumor.bam” from Public files gallery
 - Take the first read from the `AlignmentFile`
 - Inspect the fields in the `AlignedSegment`
 - Inspect the flag field
- Check out the flag for some reads
- Calculate:
 - How many unmapped reads are in the file?
 - Total number of reads
 - Number of reads with mapping quality 0
 - Average mapping quality for all the reads
 - Average mapping quality if reads with 0 mapp quality are filtered out
- Send the link to Github repo with Jupyter Notebook of executed analysis to pedjao@etf.bg.ac.rs, together with name and number of index, the mail subject should be “GI2024_DZ2”. Deadline: 10th of April, 23:59

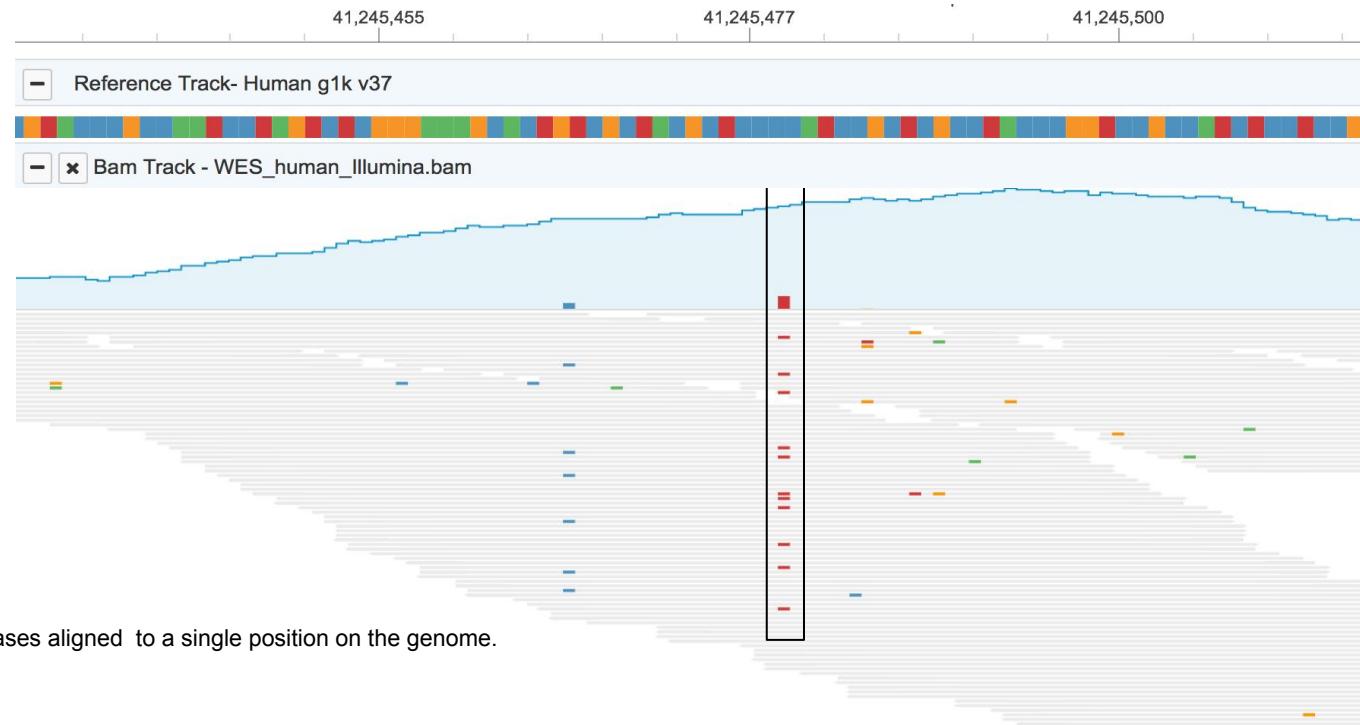
SAM file format - BWA-MEM output

coor 12345678901234 5678901234567890123456789012345
ref AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

Paired-end → r001+ TTAGATAA**AGGATA***CTG
r002+ **aaa**AGATAA***GGATA**
r003+ **gecta**AGCTAA
r004+ ATAGCT.....TCAGC
Multipart → r003- **ttagct**TAGGC
r001- CAGCGCCAT

	@SQ SN:ref LN:45
Ins & padding	r001 163 ref 7 30 8M 2I4M1D3M = 37 39 TTAGATAA AGGATA CTA *
Soft clipping	r002 0 ref 9 30 3S6M1P1I4M * 0 0 AAAAGATA AGGATA *
Splicing	r003 0 ref 9 30 5H6M * 0 0 AGCTAA * NM:i:1
Hard clipping	r004 0 ref 16 30 6M 14N5M * 0 0 ATAGCTTCAGC *
	r003 16 ref 29 30 6H5M * 0 0 TAGGC * NM:i:0
	r001 83 ref 37 30 9M = 7 -39 CAGCGCCAT *

What is a pileup?



Variant Calling

Introduction to Variant calling

- Variant calling is the process of finding differences between reference genome and observed sample
- We need aligned reads to the reference genome so we can find – “call” variants
- Different types of genomic variants

Genomic Variants

Single nucleotide variant



Deletion



Insertion



Inversion



Copy number variant



Translocation



Whole genome duplication

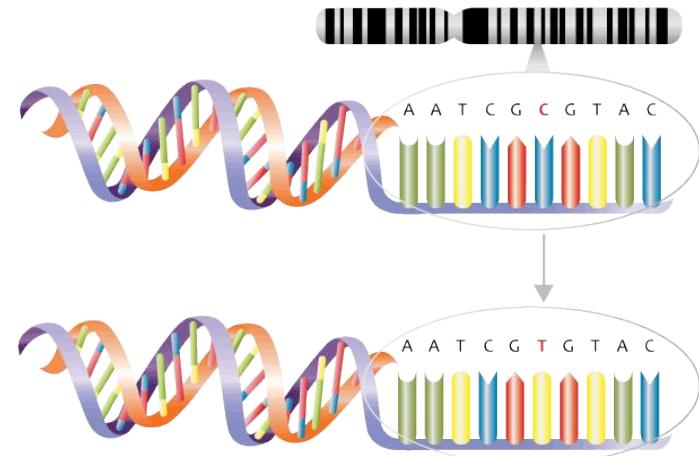


Duplication



Genomic Variants

- SNV (Single Nucleotide Variant)
- Simple ones - not a big change on the first look, but...

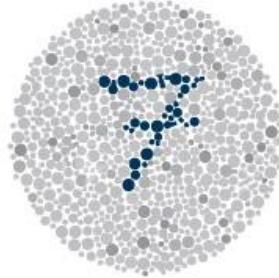


Genomic Variants

Each of those characteristics causes one SNV



LONGER EYELASHES



DALTONISM



LESS SLEEPING



SUPER STRENGTH

Genomic Variants - Breast Cancer

The BRCA2 gene provides instructions for making a protein that acts as a tumor suppressor. Tumor suppressor proteins help prevent cells from growing and dividing too rapidly or in an uncontrolled way. The BRCA2 protein is involved in repairing damaged DNA.

SNV id : rs1799954

Chromosome 13
Position 32,340,455

Cancer genotypes: CC, CT and TT

http://www.eupedia.com/genetics/cancer_related_snp.shtml

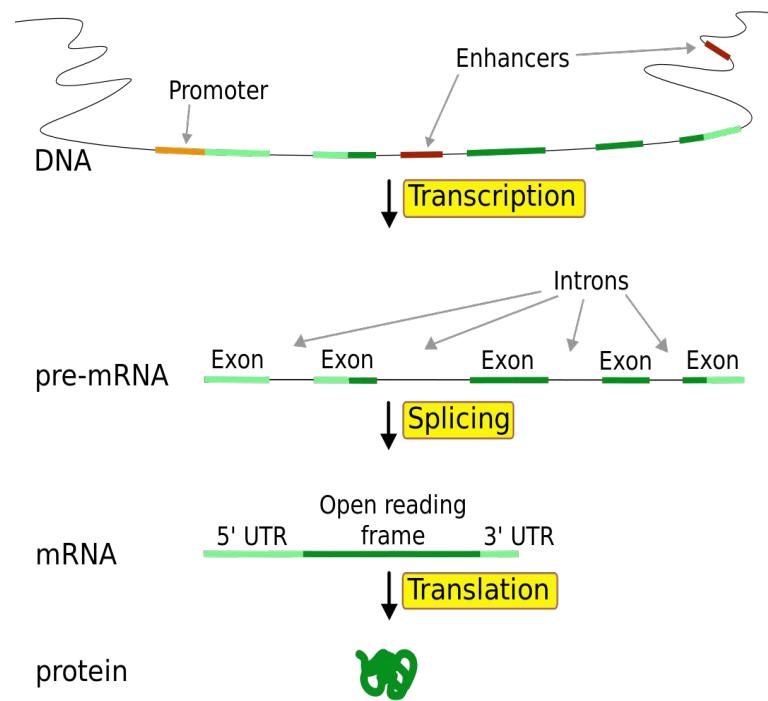
<https://www.snpedia.com/index.php/Rs1799954>

Genomic Variants

Based on the variant location,
we can predict if mutation will
have impact.

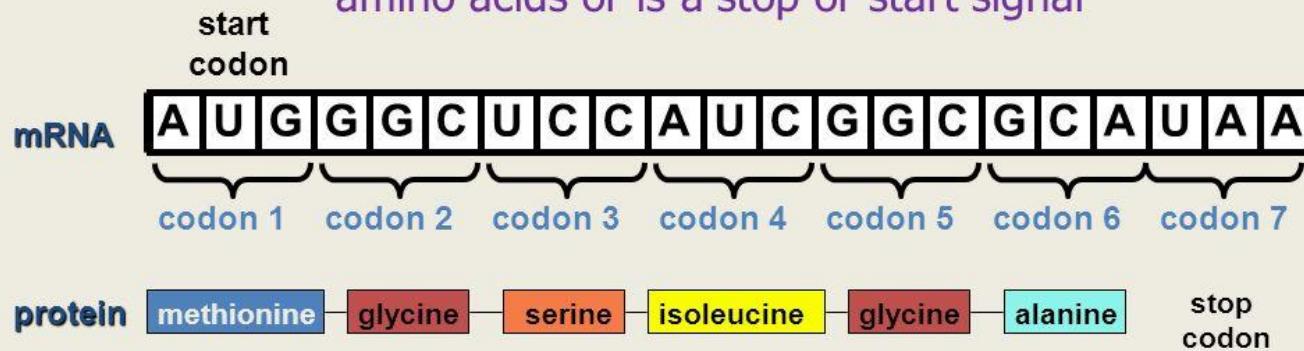


- Central dogma

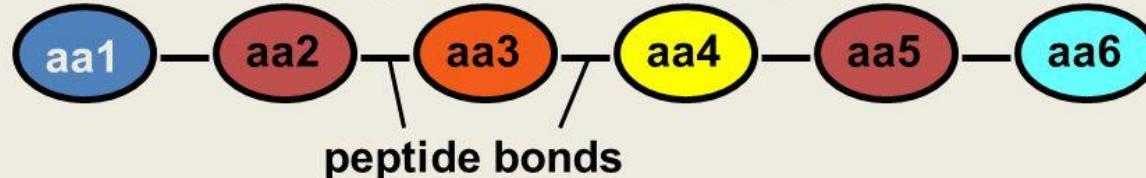


RNA to Protein

Each codon translates into one of twenty amino acids or is a stop or start signal



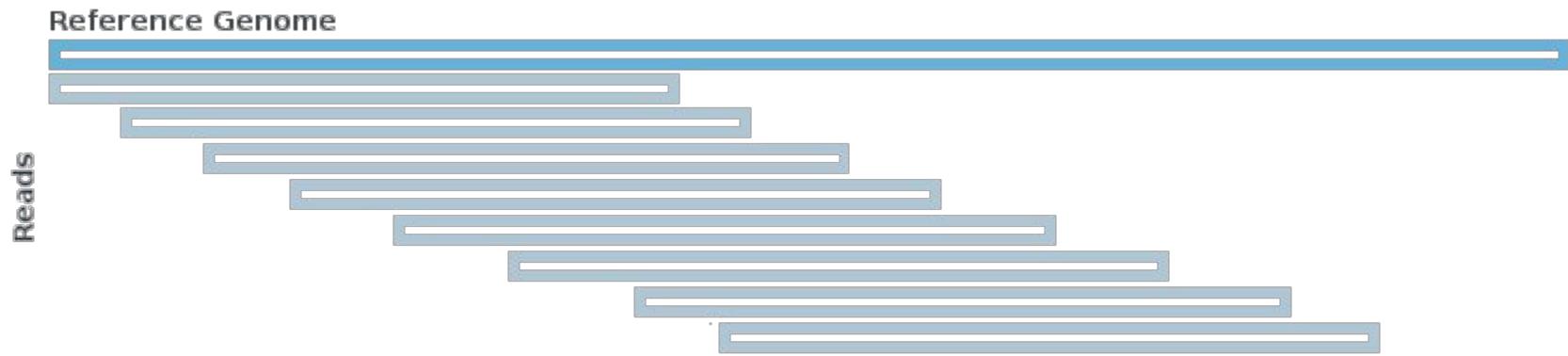
Primary structure of a protein



Genomic Variants

- Variants can have different impact on human cells and organism
- Single Nucleotide Variants(**SNV**):
 - Harmless
 - **Silent** – Usually no effect
 - Harmful:
 - **Missense** – Amino acid change
 - **Nonsense**(Start/Stop Gain/Lost) – AUG / UAG, UAA, UGA
 - Depends on the location
 - **Noncoding regions** (Promoter, Enhancer, lncRNA, miRNA...)
- Insertions/Deletions – **INDELS**
 - **In frame**
 - **Out of frame (Frameshift)**

