



The Elizabeth H.  
and James S. McDonnell III

**McDONNELL  
GENOME INSTITUTE**  
at Washington University

# GenViz Module 1: Introduction to genomic data visualization and interpretation

Malachi Griffith, Obi Griffith, Zachary Skidmore  
Genomic Data Visualization and Interpretation

April 8-12, 2019  
Freie Universität Berlin



## Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)

This is a human-readable summary of (and not a substitute for) the [license](#). [Disclaimer](#).

### You are free to:

**Share** — copy and redistribute the material in any medium or format

**Adapt** — remix, transform, and build upon the material for any purpose, even commercially.



The licensor cannot revoke these freedoms as long as you follow the license terms.

### Under the following terms:



**Attribution** — You must give [appropriate credit](#), provide a link to the license, and [indicate if changes were made](#). You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.



**ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the [same license](#) as the original.

**No additional restrictions** — You may not apply legal terms or [technological measures](#) that legally restrict others from doing anything the license permits.

# Learning objectives of the course

---

- Module 1: Introduction to genomic data visualization and interpretation
- Module 2: Using R for genomic data visualization and interpretation
- Module 3: Introduction to GenVisR
- Module 4: Expression profiling, visualization, and interpretation
- Module 5: Variant annotation and interpretation
- Module 6: Q & A, discussion, integrated assignments, and working with your own data
- Tutorials
  - Provide working examples of data visualization and interpretation
  - Self contained, self explanatory, portable

# Learning objectives of module 1

---

- **Introduction to genomic data visualization and interpretation**
- Introduction
- ‘Omics technologies
- Common genomic file formats
- Common problems
- Genome browsers
- Fundamentals of data visualization
- Best practices in visualization
- Examples of visualization in R

# Introduction

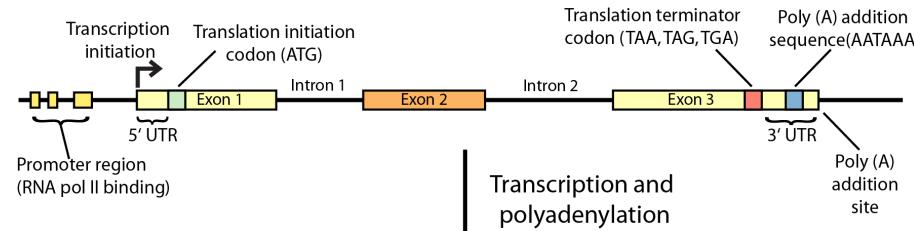
# Why do we create visualizations of genomic data?

---

- Data exploration and interpretation of results
  - QC analysis
  - Understanding whether/how an experiment worked
  - Discovery
- Communication
  - Slides for presentations
    - e.g. Keynote, Powerpoint, etc.
  - Figures for publications
    - e.g. PDFs, PNGs, etc.
      - Illustrator, Gimp, Inkscape, etc.
      - R and R Studio

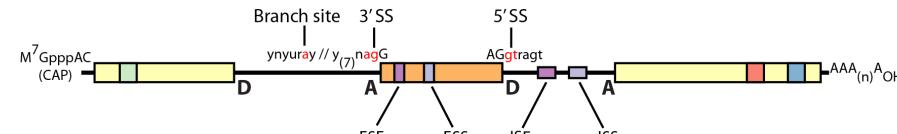
# Review of central dogma

## Double-stranded genomic DNA template



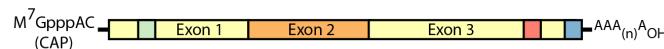
Transcription and polyadenylation

## Single-stranded pre-mRNA (nuclear RNA)



RNA processing

## Mature mRNA

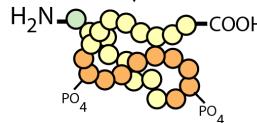


Export to cytoplasm and translation

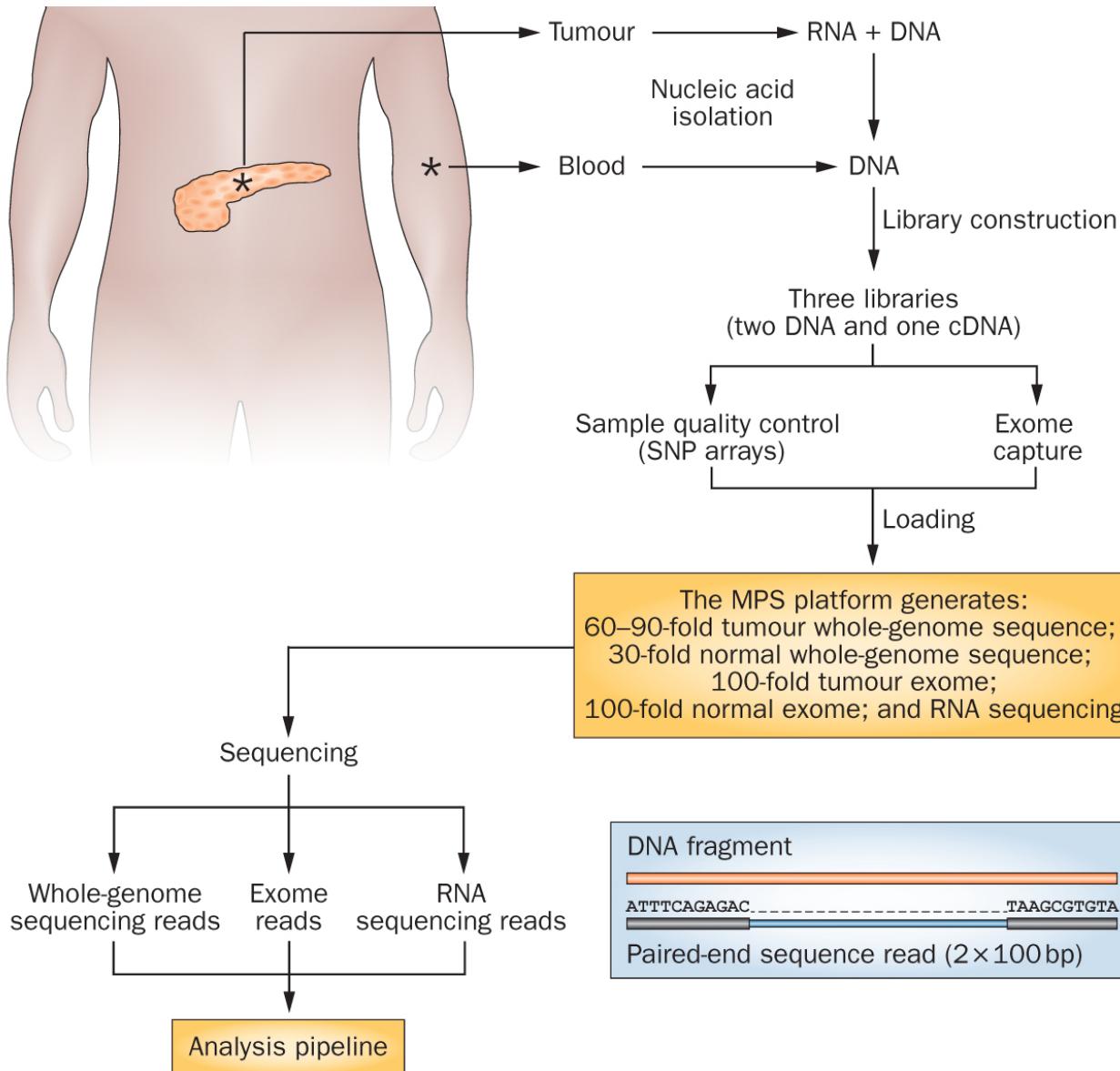
## Protein (amino acid sequence)



Folding, posttranslational modification, subcellular localization, etc.



# Genomics research has exploded with the rapid advances in DNA sequencing technologies



# Short read alignments – The fundamental unit of most omics assays



- Alignment is about fitting individual pieces (reads) into the correct part of the puzzle
- The human genome project gave us the picture on the box cover (the reference genome)
- Imperfections in how the pieces fit can indicate changes to a copy of the picture

Reference: AGCCTGAGACCGTAAAAAA**AGTCAAG**

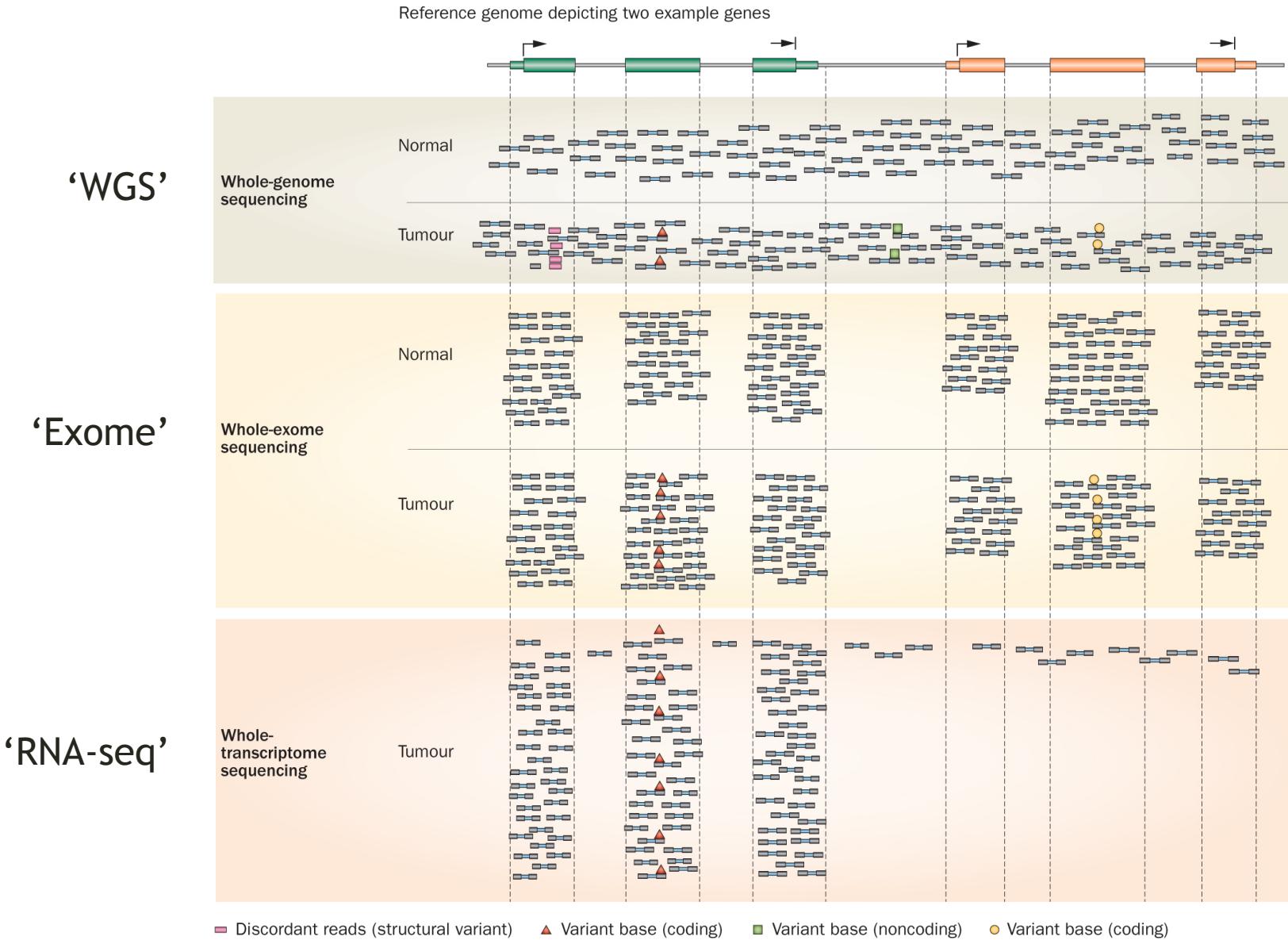
||||| ||||| ||||| |||||

GAGACCGTAAAAAC**CGTC**

A variant!

A read sequence:

# Whole genome, exome and transcriptome sequencing allows us to detect and confirm many different variant types



# Common genomic file formats

# Common genomic data file formats

---

- [FASTA](#)
- [FASTQ](#)
- [GTF](#)
- [BED](#)
- [VCF](#)
- [VEP](#)
- [MAF](#)
- [SAM/BAM/CRAM](#)
- Many, many custom data formats output by specialized tools ... often in TSV format

# Fasta – format for representing nucleic acid or amino acid sequences

```
>AY274119.3 Severe acute respiratory syndrome-related coronavirus  
isolate Tor2, complete genome
```

```
ATATTAGGTTTTACCTACCCAGGAAAAGCCAACCAACCTCGATCTCTTGTAGATCTGTTCTCTAAACGA  
ACTTTAAAATCTGTGTAGCTGTCGCTCGGCTGCATGCCTAGTGCACCTACGCAGTATAAACATAATAAA  
TTTTACTGTCGTTGACAAGAACGAGTAACCTCGTCCCTCTTCTGCAGACTGCTTACGGTTCGTCCGTGT  
TGCAGTCGATCATCAGCATACTAGGTTCTCGTCCGGTGTGACCGAAAGGTAAGATGGAGAGCCTGTTC  
TTGGGTGTCAACGAGAAAACACACGTCCAACTCAGTTGCCCTGTCCCTCAGGTTAGAGACGTGCTAGTGCG  
TGGCTTCGGGGACTCTGTGGAAGAGGCCCTATCGGAGGCACGTGAACACCTCAAAATGGCACTTGTGGT
```

...

```
>FJ882960.1 SARS coronavirus ExoN1 isolate P3pp34, complete genome  
CGATCTCTGTAGATCTGTTCTCTAAACGAACCTTAAAATCTGTGTAGCTGTCGCTCGGCTGCATGCCTA  
GTGCACCTACGCAGTATAAACATAATAAAATTACTGTCGTTGACAAGAACGAGTAACCTCGTCCCTCT  
TCTGCAGACTGCTTACGGTTCTCGTCCGTGTTGCAGTCGATCATCAGCATACTAGGTTCTCGTCCGGGTGT
```

...

