SevenBridges

Structural Variation

April 2021

Boris Majić

boris.majic@sbgenomics.com

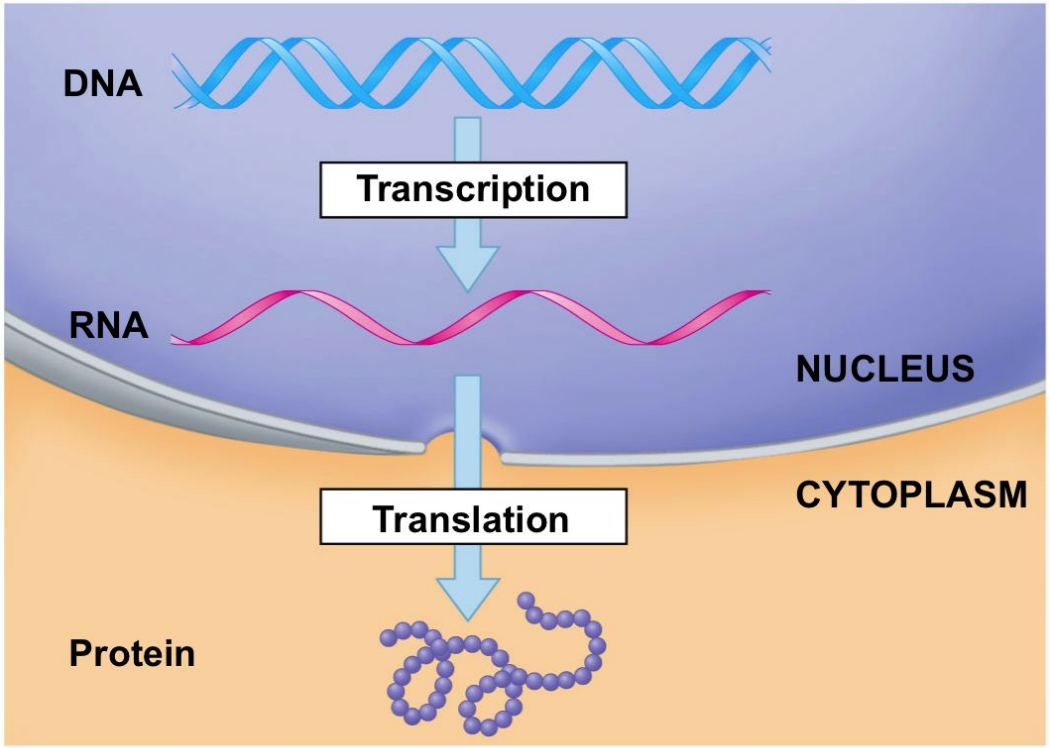
Genomic variation

Recap

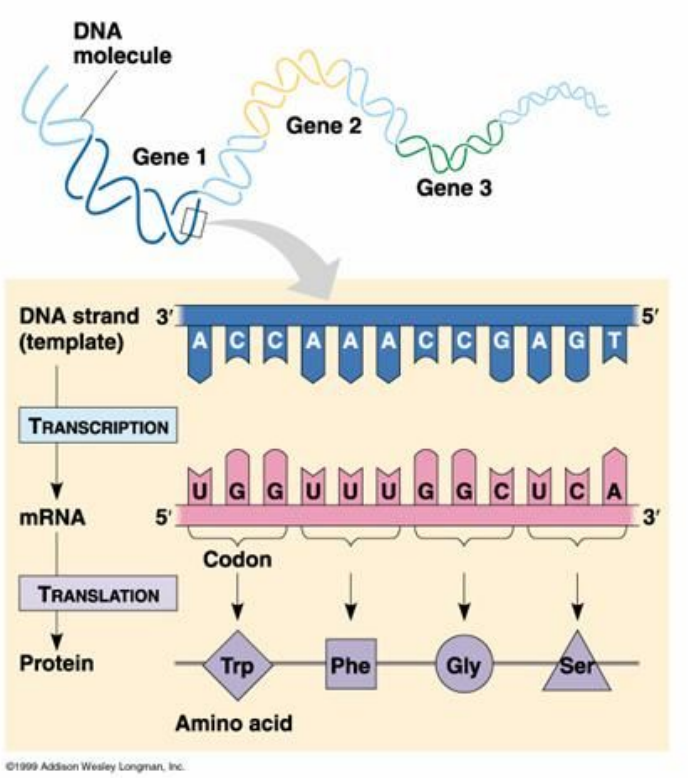
Genomic variation

- Represent **differences** between genomes which we are comparing
- Usually between a **sequenced genome** and a **reference genome**

Central dogma



Central dogma



Second base

First base	U	C	A	G	Third base
U	UUU, UUC — Phenylalanine F UUA, UUG — Leucine L	UCU, UCC, UCA, UCG — Serine S	UAU, UAC — Tyrosine Y UAA — Stop codon UAG — Stop codon	UGU, UGC — Cysteine C UGA — Stop codon UGG — Tryptophan W	U C A G
C	CUU, CUC, CUA, CUG — Leucine L	CCU, CCC, CCA, CCG — Proline P	CAU, CAC — Histidine H CAA, CAG — Glutamine Q	CGU, CGC, CGA, CGG — Arginine R	U C A G
A	AUU, AUC, AUA — Isoleucine I AUG — Methionine start codon M	ACU, ACC, ACA, ACG — Threonine T	AAU, AAC — Asparagine N AAA, AAG — Lysine K	AGU, AGC — Serine S AGA, AGG — Arginine R	U C A G
G	GUU, GUC, GUA, GUG — Valine V	GCU, GCC, GCA, GCG — Alanine A	GAU, GAC — Aspartic acid D GAA, GAG — Glutamic acid E	GGU, GGC, GGA, GGG — Glycine G	U C A G

Genomic variants

- **Single Nucleotide Variants (SNV)**
  Length: 1bp
- **Small Insertions / Deletions (small INDELS)**
  Length: up to 50bp
- **Structural Variations (SV)**
  Length: greater than 50bp

