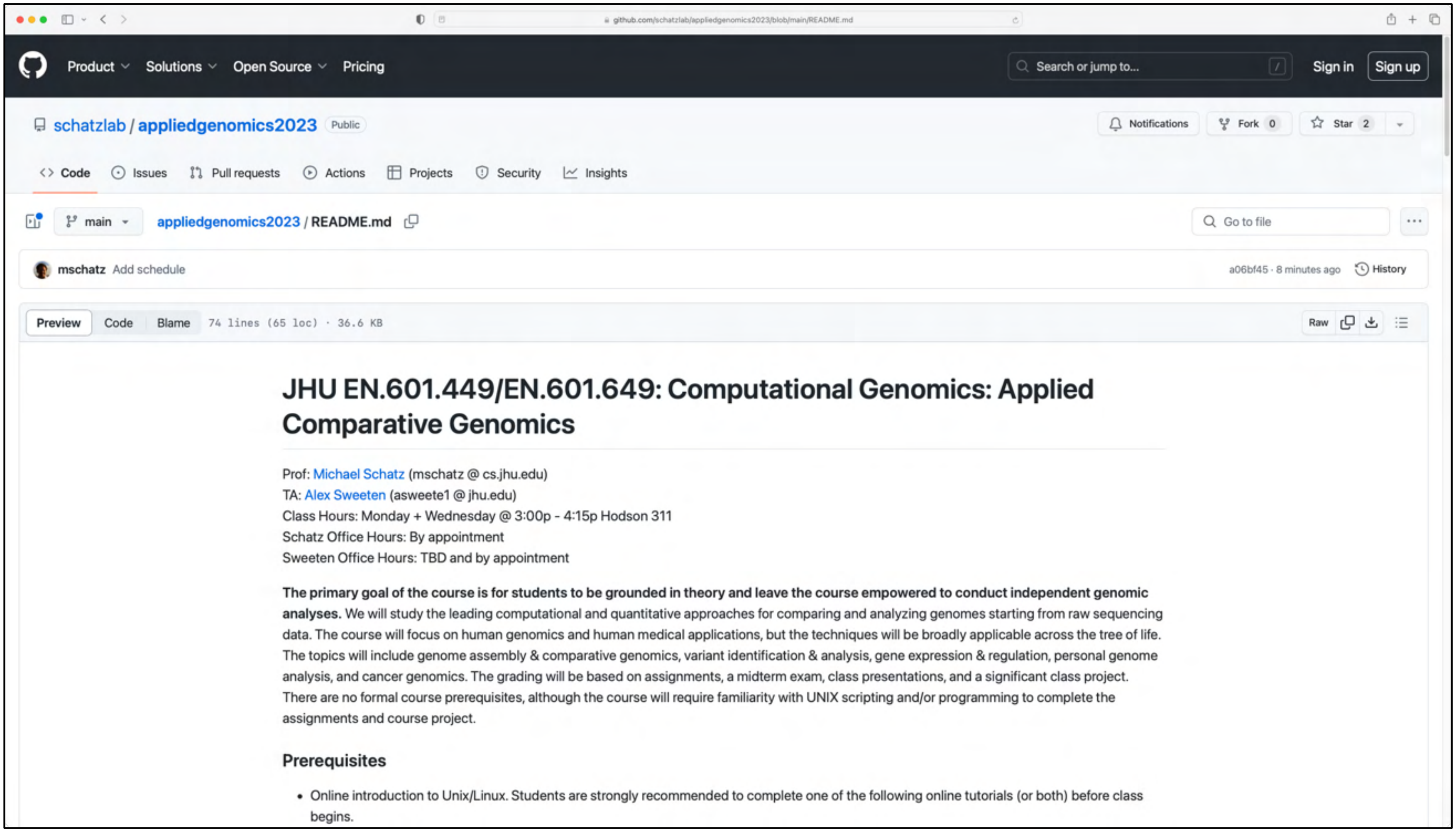
Genomic Technologies

Michael Schatz

August 30, 2022

Lecture 2: Applied Comparative Genomics

Course Webpage



JHU EN.601.449/EN.601.649: Computational Genomics: Applied Comparative Genomics

Prof: Michael Schatz (mschatz @ cs.jhu.edu)
TA: Alex Sweeten (asweete1 @ jhu.edu)
Class Hours: Monday + Wednesday @ 3:00p - 4:15p Hodson 311
Schatz Office Hours: By appointment
Sweeten Office Hours: TBD and by appointment

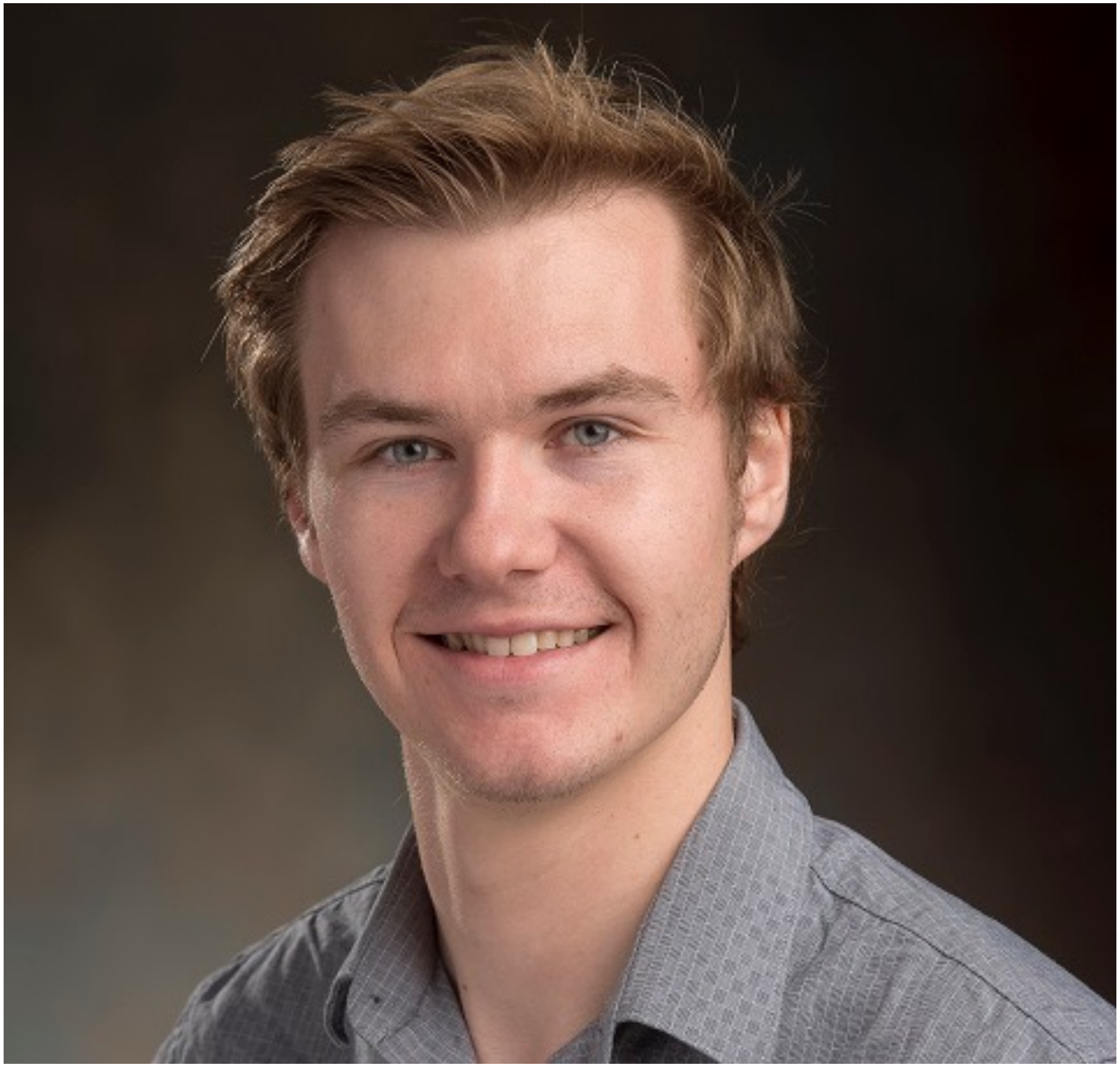
The primary goal of the course is for students to be grounded in theory and leave the course empowered to conduct independent genomic analyses. We will study the leading computational and quantitative approaches for comparing and analyzing genomes starting from raw sequencing data. The course will focus on human genomics and human medical applications, but the techniques will be broadly applicable across the tree of life. The topics will include genome assembly & comparative genomics, variant identification & analysis, gene expression & regulation, personal genome analysis, and cancer genomics. The grading will be based on assignments, a midterm exam, class presentations, and a significant class project. There are no formal course prerequisites, although the course will require familiarity with UNIX scripting and/or programming to complete the assignments and course project.

Prerequisites

- Online introduction to Unix/Linux. Students are strongly recommended to complete one of the following online tutorials (or both) before class begins.

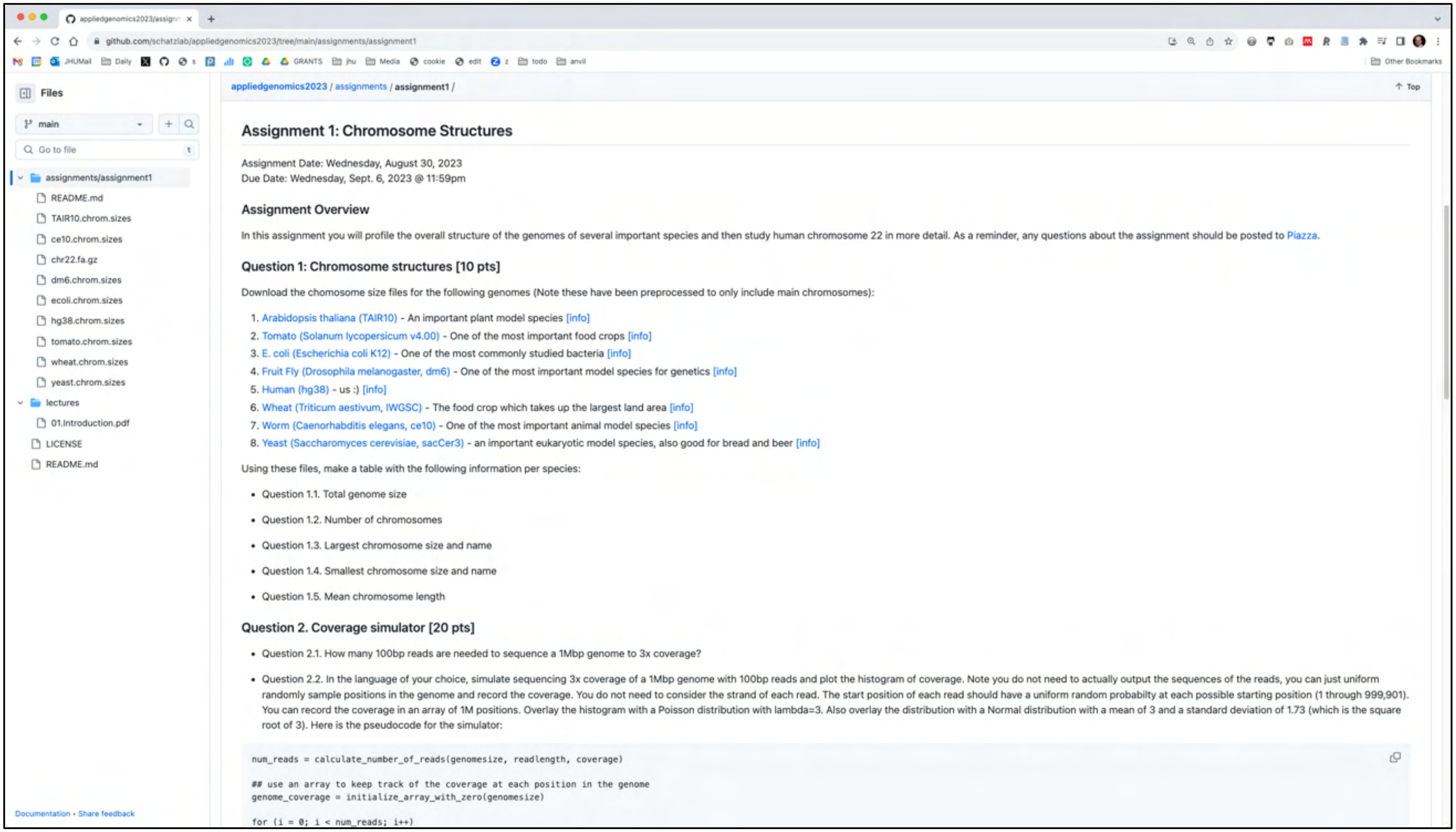
https://github.com/schatzlab/appliedgenomics2023