Ideal Variant Calling



Ideal Variant Calling

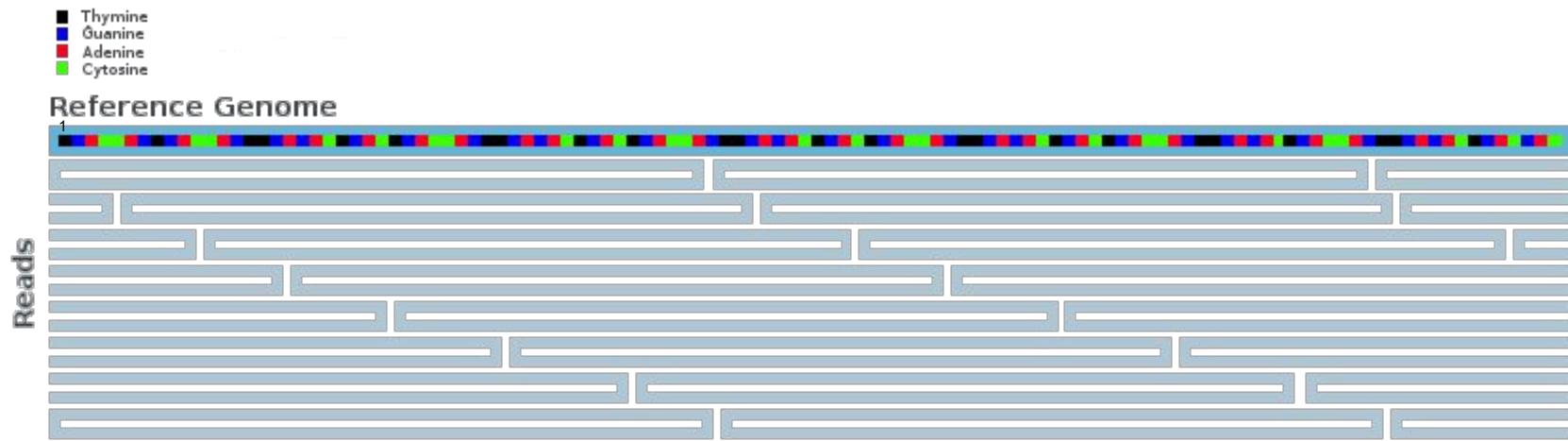


Ideally we will have uniform distribution of reads.

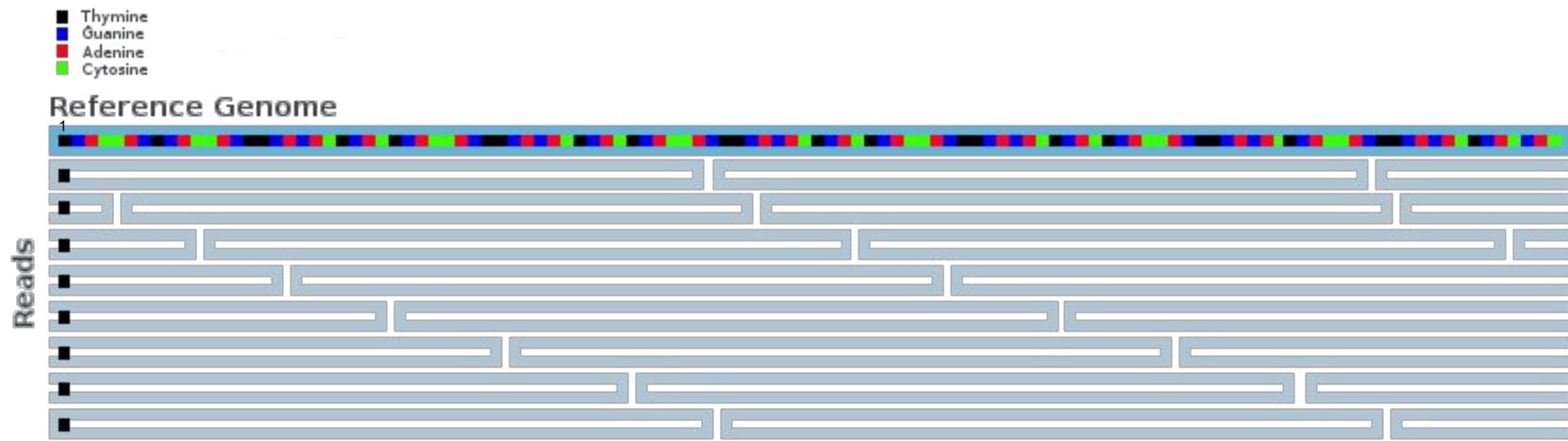
Ideal Variant Calling



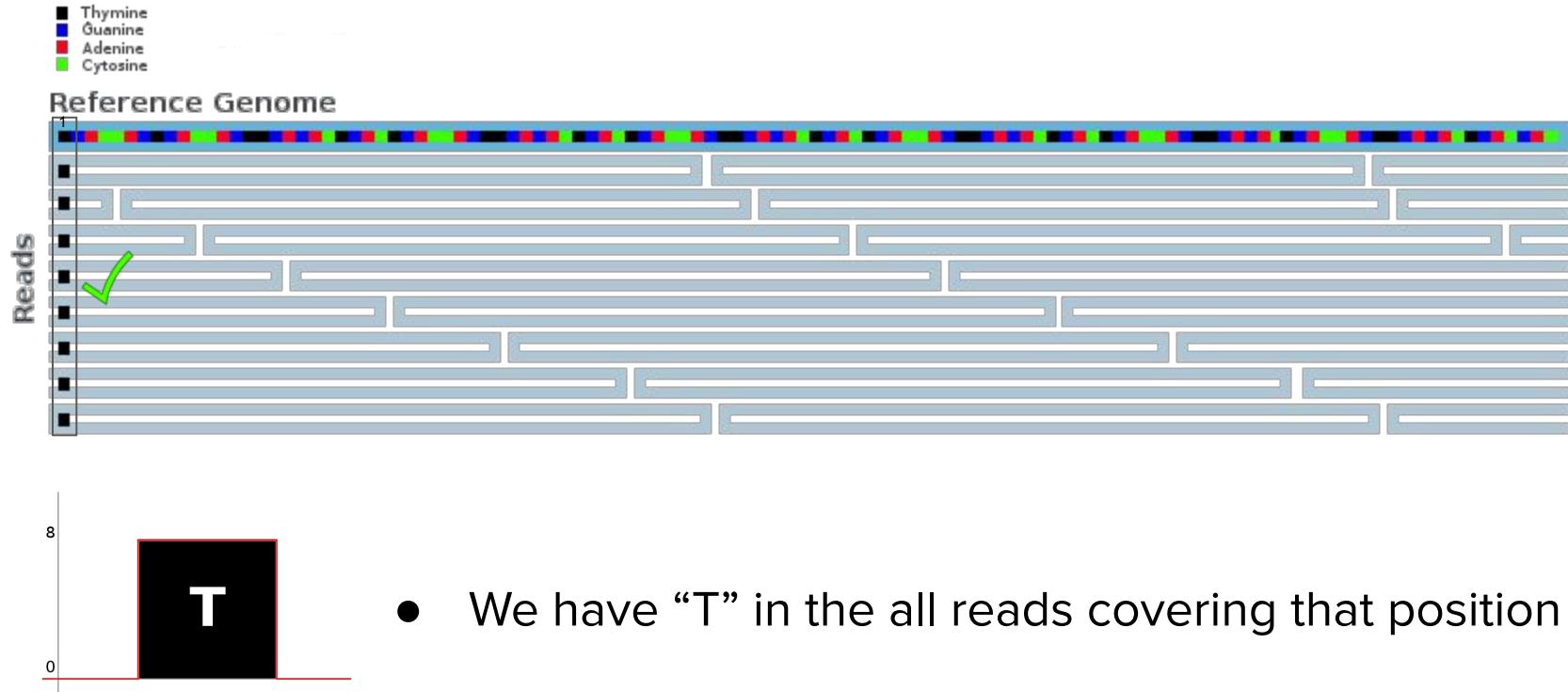
Ideal Variant Calling



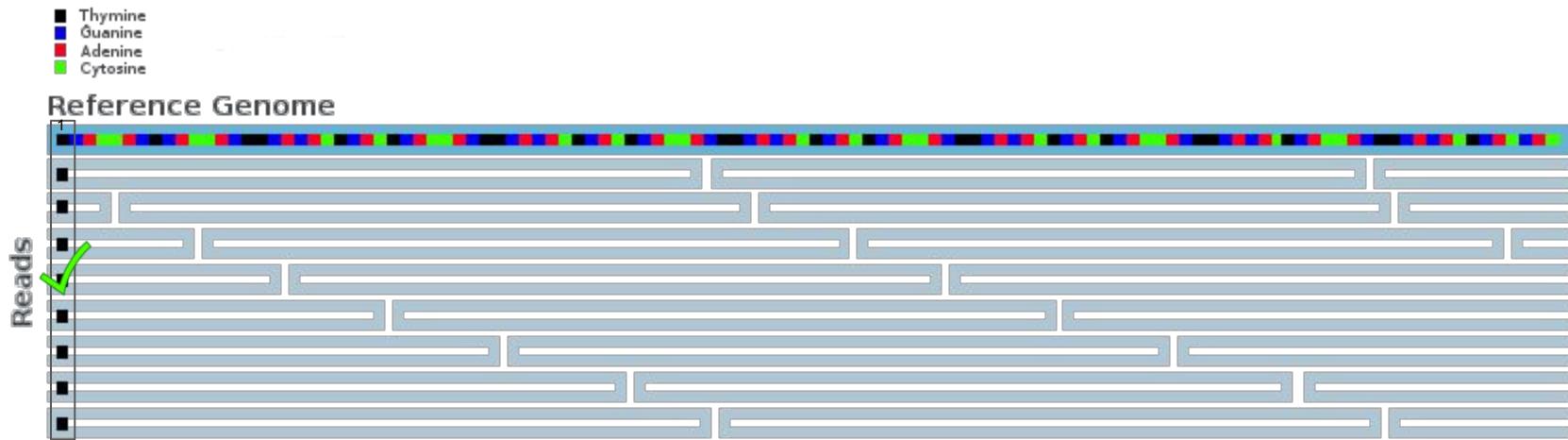
Ideal Variant Calling



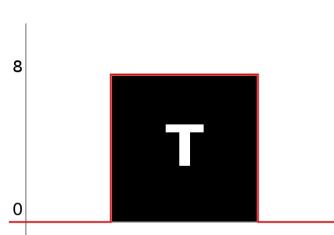
Ideal Variant Calling



Ideal Variant Calling

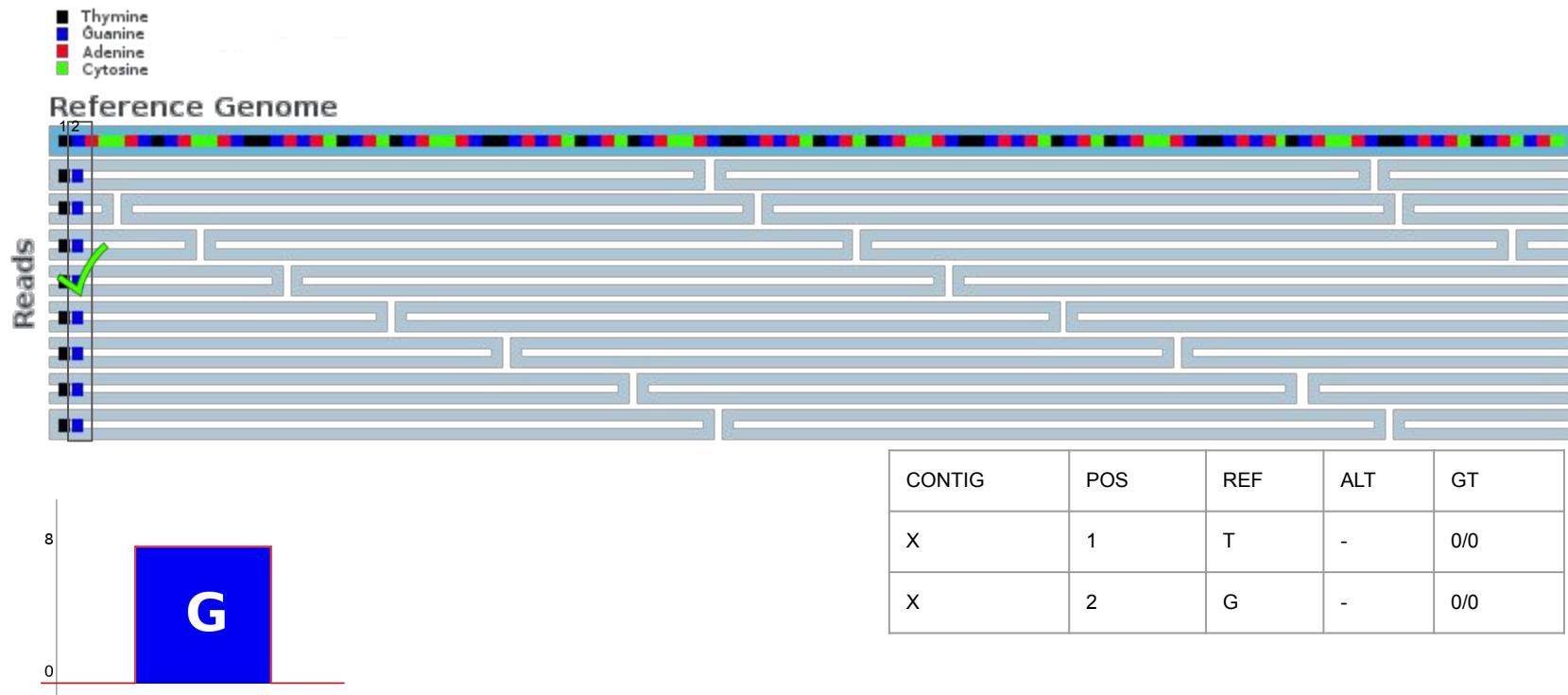


How can we represent what we have observed?