25% developmental diseases

20% developmental diseases

Genomic variants

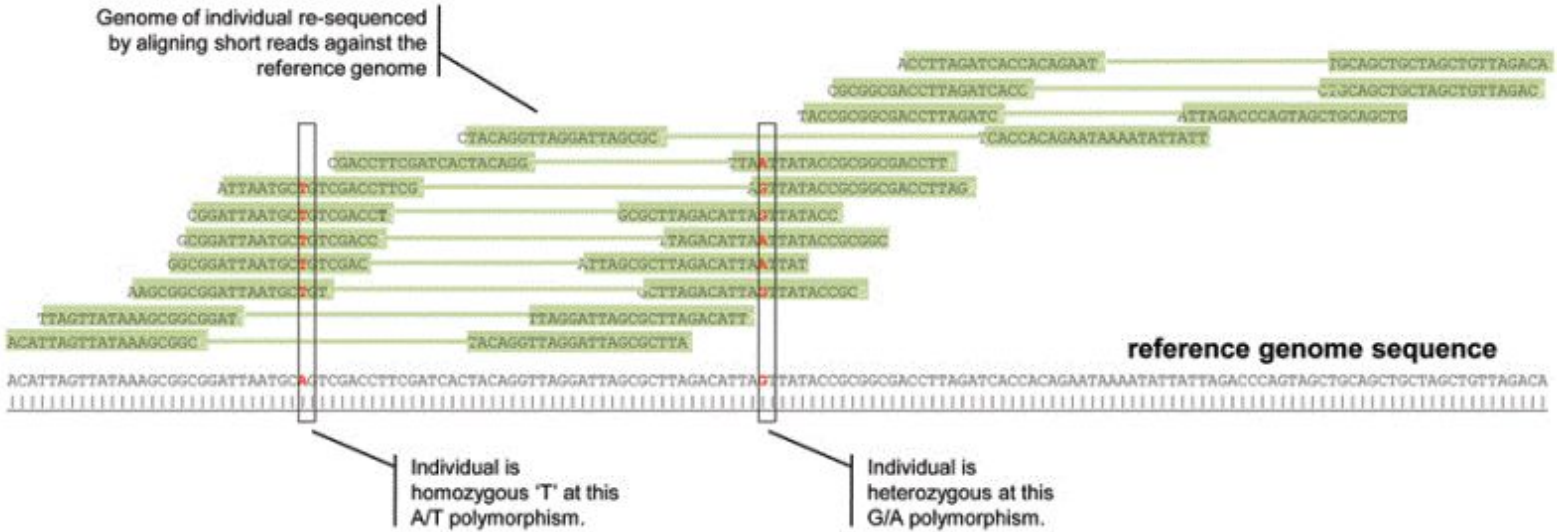
- **Single Nucleotide Variants (SNV)**
  Length: 1bp
- **Small Insertions / Deletions (small INDELS)**
  Length: up to 50bp
- **Structural Variations (SV)**
  Length: greater than 50bp

25% developmental diseases

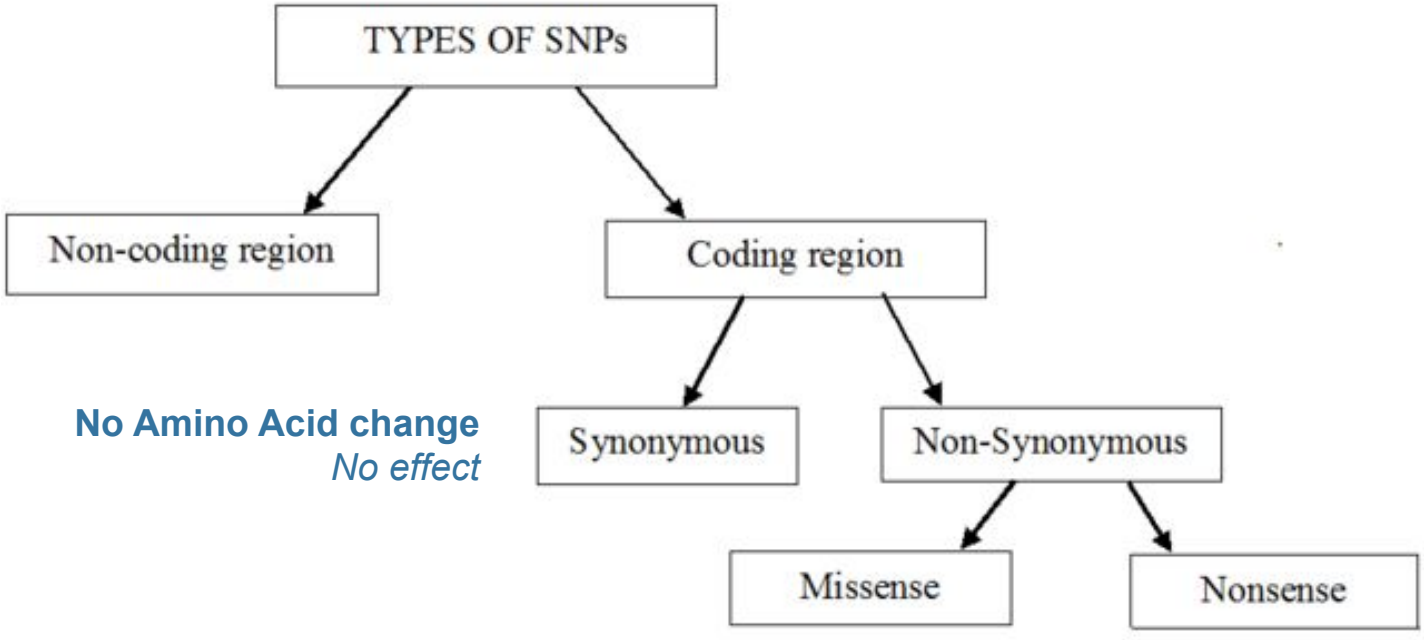
20% developmental diseases

Genomic variants

- **Single Nucleotide Variants (SNV)**
  Length: 1bp



Single Nucleotide Variants (SNV)