TA: Alex Sweeten



Check Piazza for Office Hours Poll

Assignment 1



Assignment 1: Chromosome Structures

Assignment Date: Wednesday, August 30, 2023
Due Date: Wednesday, Sept. 6, 2023 @ 11:59pm

Assignment Overview

In this assignment you will profile the overall structure of the genomes of several important species and then study human chromosome 22 in more detail. As a reminder, any questions about the assignment should be posted to Piazza.

Question 1: Chromosome structures [10 pts]

Download the chomosome size files for the following genomes (Note these have been preprocessed to only include main chromosomes):

1. Arabidopsis thaliana (TAIR10) - An important plant model species [info]
2. Tomato (Solanum lycopersicum v4.00) - One of the most important food crops [info]
3. E. coli (Escherichia coli K12) - One of the most commonly studied bacteria [info]
4. Fruit Fly (Drosophila melanogaster, dm6) - One of the most important model species for genetics [info]
5. Human (hg38) - us :) [info]
6. Wheat (Triticum aestivum, IWGSC) - The food crop which takes up the largest land area [info]
7. Worm (Caenorhabditis elegans, ce10) - One of the most important animal model species [info]
8. Yeast (Saccharomyces cerevisiae, sacCer3) - an important eukaryotic model species, also good for bread and beer [info]

Using these files, make a table with the following information per species:

- Question 1.1. Total genome size
- Question 1.2. Number of chromosomes
- Question 1.3. Largest chromosome size and name
- Question 1.4. Smallest chromosome size and name
- Question 1.5. Mean chromosome length

Question 2. Coverage simulator [20 pts]

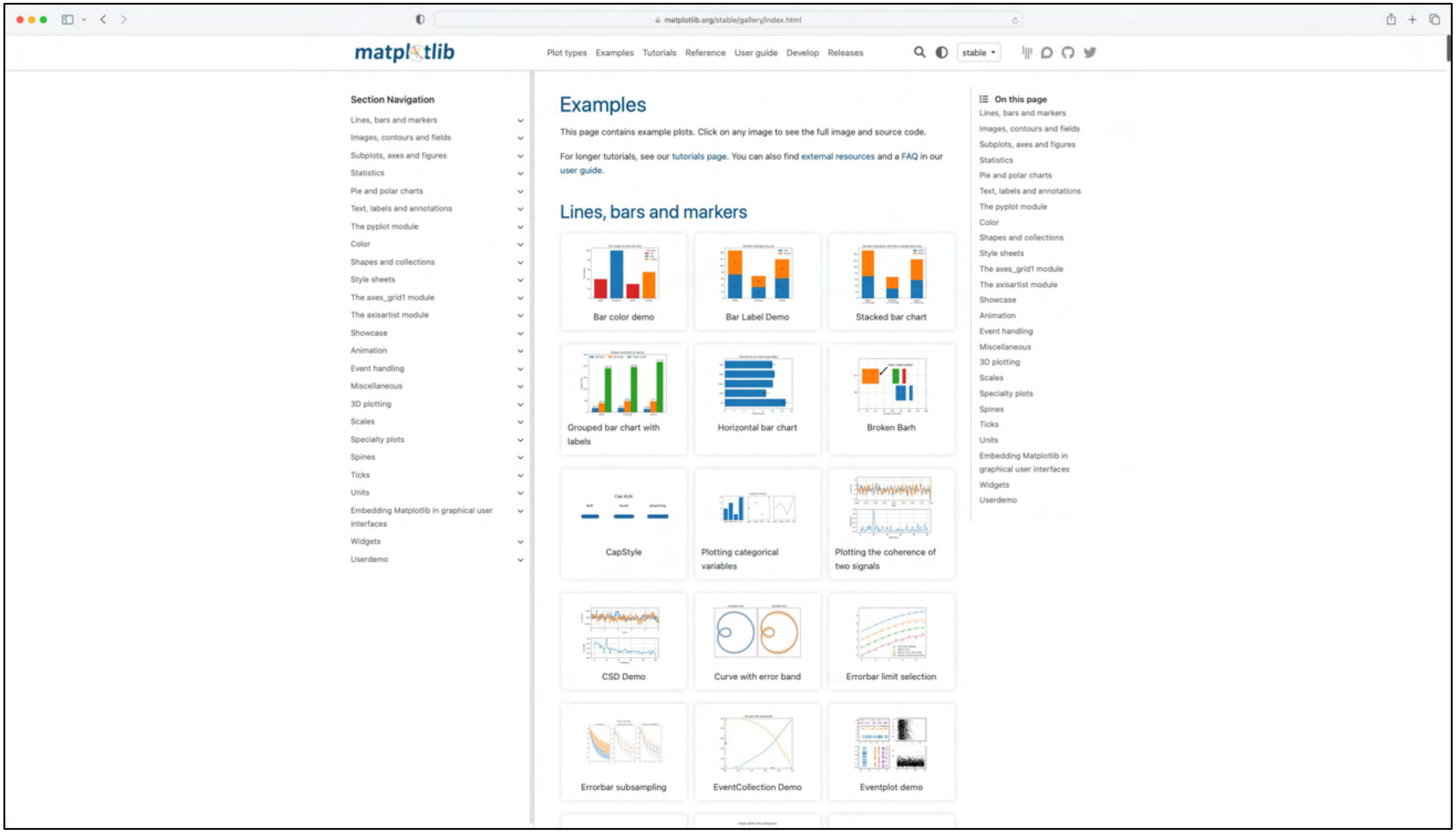
- Question 2.1. How many 100bp reads are needed to sequence a 1Mbp genome to 3x coverage?
- Question 2.2. In the language of your choice, simulate sequencing 3x coverage of a 1Mbp genome with 100bp reads and plot the histogram of coverage. Note you do not need to actually output the sequences of the reads, you can just uniform randomly sample positions in the genome and record the coverage. You do not need to consider the strand of each read. The start position of each read should have a uniform random probabilty at each possible starting position (1 through 999,901). You can record the coverage in an array of 1M positions. Overlay the histogram with a Poisson distribution with lambda=3. Also overlay the distribution with a Normal distribution with a mean of 3 and a standard deviation of 1.73 (which is the square root of 3). Here is the pseudocode for the simulator:

```
num_reads = calculate_number_of_reads(genomesize, readlength, coverage)

## use an array to keep track of the coverage at each position in the genome
genome_coverage = initialize_array_with_zero(genomesize)

for (i = 0; i < num_reads; i++)
```

https://github.com/schatzlab/appliedgenomics2023/tree/main/assignments/assignment1

Due end of day on Sept 6 (right before midnight)

Plotting in Python



Examples

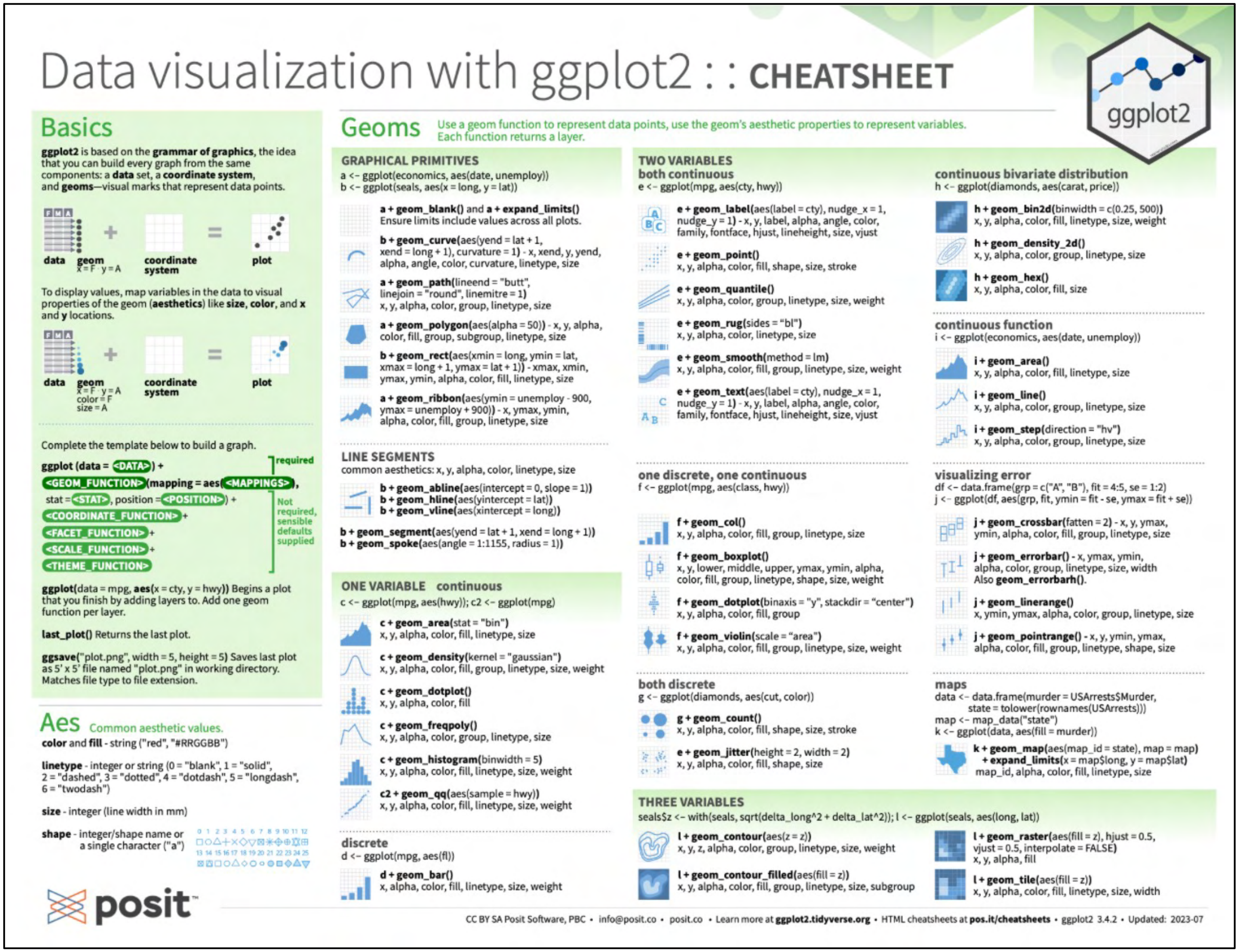
This page contains example plots. Click on any image to see the full image and source code.