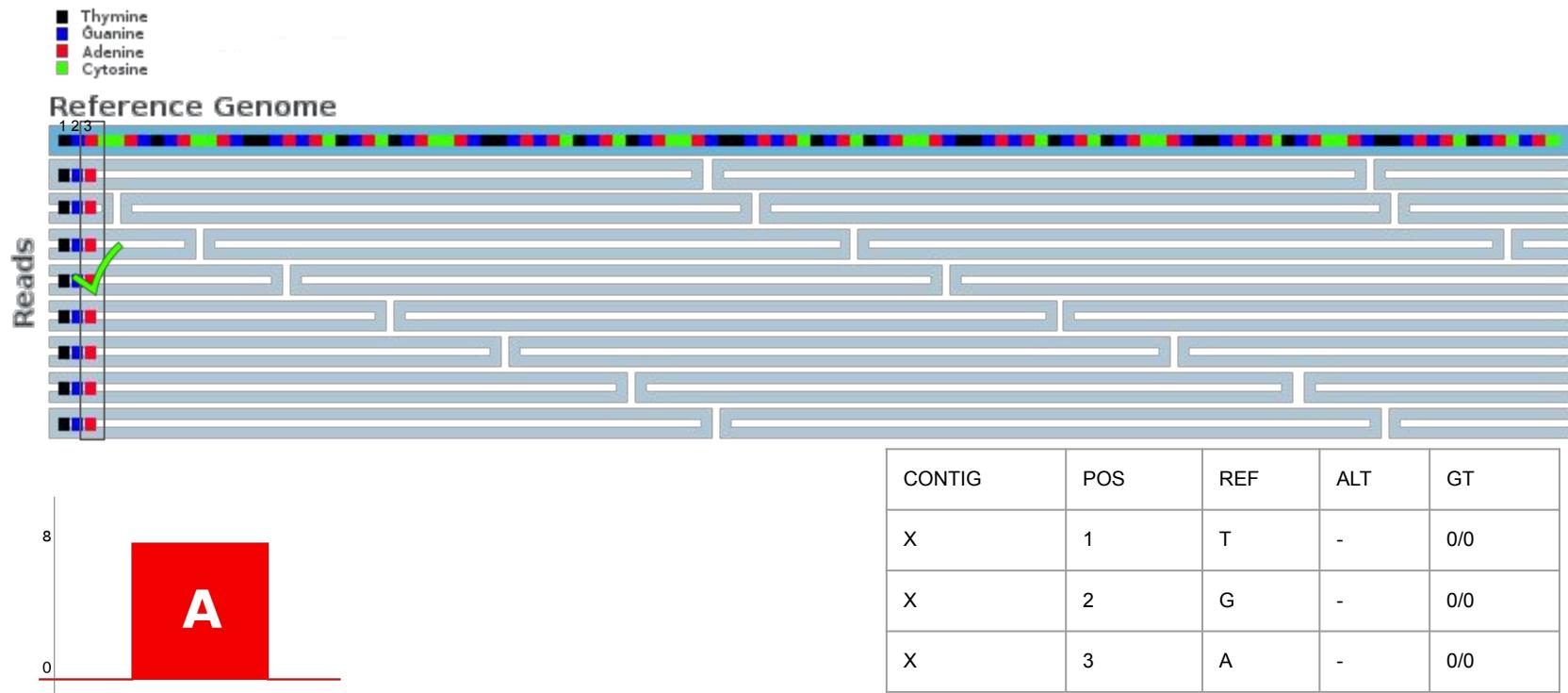


CONTIG	POS	REF	ALT	GT
X	1	T	-	0/0

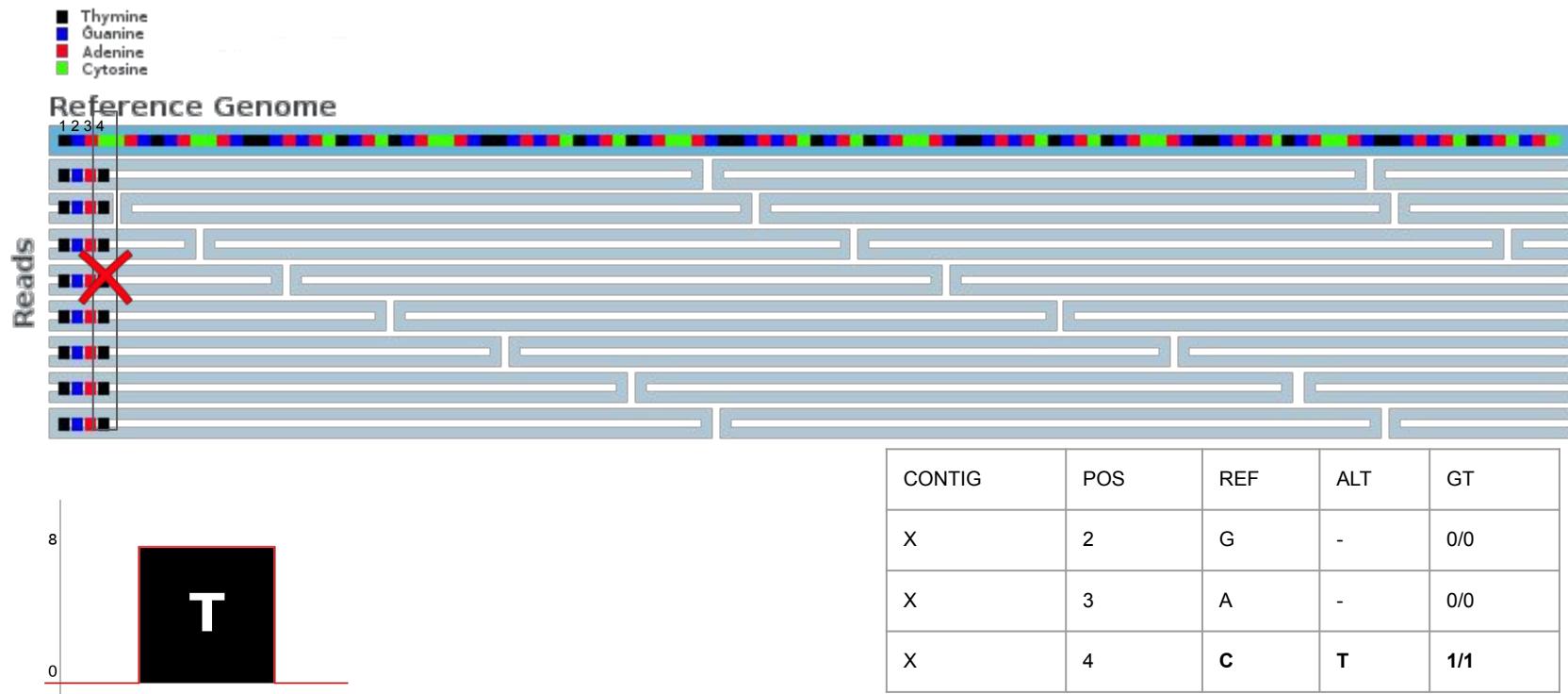
Ideal Variant Calling



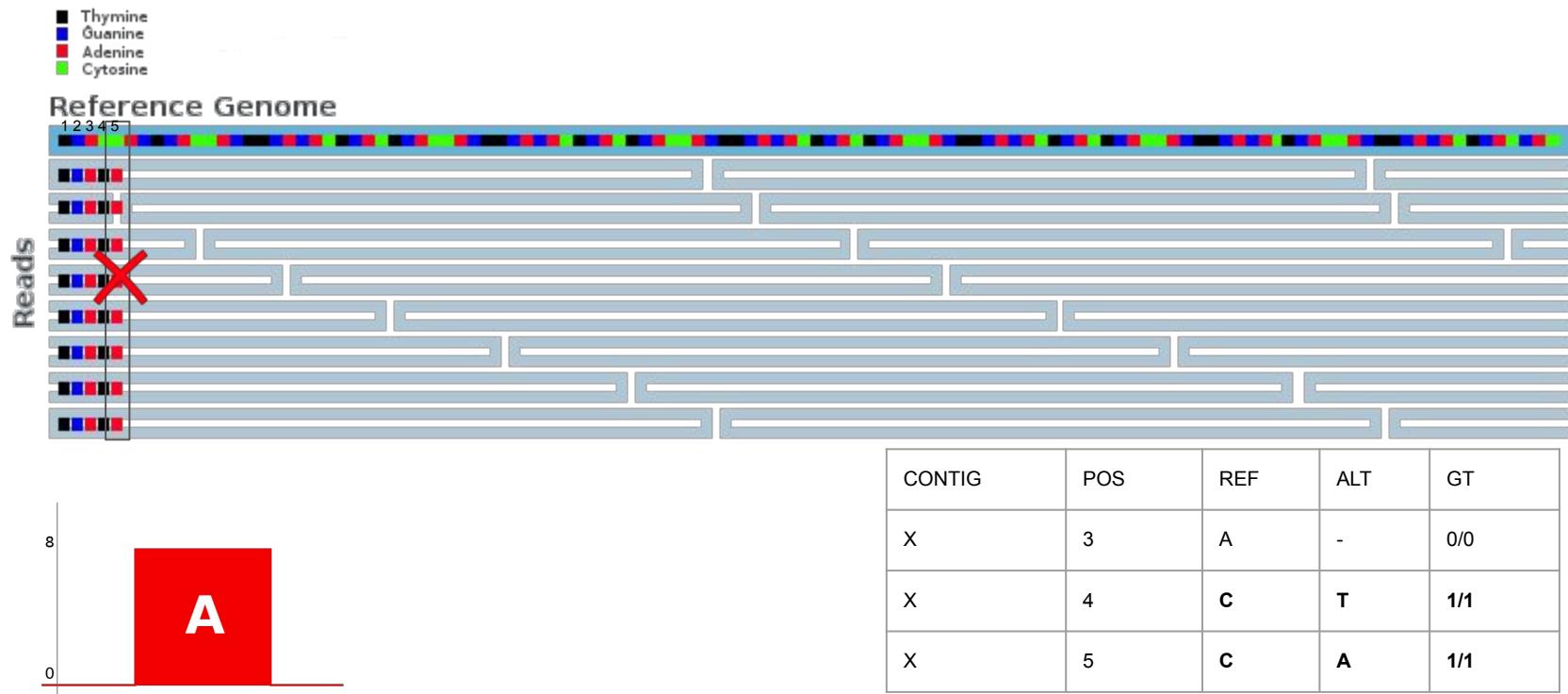
Ideal Variant Calling



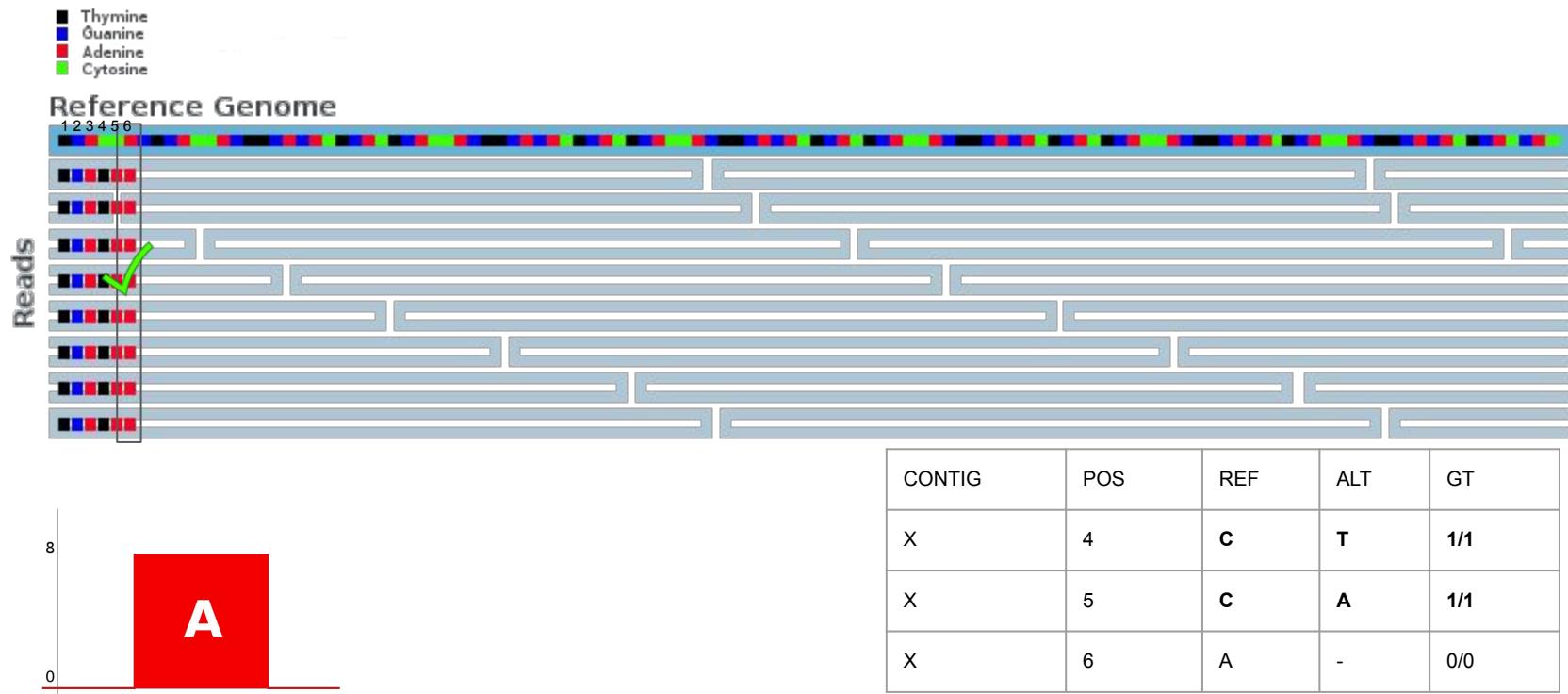
Ideal Variant Calling



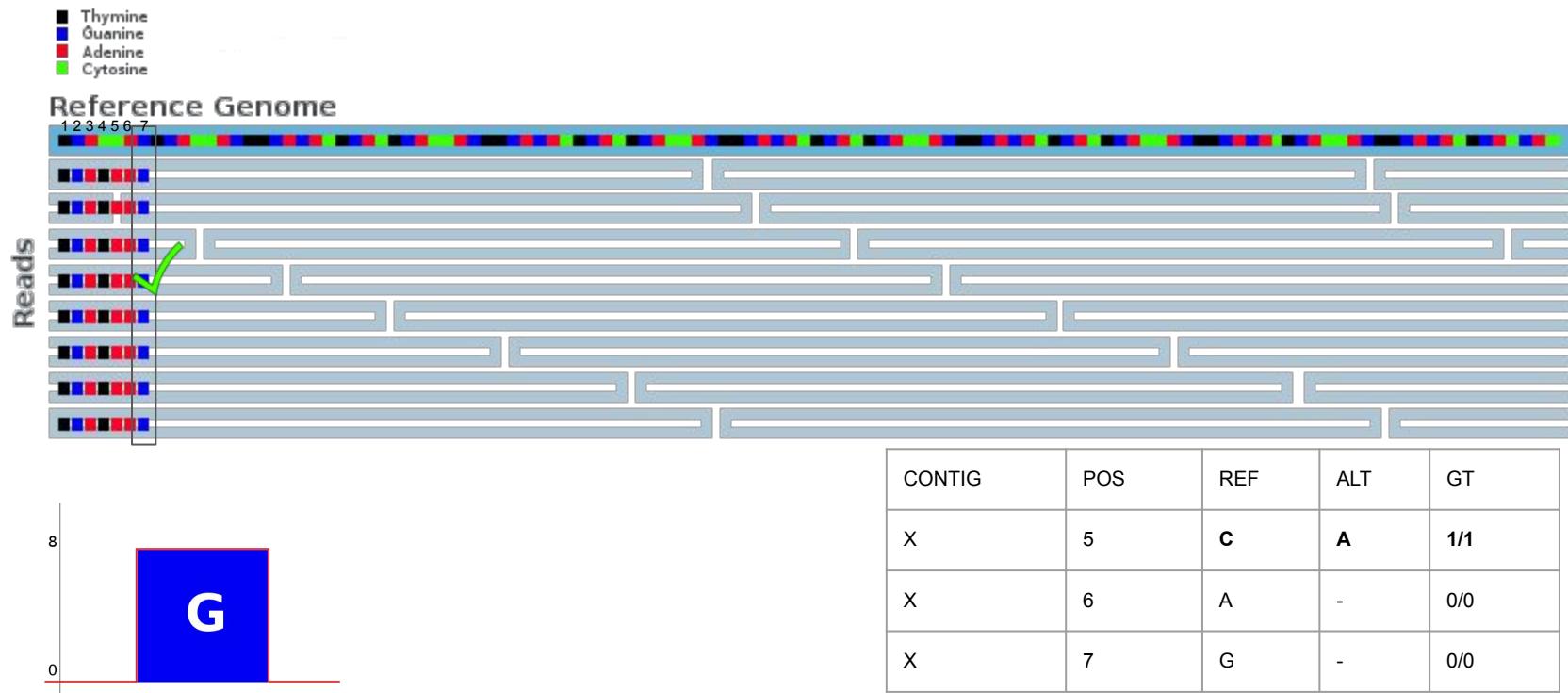
Ideal Variant Calling



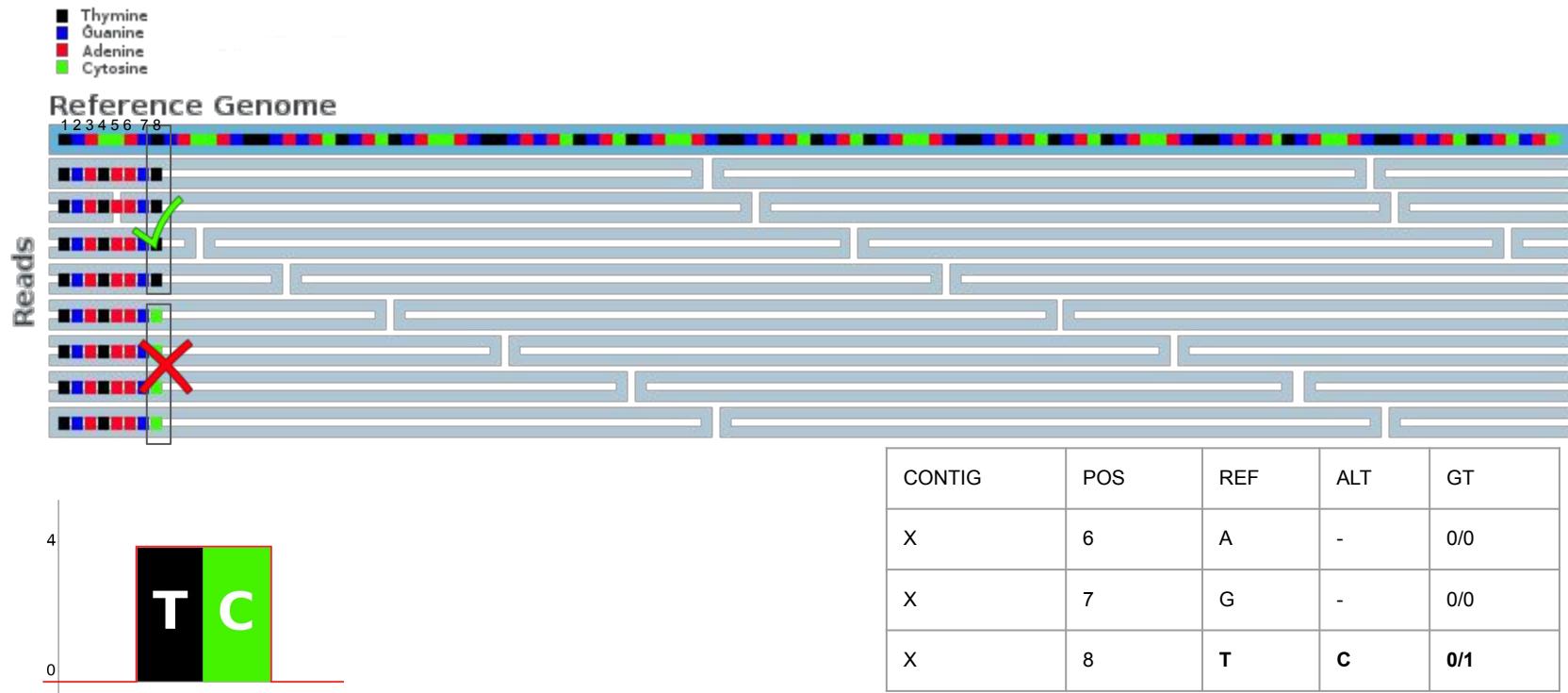
Ideal Variant Calling



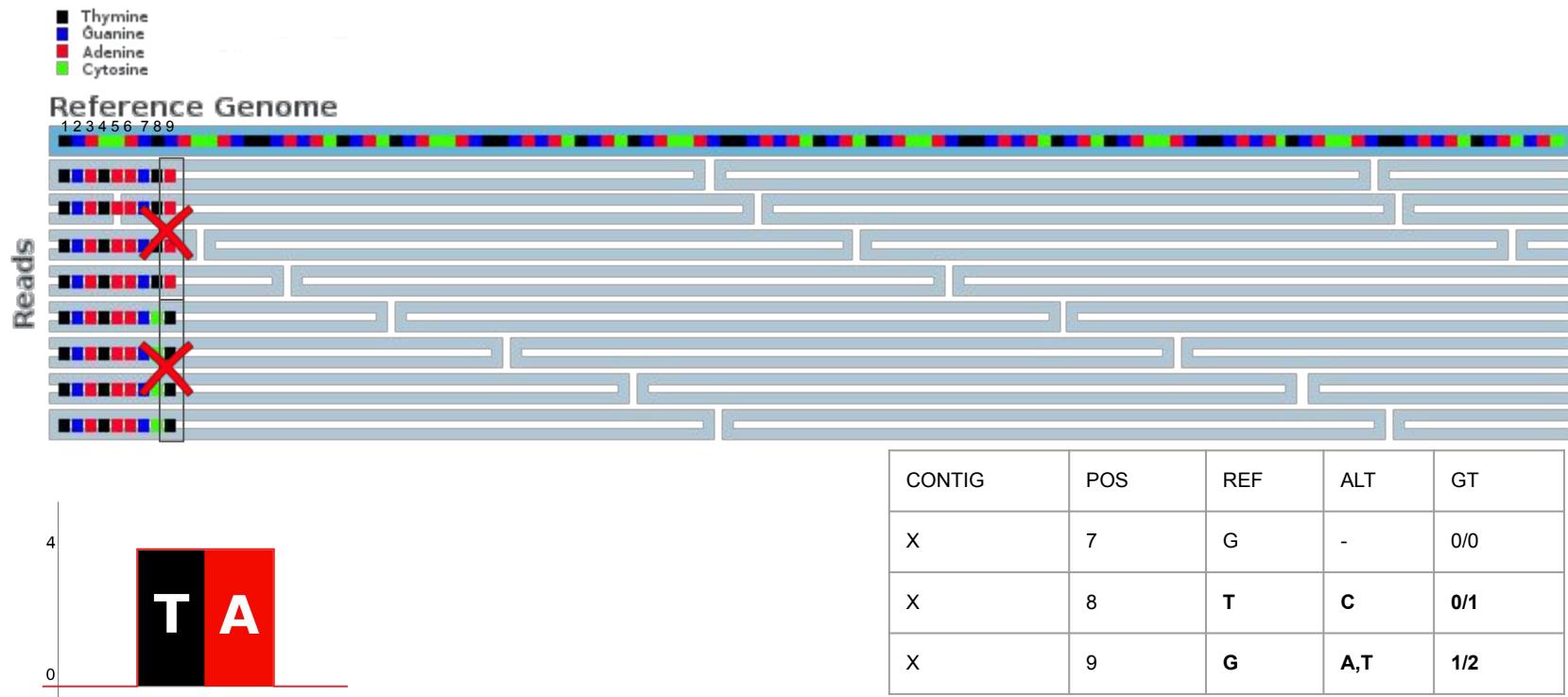
Ideal Variant Calling