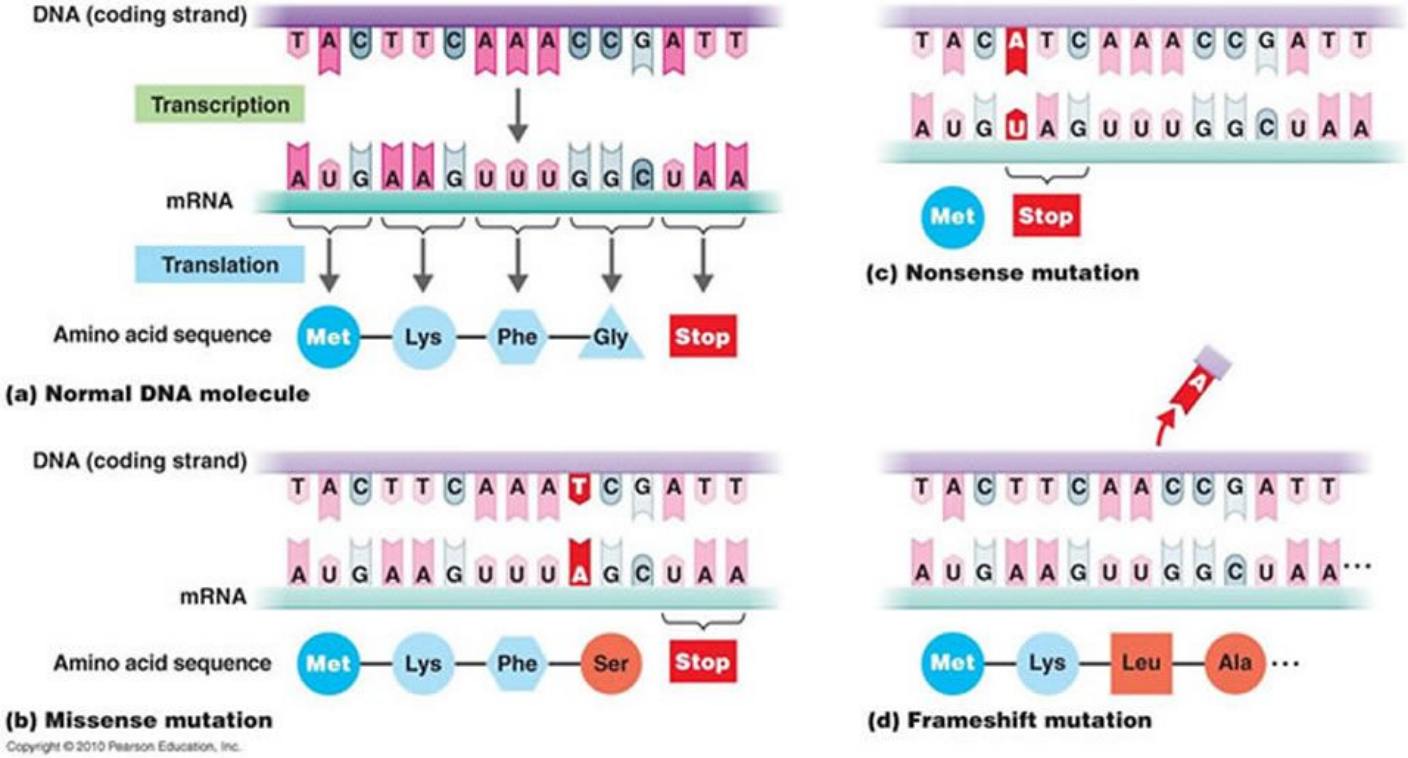


No Amino Acid change
No effect

Amino Acid change
Variable effect

Premature stop codon
Serious effect

Single Nucleotide Variants (SNV)



Genomic variants

- **Single Nucleotide Variants (SNV)**
  Length: 1bp
- **Small Insertions / Deletions (small INDELS)**
  Length: up to 50bp
- **Structural Variations (SV)**
  Length: greater than 50bp

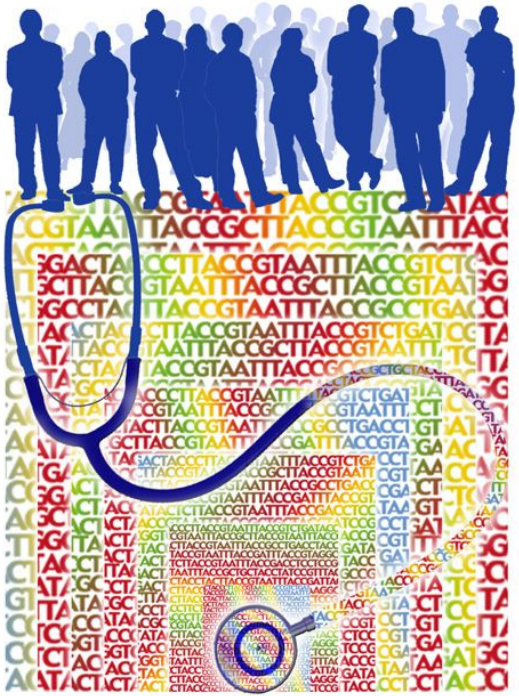
25% developmental diseases

20% developmental diseases

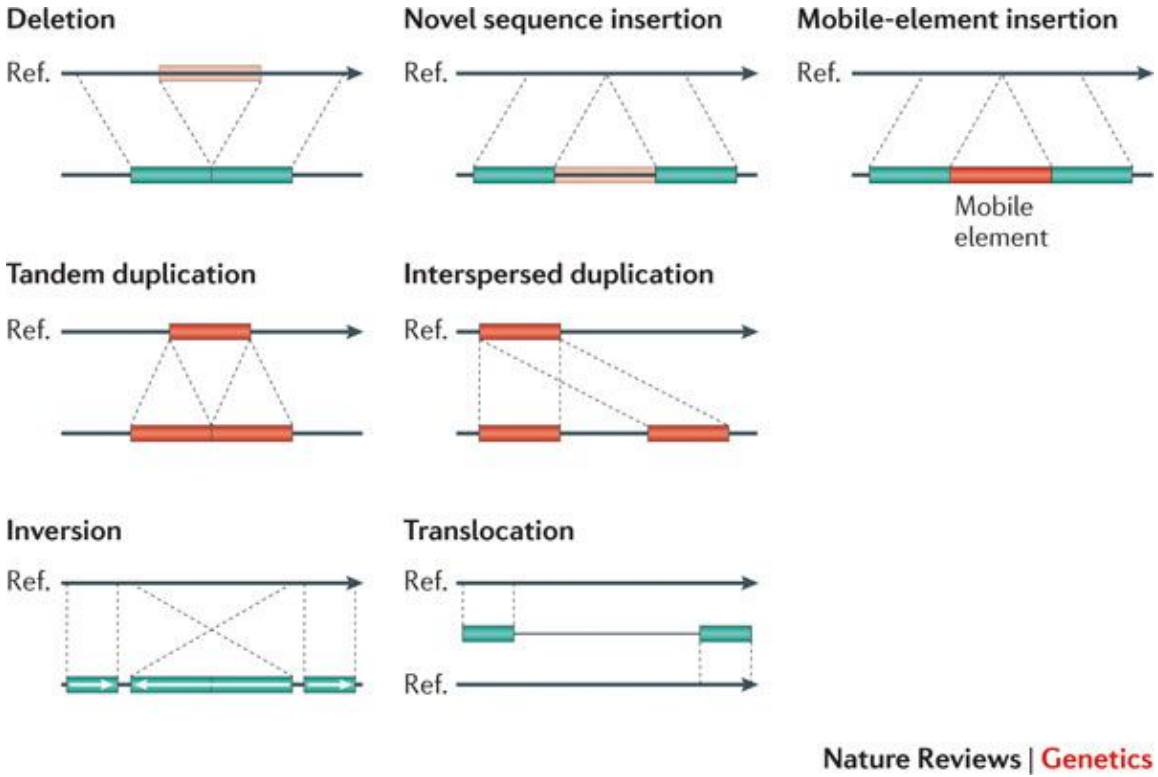
Structural variants

Structural variants (SV)

- Represent mutations in the genome **> 50bp** in length
- Human genomes **differ more** as a consequence of **structural variation (SV)** than of a single-base-pair differences (SNV)
- Approximately **20000** SVs in each human genome



Structural variants (SV)