For longer tutorials, see our tutorials page. You can also find external resources and a FAQ in our user guide.

Lines, bars and markers

https://matplotlib.org/

Plotting in R / ggplot2



Data visualization with ggplot2 : : CHEATSHEET

Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data** set, a **coordinate system**, and **geoms**—visual marks that represent data points.

F M A
data
geom
x = F · y = A
+
coordinate system
=
plot

To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.

Complete the template below to build a graph.

```
ggplot (data = <DATA>) +                                      ] required
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>),
 stat = <STAT>, position = <POSITION>) +                      ] Not
<COORDINATE_FUNCTION> +                                         required,
<FACET_FUNCTION> +                                              sensible
<SCALE_FUNCTION> +                                              defaults
<THEME_FUNCTION>                                                supplied
```

ggplot(data = mpg, **aes**(x = cty, y = hwy)) Begins a plot that you finish by adding layers to. Add one geom function per layer.

last_plot() Returns the last plot.

ggsave("plot.png", width = 5, height = 5) Saves last plot as 5' x 5' file named "plot.png" in working directory. Matches file type to file extension.

Aes Common aesthetic values.

color and **fill** - string ("red", "#RRGGBB")

linetype - integer or string (0 = "blank", 1 = "solid", 2 = "dashed", 3 = "dotted", 4 = "dotdash", 5 = "longdash", 6 = "twodash")

size - integer (line width in mm)

shape - integer/shape name or a single character ("a")

posit

Geoms Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

GRAPHICAL PRIMITIVES

a <- ggplot(economics, aes(date, unemploy))
b <- ggplot(seals, aes(x = long, y = lat))

a + geom_blank() and **a + expand_limits()**
Ensure limits include values across all plots.

b + geom_curve(aes(yend = lat + 1, xend = long + 1), curvature = 1**)** - x, xend, y, yend, alpha, angle, color, curvature, linetype, size

a + geom_path(lineend = "butt", linejoin = "round", linemitre = 1**)**
x, y, alpha, color, group, linetype, size

a + geom_polygon(aes(alpha = 50)**)** - x, y, alpha, color, fill, group, subgroup, linetype, size

b + geom_rect(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1)**)** - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

a + geom_ribbon(aes(ymin = unemploy - 900, ymax = unemploy + 900)**)** - x, ymax, ymin, alpha, color, fill, group, linetype, size

LINE SEGMENTS

common aesthetics: x, y, alpha, color, linetype, size

b + geom_abline(aes(intercept = 0, slope = 1)**)**
b + geom_hline(aes(yintercept = lat)**)**
b + geom_vline(aes(xintercept = long)**)**

b + geom_segment(aes(yend = lat + 1, xend = long + 1)**)**
b + geom_spoke(aes(angle = 1:1155, radius = 1)**)**

ONE VARIABLE continuous

c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)

c + geom_area(stat = "bin"**)**
x, y, alpha, color, fill, linetype, size

c + geom_density(kernel = "gaussian"**)**
x, y, alpha, color, fill, group, linetype, size, weight

c + geom_dotplot()
x, y, alpha, color, fill

c + geom_freqpoly()
x, y, alpha, color, group, linetype, size

c + geom_histogram(binwidth = 5**)**
x, y, alpha, color, fill, linetype, size, weight

c2 + geom_qq(aes(sample = hwy)**)**
x, y, alpha, color, fill, linetype, size, weight

discrete

d <- ggplot(mpg, aes(fl))

d + geom_bar()
x, alpha, color, fill, linetype, size, weight

TWO VARIABLES

both continuous

e <- ggplot(mpg, aes(cty, hwy))

e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1**)** - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

e + geom_point()
x, y, alpha, color, fill, shape, size, stroke

e + geom_quantile()
x, y, alpha, color, group, linetype, size, weight

e + geom_rug(sides = "bl"**)**
x, y, alpha, color, linetype, size

e + geom_smooth(method = lm**)**
x, y, alpha, color, fill, group, linetype, size, weight

e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1**)** - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

one discrete, one continuous

f <- ggplot(mpg, aes(class, hwy))

f + geom_col()
x, y, alpha, color, fill, group, linetype, size

f + geom_boxplot()
x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

f + geom_dotplot(binaxis = "y", stackdir = "center"**)**
x, y, alpha, color, fill, group

f + geom_violin(scale = "area"**)**
x, y, alpha, color, fill, group, linetype, size, weight

both discrete

g <- ggplot(diamonds, aes(cut, color))

g + geom_count()
x, y, alpha, color, fill, shape, size, stroke

e + geom_jitter(height = 2, width = 2**)**
x, y, alpha, color, fill, shape, size

continuous bivariate distribution

h <- ggplot(diamonds, aes(carat, price))

h + geom_bin2d(binwidth = c(0.25, 500)**)**
x, y, alpha, color, fill, linetype, size, weight

h + geom_density_2d()
x, y, alpha, color, group, linetype, size

h + geom_hex()
x, y, alpha, color, fill, size

continuous function

i <- ggplot(economics, aes(date, unemploy))

i + geom_area()
x, y, alpha, color, fill, linetype, size

i + geom_line()
x, y, alpha, color, group, linetype, size

i + geom_step(direction = "hv"**)**
x, y, alpha, color, group, linetype, size

visualizing error

df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
j <- ggplot(df, aes(grp, fit, ymin = fit - se, ymax = fit + se))

j + geom_crossbar(fatten = 2**)** - x, y, ymax, ymin, alpha, color, fill, group, linetype, size

j + geom_errorbar() - x, ymax, ymin, alpha, color, group, linetype, size, width
Also **geom_errorbarh()**.

j + geom_linerange()
x, ymin, ymax, alpha, color, group, linetype, size

j + geom_pointrange() - x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

maps

data <- data.frame(murder = USArrests$Murder,
      state = tolower(rownames(USArrests)))
map <- map_data("state")
k <- ggplot(data, aes(fill = murder))

k + geom_map(aes(map_id = state), map = map**)**
  + expand_limits(x = map$long, y = map$lat**)**
map_id, alpha, color, fill, linetype, size

THREE VARIABLES

seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2)); l <- ggplot(seals, aes(long, lat))

l + geom_contour(aes(z = z)**)**
x, y, z, alpha, color, group, linetype, size, weight

l + geom_contour_filled(aes(fill = z)**)**
x, y, alpha, color, fill, group, linetype, size, subgroup

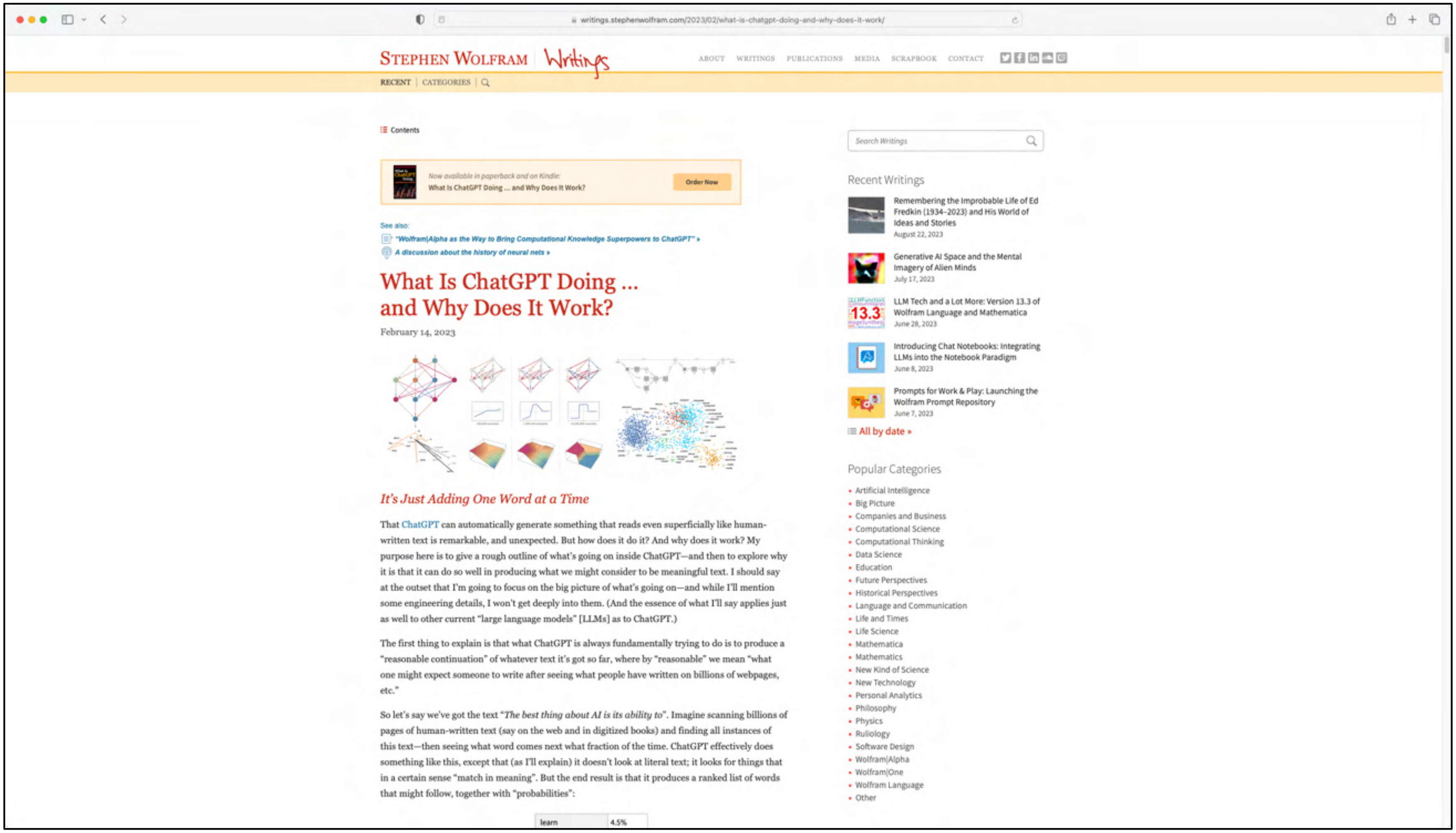
l + geom_raster(aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE**)**
x, y, alpha, fill

l + geom_tile(aes(fill = z)**)**
x, y, alpha, color, fill, linetype, size, width

• info@posit.co • posit.co • Learn more at **ggplot2.tidyverse.org** • HTML cheatsheets at **pos.it/cheatsheets** • ggplot2 3.4.2 • Updated: 2023-07

https://ggplot2.tidyverse.org/

What is ChatGPT and Why Does it Work?



writings.stephenwolfram.com/2023/02/what-is-chatgpt-doing-and-why-does-it-work/

Contents

See also:

"Wolfram|Alpha as the Way to Bring Computational Knowledge Superpowers to ChatGPT" »

A discussion about the history of neural nets »

What Is ChatGPT Doing ... and Why Does It Work?