First line starts with “>” header or “Comment”; used as a summary/description, often starting with unique accession/identifier

Subsequent lines contain sequence

- Interleaved: sequence broken into multiple lines of characters
- Sequential: entire sequence on a single line

Multiple sequence FASTA obtained by simply concatenating multiple FASTA records together

# Fastq – format for representing raw sequence – base calls and quality values

```
@HWUSI-EAS100R:6:73:941:1973#0/1
```

```
CTTTTTATTGGTCTGACTGGGTTGATTCAAAA
```

```
+
```

```
CCCFFFFFHHHGJHIIJHIHIIIFHIJJJJJJGIBBFGE
```

```
@HWUSI-EAS100R:6:2303:11793:37095#0/1
```

```
ATGAATTATAAGGGCTGTATTTAATTTGCATTTAA
```

```
+
```

```
@@??BDDFFF<FHEGFFGGIEBGHIIIIIBEHIIGIH<FHE
```

First line starts with "@" header or "Comment"; followed by sequence identifier and optional description

Sequence line

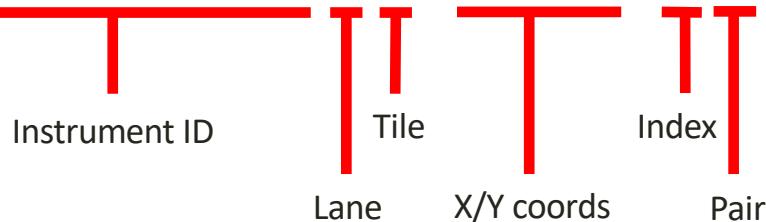
Spacer line

Quality values

Next sequence record

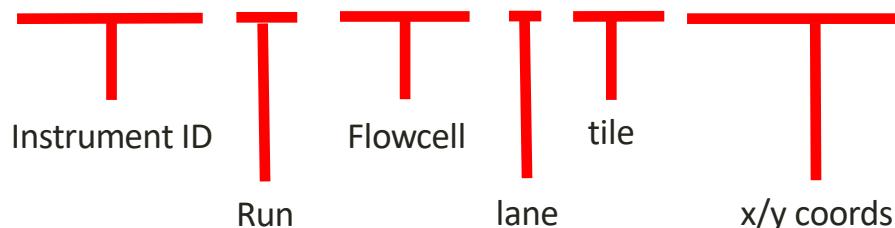
# Read naming conventions

@HWUSI-EAS100R:6:73:941:1973#0/1



Filter  
status

@EAS139:136:FC706VJ:2:2104:15343:197393:GATTACT+GTCTTAAC 1:Y:0:ATCACG

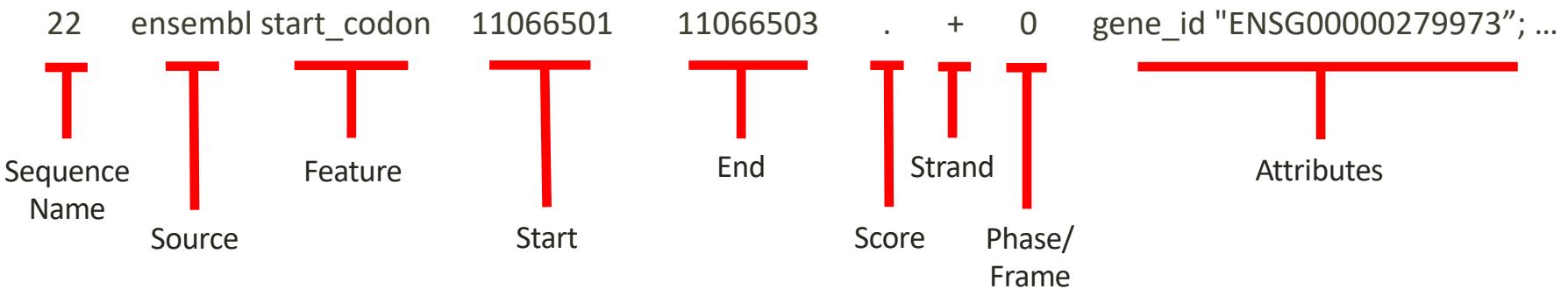


# GFF/GTF - representing sequence features

---

- GFF – General/Generic Feature Format; Gene Finding Format
  - Two versions in wide use
    - GFF2 (see also GTF)
    - GFF3
      - Added formal support for multiple levels (and direction) of hierarchy  
(e.g., gene -> transcript -> exon)
- GTF – Gene Transfer Format
  - An extension of GFF2
- GFF2, GFF3 and GTF are all tab-separate files with 9 fields
  - Differing content in 9<sup>th</sup> column

# Ensembl GTF example record

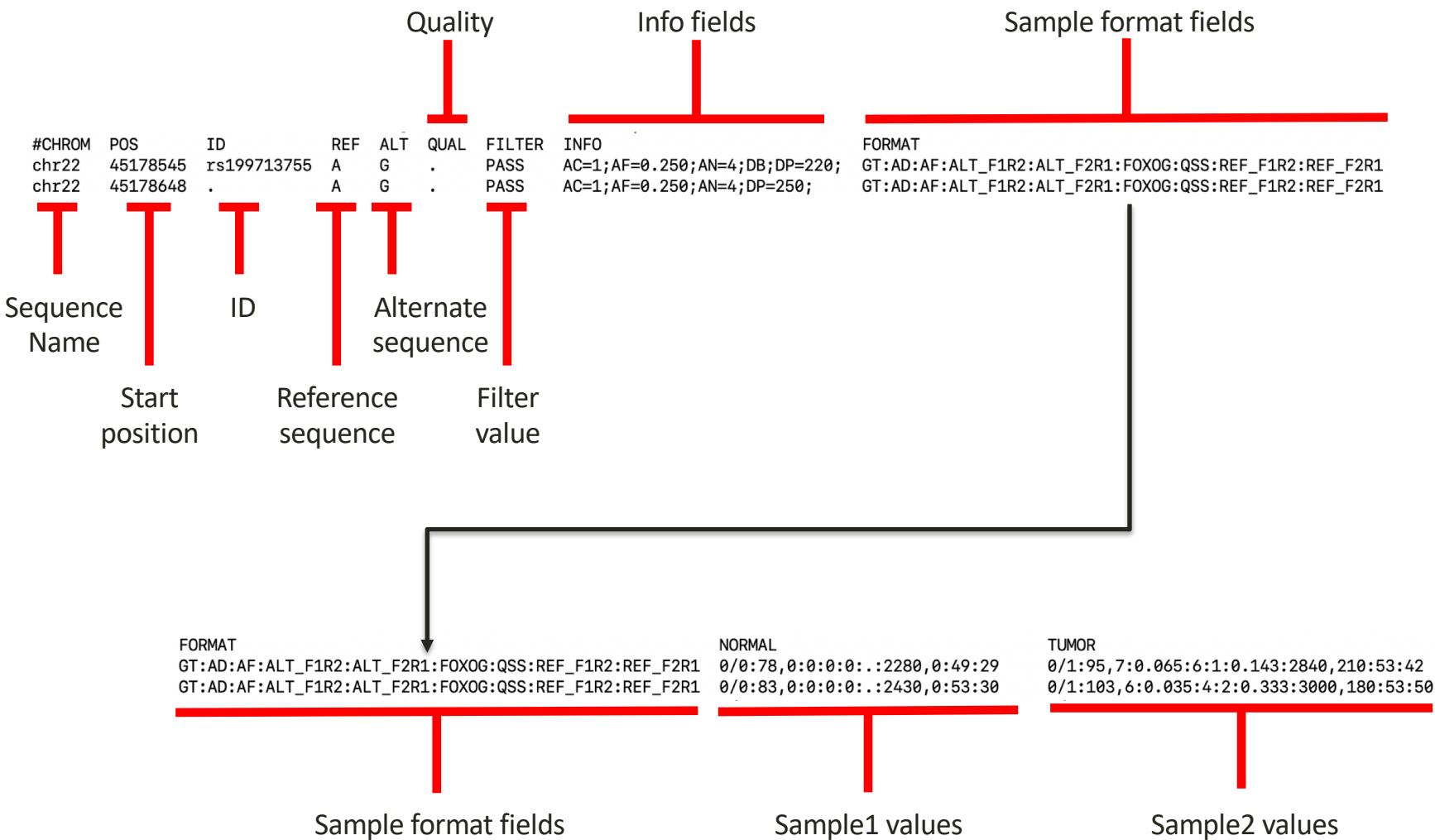


## Example of attributes string:

```
gene_id "ENSG00000279973"; gene_version "1"; transcript_id "ENST00000624155"; transcript_version "1";  
exon_number "1"; gene_name "BAGE5"; gene_source "ensembl"; gene_biotype "protein_coding"; transcript_name  
"BAGE5-201"; transcript_source "ensembl"; transcript_biotype "protein_coding"; tag "basic"; transcript_support_level  
"1";
```

Note: there will be many GTF records/rows per transcript per gene (UTRs, start\_codon, exons, etc)

# VCF file



Details of the VCF file format: [hts-specs](#), [VCF-v4.2.pdf](#)

# Example of SAM/BAM file format

## Example SAM/BAM header section (abbreviated)

```
mgriffit@linus270 ~> samtools view -H /gscmnt/gc13001/info/model_data/2891632684/build136494552/alignments/136080019.bam | grep -P "SN\|22\|HD\|RG\|PG"
@HD VN:1.4 SO:coordinate
@SQ SN:22 LN:51304566 UR:ftp://ftp.ncbi.nih.gov/genbank/organisms/vertebrates_mammals/Homo_sapiens/GRCh37/special_requests/GRCh37-lite.fa.gz AS:GRCh37-lite M5:a718aca6135fdca8357d5bfe
4211dd SP:Homo sapiens
@RG ID:2888721359 PL:illumina PU:D1BA4ACXX.3 LB:H_KA-452198-0817007-cDNA-3-lib1 PI:365 DS:paired end DT:2012-10-03T19:00:00-0500 SM:H_KA-452198-0817007 CN:WUGSC
@PG ID:2888721359 VN:2.0.8 CL:tophat --library-type fr-secondstrand --bowtie-version=2.1.0
@PG ID:MarkDuplicates PN:MarkDuplicates PP:2888721359 VN:1.85(exported) CL:net.sff.picard.sam.MarkDuplicates INPUT=/gscmnt/gc13001/info/build_merged_alignments/merged-alignment-blade10-2-5.gsc.wustl.edu-jwalker-15434-136080019/scratch-IlG6Y/H_KA-452198-0817007-cDNA-3-lib1-2888360300.bam OUTPUT=/gscmnt/gc13001/info/build_merged_alignments/merged-alignment-blade10-2-5.gsc.wustl.edu-jwalker-15434-136080019/scratch-IlG6Y/H_KA-452198-0817007-cDNA-3-lib1-2888360300-post_dup.bam METRICS_FILE=/gscmnt/gc13001/info/build_merged_alignments/merged-alignment-blade10-2-5.gsc.wustl.edu-jwalker-15434-136080019/staging-liuJS/H_KA-452198-0817007-cDNA-3-lib1-2888360300.metrics REMOVE_DUPLICATES=false ALLOW_SORTED=true MAX_FILE_HANDLES_FOR_READ_ENDS_MAP=9500 TMP_DIR=/gscmnt/gc13001/info/build_merged_alignments/merged-alignment-blade10-2-5.gsc.wustl.edu-jwalker-15434-136080019/scratch-IlG6Y/ VALIDATION_STRINGENCY=SILENT MAX_RECORDS_IN_RAM=500000 PROGRAM_RECORD_ID=MarkDuplicates PROGRAM_GROUP_NAME=MarkDuplicates MAX_SEQUENCES_FOR_DISK_READ_ENDS_MAP=50000 SORTING_COLLECTION_SIZE_RATIO=0.25 READ_NAME_REGEX=[a-zA-Z0-9]+:[0-9]+:[0-9]+:[0-9]+.* OPTICAL_DUPLICATE_PIXEL_DISTANCE=100 VERBOSITY=INFO QUIET=false COMPRESSION_LEVEL=5 CREATE_INDEX=false CREATE_MDS_FILE=false
mgriffit@linus270 ~>
```

## Example SAM/BAM alignment section (only 10 alignments shown)

# Introduction to the SAM/BAM format

- The specification
  - <http://samtools.sourceforge.net/SAM1.pdf>
- The SAM format consists of two sections:
  - Header section
    - Used to describe source of data, reference sequence, method of alignment, etc.
  - Alignment section
    - Used to describe the read, quality of the read, and nature alignment of the read to a region of the genome
- BAM is a compressed version of SAM
  - Compressed using lossless BGZF format
  - Other BAM compression strategies are a subject of research. See 'CRAM' format for example
- BAM files are usually 'indexed'
  - A '.bai' file will be found beside the '.bam' file
  - Indexing aims to achieve fast retrieval of alignments overlapping a specified region without going through the whole alignments. BAM must be sorted by the reference ID and then the leftmost coordinate before indexing