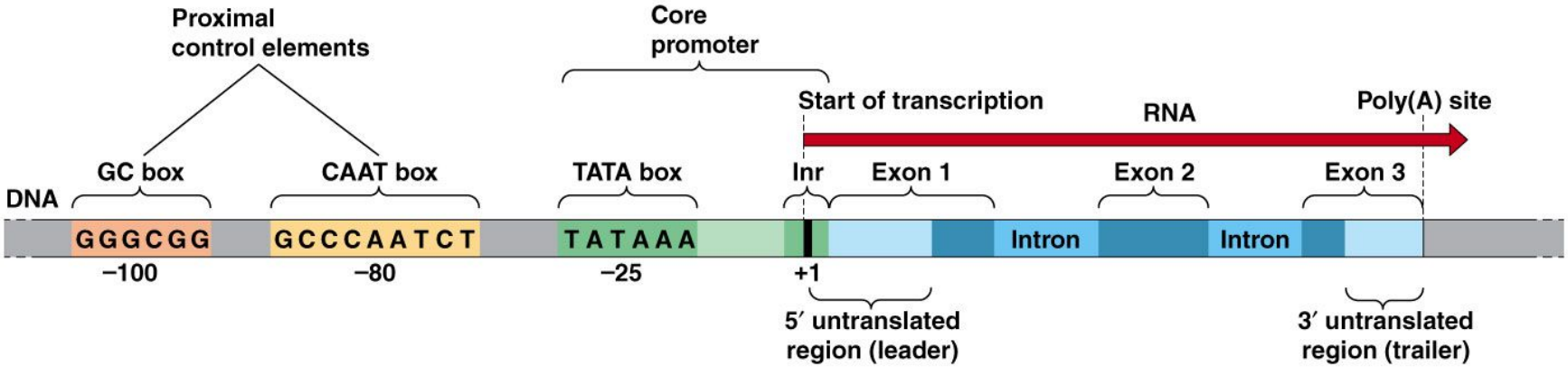


Nature Reviews | Genetics

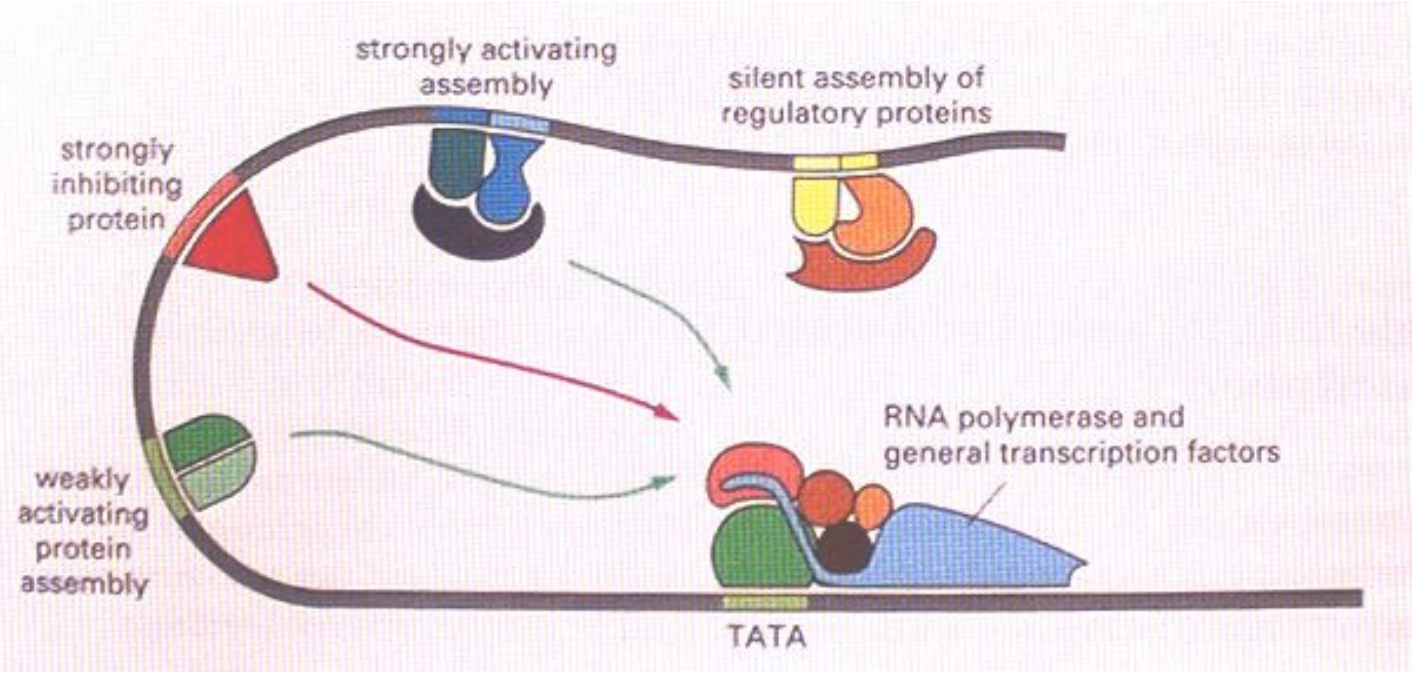
Effects of SV on the genome

- **Complete loss/gain** of a particular **region/gene**
- **Disruption of local interactions** in the genome
  - Increase/decrease expression of a gene
- **Disruption of global interactions** in the genome
  - Interaction with remote elements in the genome
  - Altering positions of chromosomes in the nucleus