February 14, 2023

It's Just Adding One Word at a Time

That ChatGPT can automatically generate something that reads even superficially like human-written text is remarkable, and unexpected. But how does it do it? And why does it work? My purpose here is to give a rough outline of what's going on inside ChatGPT—and then to explore why it is that it can do so well in producing what we might consider to be meaningful text. I should say at the outset that I'm going to focus on the big picture of what's going on—and while I'll mention some engineering details, I won't get deeply into them. (And the essence of what I'll say applies just as well to other current "large language models" [LLMs] as to ChatGPT.)

The first thing to explain is that what ChatGPT is always fundamentally trying to do is to produce a "reasonable continuation" of whatever text it's got so far, where by "reasonable" we mean "what one might expect someone to write after seeing what people have written on billions of webpages, etc."

So let's say we've got the text "*The best thing about AI is its ability to*". Imagine scanning billions of pages of human-written text (say on the web and in digitized books) and finding all instances of this text—then seeing what word comes next what fraction of the time. ChatGPT effectively does something like this, except that (as I'll explain) it doesn't look at literal text; it looks for things that in a certain sense "match in meaning". But the end result is that it produces a ranked list of words that might follow, together with "probabilities":

learn	4.5%

Search Writings

Recent Writings

Remembering the Improbable Life of Ed Fredkin (1934–2023) and His World of Ideas and Stories
August 22, 2023

Generative AI Space and the Mental Imagery of Alien Minds
July 17, 2023

LLM Tech and a Lot More: Version 13.3 of Wolfram Language and Mathematica
June 28, 2023

Introducing Chat Notebooks: Integrating LLMs into the Notebook Paradigm
June 8, 2023

Prompts for Work & Play: Launching the Wolfram Prompt Repository
June 7, 2023

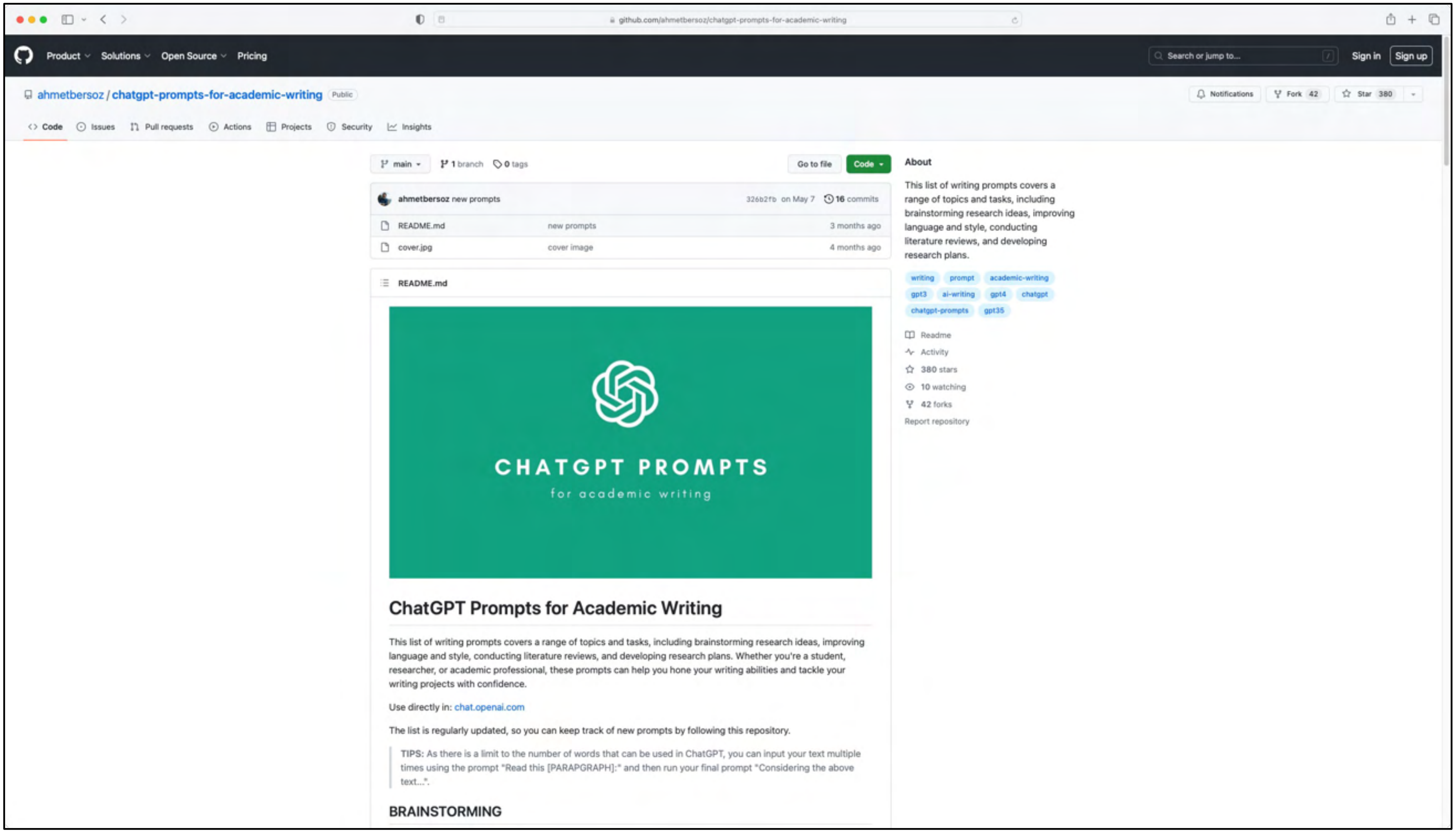
All by date »

Popular Categories

- Artificial Intelligence
- Big Picture
- Companies and Business
- Computational Science
- Computational Thinking
- Data Science
- Education
- Future Perspectives
- Historical Perspectives
- Language and Communication
- Life and Times
- Life Science
- Mathematica
- Mathematics
- New Kind of Science
- New Technology
- Personal Analytics
- Philosophy
- Physics
- Ruliology
- Software Design
- Wolfram|Alpha
- Wolfram|One
- Wolfram Language
- Other

https://writings.stephenwolfram.com/2023/02/what-is-chatgpt-doing-and-why-does-it-work/

ChatGPT Prompts for Academic Writing



ahmetbersoz new prompts 326b2fb on May 7 16 commits

README.md new prompts 3 months ago

cover.jpg cover image 4 months ago

README.md

About

This list of writing prompts covers a range of topics and tasks, including brainstorming research ideas, improving language and style, conducting literature reviews, and developing research plans.

writing prompt academic-writing gpt3 ai-writing gpt4 chatgpt chatgpt-prompts gpt35

ChatGPT Prompts for Academic Writing

This list of writing prompts covers a range of topics and tasks, including brainstorming research ideas, improving language and style, conducting literature reviews, and developing research plans. Whether you're a student, researcher, or academic professional, these prompts can help you hone your writing abilities and tackle your writing projects with confidence.

Use directly in: chat.openai.com

The list is regularly updated, so you can keep track of new prompts by following this repository.

> TIPS: As there is a limit to the number of words that can be used in ChatGPT, you can input your text multiple times using the prompt "Read this [PARAPGRAPH]:" and then run your final prompt "Considering the above text...".

BRAINSTORMING

https://github.com/ahmetbersoz/chatgpt-prompts-for-academic-writing

Unsolved Questions in Biology

- What is your genome sequence?
- How does your genome compare to my genome?

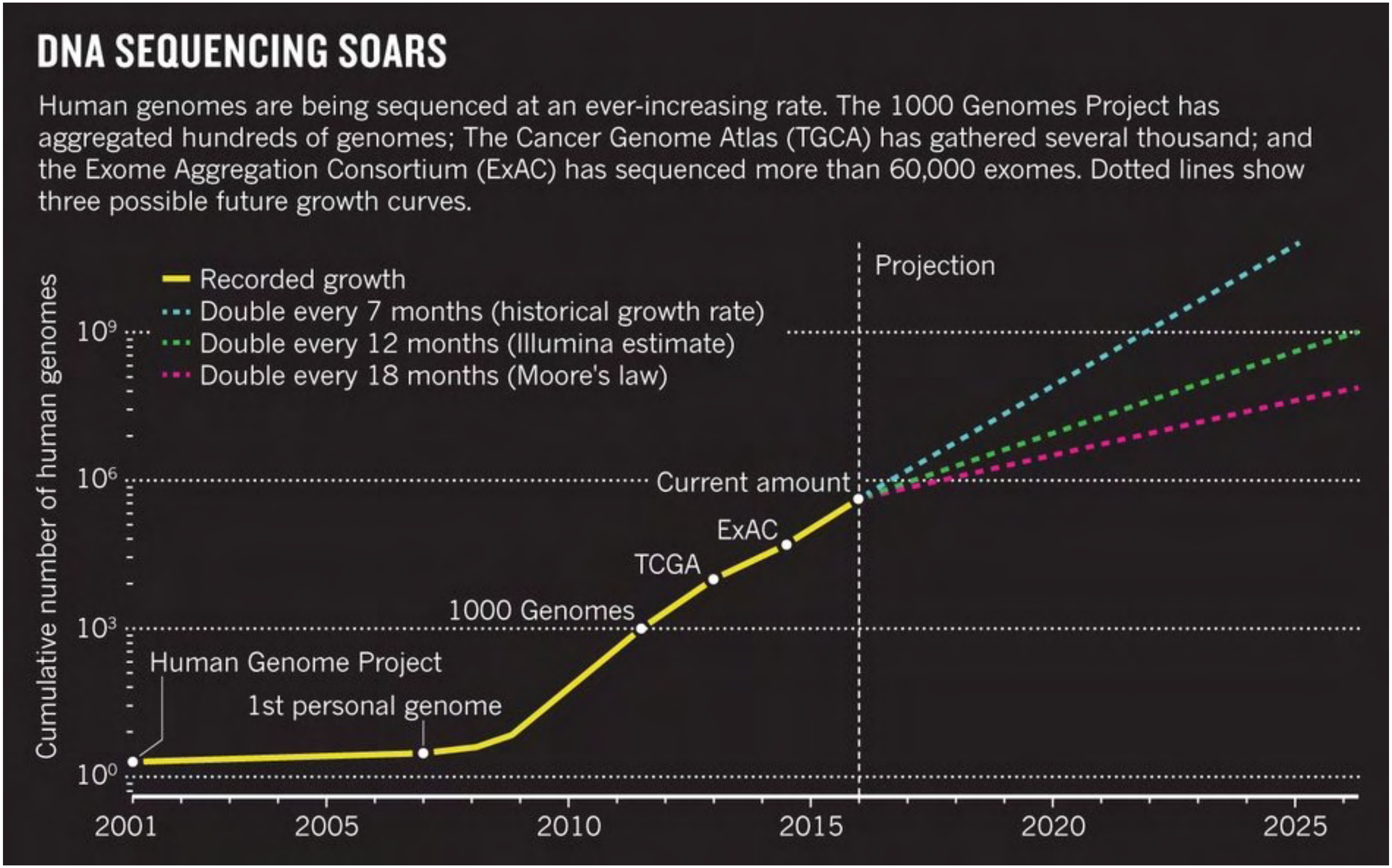
- Where are the genes and how active are they?
- How does gene activity change during development?
- How does splicing change during development?