# SAM/BAM header section

---

- Used to describe source of data, reference sequence, method of alignment, etc.
- Each section begins with character ‘@’ followed by a two-letter record type code. These are followed by two-letter tags and values
  - @HD The header line
    - VN: format version
    - SO: Sorting order of alignments
  - @SQ Reference sequence dictionary
    - SN: reference sequence name
    - LN: reference sequence length
    - SP: species
  - @RG Read group
    - ID: read group identifier
    - CN: name of sequencing center
    - SM: sample name
  - @PG Program
    - PN: program name
    - VN: program version

# SAM/BAM alignment section

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	[!-?A-~]{1,255}	Query template NAME
★ 2	FLAG	Int	[0,2 <sup>16</sup> -1]	bitwise FLAG
3	RNAME	String	\*  [!-()+-<>-~] [!-~]*	Reference sequence NAME
4	POS	Int	[0,2 <sup>29</sup> -1]	1-based leftmost mapping POSition
5	MAPQ	Int	[0,2 <sup>8</sup> -1]	MAPping Quality
★ 6	CIGAR	String	\*  ([0-9]+[MIDNSHPX=])+	CIGAR string
7	RNEXT	String	\* =  [!-()+-<>-~] [!-~]*	Ref. name of the mate/next segment
8	PNEXT	Int	[0,2 <sup>29</sup> -1]	Position of the mate/next segment
9	TLEN	Int	[-2 <sup>29</sup> +1,2 <sup>29</sup> -1]	observed Template LENGTH
10	SEQ	String	\*  [A-Za-z.=.]+	segment SEQuence
11	QUAL	String	[!-~]+	ASCII of Phred-scaled base QUALity+33

## Example values

```
1 QNAME e.g. HWI-ST495_129147882:1:2302:10269:12362 (QNAME)
2 FLAG e.g. 99
3 RNAME e.g. 1
4 POS e.g. 11623
5 MAPQ e.g. 3
6 CIGAR e.g. 100M
7 RNEXT e.g. =
8 PNEXT e.g. 11740
9 TLEN e.g. 217
10 SEQ e.g. CCTGTTCTCCACAAAGTGTACTTTGGATTTGCCAGTCTAACAGGTGAAGCCCTGGAGATTCTATTAGTGATTGGCTGGGCCTGCCATGT
11 QUAL e.g. CCCFFFFFHJHJIJFIJJJJJJJJJHJJHJJJJJJGGHJHJJJJJJJJGHGGIJJJJJJJIJEEHHHFFFFCDCDDDDDDB@ACDD
```

# SAM/BAM flags explained

- <http://broadinstitute.github.io/picard/explain-flags.html>
- 12 bitwise flags describing the alignment
- These flags are stored as a binary string of length 11 instead of 11 columns of data
- Value of '1' indicates the flag is set. e.g. 00100000000
- All combinations can be represented as a number from 1 to 2048 (i.e.  $2^{11}-1$ ). This number is used in the BAM/SAM file. You can specify 'required' or 'filter' flags in samtools view using the '-f' and '-F' options respectively

Bit	Description
1	0x1 template having multiple segments in sequencing
2	0x2 each segment properly aligned according to the aligner
4	0x4 segment unmapped
8	0x8 next segment in the template unmapped
16	0x10 SEQ being reverse complemented
32	0x20 SEQ of the next segment in the template being reverse complemented
64	0x40 the first segment in the template
128	0x80 the last segment in the template
256	0x100 secondary alignment
512	0x200 not passing filters, such as platform/vendor quality controls
1024	0x400 PCR or optical duplicate
2048	0x800 supplementary alignment

Note that to maximize confusion, each bit is described in the SAM specification using its hexadecimal representation (i.e., '0x10' = 16 and '0x40' = 64).

# CIGAR strings explained

Op	BAM	Description
M	0	alignment match (can be a sequence match or mismatch)
I	1	insertion to the reference
D	2	deletion from the reference
N	3	skipped region from the reference
S	4	soft clipping (clipped sequences present in SEQ)
H	5	hard clipping (clipped sequences NOT present in SEQ)
P	6	padding (silent deletion from padded reference)
=	7	sequence match
X	8	sequence mismatch

- The CIGAR string is a sequence of base lengths and associated ‘operations’ that are used to indicate which bases align to the reference (either a match or mismatch), are deleted, are inserted, represent introns, etc.
- e.g. 81M859N19M
  - A 100 bp read consists of: 81 bases of alignment to reference, 859 bases skipped (an intron), 19 bases of alignment

# CRAM files

- CRAM is an ultra-compressed version of a BAM file
  - Usually between 30-60% smaller than the corresponding BAM
- Stores “diffs” from the reference genome
  - requires the matching reference genome to restore original data!
- Base quality binning may be used as well
- Some tools still require conversion back to bam

Quality Score Bins	Example of Empirically Mapped Quality Scores*
N (no call)	N (no call)
2–9	6
10–19	15
20–24	22
25–29	27
30–34	33
35–39	37
≥ 40	40

By replacing the quality scores between 19 and 25 with a new score of 22, data storage space is conserved.

\*The mapped quality score of each bin (except “N”) is subject to change depending on individual Q-tables.

# Introduction to the BED format

---

- When working with BAM files, it is very common to want to examine a focused subset of the reference genome
  - e.g. the exons of a gene
- These subsets are commonly specified in ‘BED’ files
  - <https://genome.ucsc.edu/FAQ/FAQformat.html#format1>
- Many BAM manipulation tools accept regions of interest in BED format
- Basic BED format (tab separated):
  - Chromosome name, start position, end position
  - Coordinates in BED format are 0 based

# Manipulation of alignment, region and variant files

- Several tools are used ubiquitously in sequence analysis to manipulate these files
- SAM/BAM files
  - samtools
  - bamtools
  - picard
- BED files
  - bedtools
  - bedops
- VCF files
  - vcftools
  - bcftools

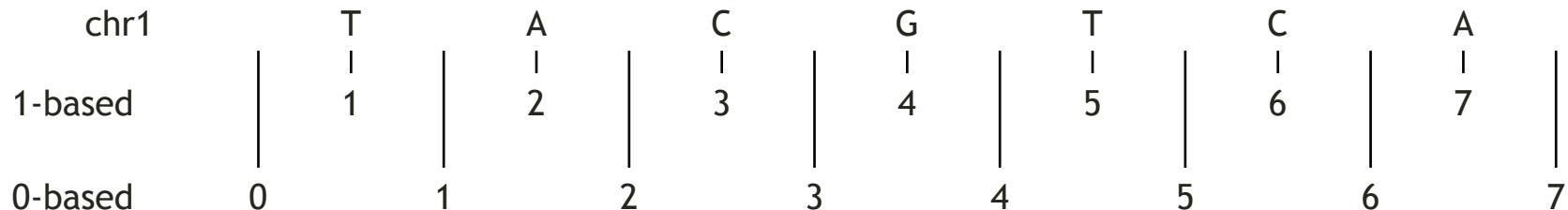
# Common problems

# Common sources of confusion

---

- Genomic coordinate systems
- Genome builds
- Variant representation

# Genomic coordinates – 1 vs 0 based



	1-based	0-based
Indicate a single nucleotide	chr1:4-4 G	chr1:3-4 G
Indicate a range of nucleotides	chr1:2-4 ACG	chr1:1-4 ACG
Indicate a single nucleotide variant	chr1:5-5 T/A	chr1:4-5 T/A

- **1-based** : Single nucleotides, variant positions, or ranges are specified directly by their corresponding nucleotide numbers
  - GFF, SAM, VCF, Ensembl browser, ...
- **0-based**: Single nucleotides, variant positions, or ranges are specified by the coordinates that flank them
  - BED, BAM, UCSC browser, ...

# Genome builds

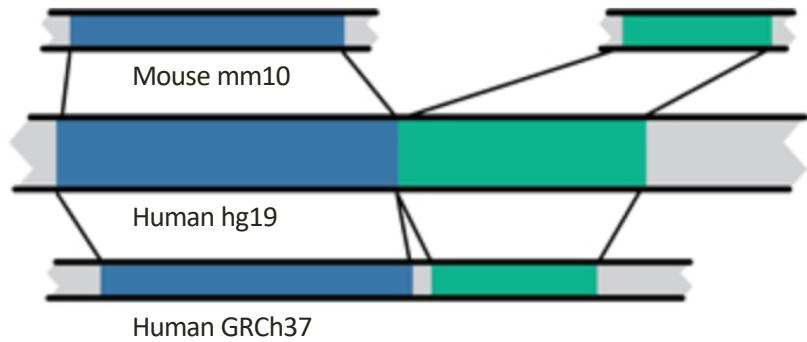
## Reference Genome builds

Current human: GRCh38, hg38, b38  
alternate: GRCh38v2\_ccdg

Previous human: GRCh37, hg19, b37

Current mouse: GRCm38, mm10

## Lift-over



# Variant shifting (alignment) and parsimony/trimming

Reference and alternative alleles of a CA short tandem repeat (STR)

REF  
ALT

GGGCACACAC**CA**GGG  
GGGCACACACAGGG

↑ CA deletion from the reference

Genome Reference		Variant Call Format		
REF	ALT	POS	REF	ALT
	GGGCACACACAGGG	8	CA	.
REF	CA			Not left aligned and alternate allele is empty
ALT	.			Not left aligned but parsimonious
REF	CAC	6	CAC	C
ALT	C			Not right trimmed
REF	GCACA	3	GCACA	GCA
ALT	GCA			Not right trimmed
REF	GGCA	2	GGCA	GG
ALT	GG			Not left trimmed
REF	GCA	3	GCA	G
ALT	G			Normalized (left aligned & parsimonious)

Alleles represented against the human genome reference. Allele pairs are colored the same, all are representations of the same variant.

Alleles represented in Variant Call Format, all are representations of the same variant.

**Parsimony:** representing variant in as few nucleotides as possible without reducing the length of any allele to 0

**Left (right) aligning =** shifting the start position of a variant as far to the left (right) as possible

The five colors above represent five different ways of showing the same variant (and how it would be represented in a VCF, though only the last representation is correct according to the VCF specification)

# Genome browsers

# Genome browsers - Ensembl

**e!Ensembl** BLAST/BLAT | BioMart | Tools | Downloads | Help & Documentation | Blog | Mirrors

Search: All species  for   
e.g. BRCA2 or rat 5:62797383-63627669 or rs699 or coronary heart disease

### Browse a Genome

Ensembl is a genome browser for vertebrate genomes that supports research in comparative genomics, evolution, sequence variation and transcriptional regulation. Ensembl annotates genes, computes multiple alignments, predicts regulatory function and collects disease data. Ensembl tools include BLAST, BLAT, BioMart and the Variant Effect Predictor (VEP) for all supported species.

### Favourite genomes

 **Human**  
GRCh38.p10

 **Mouse**  
GRCm38.p5

 **Zebrafish**  
GRCz10

[Edit favourites](#)

### All genomes

-- Select a species --

[View full list of all Ensembl species](#)

### Find a Data Display

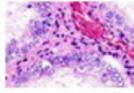
Not sure how to find the data visualisation you need? With our new [Find a Data Display](#) page, you can choose a gene, region or variant and then browse a selection of relevant visualisations

[Try it now!](#)