Disruption of local interactions

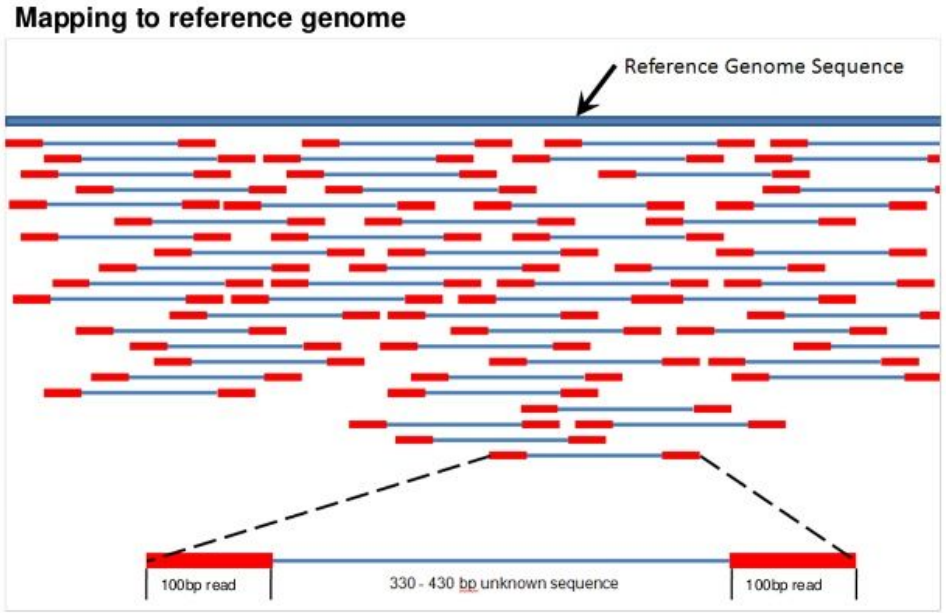


Disruption of global interactions



NGS short reads - recap

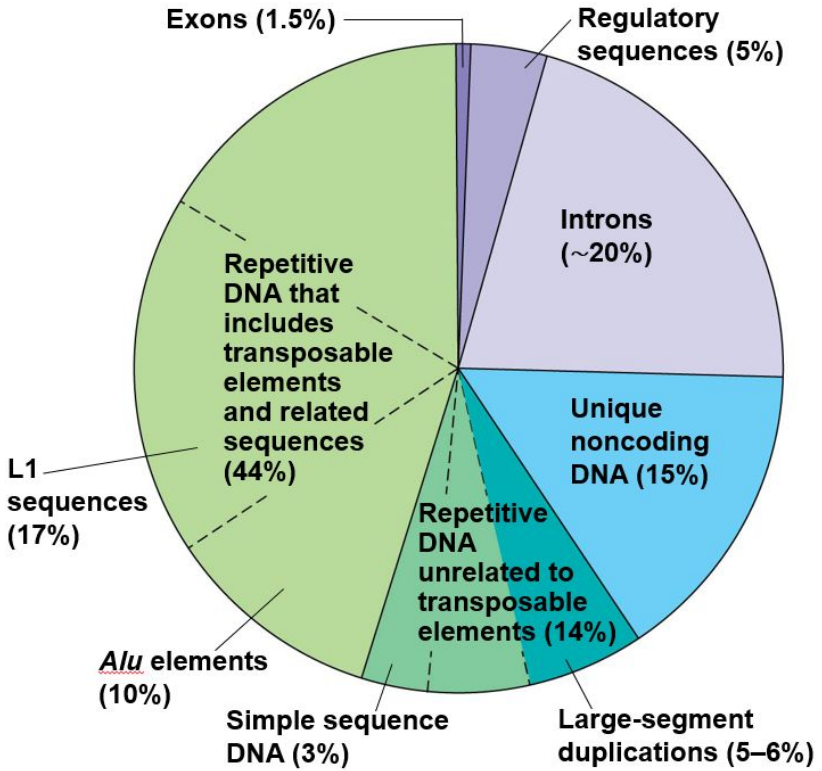
- Fragment size roughly **400-700bp**
- Paired-end (**PE**) reads **100-150bp** in length



Adapted from wikipedia

Genome structure

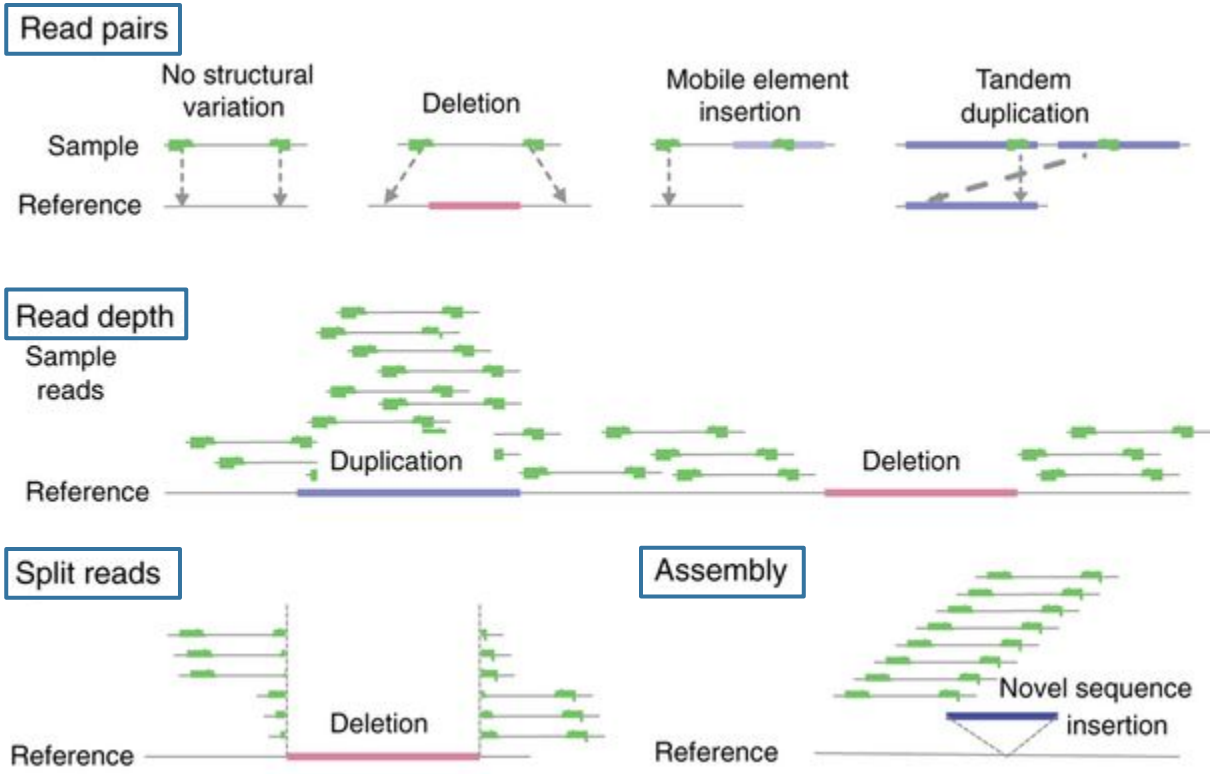
- **60% of the genome** is made of **repetitive sequences**
- Difficult to uniquely map a read to the correct position in the genome



SV detection - drawbacks

- Repetitive DNA
- Short reads (100-150bp)
- Short fragment size (distance between paired reads)

SV detection using short reads NGS



SV encoded in VCF

#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT NA00001

1 2827693 . CCGTGGATGCGGGGACCCGCATCCCCTCTCCCTTCACAGCTGAGTGACCCACATCCCCTCTCCCCTCGCA C . PASS **SVTYPE=DEL**;END=2827680;BKPTID=Pindel_LCS_D1099159;HOMLEN=1;HOMSEQ=C;**SVLEN=-66** GT:GQ 1/1:13.9