- How does methylation change during development?
- How does chromatin change during development?
- How does is your genome folded in the cell?
- Where do proteins bind and regulate genes?

- What virus and microbes are living inside you?
- How do your mutations relate to disease?
- What drugs and treatments should we give you?

- ***Plus thousands and thousands more***

Sequencing Capacity



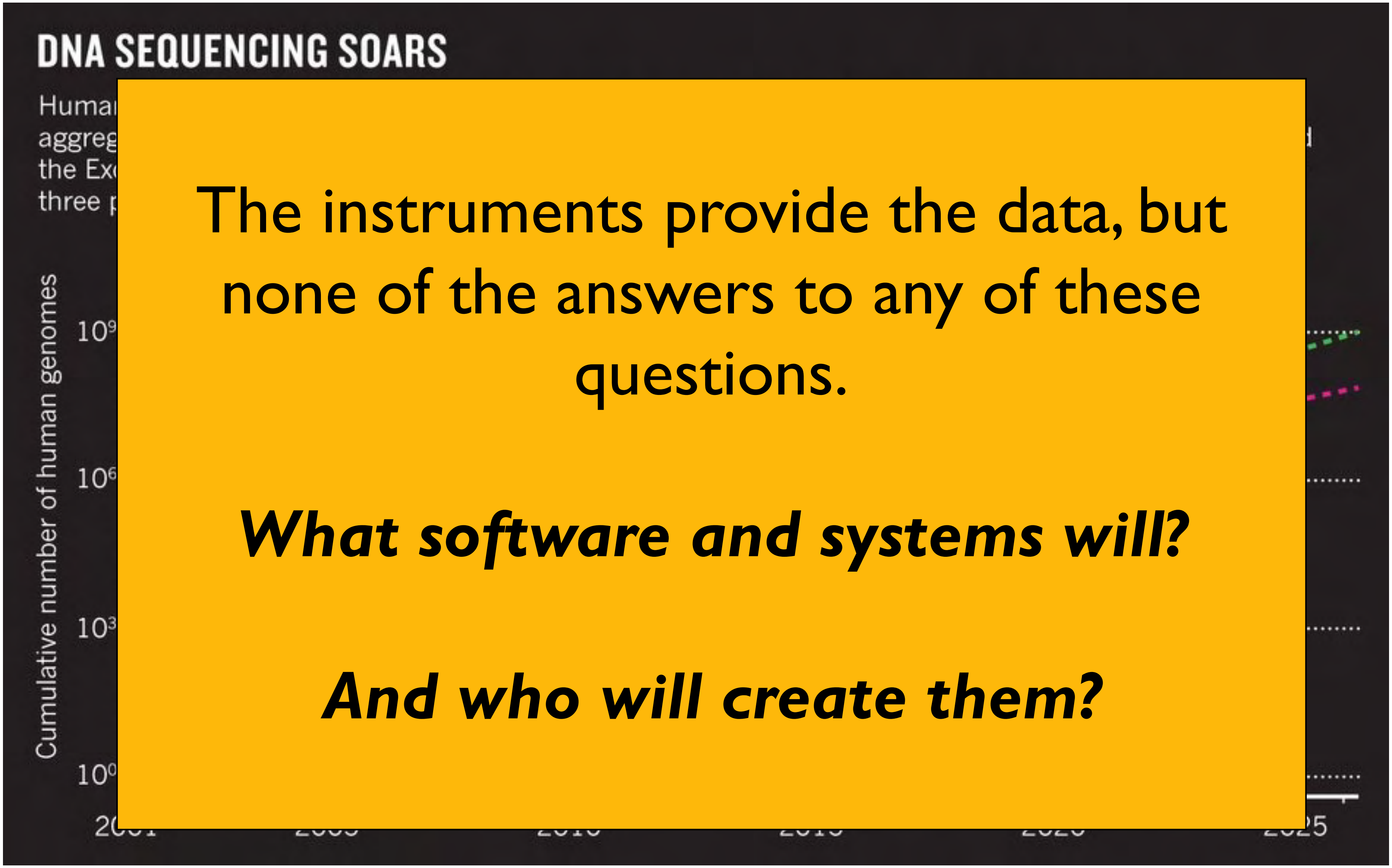
DNA SEQUENCING SOARS

Human genomes are being sequenced at an ever-increasing rate. The 1000 Genomes Project has aggregated hundreds of genomes; The Cancer Genome Atlas (TGCA) has gathered several thousand; and the Exome Aggregation Consortium (ExAC) has sequenced more than 60,000 exomes. Dotted lines show three possible future growth curves.

Big Data: Astronomical or Genomical?

Stephens, Z, et al. (2015) PLOS Biology DOI: 10.1371/journal.pbio.1002195

Sequencing Capacity



The instruments provide the data, but none of the answers to any of these questions.

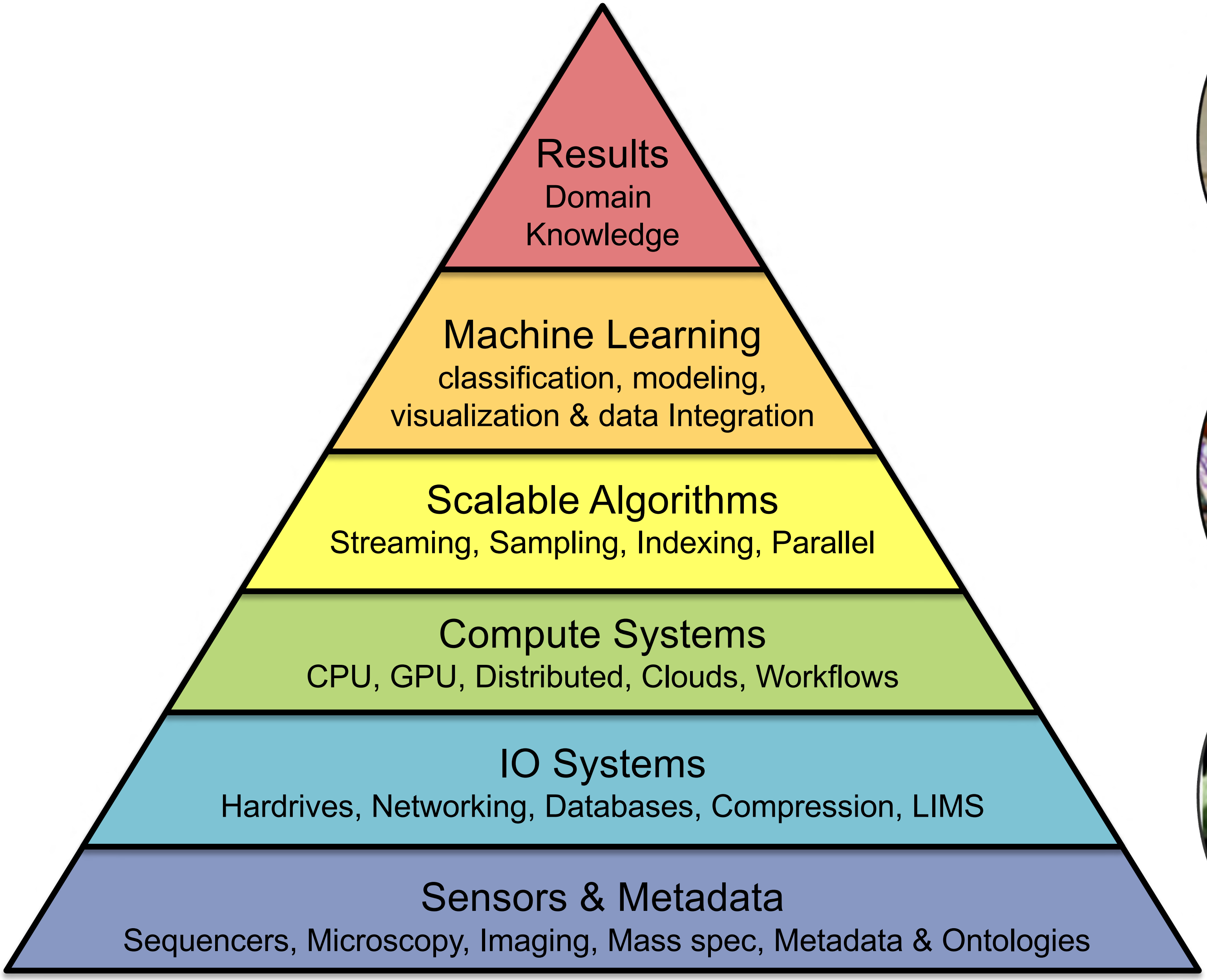
What software and systems will?

And who will create them?

Big Data: Astronomical or Genomical?