Ideal Variant Calling



Ideal Variant Calling

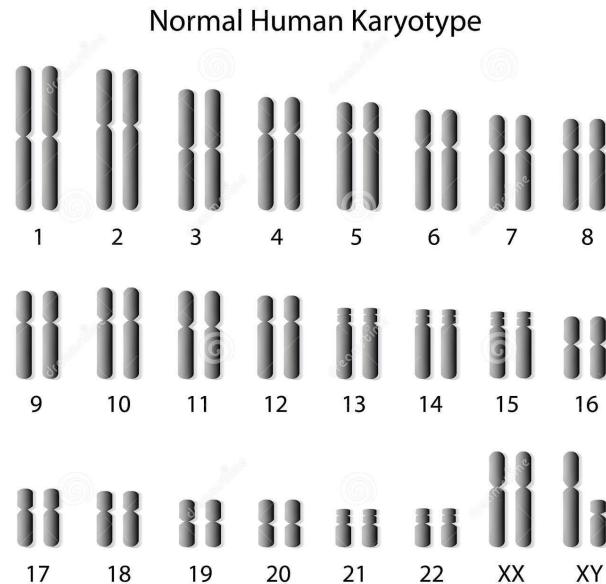


Variant Calling

Two possible cases:

1. All of the bases in pileup are the same nucleotide [A,T,C,G]
2. Different nucleotides exist in the pileup

- In the simplest case, assume diploidy. There can be only two alleles at a site
- If there are more than two different letters in the pileup we will only consider the most common two (assume others are errors and discard them)