2 321682 . T <DEL> 6 PASS IMPRECISE;**SVTYPE=DEL**;END=321887;**SVLEN=-105**;CIPOS=-56,20;CIEND=-10,62 GT:GQ 0/1:12

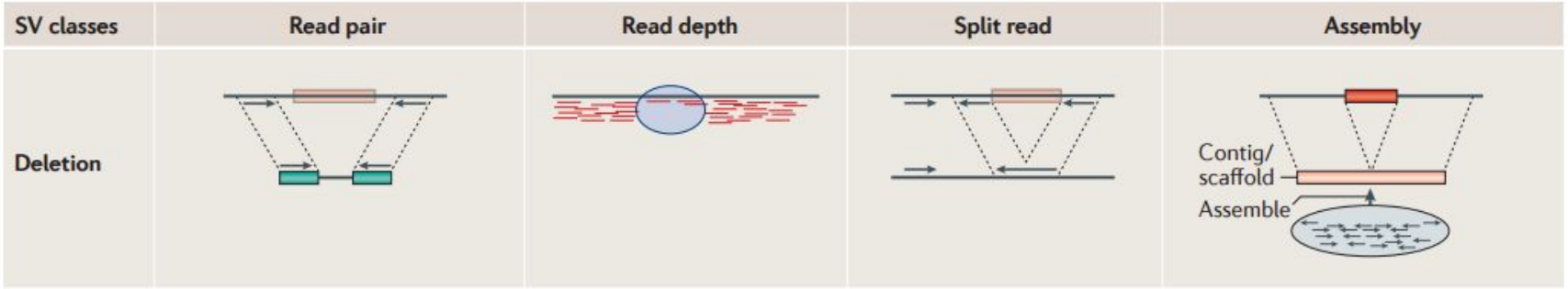
3 12665100 . A <DUP> 14 PASS IMPRECISE;**SVTYPE=DUP**;END=12686200;**SVLEN=21100**;CIPOS=-500,500;CIEND=-500,500 GT:GQ:CN:CNQ ./.:0:3:16.2

SV classification

- **Balanced SVs -** *No change in length of the genome*
  - Inversions
  - Translocations
- **Unbalanced SVs** - *Alteration of genome length*
  - Insertions
  - **CNV** (copy number variation) - deletions, duplications

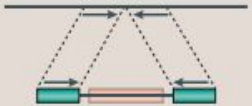
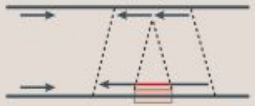
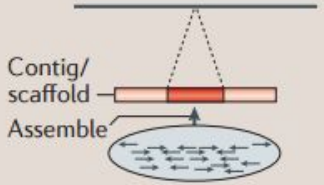
SV - Deletions

- **Read pair** - increased interpair mapping distance
- **Read depth** - fewer reads
- **Split read** - single read is “merged” from two segments surrounding deletion
- **Assembly -** assembled sequence shows “gap”



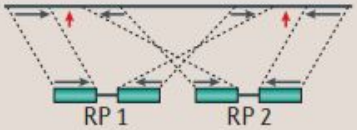
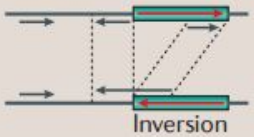
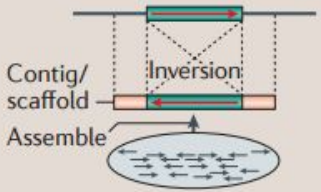
SV - Insertions

- **Read pair** - decreased interpair mapping distance
- **Read depth** - not applicable
- **Split read** - single read is split into two segments surrounding novel insertion sequence
- **Assembly -** assembled sequence contains novel sequence

SV classes	Read pair	Read depth	Split read	Assembly
Novel sequence insertion		Not applicable		

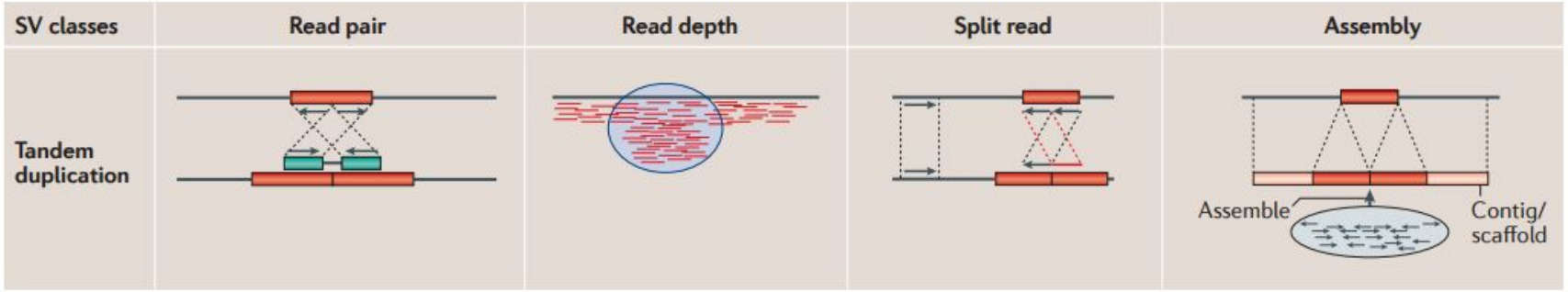
SV - Inversions

- **Read pair** - aberrant mapping and interpair distance
- **Read depth** - not applicable
- **Split read** - single read is split into two segments one of which is inverted
- **Assembly -** assembled sequence with inverted sequence

SV classes	Read pair	Read depth	Split read	Assembly
Inversion		Not applicable		

SV - Duplication

- **Read pair** - aberrant mapping and interpair distance
- **Read depth** - increased read depth
- **Split read** - single read is split into two segments one of which is inverted
- **Assembly -** assembled sequence with inverted sequence



SV detection using long reads

- **Pros**:
  - Ability for reads to span over entire variant
- **Cons**:
  - Higher error rate
  - Inability to detect inversions due to singe-end approach
- Still ineffective for extremely long variation

CNV copy number variants

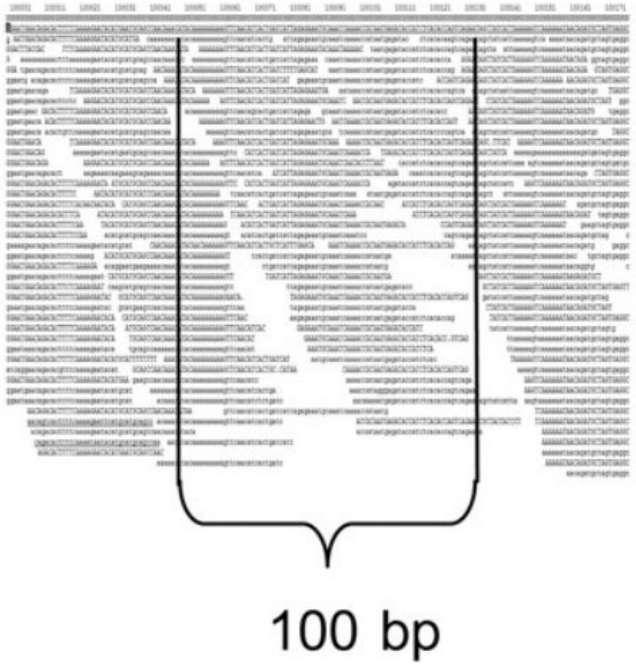
- Calling **CNVs** from whole genome data (WGS)
- Sensitive and accurate detection of copy number variants using **read depth of coverage**

CNV calling using read depth

- Align whole-genome sequences (high-coverage)
- Filter out reads with low mapping quality (PHRED < 30)
- Count read depth in windows (100bp)
- Adjust read-depth according to GC content of window
- Combine neighboring windows to maximize score