Stephens, Z, et al. (2015) PLOS Biology DOI: 10.1371/journal.pbio.1002195

Comparative Genomics Technologies



Results
Domain Knowledge

Machine Learning
classification, modeling, visualization & data Integration

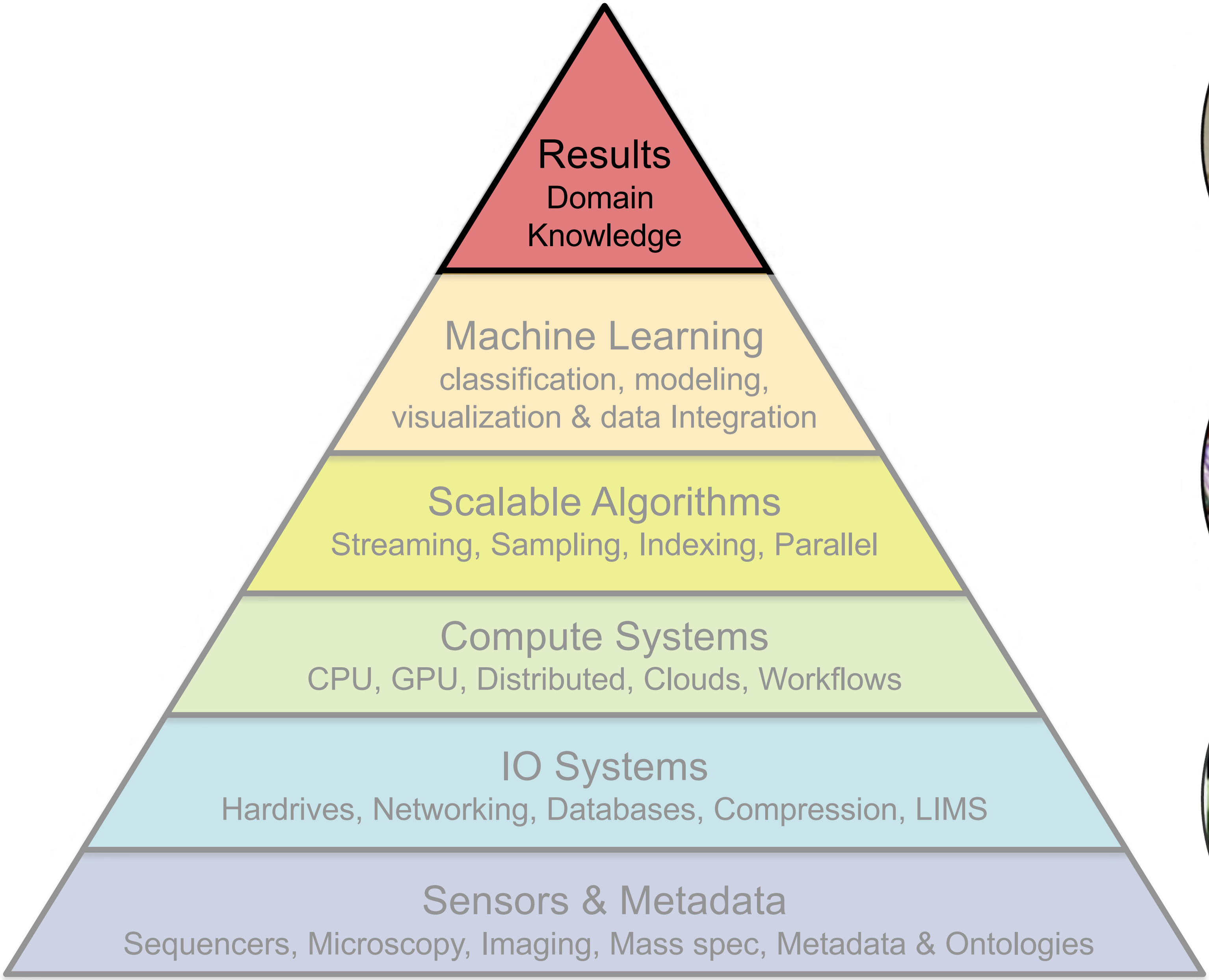
Scalable Algorithms
Streaming, Sampling, Indexing, Parallel

Compute Systems
CPU, GPU, Distributed, Clouds, Workflows

IO Systems
Hardrives, Networking, Databases, Compression, LIMS

Sensors & Metadata
Sequencers, Microscopy, Imaging, Mass spec, Metadata & Ontologies

Comparative Genomics Technologies



Results
Domain Knowledge

Machine Learning
classification, modeling, visualization & data Integration

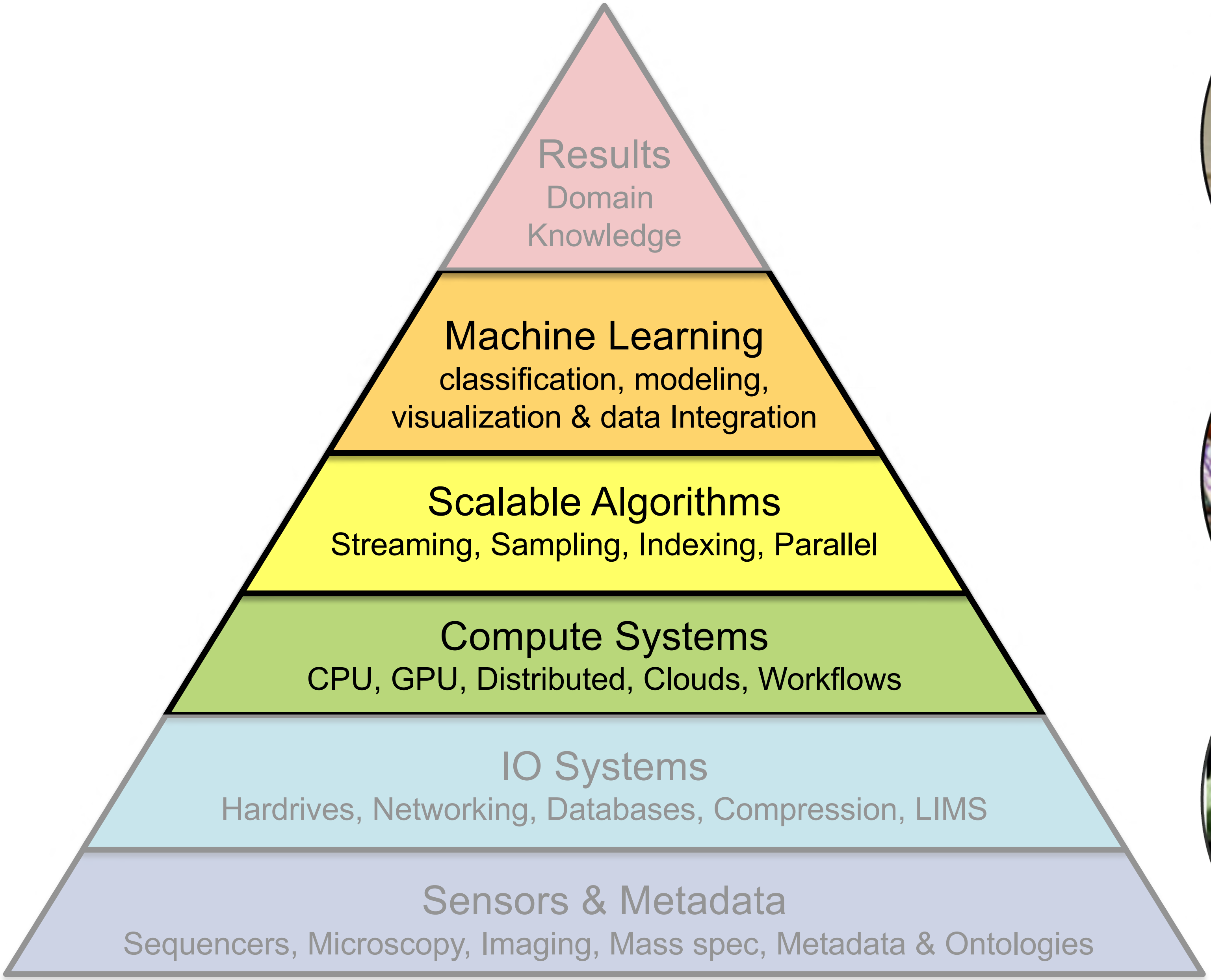
Scalable Algorithms
Streaming, Sampling, Indexing, Parallel

Compute Systems
CPU, GPU, Distributed, Clouds, Workflows

IO Systems
Hardrives, Networking, Databases, Compression, LIMS

Sensors & Metadata
Sequencers, Microscopy, Imaging, Mass spec, Metadata & Ontologies

Comparative Genomics Technologies



Results
Domain
Knowledge

Machine Learning
classification, modeling,
visualization & data Integration

Scalable Algorithms
Streaming, Sampling, Indexing, Parallel

Compute Systems
CPU, GPU, Distributed, Clouds, Workflows

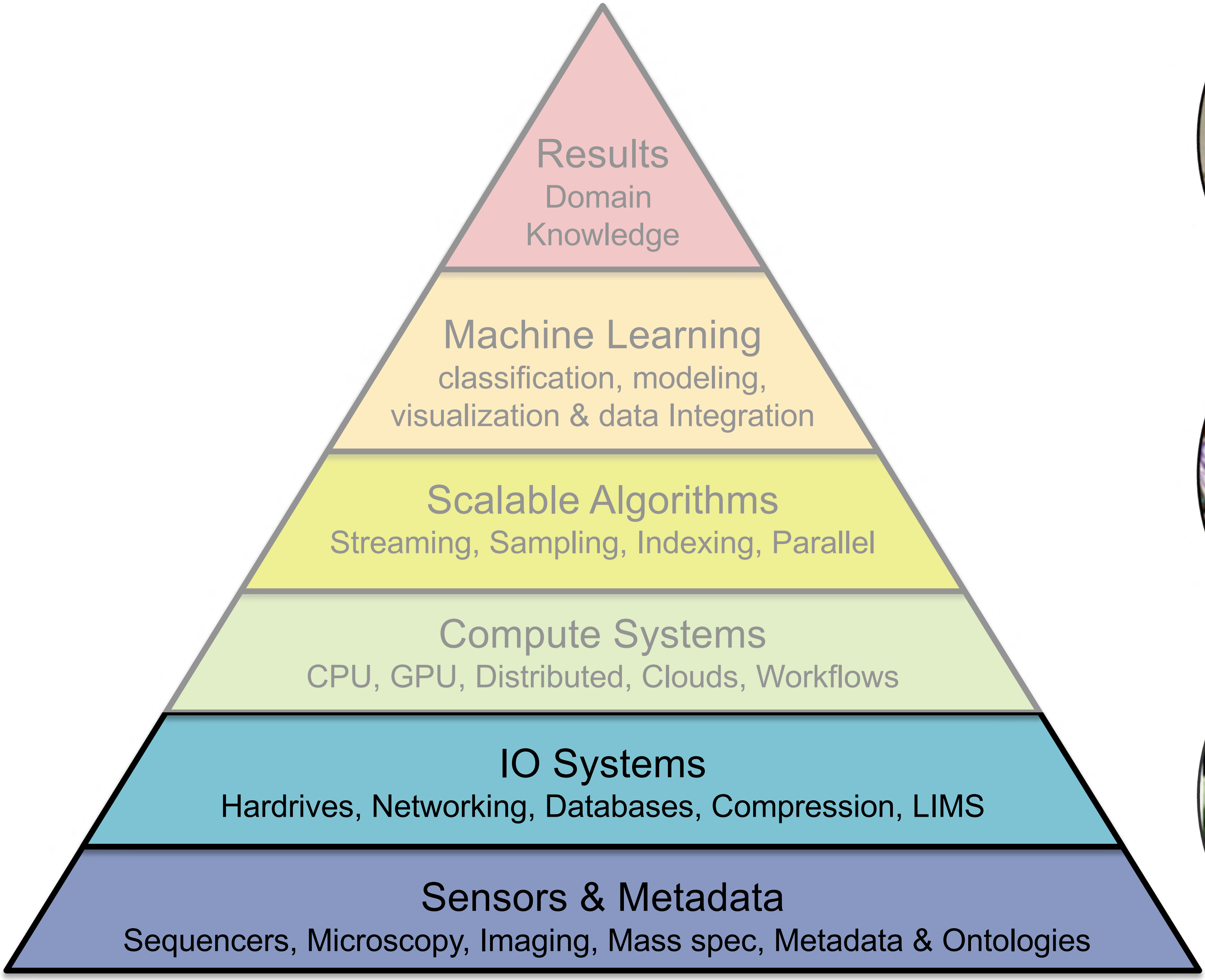
IO Systems
Hardrives, Networking, Databases, Compression, LIMS