### Variant Effect Predictor



### Gene expression in different tissues



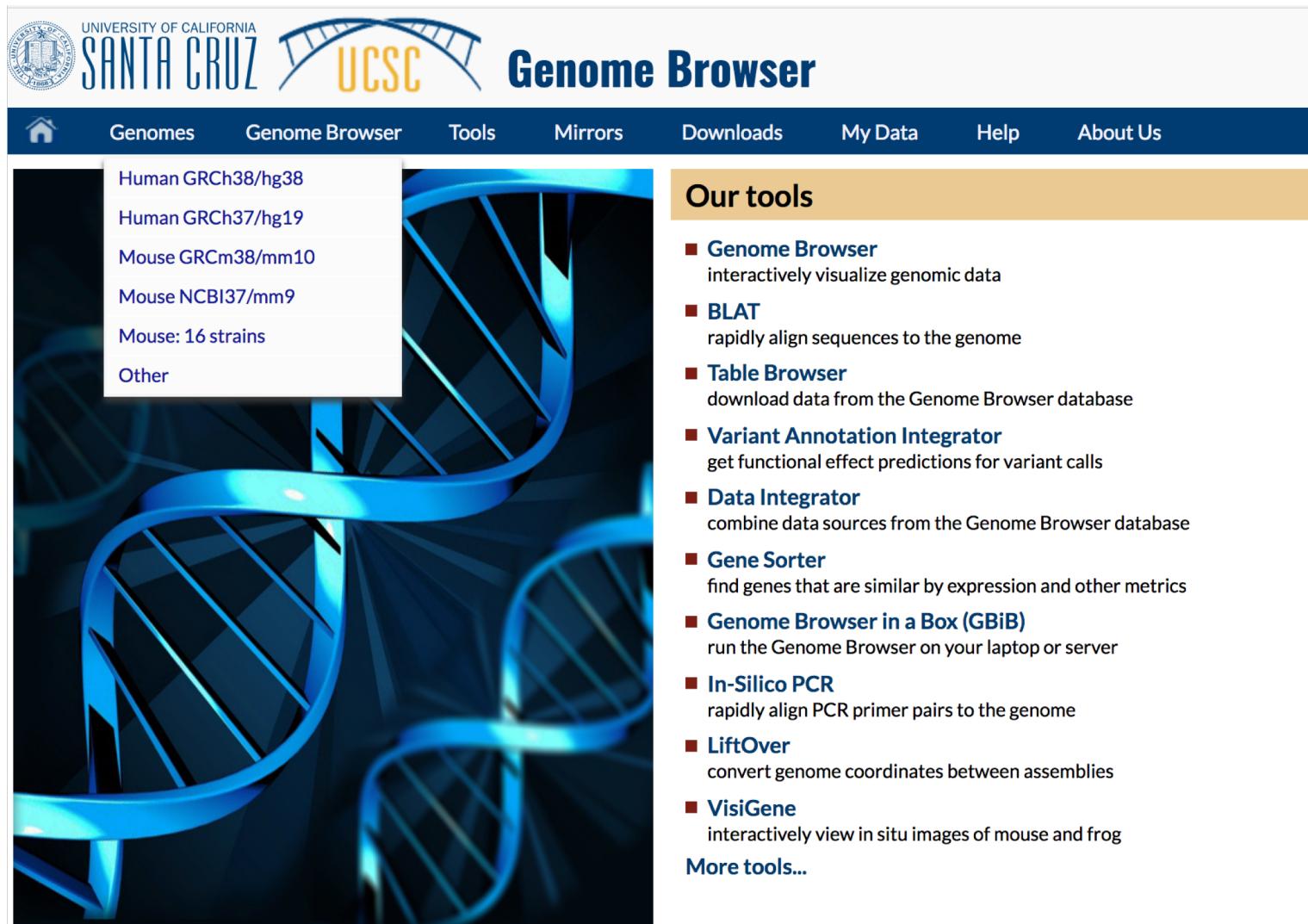
### Retrieve gene sequence

```
GCCTGACTTCGGGTGG;  
GGGCTTGTGGCGCGAGC;  
GGGCTCTGCTGGGCT;  
AOGGGACAGATTTGTGA;  
CACCTCTGGAGCGGTT;  
CCCAGTCAGCGTGGCG;
```

### Compare genes across species



# Genome browsers - UCSC

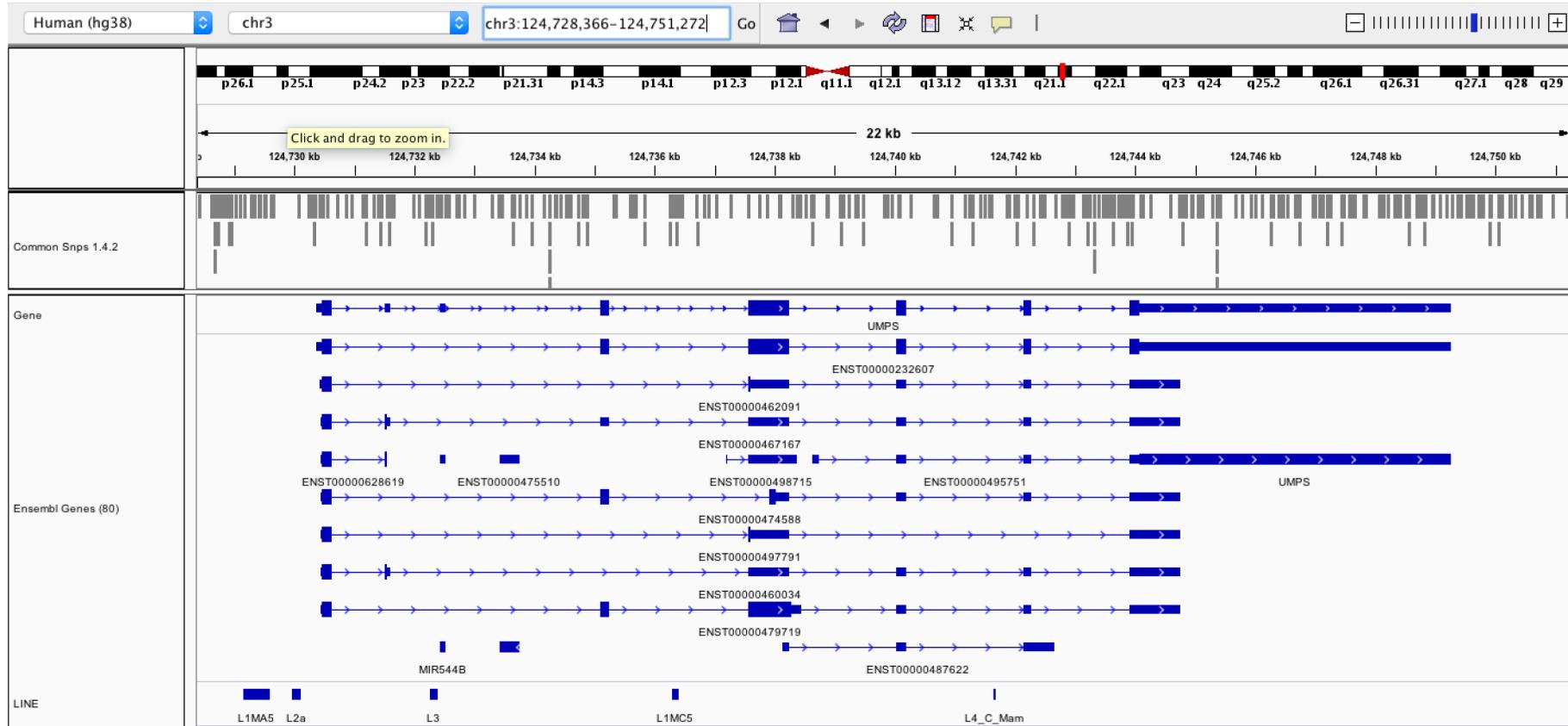


The image shows the homepage of the UCSC Genome Browser. At the top left is the University of California Santa Cruz logo. To its right is the text "UNIVERSITY OF CALIFORNIA SANTA CRUZ" above the "UCSC" logo, which features a stylized bridge arch. To the right of the logo is the title "Genome Browser". Below the header is a navigation bar with links: Home, Genomes, Genome Browser, Tools, Mirrors, Downloads, My Data, Help, and About Us. The "Genome Browser" link is highlighted with a blue background. On the left side, there is a sidebar with a list of genome assemblies: Human GRCh38/hg38, Human GRCh37/hg19, Mouse GRCm38/mm10, Mouse NCBI37/mm9, Mouse: 16 strains, and Other. The main content area features a large blue DNA helix graphic. To the right of the sidebar is a section titled "Our tools" with a yellow background. This section lists ten tools with descriptions:

- **Genome Browser**  
interactively visualize genomic data
- **BLAT**  
rapidly align sequences to the genome
- **Table Browser**  
download data from the Genome Browser database
- **Variant Annotation Integrator**  
get functional effect predictions for variant calls
- **Data Integrator**  
combine data sources from the Genome Browser database
- **Gene Sorter**  
find genes that are similar by expression and other metrics
- **Genome Browser in a Box (GBiB)**  
run the Genome Browser on your laptop or server
- **In-Silico PCR**  
rapidly align PCR primer pairs to the genome
- **LiftOver**  
convert genome coordinates between assemblies
- **VisiGene**  
interactively view *in situ* images of mouse and frog

[More tools...](#)

# Genome browsers - IGV



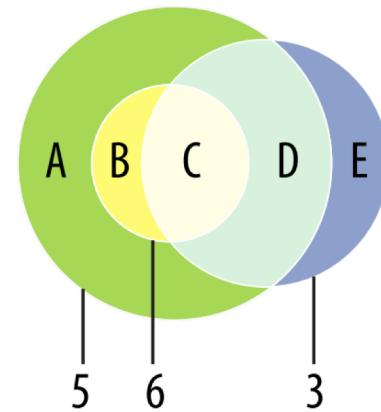
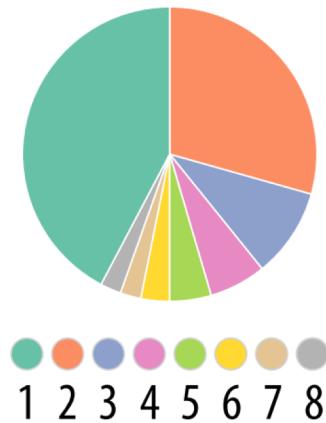
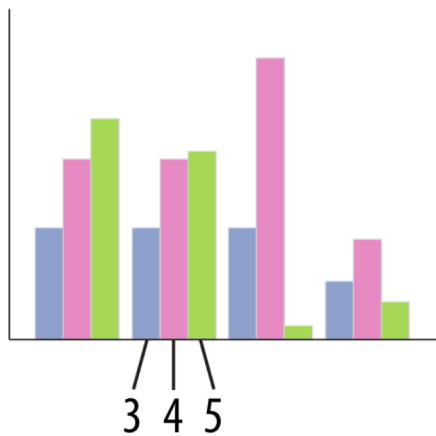
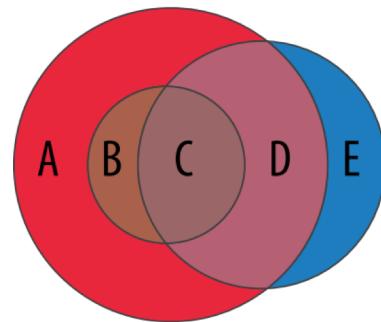
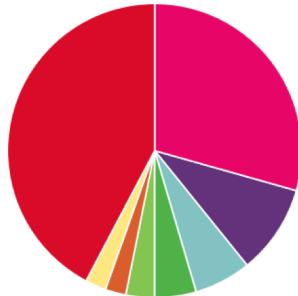
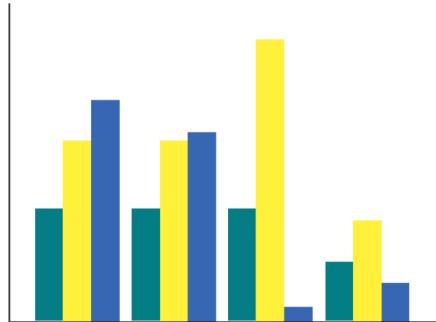
# Fundamentals of data visualization

# Fundamentals of data visualization

---

- Where to learn more about the art and science of visualization:
  - Collection of 40 Nature Methods articles on data visualization:  
[“Points of View” Articles](#)
  - [Visual design principles lecture](#)

# Which series is more effective (top or bottom)?

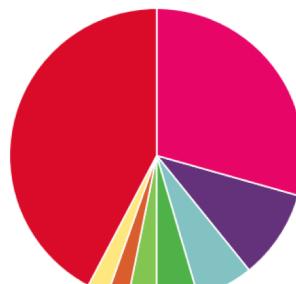


# Which series is more effective (top or bottom)?

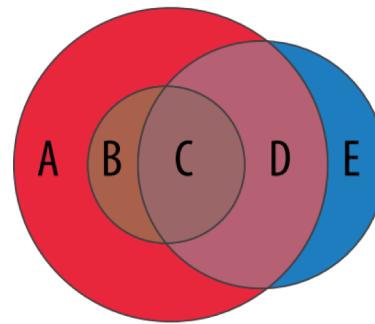
*one color dominates*



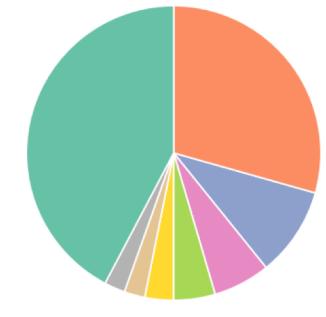
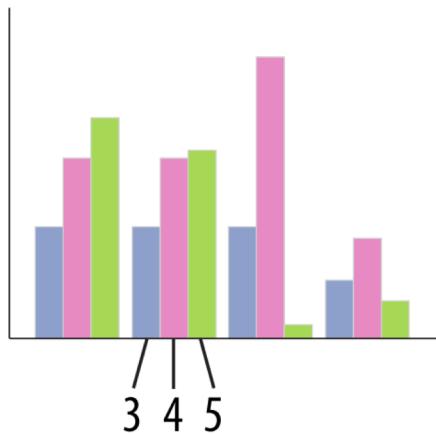
*difficult to distinguish*