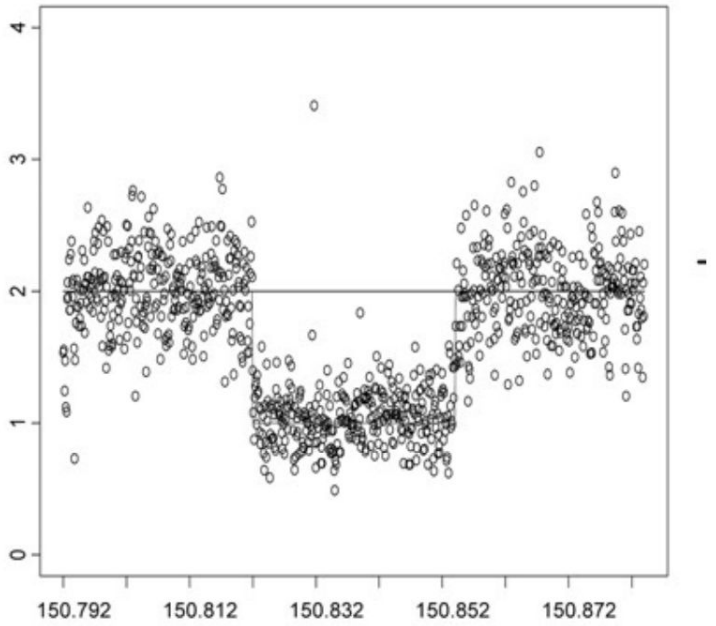
CNV calling using read depth

- Align whole-genome sequences (high-coverage)
- Filter out reads with low mapping quality (PHRED < 30)
- Count read depth in windows (100bp)

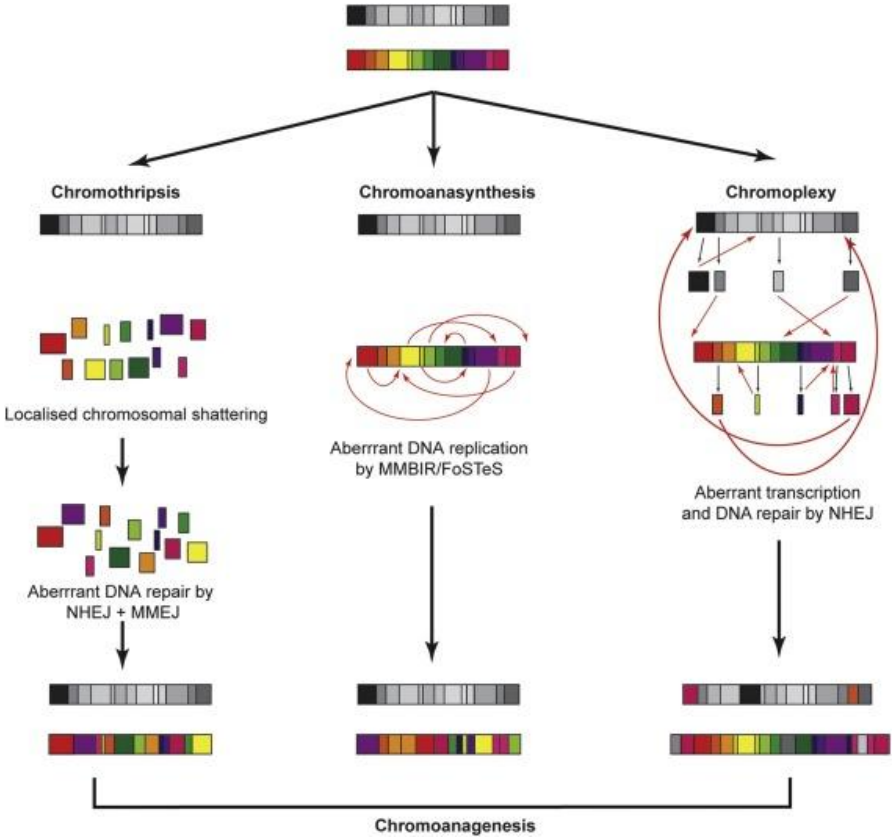


CNV calling using read depth

- **Event detection**
  A deletion or duplication is evident as a decrease or increase across multiple consecutive windows