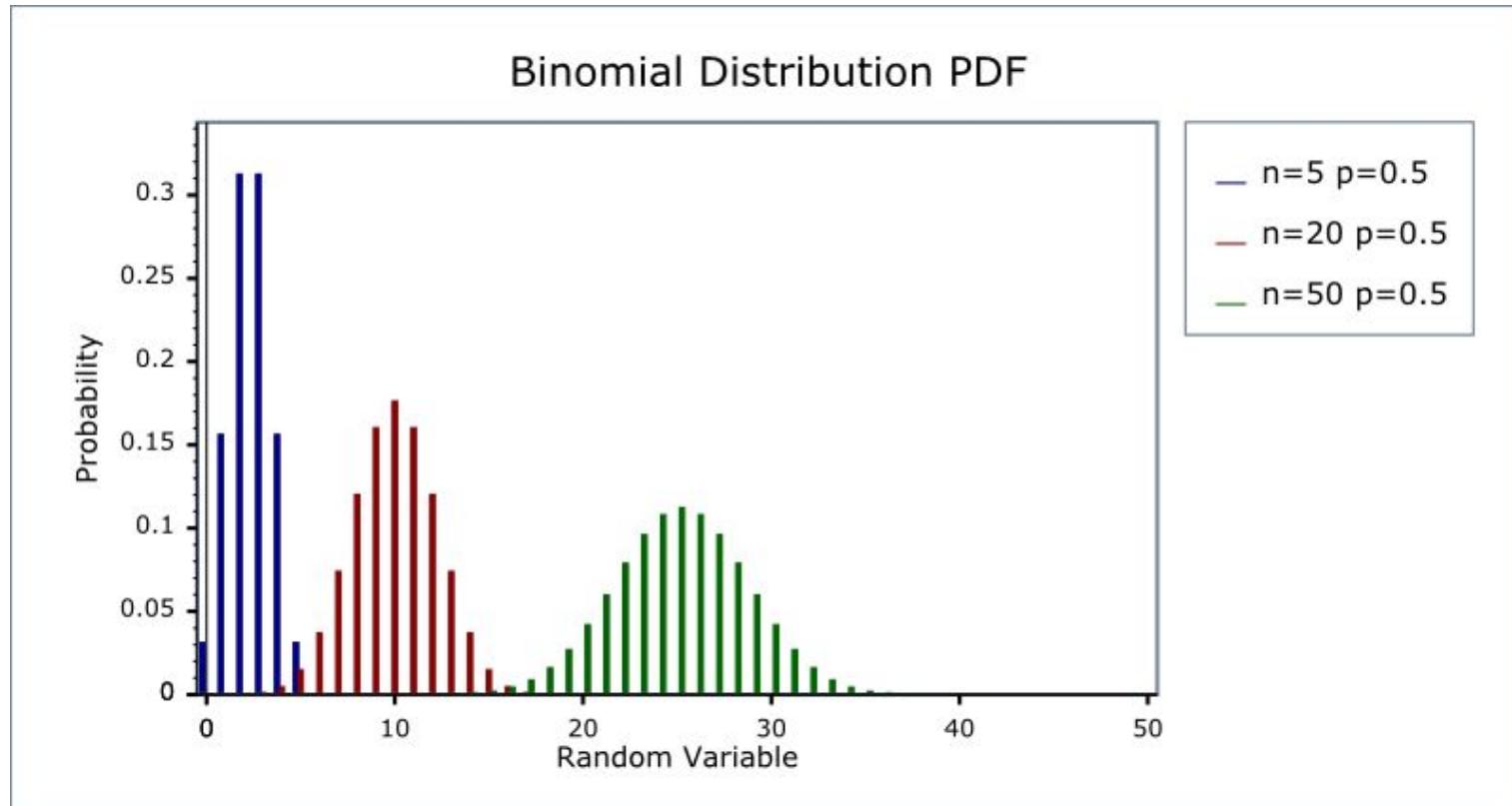


Binomial distribution

- Models the number of successes in a sequence of yes/no experiments
- Parameters:
 - n - number of trials
 - p - probability of a success in a single trial
 - Probability that K out of n trials will be success

$$f(k; n, p) = \Pr(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}$$

Binomial distribution



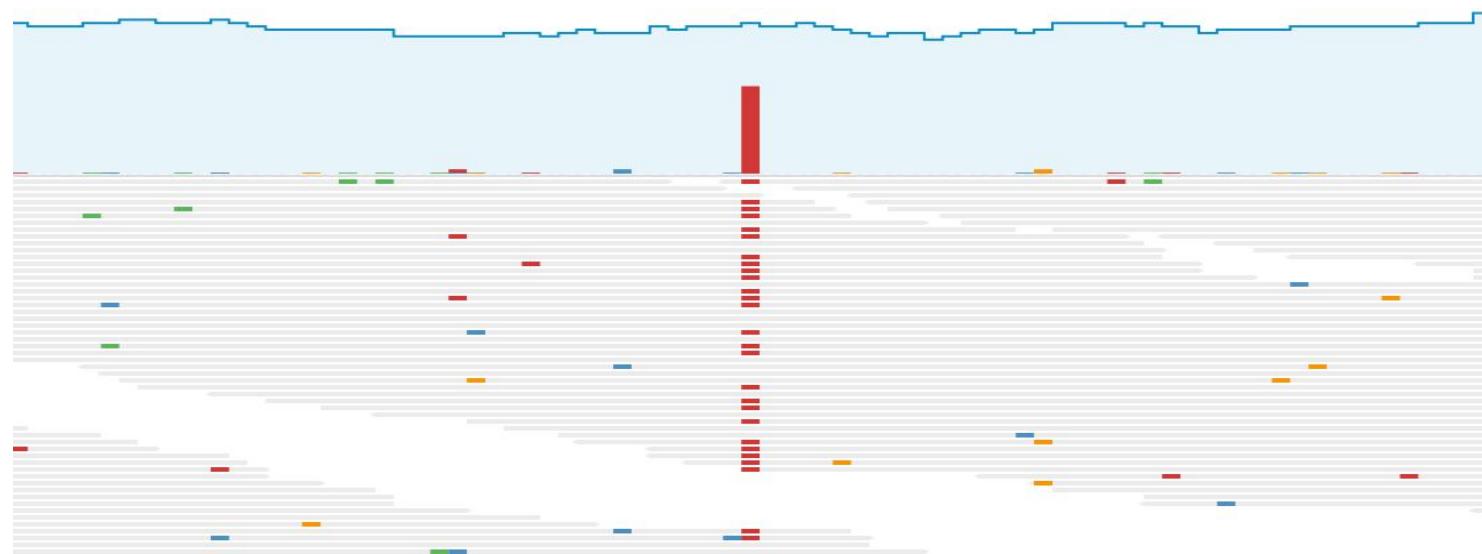
Variant Calling

- So, when we have two letters in the pileup, what should we call?
 - Let's call the two "letters" **b** and **b'** ($b, b' \in [A, C, T, G]$)
 - Let **n** be the total number of bases, and **k** number of **b'** bases
 - Three possible explanations for the pileup:
 - Genotype is bb ; k bases are errors, $n-k$ are correct
 - Genotype is $b'b'$; $n-k$ bases are errors, k are correct
 - Genotype is bb' ; all n bases are correct
 - Now we need to find the probabilities of these three cases
Will pick the most probable one!

Variant Calling – advance

- We assumed a flat error rate
 - But we have Base qualities from the sequencer
 - Machine-specific error profiles
- We can look at mapping qualities
 - Mapping errors are a big source of errors
- We can look at haplotypes
 - Errors don't segregate nicely
- Population-based methods
 - Separate variant calling from genotyping

Variant calling results – check out BAM file



CHR	POS	REF	ALT	FORMAT	NA12877
1	14125	T	C	GT, VAF	0/1, 0.6

Variant calling results

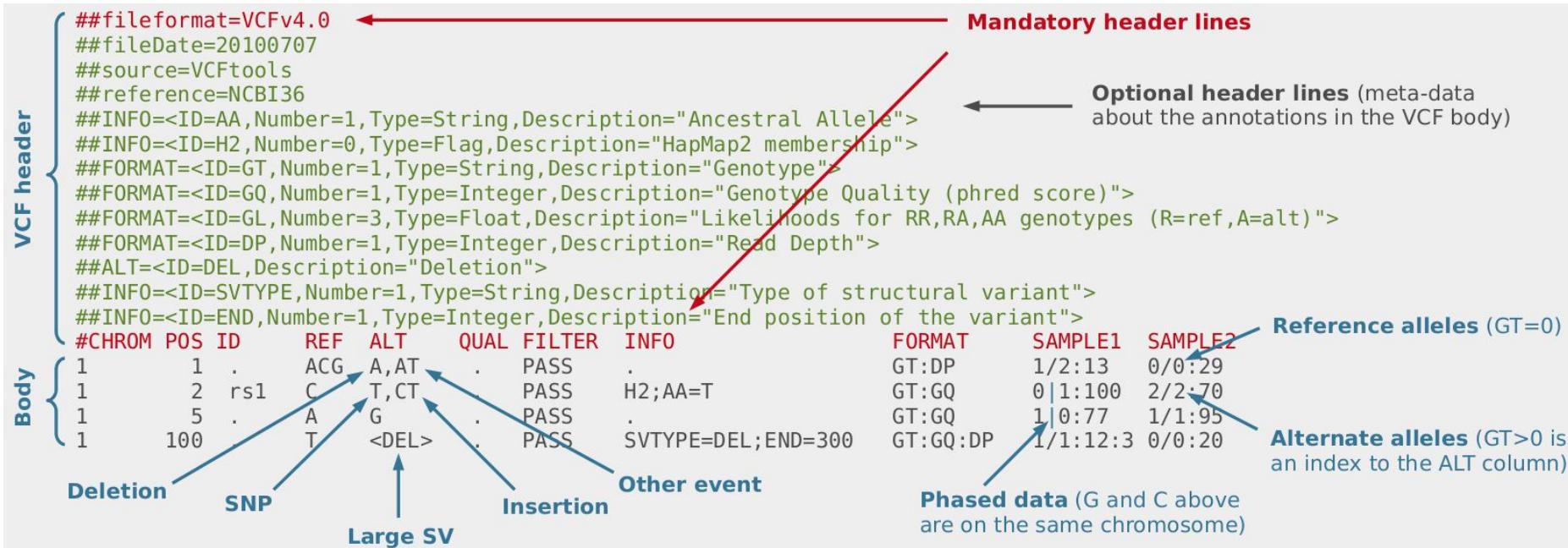
- The result of Variant Calling is a file in VCF format, which contains mutations
- A plain text file format for storing variant data
- A number of line starting with ## -the header
- Main header line:
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT SAMPLE1
- This is followed by the actual variant data, one entry per line
22 10001 . A C 40 PASS DP=14 GT 0/1
- More than one sample can be in one line
- For details: <http://samtools.github.io/hts-specs/VCFv4.2.pdf>

Variant calling results

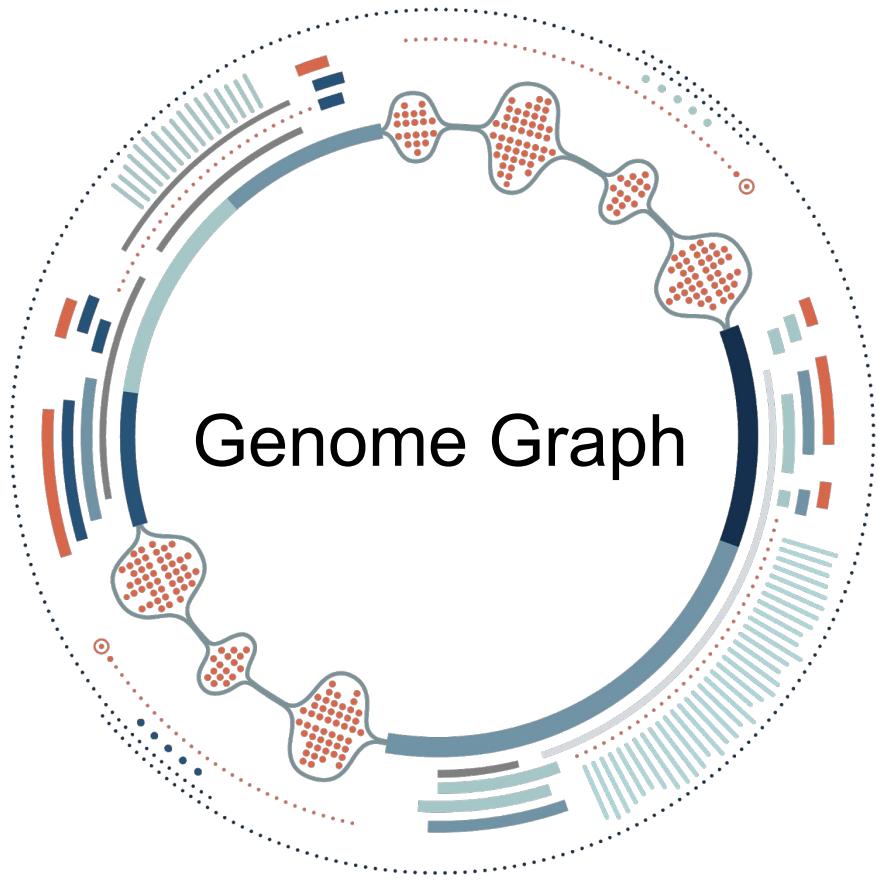
- Example of VCF format
- Each row represents one mutation

CHR	POS	REF	ALT	FORMAT	NA12878
1	14300	A	G	GT, VAF	0/1, 0.4
2	15367	A	C	GT, VAF	1/1, 0.9
3	25612	C	G,A	GT, VAF	1/2, ?
5	5632	TA	T	GT, VAF	0/1, 0.5
7	7824	T	TA	GT, VAF	1/1, 0.8

Variant Calling Format File



GRAF Genome



SB GRAF™

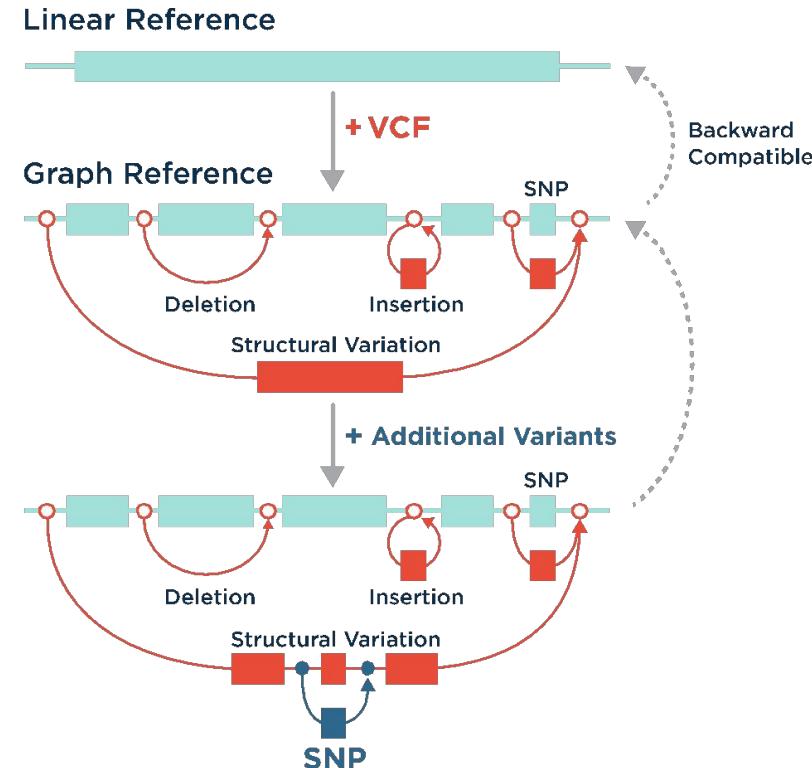
Genome graphs and their
applications

What is a Genome Graph Reference?