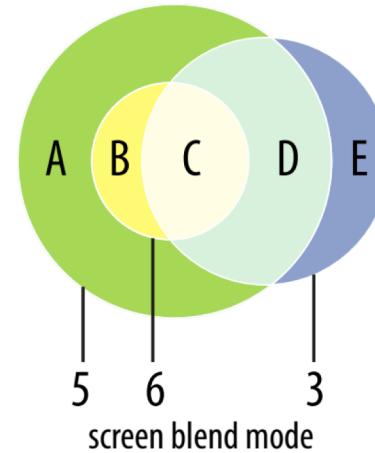
*murky*



*recolored with Brewer palettes*

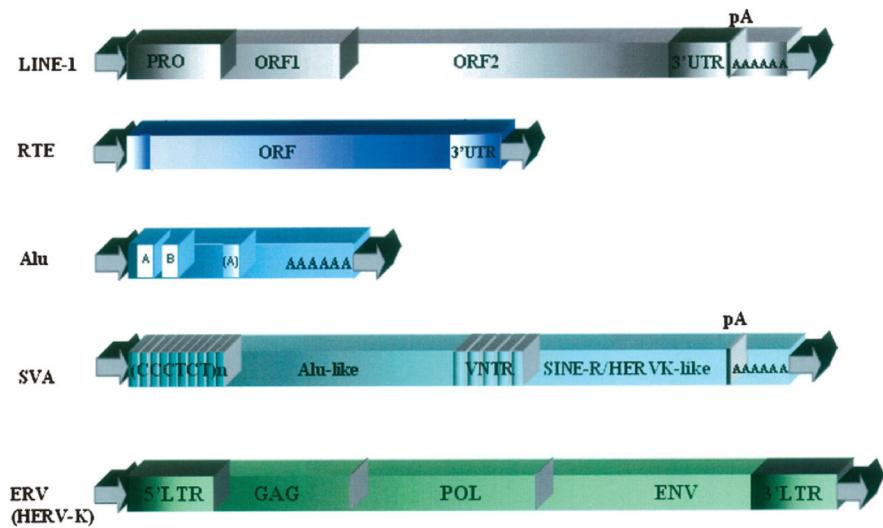
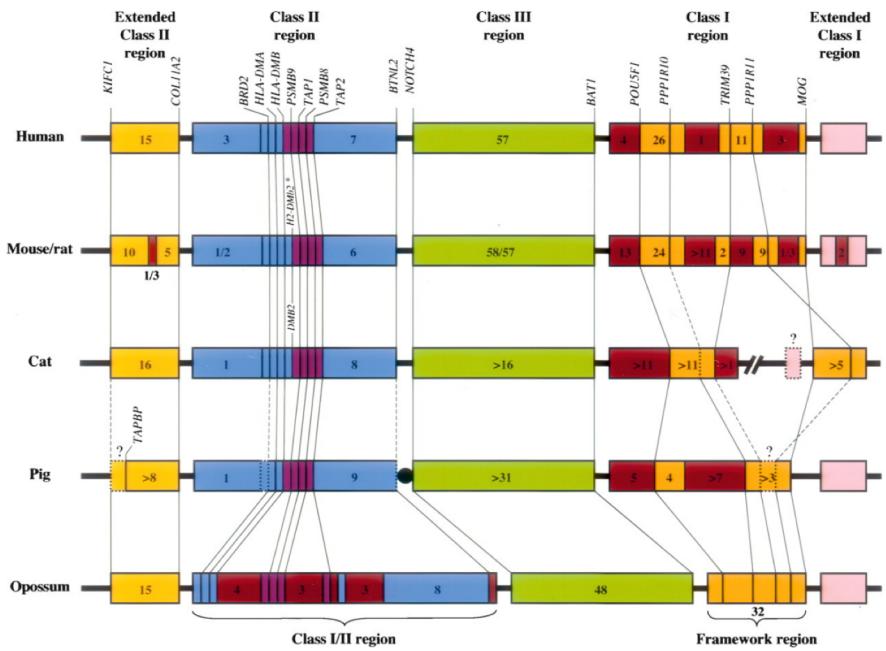


1 2 3 4 5 6 7 8  
set2

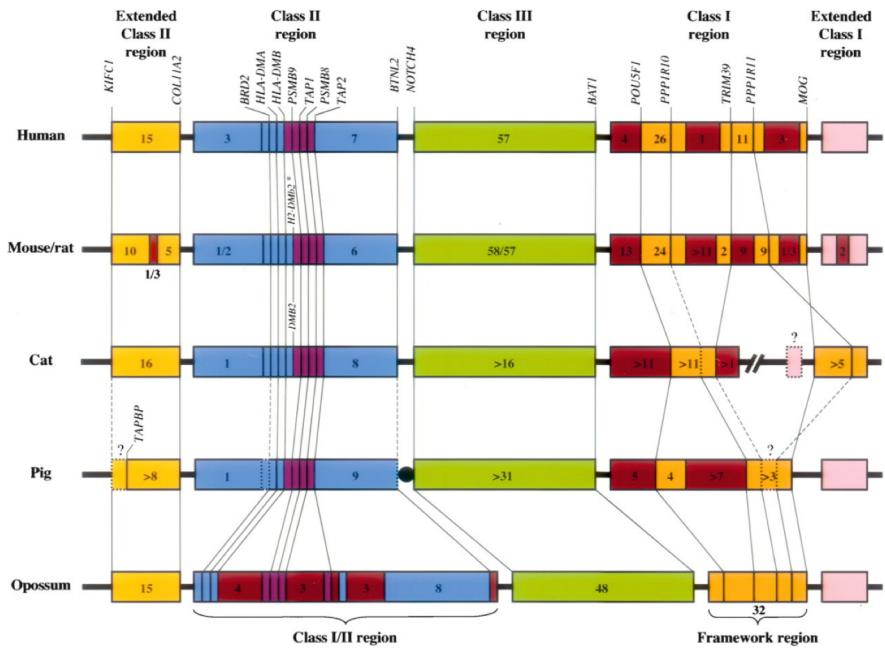


5 6 3  
screen blend mode

# Which is more effective (left or right?)

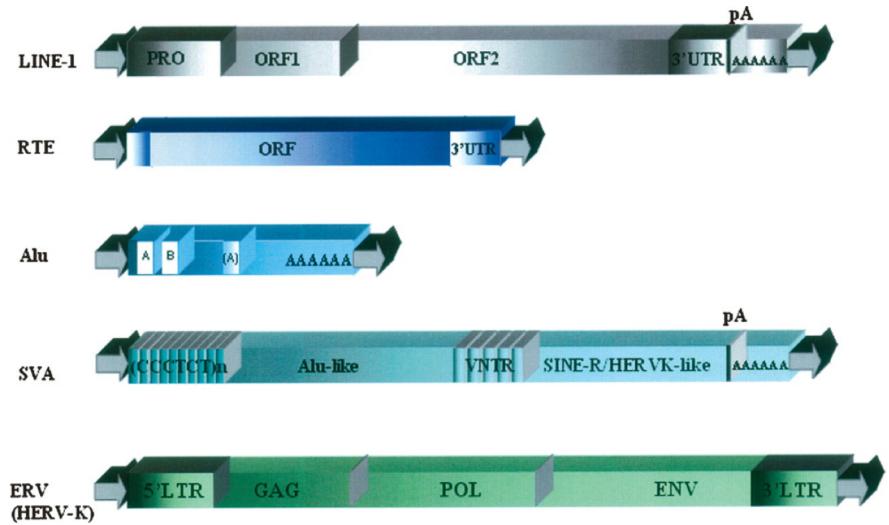


# Which is more effective (left or right?)



Excellent organization and consistency. Vertical lines cue continuity. Good use of color.

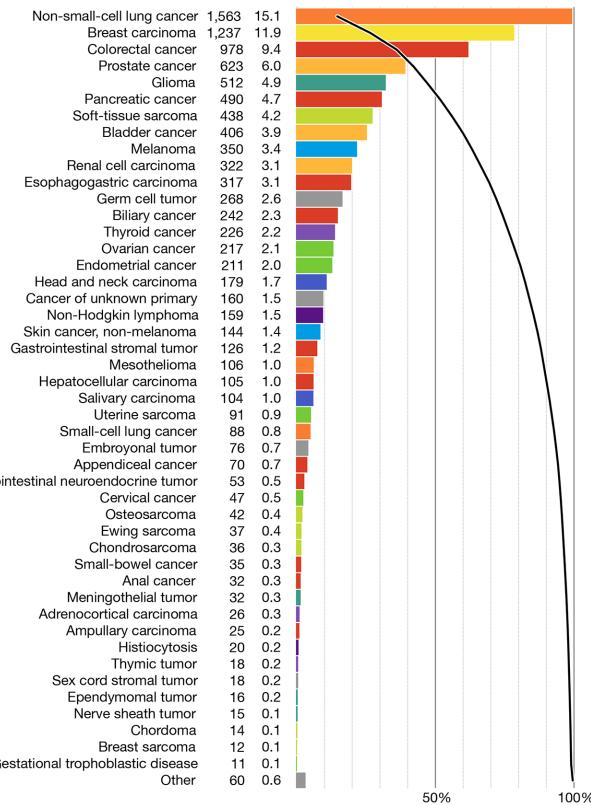
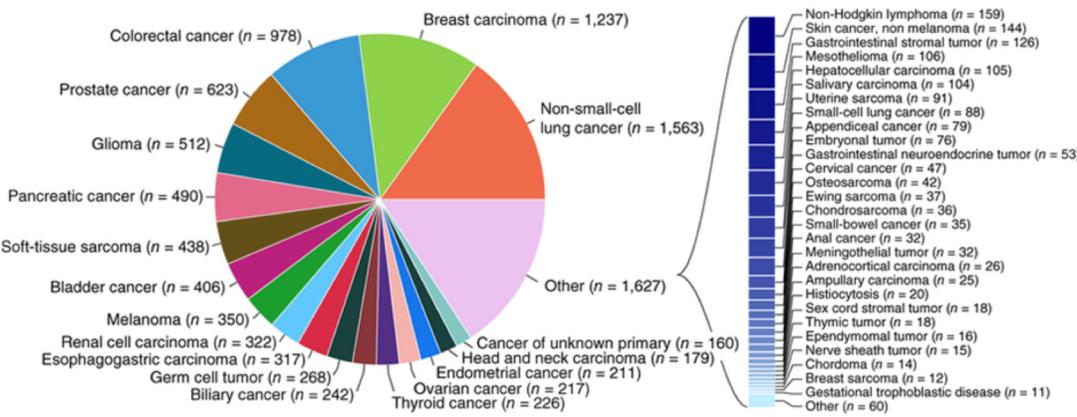
Samollow, P.B., The opossum genome: insights and opportunities from an alternative mammal. *Genome Res.*, 2008. 18(8): p. 1199-215.



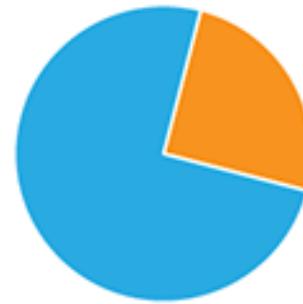
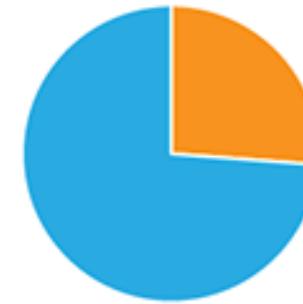
Chartjunk plentiful. Screaming ornamental and redundant elements. Text inconsistent and illegible.

Gentles, A.J., et al., Evolutionary dynamics of transposable elements in the short-tailed opossum *Monodelphis domestica*. *Genome Res.*, 2007. 17(7): p. 992-1004.