Chromothripsis

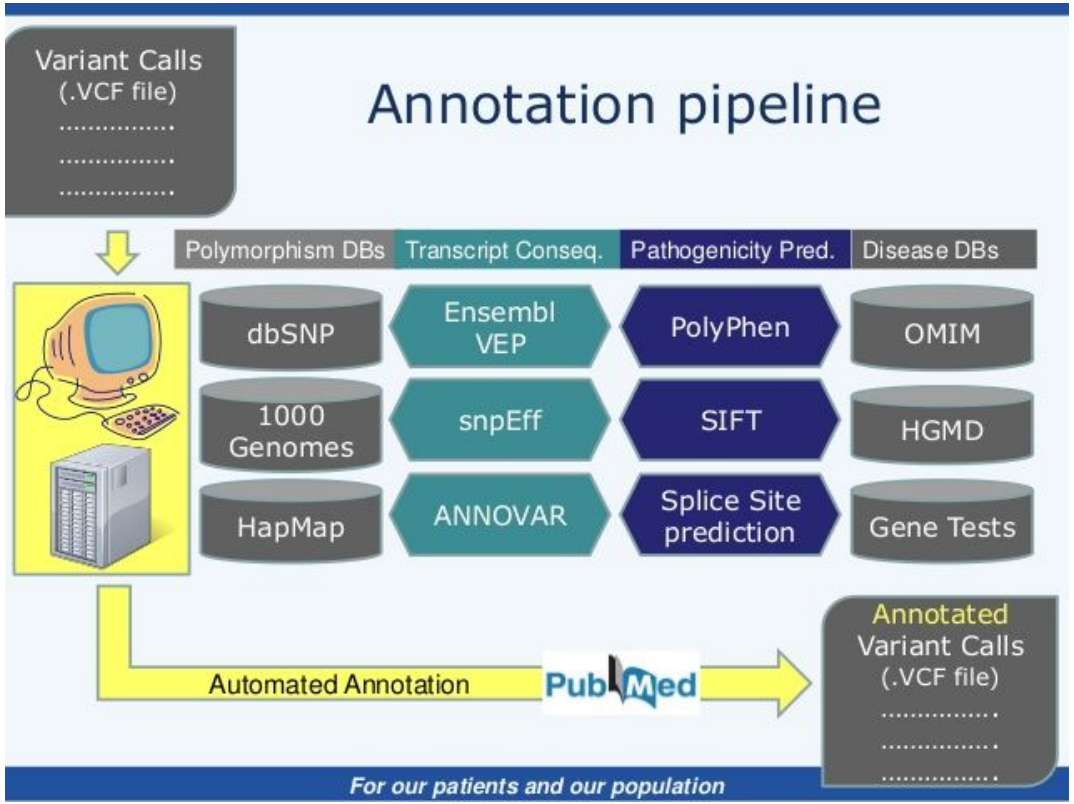


Variants annotation

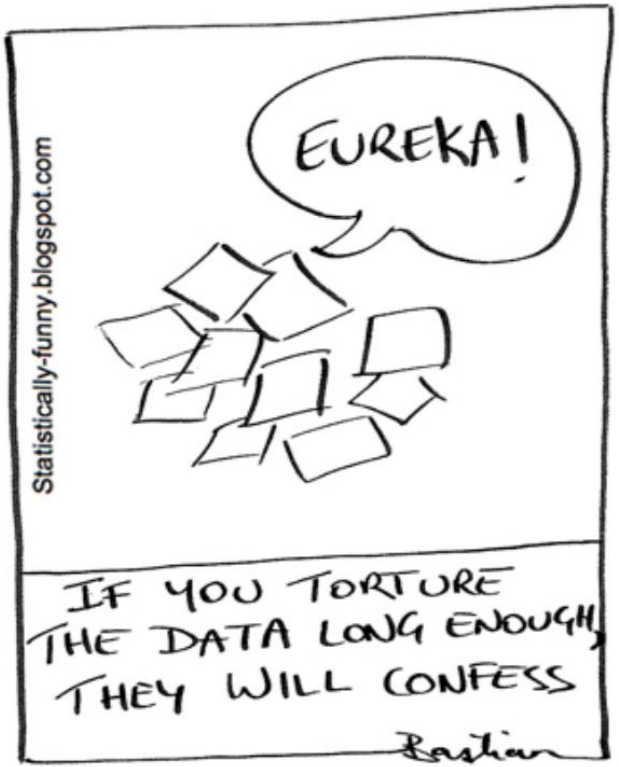
Variants annotation

- Identify the **gene(s)** that **overlaps** with the **variant**
- Determine whether the variant is located in an **exon**
- If the variant is an **SNV**, determine whether the encoded amino acid is changed, if so annotate as missense
- If the variant is located right before or after an exon/intron boundary, annotate as splicing

Variants annotation pipeline



Variant calling in short



Additional links

- Genome Sequencing and Structural Variation
- Encoding structural variants in VCF format
- Variant calling and annotation
- A geometric approach for classification and comparison of structural variants
- Structural variation in the human genome

SV - Deletions Exercise

- Simplified deletion detection example based on **read depth** and **split reads**
- Find breakend candidates using split reads
- Detect SV type using read depth

SV - Deletions Exercise

- Simplified deletion detection example based on **read depth** using pysam:
  - Load BAM file

```
alignment = pysam.AlignmentFile("/sbgenomics/project-files/simulated_somatic.bam", "rb")
```

  - Plot read depth

```
alignments = alignment.fetch('20', 100, 200)
```

  - Find deletions

SV - Deletions Exercise

- Deletion detection based on split reads:
  - Locate soft clip locations
- CIGAR string

```
for read in alignments:
    if 'S' in read.cigarstring:
```

- 73M27S
  - U read-u imamo prvo 73 matcha

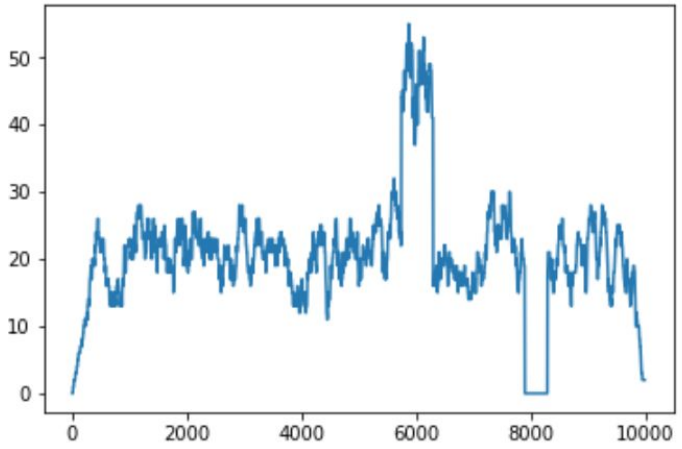
M	BAM_CMATCH	0
I	BAM_CINS	1
D	BAM_CDEL	2
N	BAM_CREF_SKIP	3
S	BAM_CSOFT_CLIP	4
H	BAM_CHARD_CLIP	5
P	BAM_CPAD	6
=	BAM_CEQUAL	7
X	BAM_CDIFF	8
B	BAM_CBACK	9

```
import pysam
import matplotlib.pyplot as plt
```

```
# Read BAM file
alignment = pysam.AlignmentFile("/sbgenomics/project-files/simulated_somatic.bam", "rb")
```

```
# Make read depth chart
interval_length = 5
reference_length = alignment.lengths[0]
intervals = [i*interval_length for i in range(round(reference_length / interval_length))]
read_depth = [
    len(list(alignment.fetch('20', start, end)))
    for start, end in zip(intervals[1:-1], intervals[2:])
]
```

```
plt.plot(intervals[1:-1], read_depth)
plt.show()
```



```
# Making a simple duplication and deletion caller:
average_coverage = sum(read_depth)/len(read_depth)
previous_depth = read_depth[0]
deletion_start = 0
duplication_start = 0

SVs = {
    'DEL': [],
    'DUP': []
}
# Kako cemo da definisemo pocetak SV?
# Hajde da se dogovorimo da je neophodna promena u read depth-u od bar 30% average coverage-a
# Probajte i sa drugim vrednostima
threshold = 0.3 * average_coverage

for curr_bin, depth in enumerate(read_depth):
    if depth - previous_depth < -threshold and not(deletion_start):
        if duplication_start:
            SVs['DUP'].append((duplication_start * interval_length, curr_bin * interval_length))
            duplication_start = 0
        else:
            deletion_start = curr_bin
    if depth - previous_depth > threshold and not(duplication_start):
        if deletion_start:
            SVs['DEL'].append((deletion_start * interval_length, curr_bin * interval_length))
            deletion_start = 0
        else:
            duplication_start = curr_bin
    previous_depth = depth
```

SV detection using split reads

```
alignments = alignment.fetch('20')
breakpoints = []
for read in alignments:
    if 'S' in read.cigarstring and not read.is_secondary:
        cigar = read.cigarstring
        start = read.reference_start
        if cigar.find('M') < cigar.find('S'):
            location = int(cigar.split('M')[0])
            breakpoints.append(start + location)
        elif cigar.find('S') < cigar.find('M'):
            breakpoints.append(start + 1)

# Ovakav pristup nam ne govori da li se breakend (mesto pucanja hromozoma)
# odnosi na pocetak ili kraj varijante, ni koji je tip varijante u pitanju.
# Za odgovor na ova pitanja bilo bi potrebno malo izmeniti algoritam, ili
# ga kombinovati sa drugim algoritmima detekcije strukturnih varijanti
print(set(breakpoints))
```

```
{8300, 6300, 5750, 6297, 7900}
```