A **Genome Graph Reference** better captures genetic diversity, from virtually **any population of interest**, compared to linear reference genomes.

Genome Graph References can thus **incorporate more genetic variation** than linear references, facilitating accurate variant identification for downstream analyses.

Graph edges can store relevant information such as allele frequency, genotype, etc. which are used during alignment, variant calling and/or any tertiary analyses.



Why GRAF?

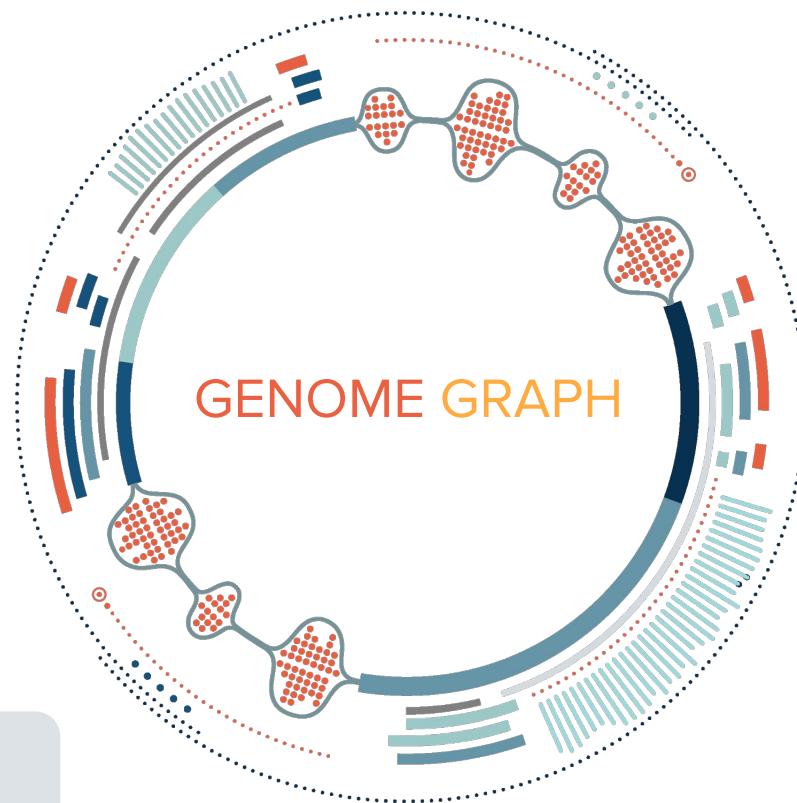
Unbiased: Any known genetic variation can be explicitly represented.

Self-Improving: More data makes for better results.

Tailored: References incorporating relevant alleles for specific analyses.

Efficient, robust: Faster and more accurate than current popular pipelines.

Cost-effective: One of the lowest processing cost per genome amongst other WGS pipelines.



Publication:

Rakocevic, Goran, et al. Fast and accurate genomic analyses using genome graphs. *Nature Genetics*, 2019.

<https://www.nature.com/articles/s41588-018-0316-4>

Graph Genomes Alleviate Reference Bias in Linear References

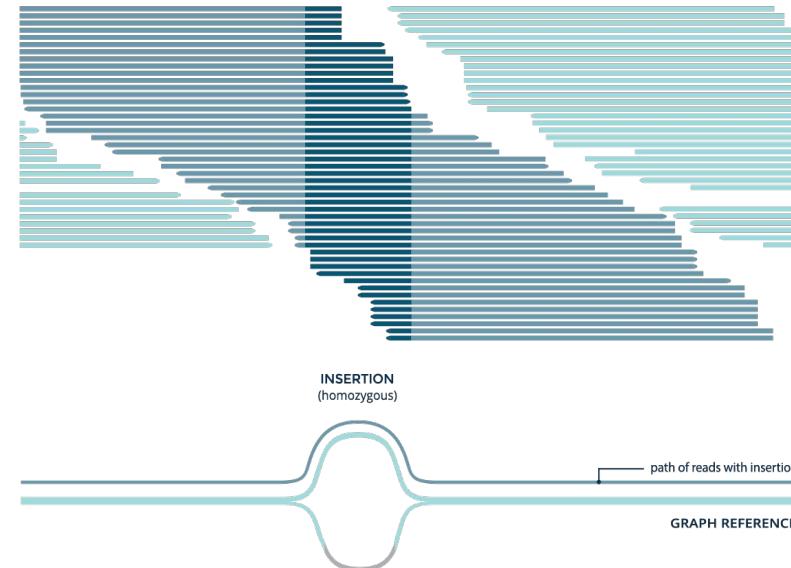
- Representative of only a few individuals; Genetic diversity not captured
- **Reference Bias:** Alignment inaccuracy for reads containing non-reference sequences



LINEAR REFERENCE

Canonical Linear reference
doesn't contain insertion

Only a few reads containing the insertion are properly mapped, with some incorrectly mapped reads.



INSERTION
(homozygous)

path of reads with insertion

GRAPH REFERENCE

Reads are properly mapped, since the Graph reference includes information about the insertion

Pan-Genome Graph Data Sources

Population Genomics Databases

- The 1000 Genomes Project
 - Simons Genome Diversity Project
 - gnomAD

SNP and SV Databases

- International HapMap Project
 - Mills *et al.* Indels
 - Single Nucleotide Polymorphism Database (dbSNP) - with QC+filtering
 - GRCh38 alternate contigs

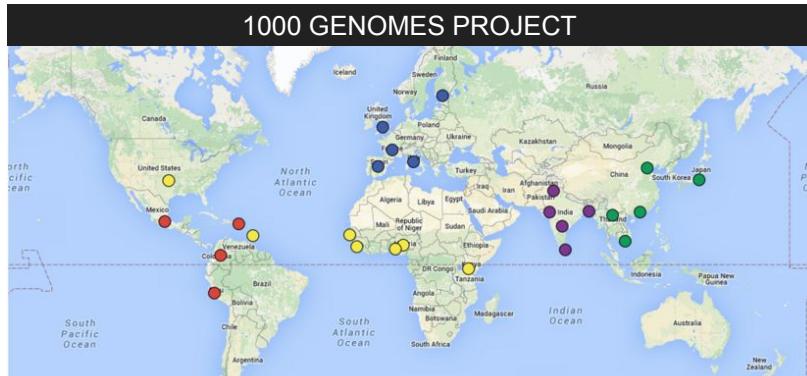


Image credit: 1000 Genome: <http://www.internationalgenome.org/>

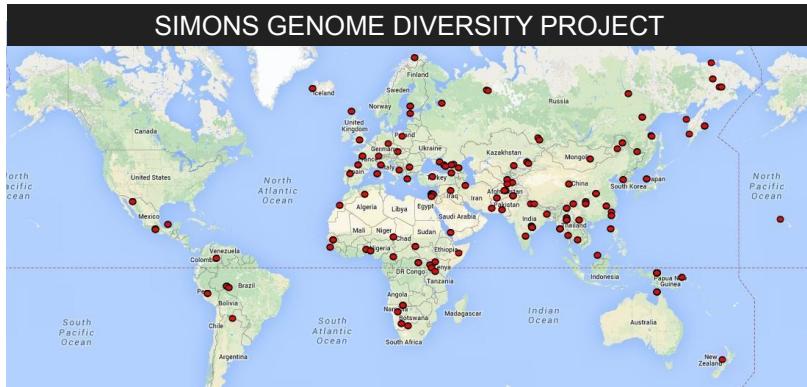
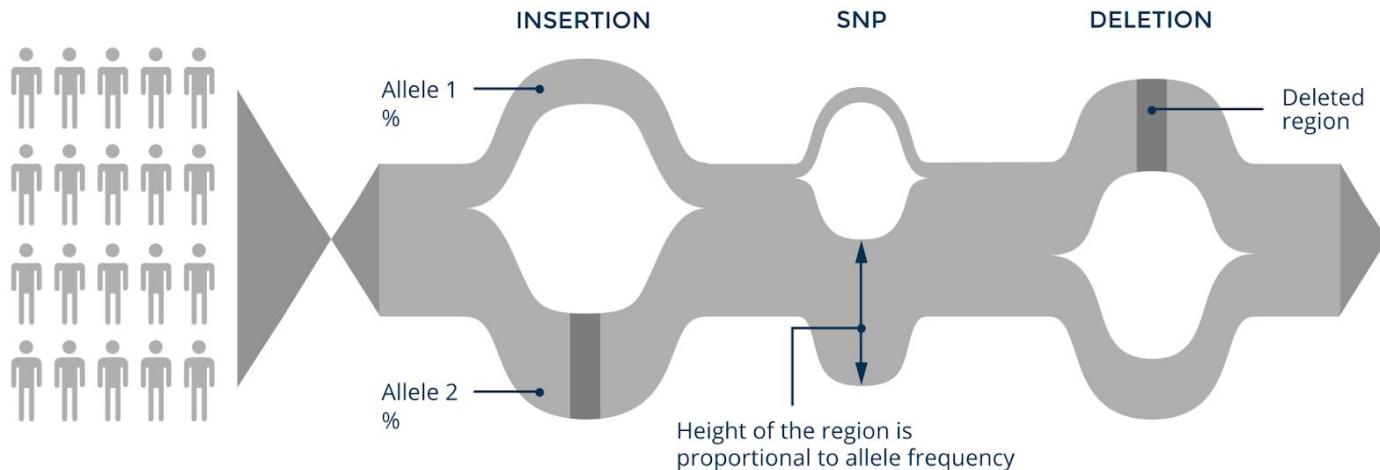


Image credit: Simons: <https://www.simonsfoundation.org/simons-genome-diversity-project/>

Population Graph References

- Captures **genetic diversity** of specific populations
- Enables more **accurate read alignment and variant calling** for specific genomic contexts
- **Custom population graph** construction services.

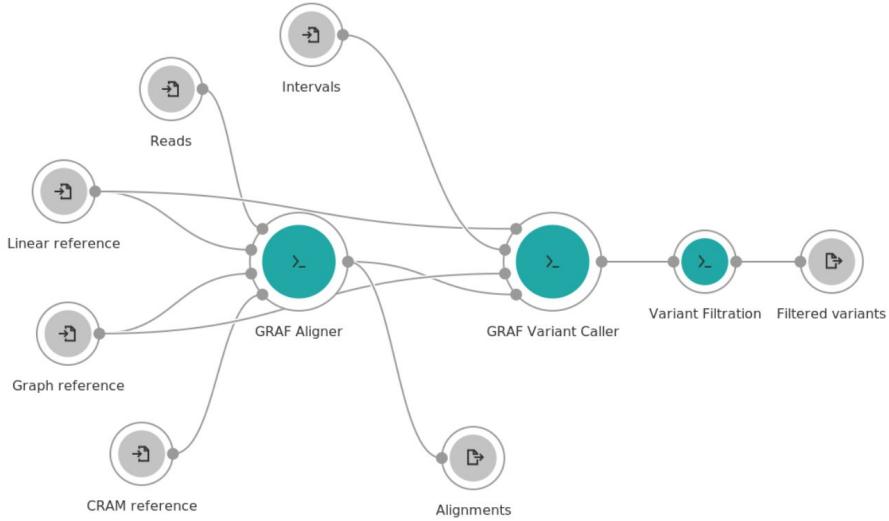


Out-of-the-box GRAF Workflows

GRAF Germline Variant Workflow

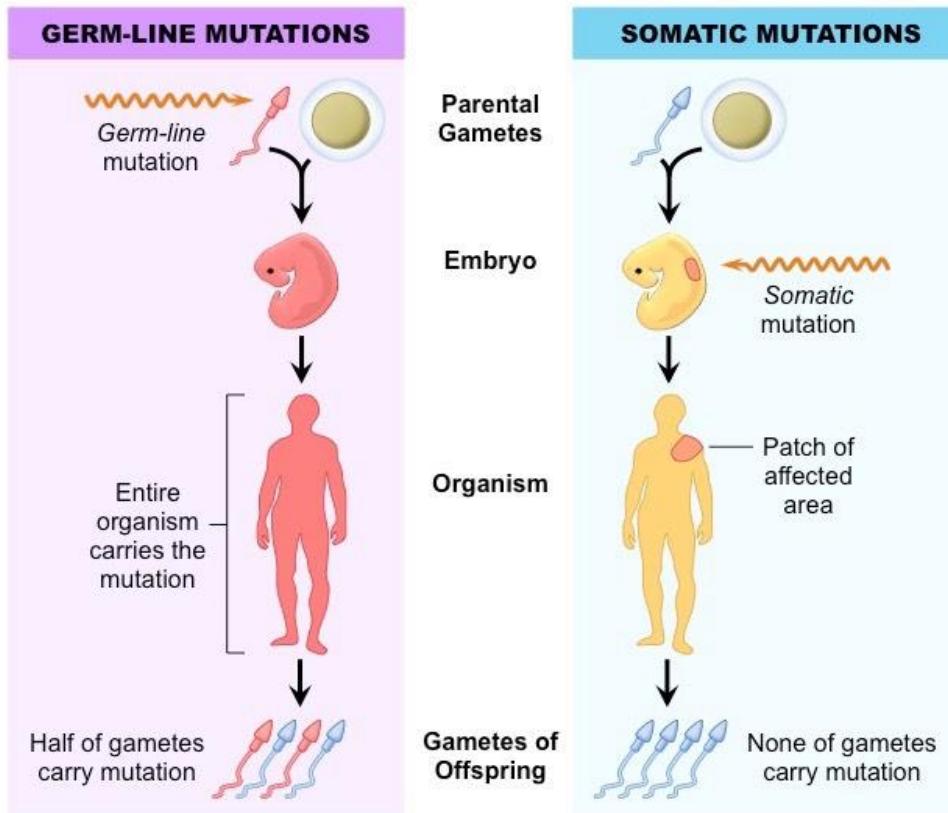
- Graph-aware read aligner
- Graph-aware germline variant caller
- Compatible with Pan-Genome, Population, Disease Graph References
- Available to **any** customer with a platform licence.
- Freely available on all our academic platforms (CGC, CAVATICA, NHLBI).