# Which is more effective (left or right?)



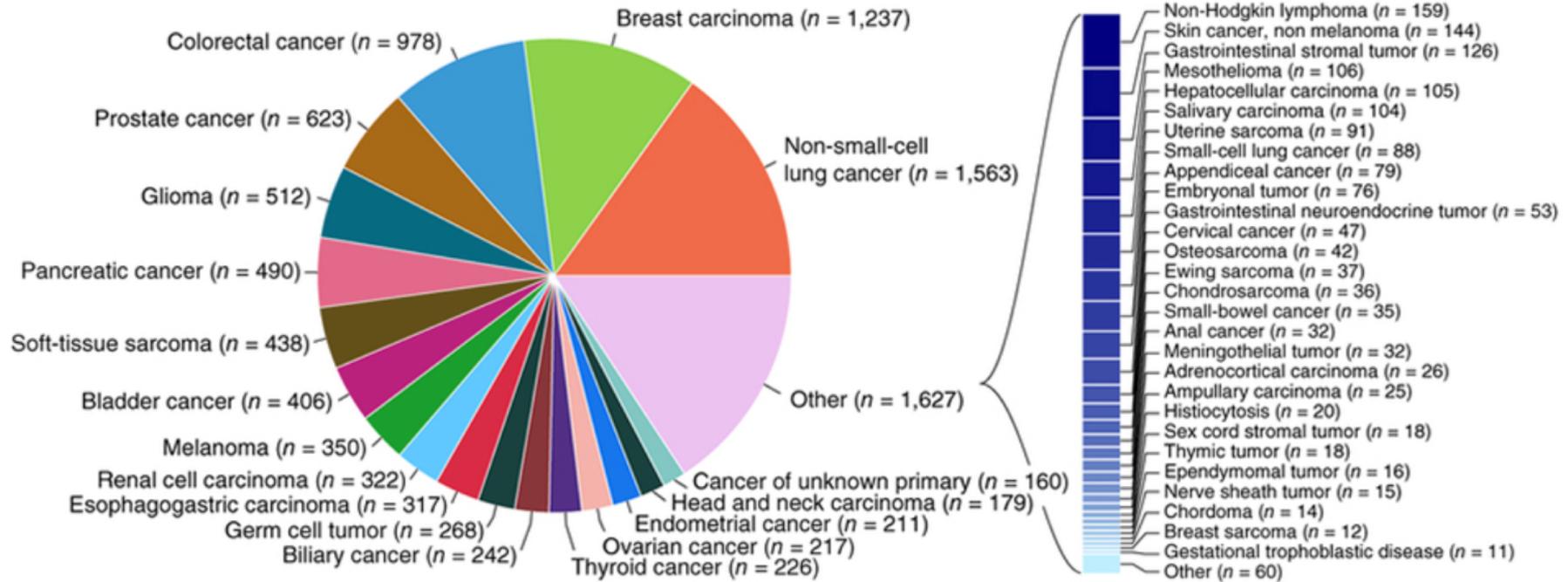
# When to use a pie chart



Pie charts are good at precisely showing 1:3 proportions

...but not if the slice is rotated

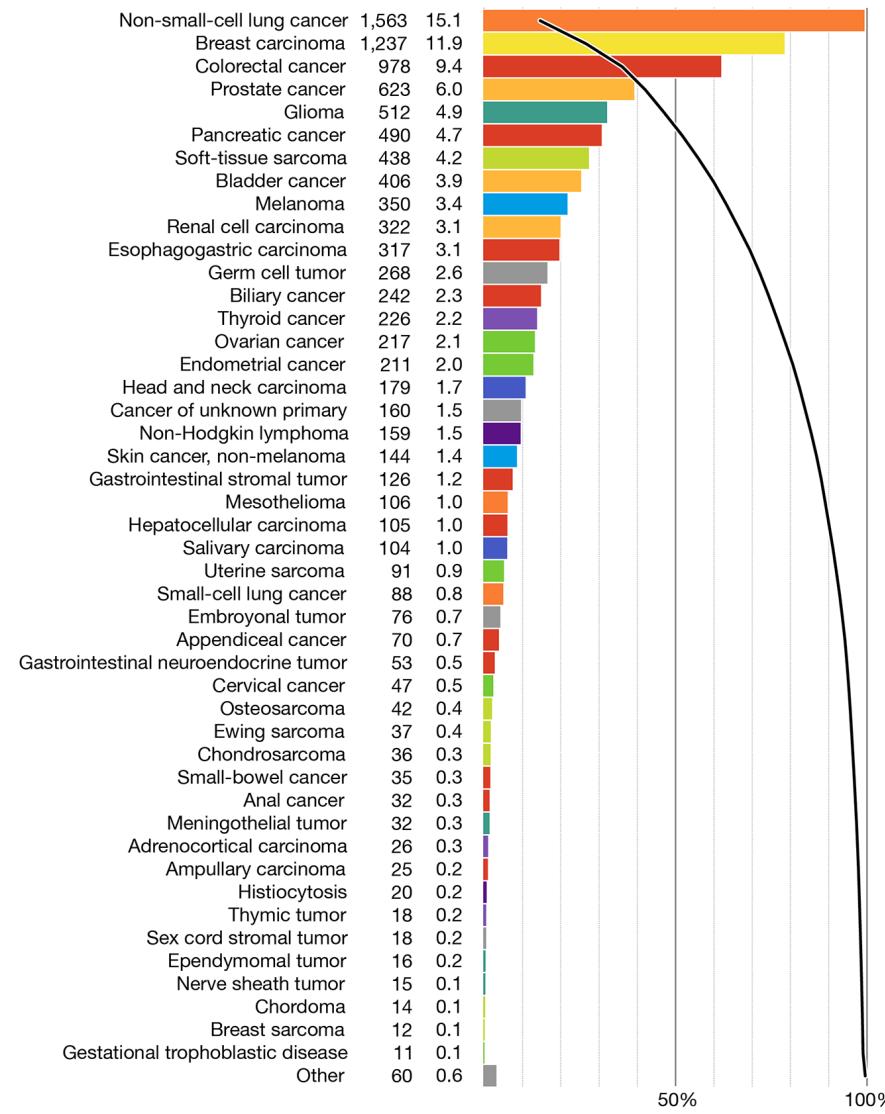
# When to not use a pie chart



- Hard to judge proportions
- Poor use of color
- Hard to read labels
- Over  $\frac{1}{2}$  of the categories had to be broken out of the pie chart

Ahmet Zehir et al. (2017) Nature Medicine doi:10.1038/nm.4333

# Same data with a redesigned approach



# Selected articles on fundamentals of data viz

- Visualizing samples with box plots
- Circos plots
- When to use (and not use) pie charts
- Resources for choosing colors
  - <http://colorbrewer2.org/>
  - <http://mkweb.bcgsc.ca/color/>
  - Understanding and using Color Palettes
  - Color palettes for color blindness
  - Names for >9000 colors
    - Including 40 beer colors
- Credit to Martin Krzywinski for his extensive work in this area and many of the above resources

# Best practices in visualization

# Best practices from this workshop

---

- \*Always\* label axes
- Consider readability of font size
- Avoid vertical or angled text if possible
- Avoid unnecessary use of color, point shapes, etc.
- Choose colors wisely
- If individual data points are being plotted and have started to really pile up on top of each other consider using a density function
- Always be transparent about what data manipulation is taking place (e.g. log scale, filtering of outliers, etc.)

# Best practices from the experts

---

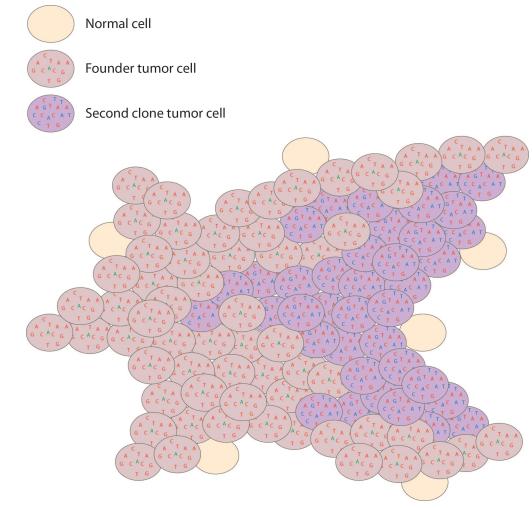
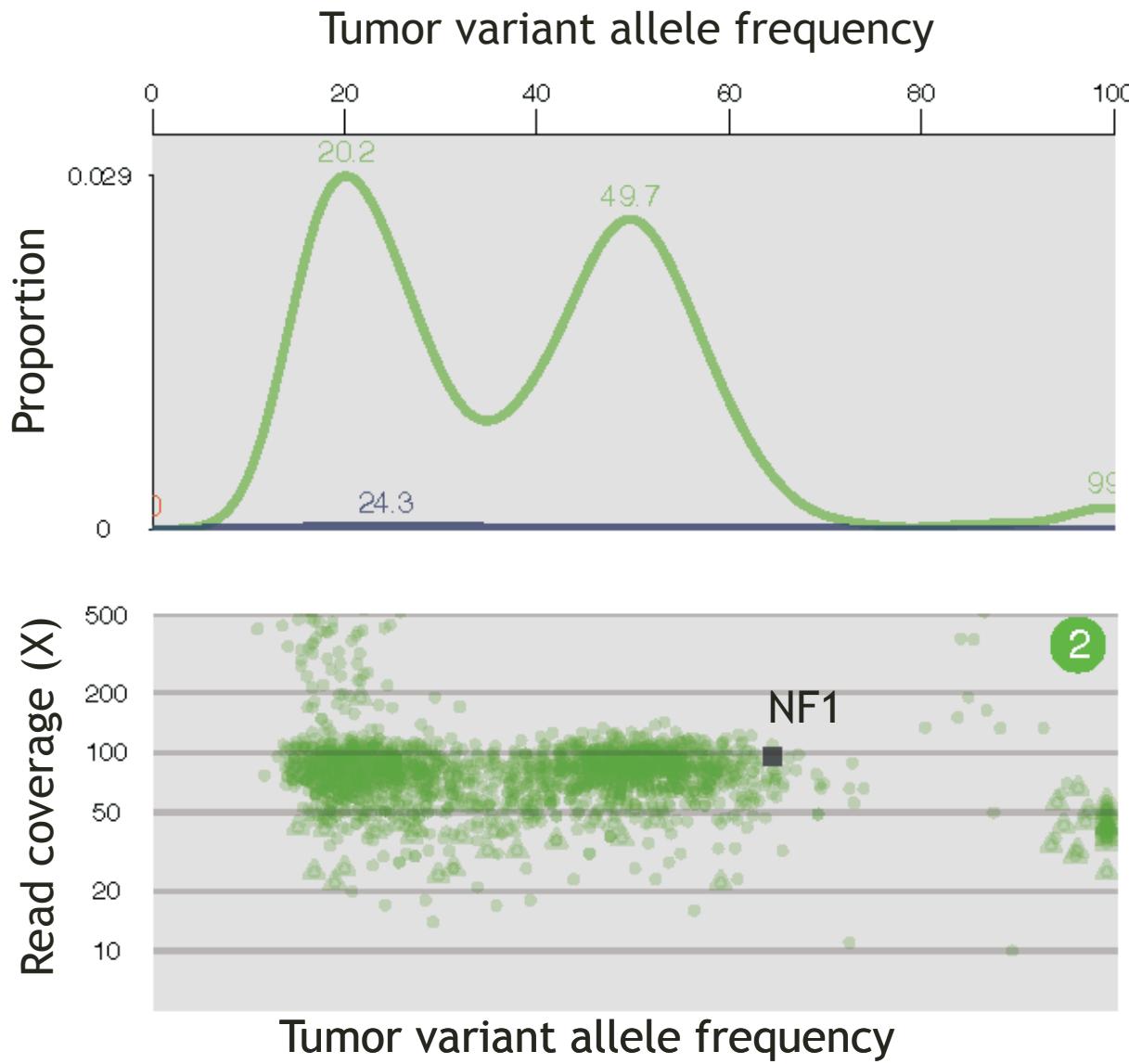
## Ten Simple Rules for Better Figures (Rougier et al. 2014):

Scientific visualizations should act as a “a graphical interface between people and data”. Try to follow these rules.

1. Know Your Audience
2. Identify Your Message
3. Adapt the Figure to the Support Medium
4. Captions Are Not Optional
5. Do Not Trust the Defaults
6. Use Color Effectively
7. Do Not Mislead the Reader
8. Avoid “Chartjunk”
9. Message Trumps Beauty
10. Get the Right Tool