Sensors & Metadata
Sequencers, Microscopy, Imaging, Mass spec, Metadata & Ontologies

Selected Topics

- Genome assembly, whole genome alignment
- Full text indexing: Suffix Trees, Suffix Arrays, FM-index
- Dynamic Programming: Edit Distance, sequence similarity
- Read mapping & Variant identification
- Gene Finding: HMMs, Plane-sweep algorithms
- RNA-seq: mapping, assembly, quantification
- ChIP-seq: Peak finding, motif finding
- Methylation-seq: Mapping, CpG island detection
- HiC: Domain identification, scaffolding
- Chromatin state analysis: ChromHMM
- Scalable genomics: Cloud computing, scalable data structures
- Population & single cell analysis: clustering, pseudotime
- Disease analysis, cancer genomics, Metagenomics
- Deep learning in genomics

Comparative Genomics Technologies



Results
Domain Knowledge

Machine Learning
classification, modeling, visualization & data Integration

Scalable Algorithms
Streaming, Sampling, Indexing, Parallel

Compute Systems
CPU, GPU, Distributed, Clouds, Workflows

IO Systems
Hardrives, Networking, Databases, Compression, LIMS

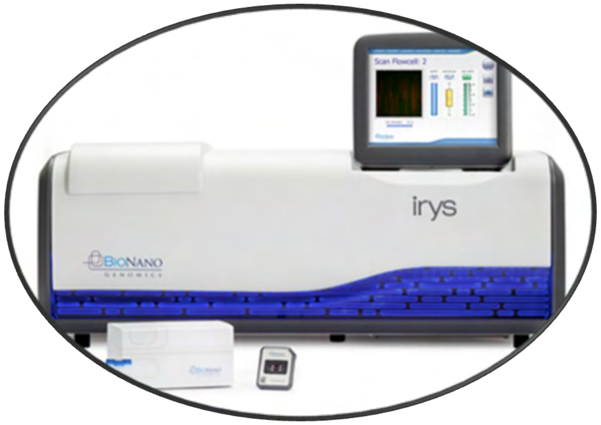
Sensors & Metadata
Sequencers, Microscopy, Imaging, Mass spec, Metadata & Ontologies

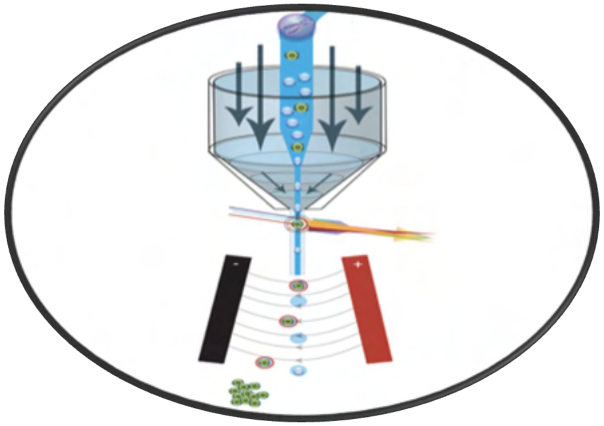
Genomics Arsenal in the year 2023

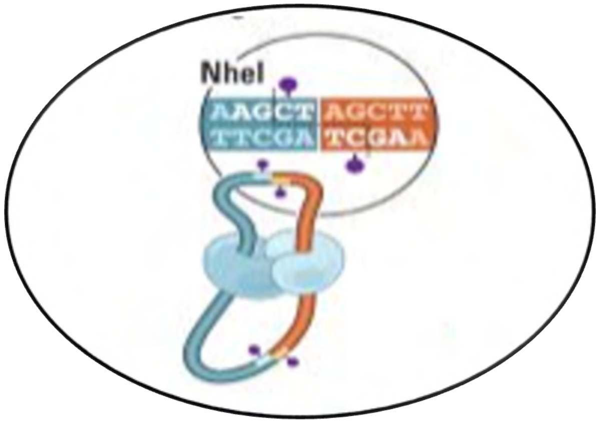
Sample Preparation

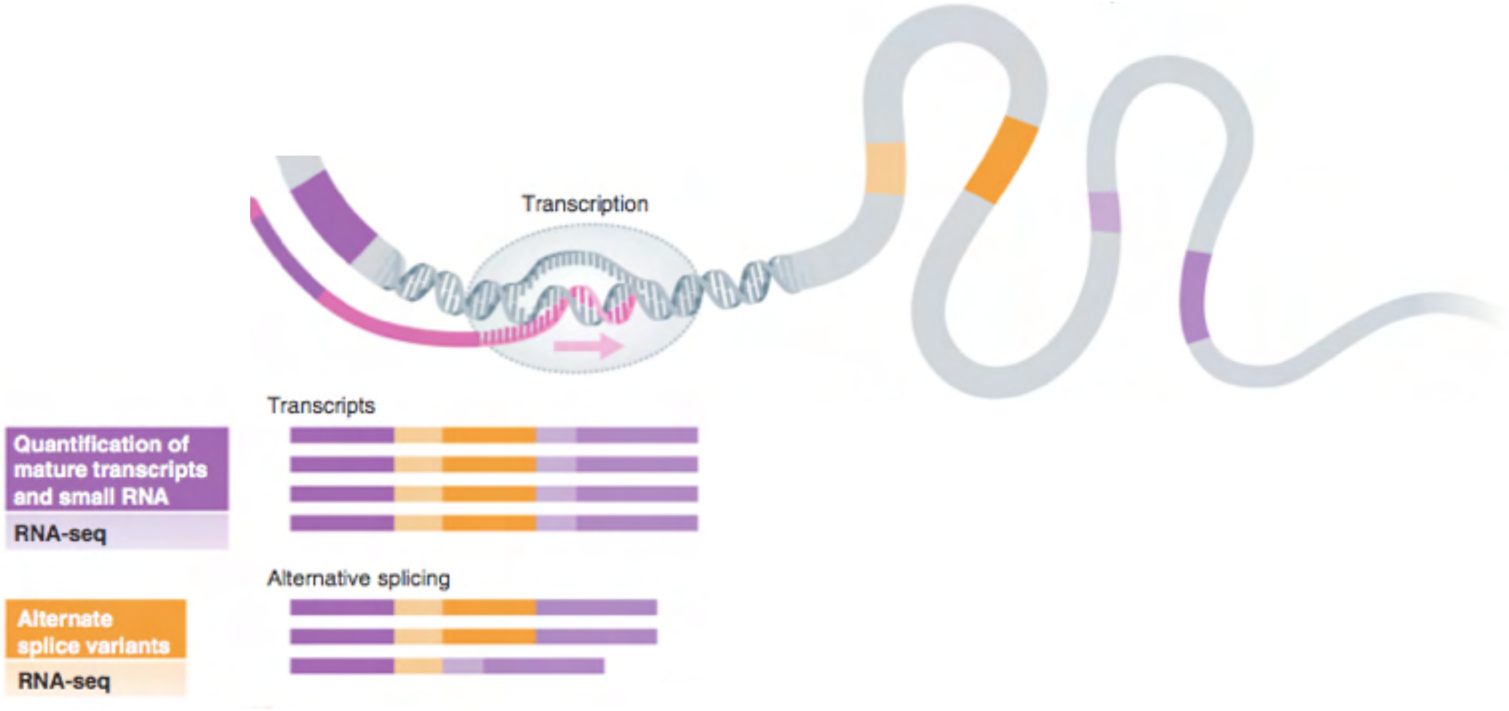
Sequencing

Chromosome Mapping

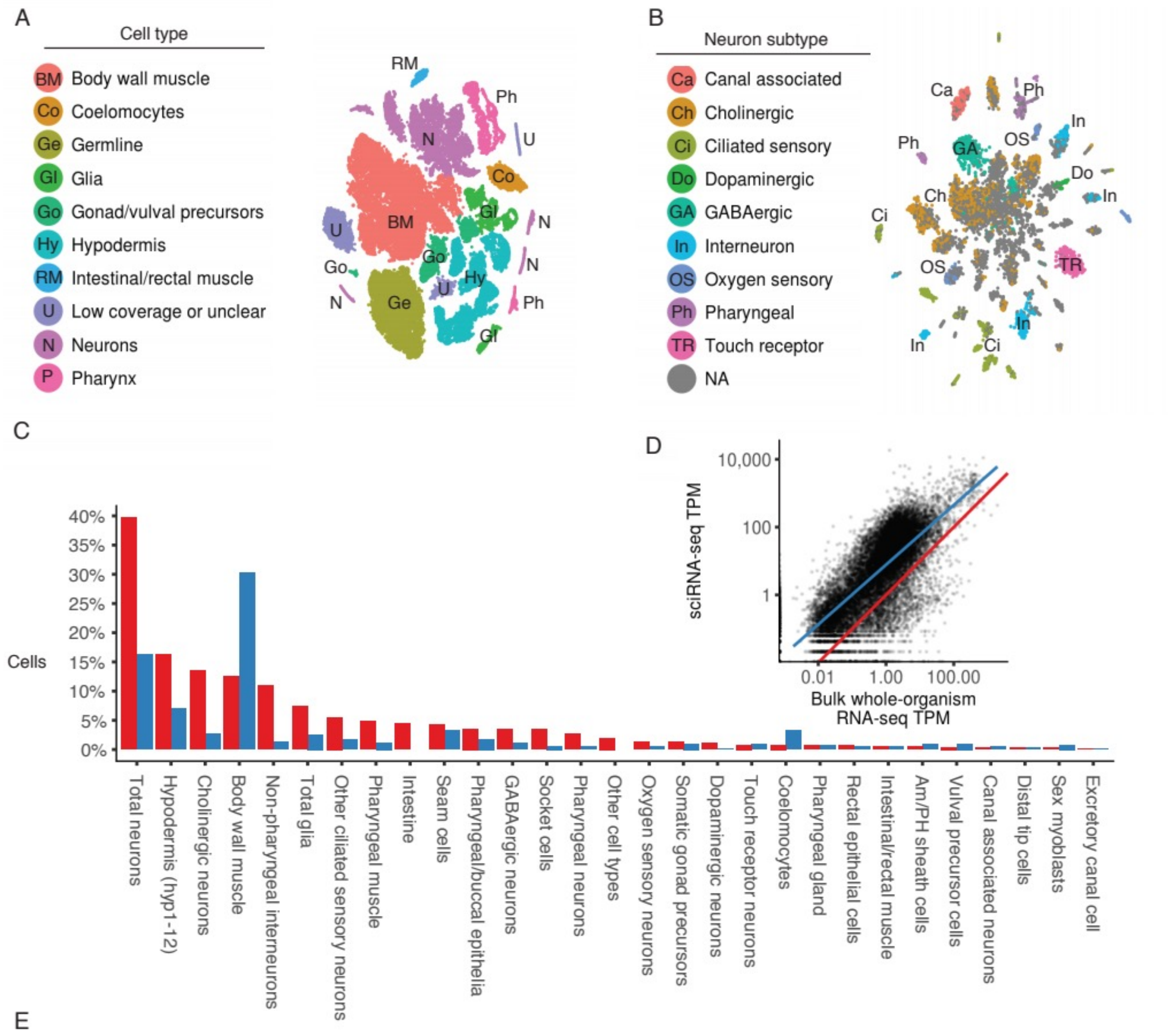







Transcription
Transcripts
Quantification of mature transcripts and small RNA
RNA-seq
Alternative splicing
Alternate splice variants
RNA-seq