Workflows are **fully compatible** with existing data formats such as FASTQ, CRAM, BAM, VCF and GVCF.

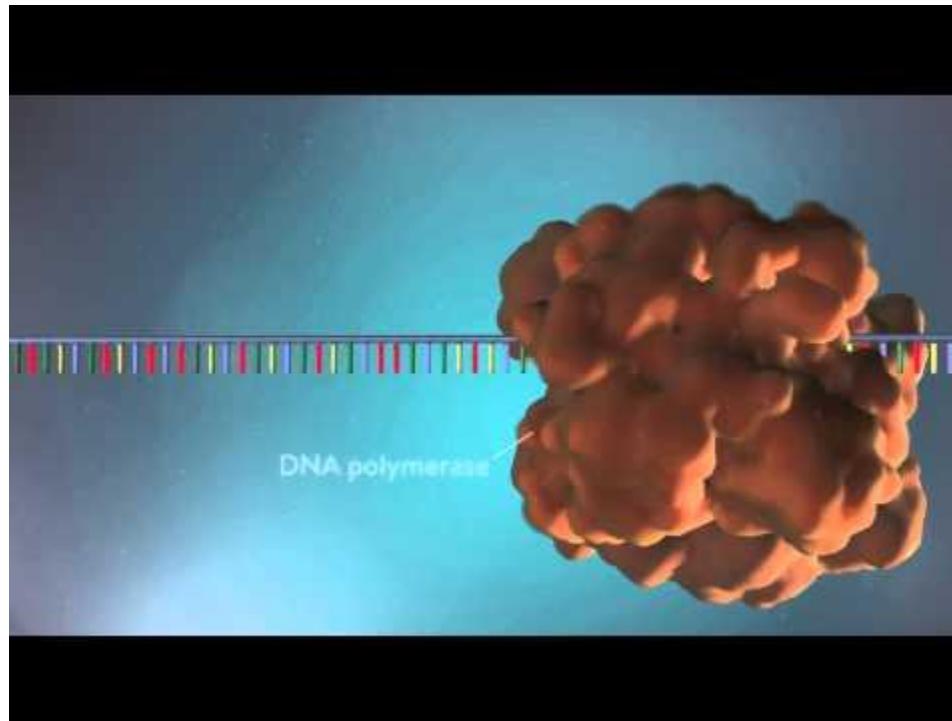


Computational Cancer Analysis

Variants?



DNA replication



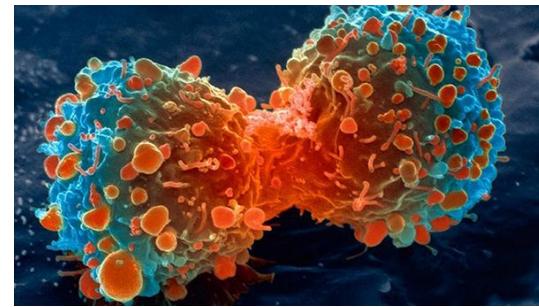
What is cancer?

Mutation during DNA replication can fall to:

1. Intron (no change)
2. Important gene (cell dies, organism lives)
3. Gene that stops cell division (cell lives, organism...)

What causes cancer (increases probability of mutation)?

1. EM radiation
2. Chemical agents
3. Free radicals
4. Genetic factors
5. Infections (viruses)



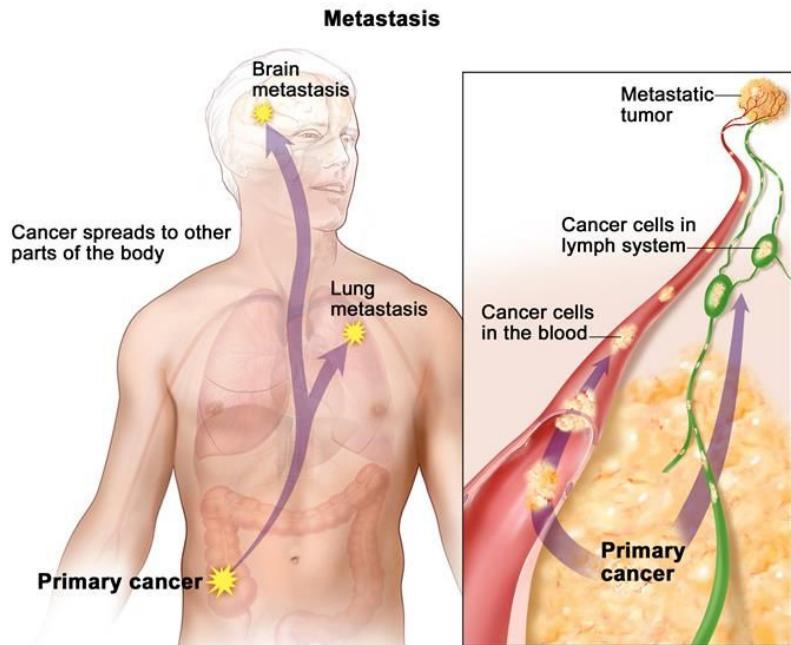
A dividing lung cancer cell.

Credit: [National Institutes of Health](#)

What is metastasis?

Body's cells begin to divide without stopping and spread into surrounding tissues

Cancer cells - ignore signals that normally tell cells to stop dividing or that begin a process known as programmed cell death, or **apoptosis**, which the body uses to get rid of unneeded cells

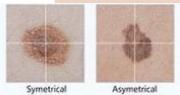


© 2014 Terese Winslow LLC
U.S. Govt. has certain rights

Increased epithelial cell division leading to metastasis

A
B
C
D
E

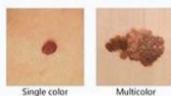
Asymmetry: Moles that have asymmetrical appearance



Border: A mole that has blurry and/or jagged edges



Color: A mole that has more than one colour



Diameter: Moles with a diameter larger than a pencil eraser (6 mm or 1/4 inch)



Evolution: A mole that has gone through sudden changes in size, shape or colour



ADDITIONAL MUTATIONS

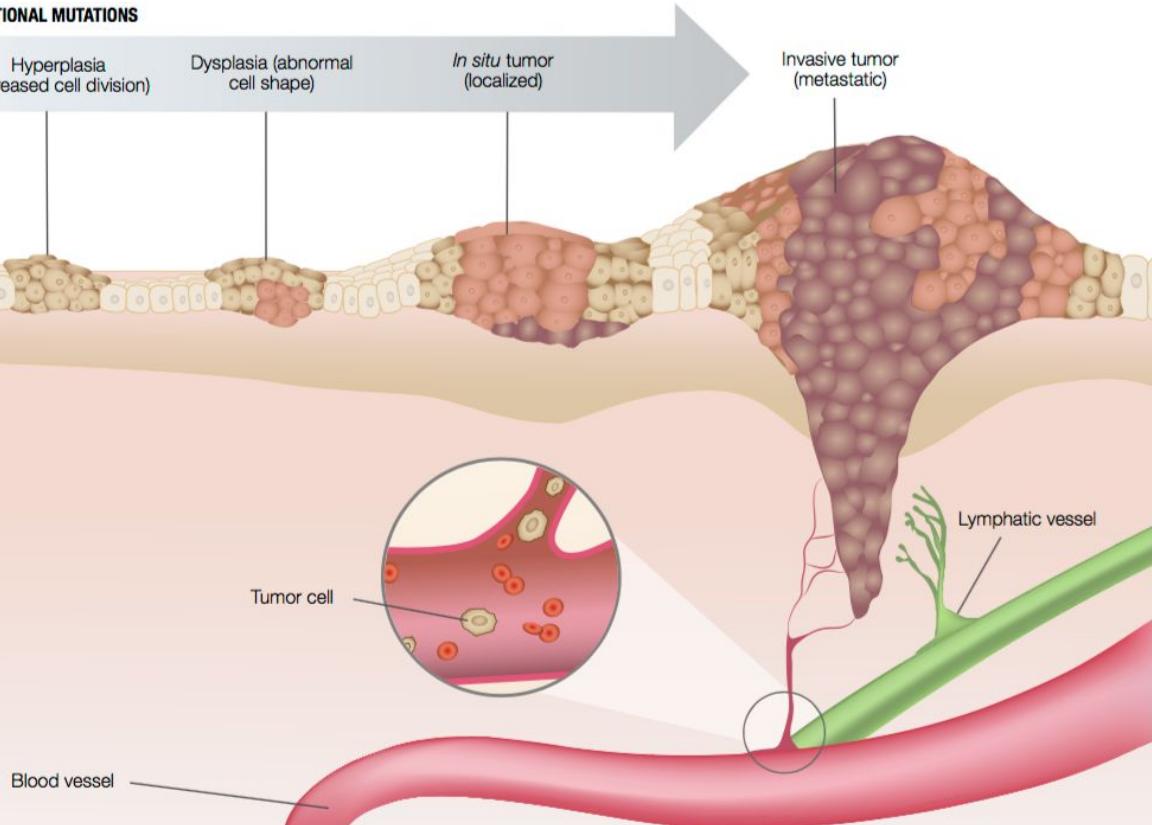
Cell with genetic mutation

Hyperplasia
(increased cell division)

Dysplasia (abnormal cell shape)

In situ tumor
(localized)

Invasive tumor
(metastatic)



"Drivers" of Cancer

Cancer is a genetic disease that is caused by changes to genes which control the way our cells function, especially how they grow and divide:

1. **Abnormal growth (proto-oncogenes)** Cellular growth mechanism is damaged and cell starts to multiply uncontrollably
2. **Damaged control mechanism** (tumor suppressors) - Cells with certain alterations in tumor suppressor genes may divide in an uncontrolled manner (TP53 - Apoptosis)
3. **Damaged DNA repair mechanism** (Accumulated errors in this group of genes can lead to uncontrollable proliferation)

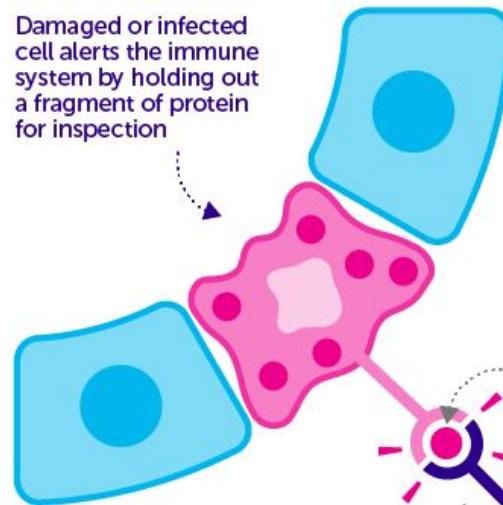
Cancer cells

Our body develops thousands cancer cells every day.

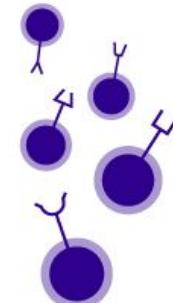
OMG! OMG! OMG!