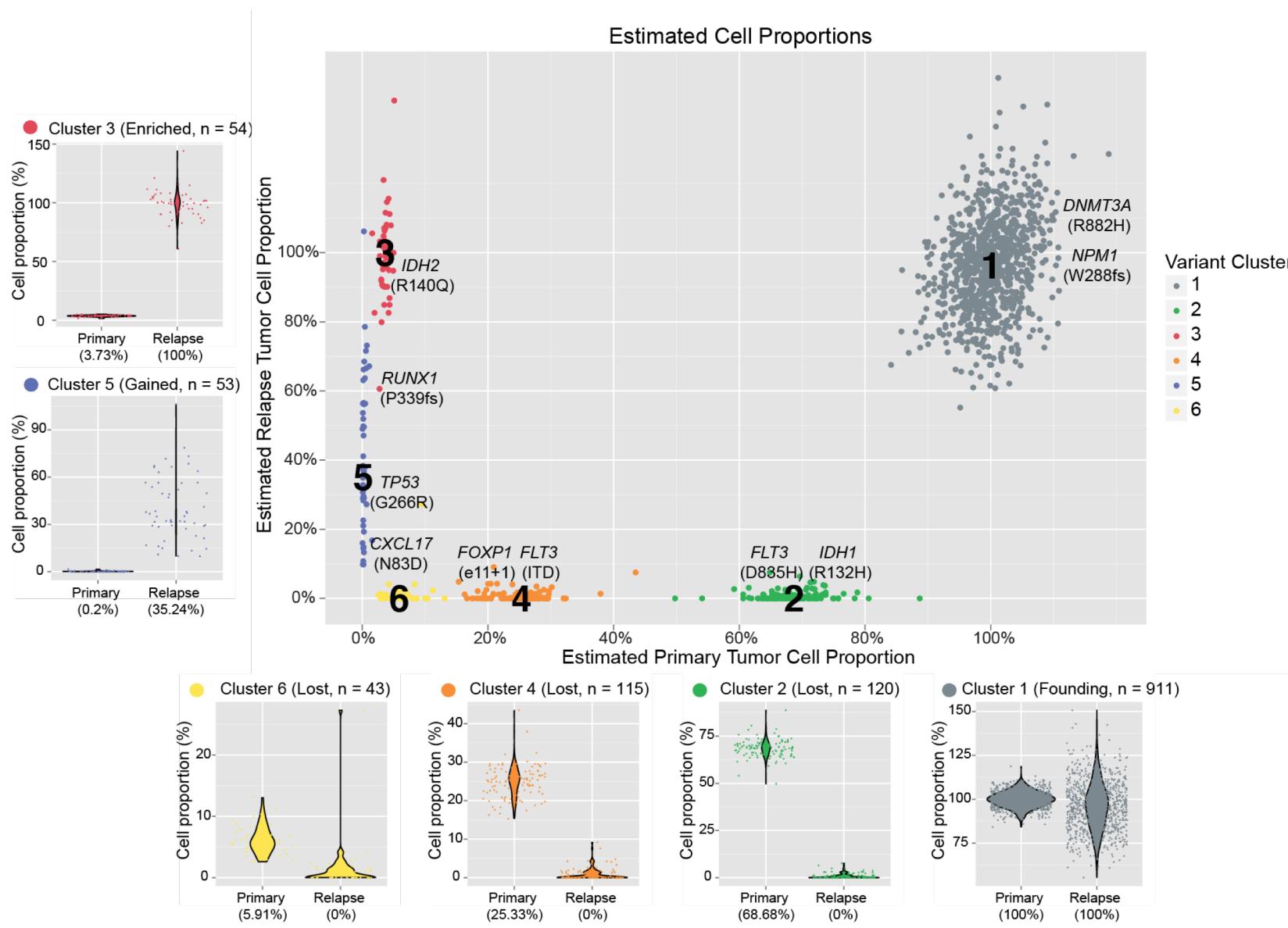
## Example visualizations using R

# Histogram and scatterplot to define cell populations

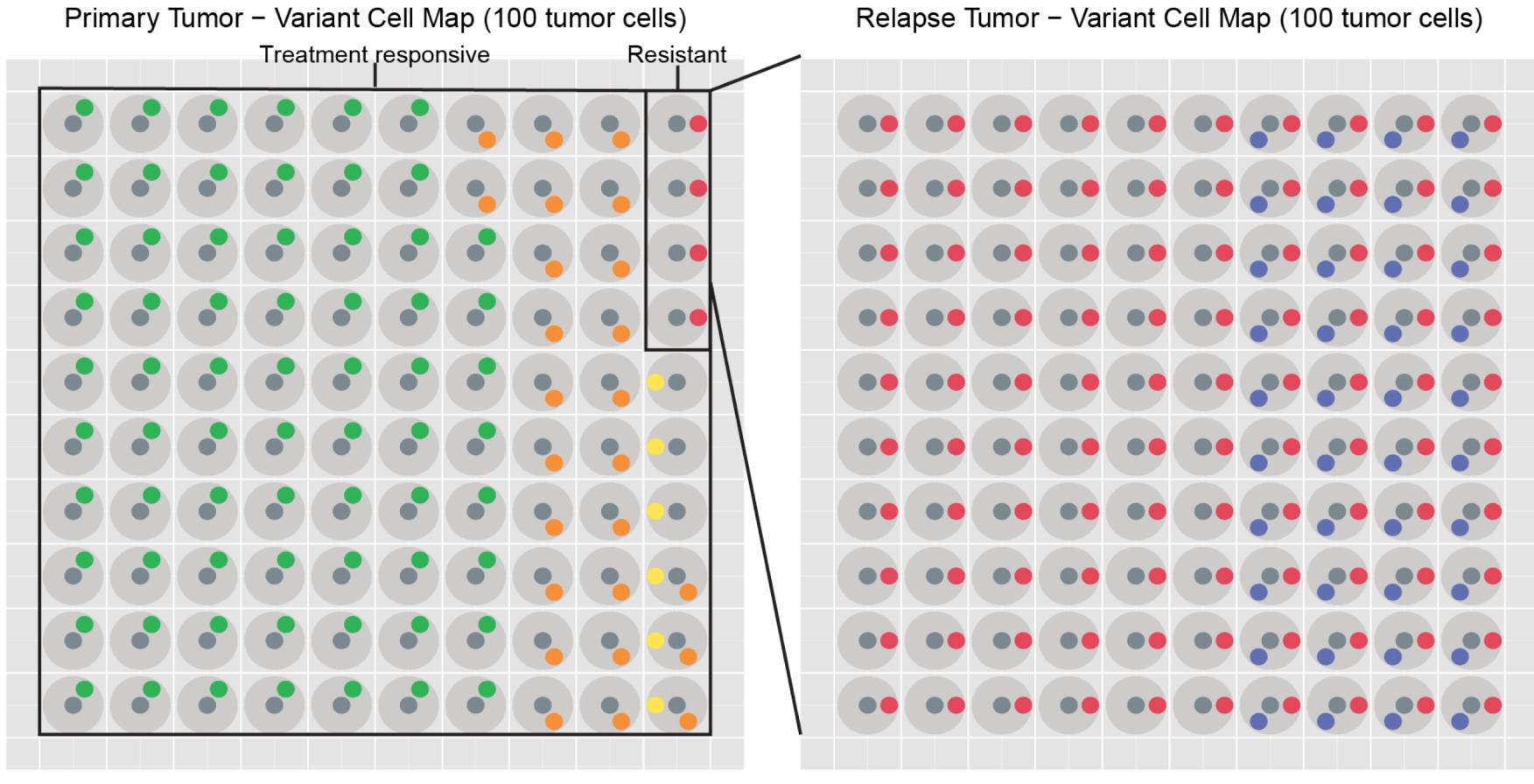


# WGS data

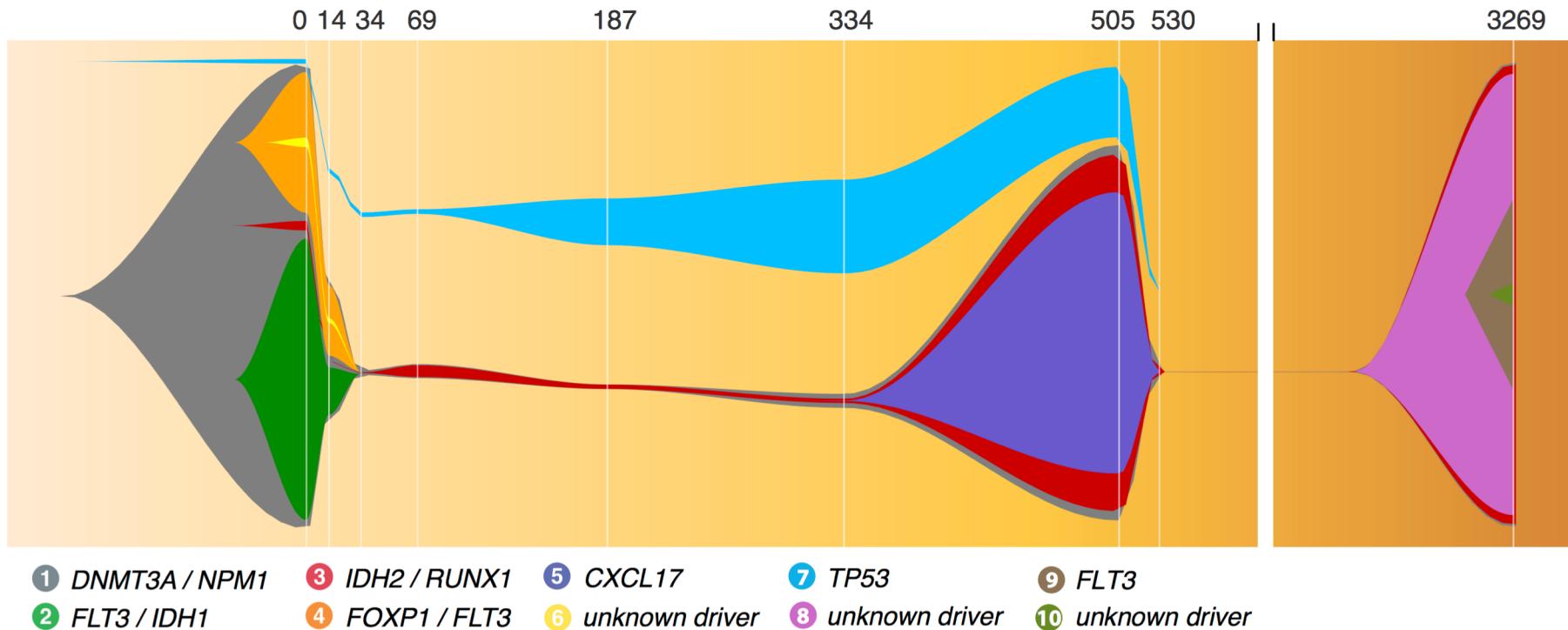
# Clustering to define two distinct cell populations



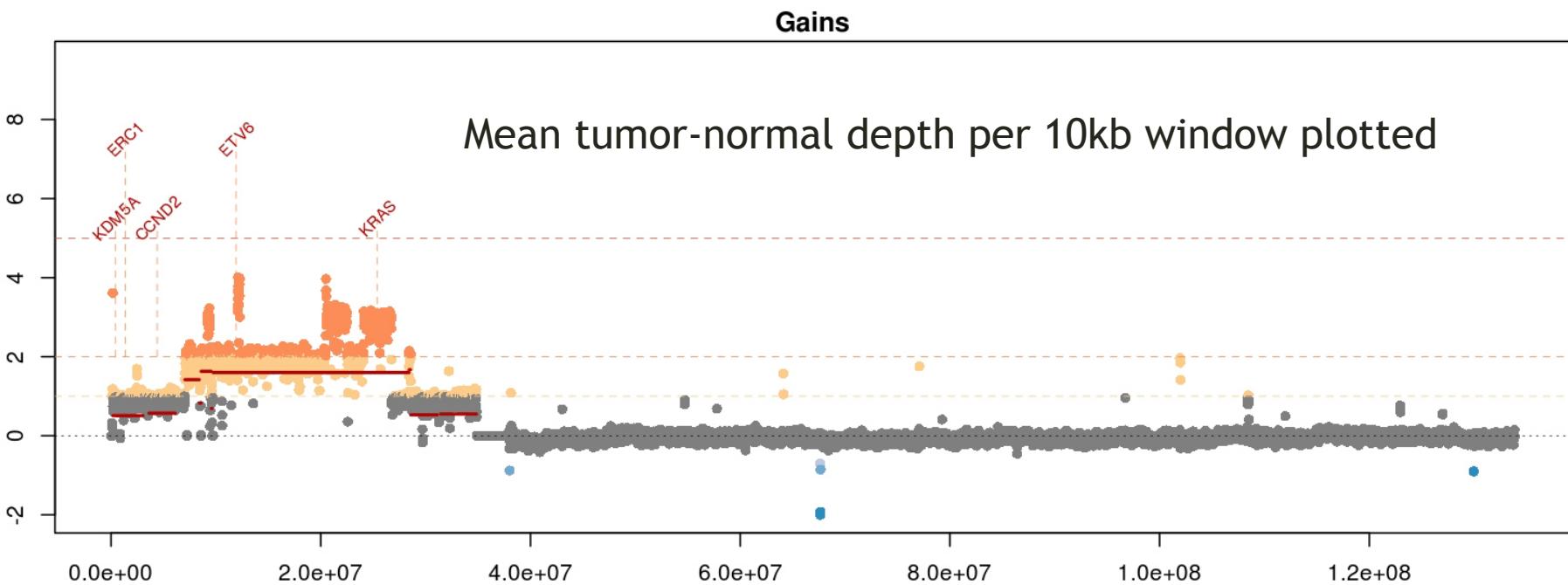
# Using a “cell map” to represent the variant clusters in 100 hypothetical cells of a tissue



# A ‘fish’ plot is used to represent sub-clones lost and gained over time

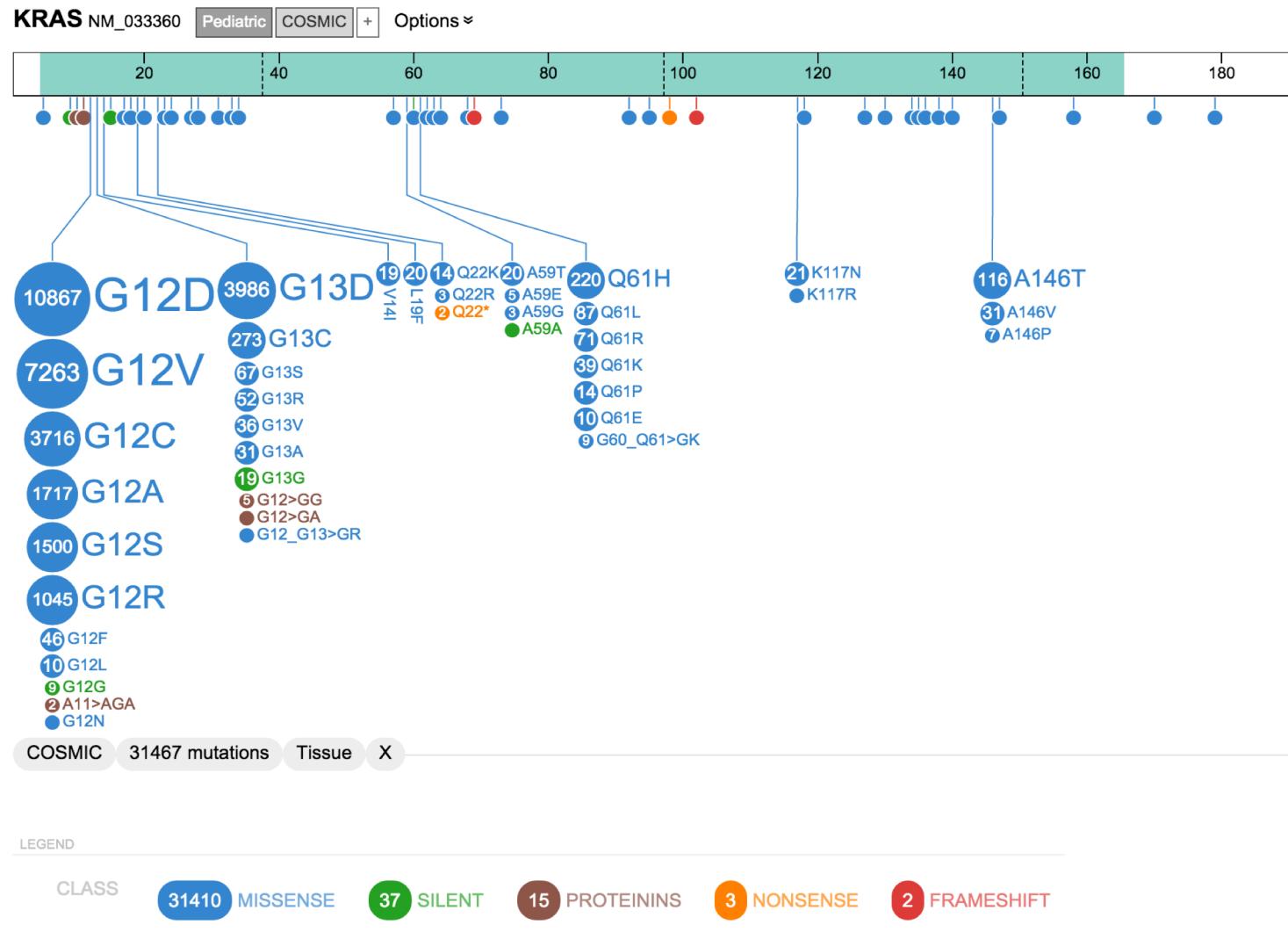


# Visualizing copy number variation



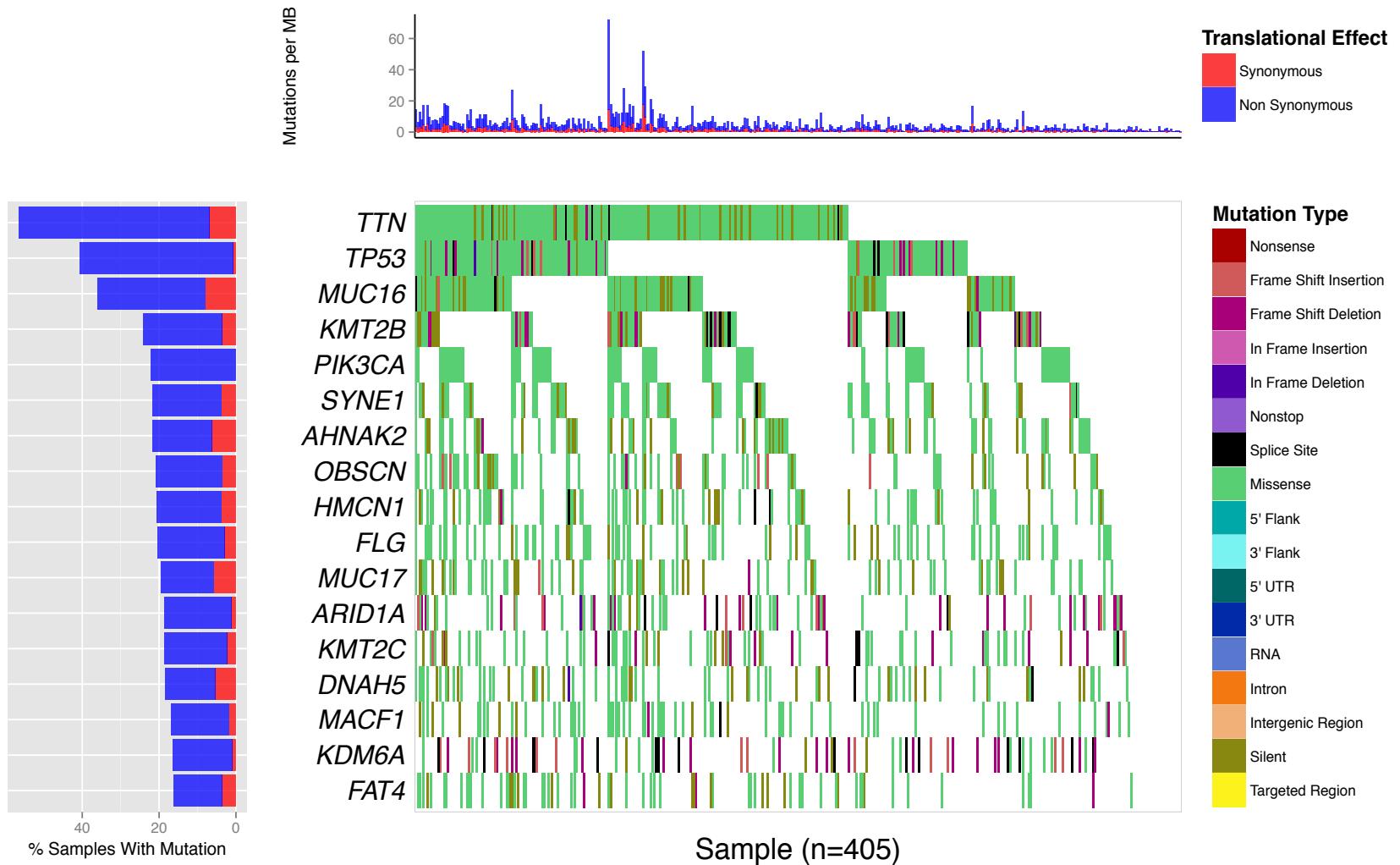
KRAS amplification in a metastatic breast cancer

# A ‘lollipop’ plot is used to variant recurrence in a gene



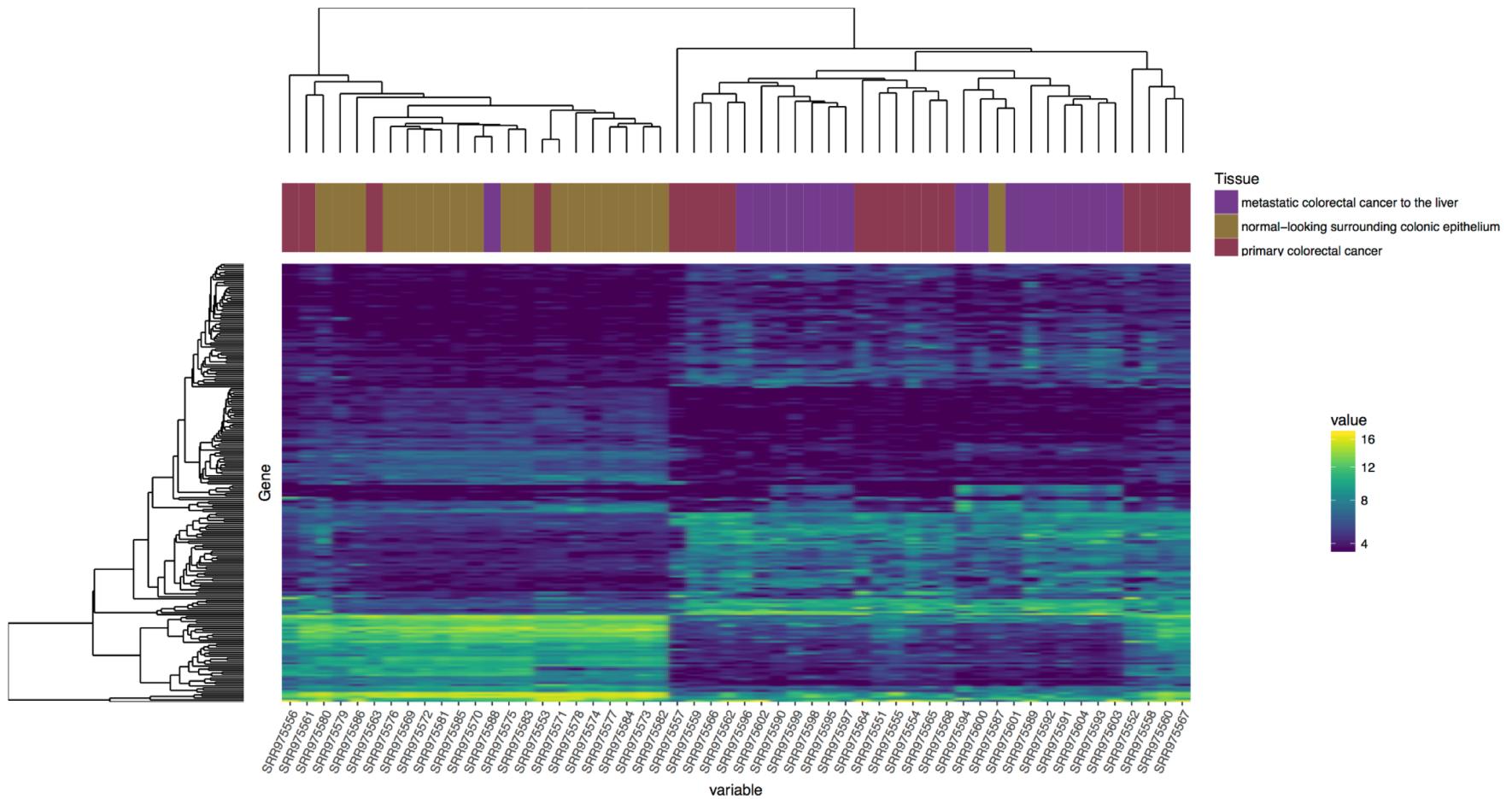
\* Actually this one used “ProteinPaint” a web based tool we will try

# A ‘waterfall’ plot is one way to visualize the pattern of recurrence in a cohort



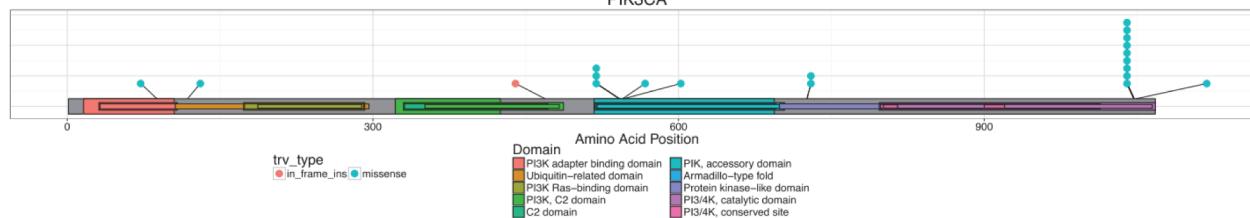
<https://github.com/griffithlab/GenVisR>

Heatmaps are a common way to simultaneously visualize multiple features of a dataset

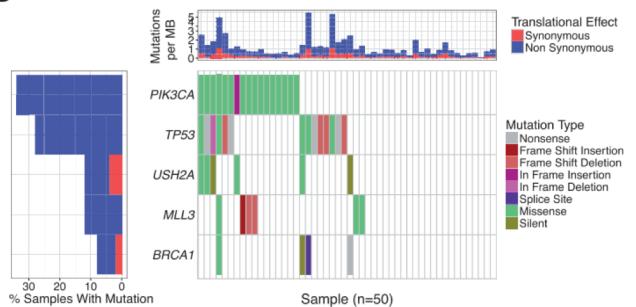


# GenVisR was created to help others make common genomic visualizations

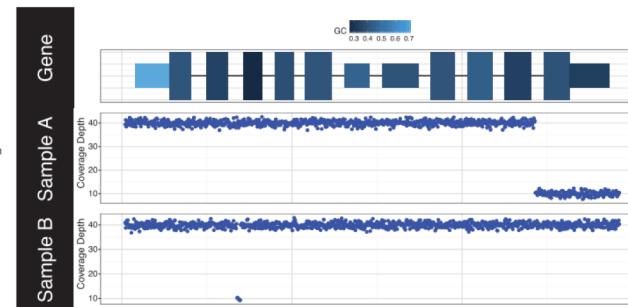
A



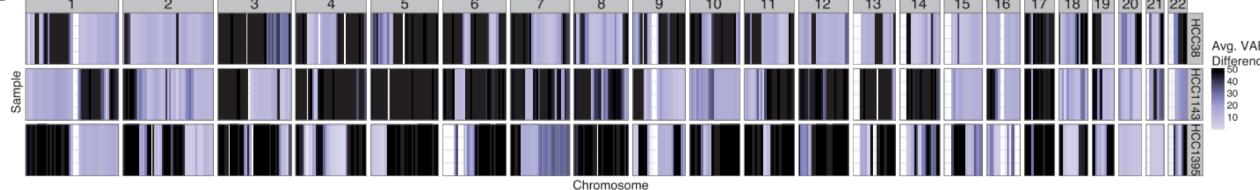
B



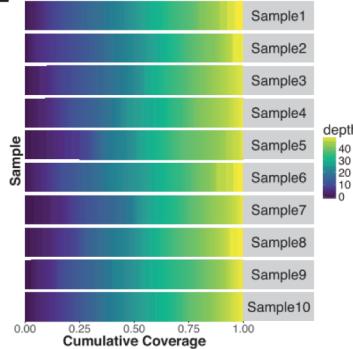
C



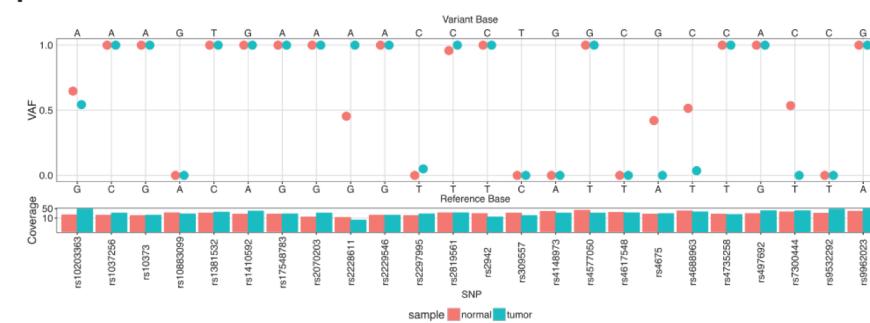
D



E



F



# Introduction to GenViz course site

---

[www.genviz.org](http://www.genviz.org)

# Tutorial for module 1

---

[www.genviz.org/course/#module-1](http://www.genviz.org/course/#module-1)