IDENTIFYING THE ENEMY

Damaged or infected cell alerts the immune system by holding out a fragment of protein for inspection



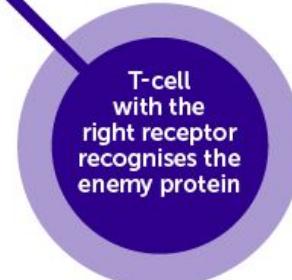
Each of your millions of T-cells has a slightly different receptor. Each detects different proteins



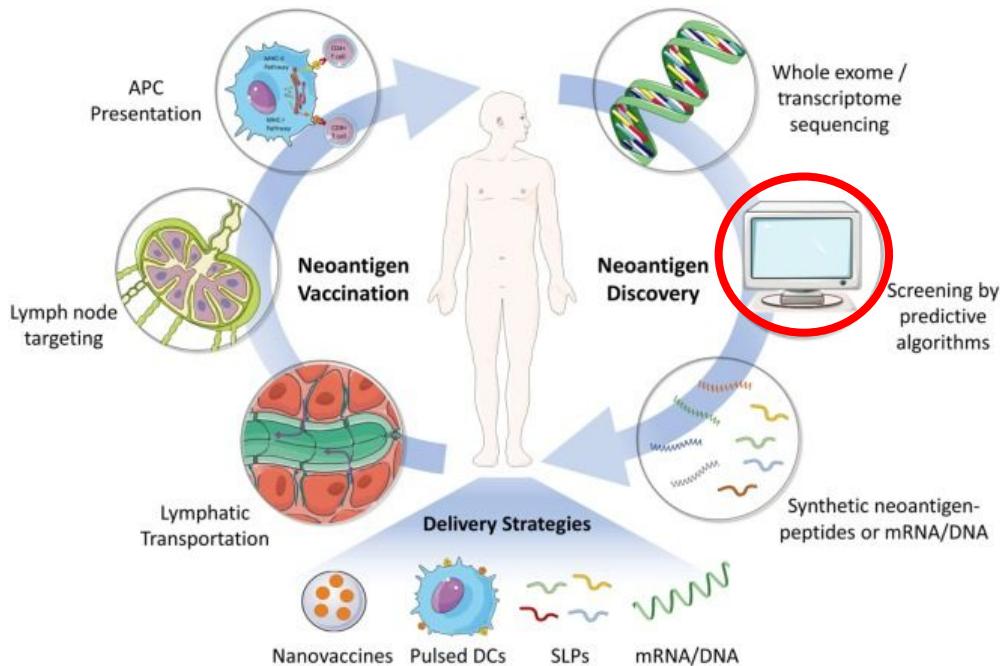
Fragment of protein

T-cell receptor

T-cell detects and destroys damaged or infected cell



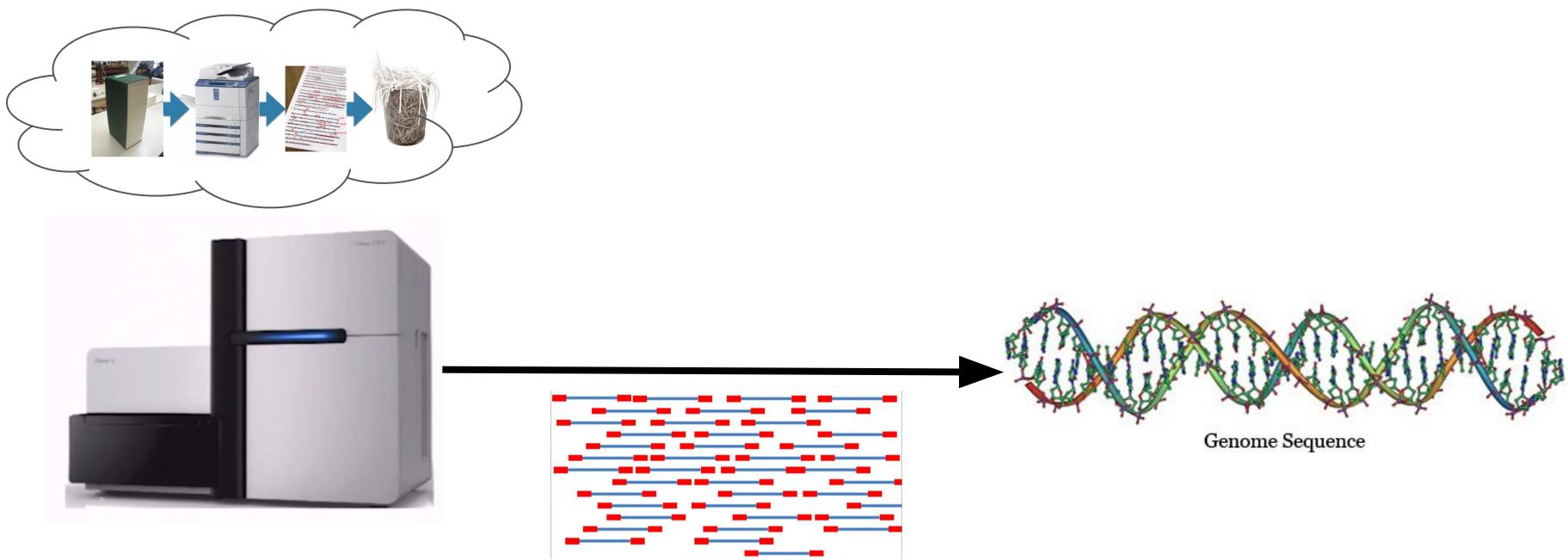
Neoantigens



- Neoantigens - proteins presented only by cancer cells
- When neoantigen is known -> immune T-cells can be “programmed” to destroy cancer cells
- These unique cancer markers could be a key to developing a new generation of personalized, targeted cancer immunotherapies

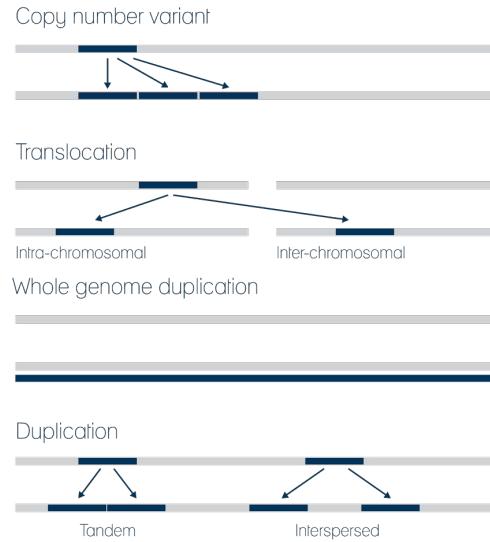
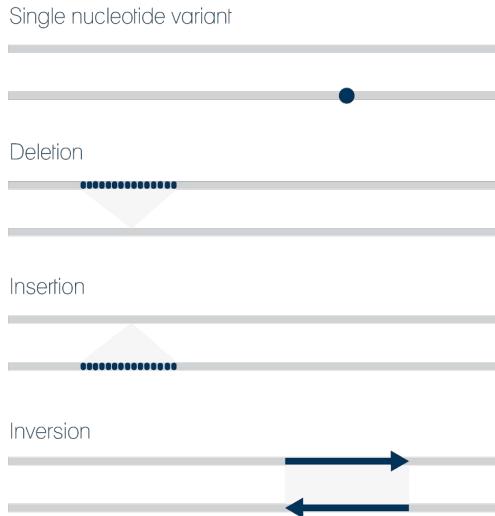
How can we discover neoantigens?

1. Reconstruct DNA of tumor and normal tissue

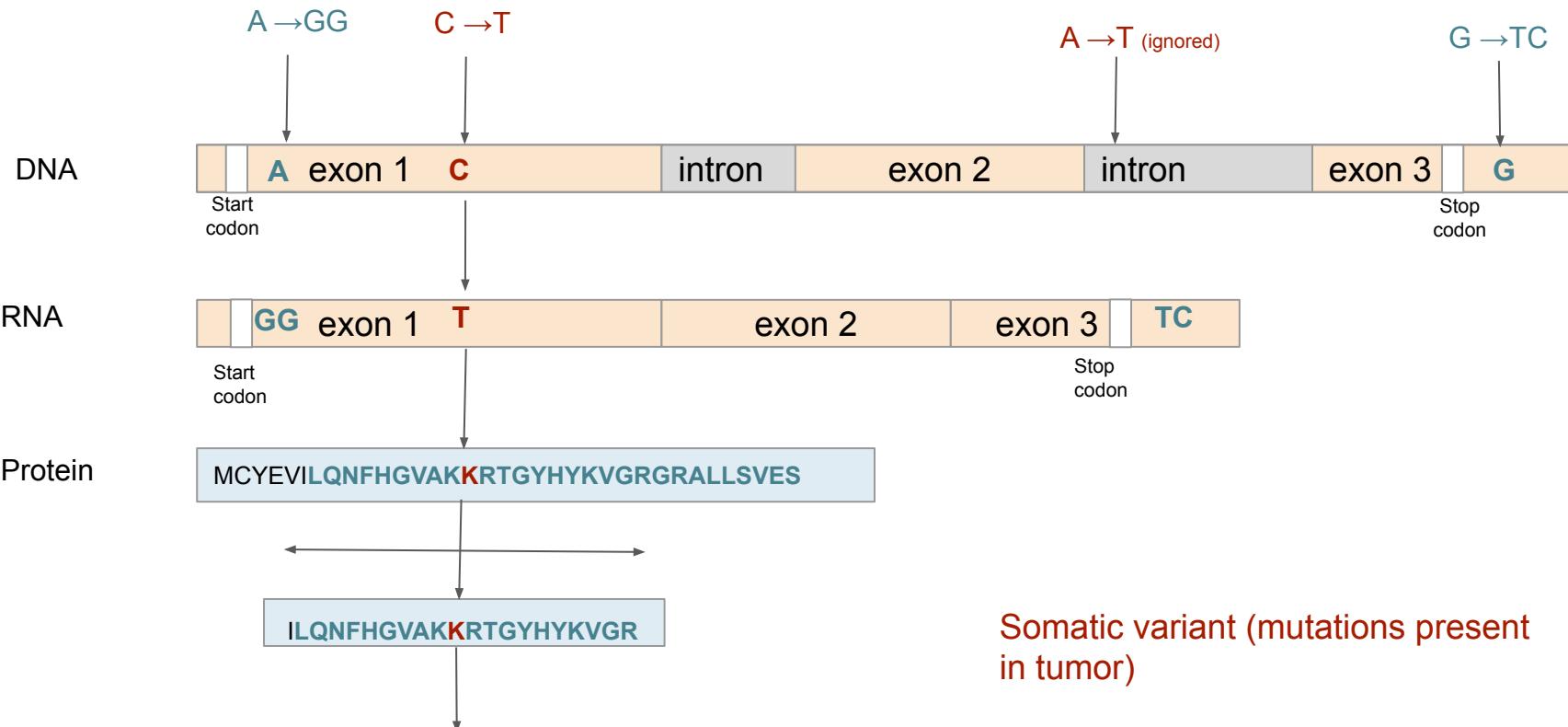


How can we discover neoantigens?

2. Compare DNA from Tumor and Normal tissue



How can we discover neoantigens?



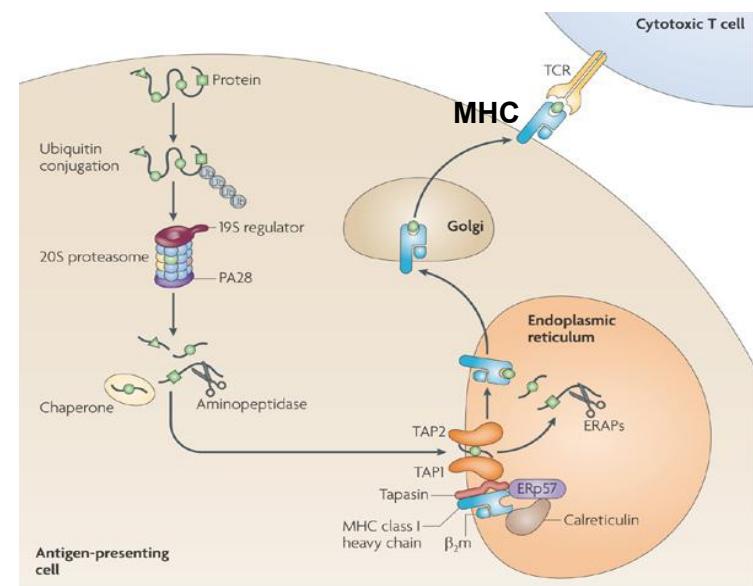
3. Protein extraction

How can we discover neoantigens?

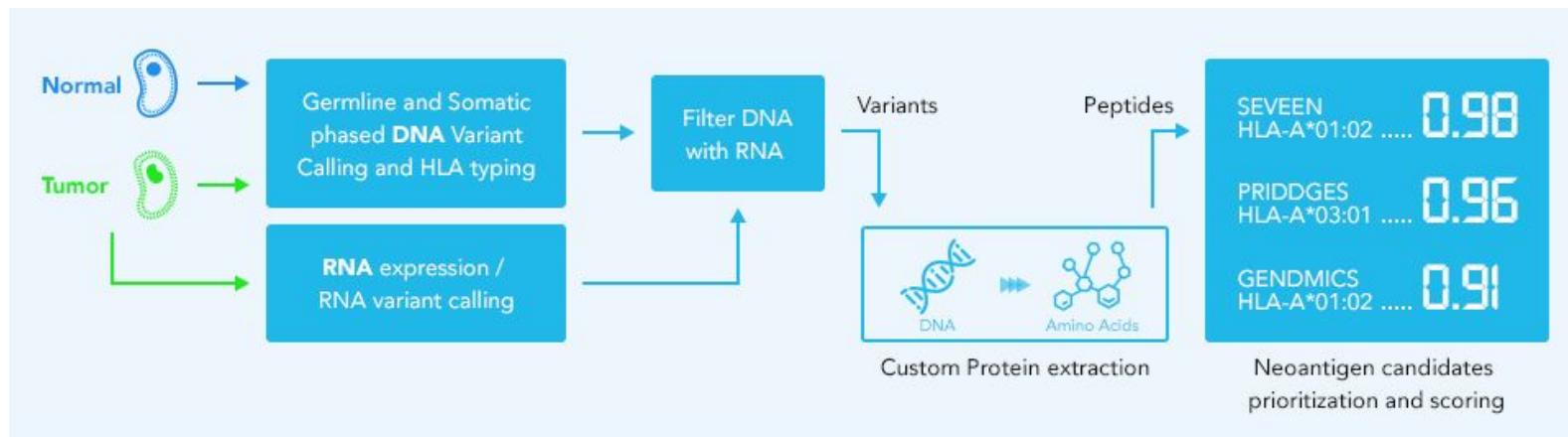
4. Discover HLA type from genome
(translates to MHC molecule)

5. Perform scoring of protein-HLA sets
(Iedb processing tools)

Mutation	HLA type	peptide	NetMHC score	Pickpocket score	NetCTLpan score	RNA expression
1_111957245_C_A	HLA-A*02:01	MMLSSSPV	0.881	0.633	1.09815	11.5
8_144392368_T_C	HLA-A*02:01	WLLEKLEQL	0.828	1.097	1.06133	12.5
17_28537638_C_T	HLA-A*02:01	VLDEFPHV	0.836	0.374	1.06015	23



Neoantigen workflow



References

How the Immune System Works - Lauren Sompayrac

[Neoantigen discovery webinar](#)