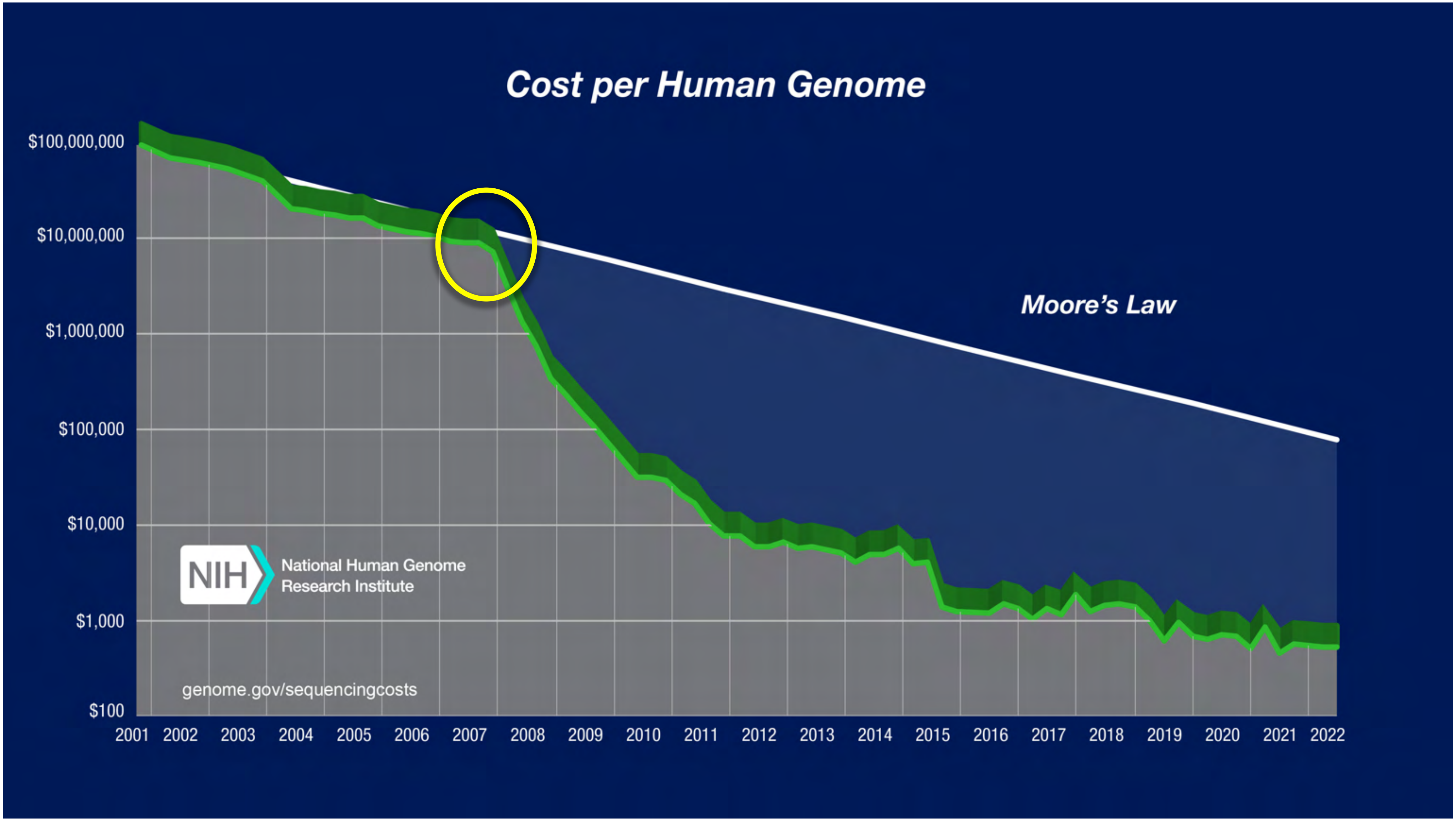
Soon et al., Molecular Systems Biology, 2013



Comprehensive single-cell transcriptional profiling of a multicellular organism

Cao, *et al.* (2017) *Science.* doi: 10.1126/science.aam8940

Cost per Genome

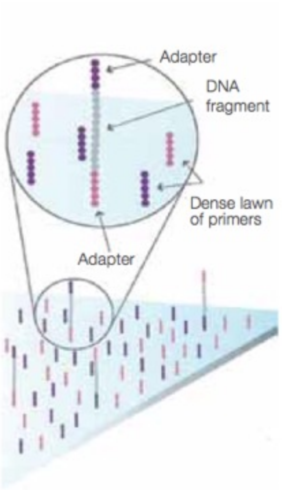


https://www.genome.gov/about-genomics/fact-sheets/Sequencing-Human-Genome-cost

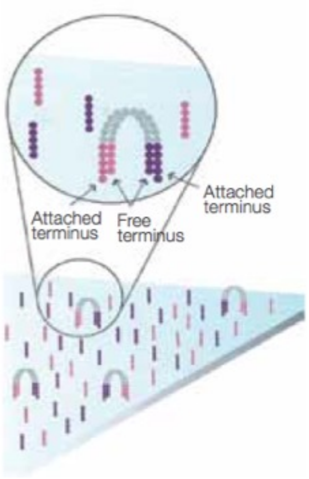
Second Generation Sequencing

Illumina NovaSeq 6000
Sequencing by Synthesis

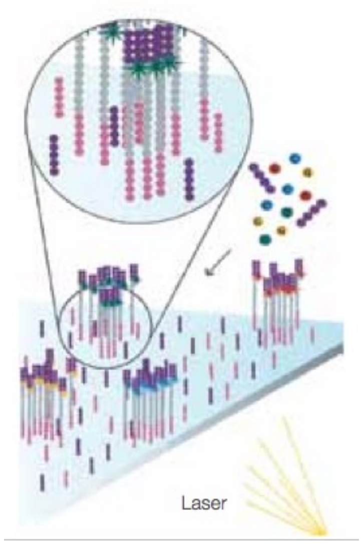
>3Tbp / day
(JHU has 4 of these!)



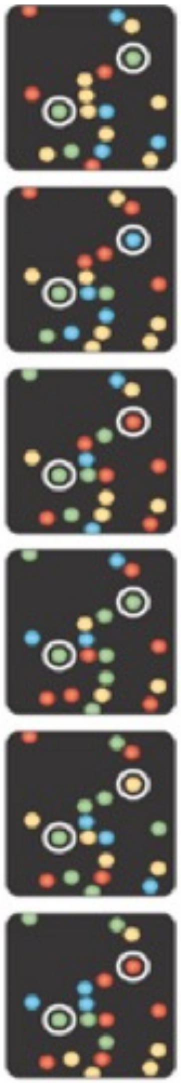
1. Attach



2. Amplify



3. Image

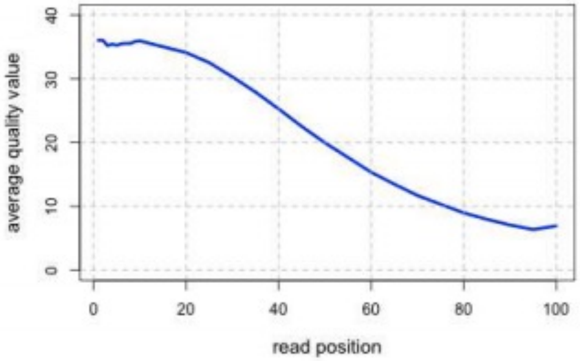


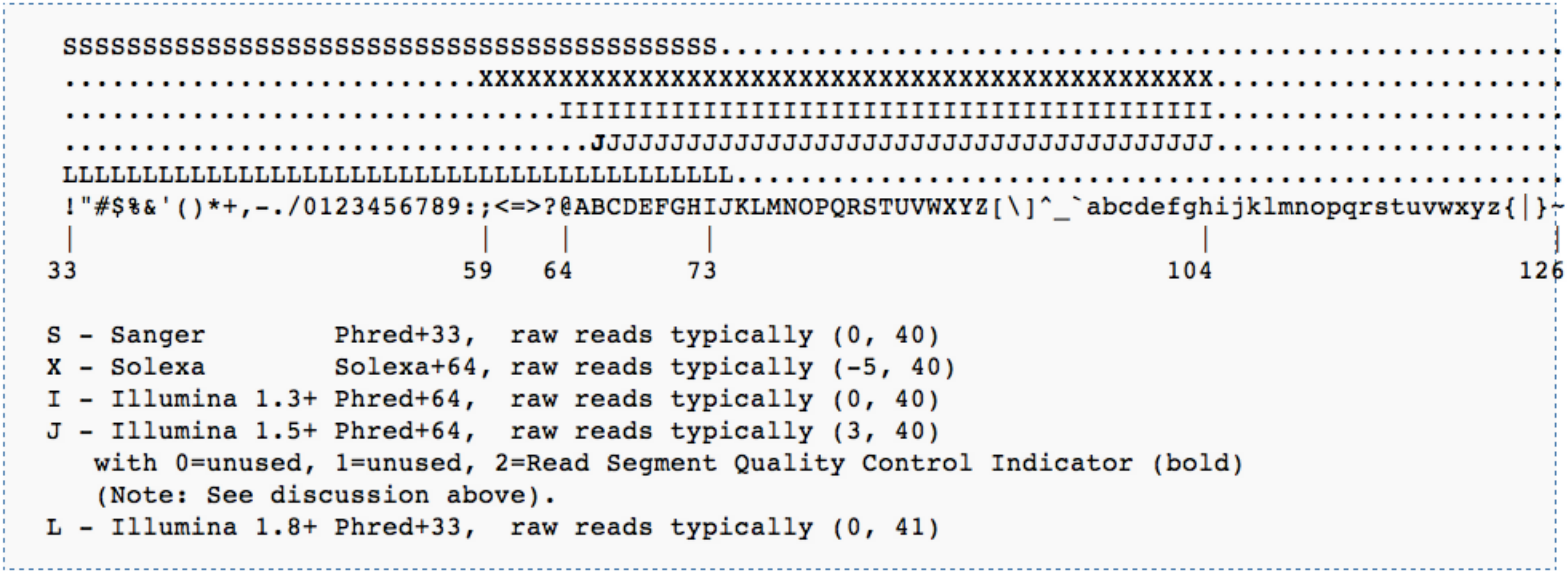
Metzker (2010) Nature Reviews Genetics 11:31-46
https://www.youtube.com/watch?v=fCd6B5HRaZ8

Illumina Quality

QV	p_{error}
40	1/10000
30	1/1000
20	1/100
10	1/10

$$Q_{\text{sanger}} = -10 \log_{10} p$$

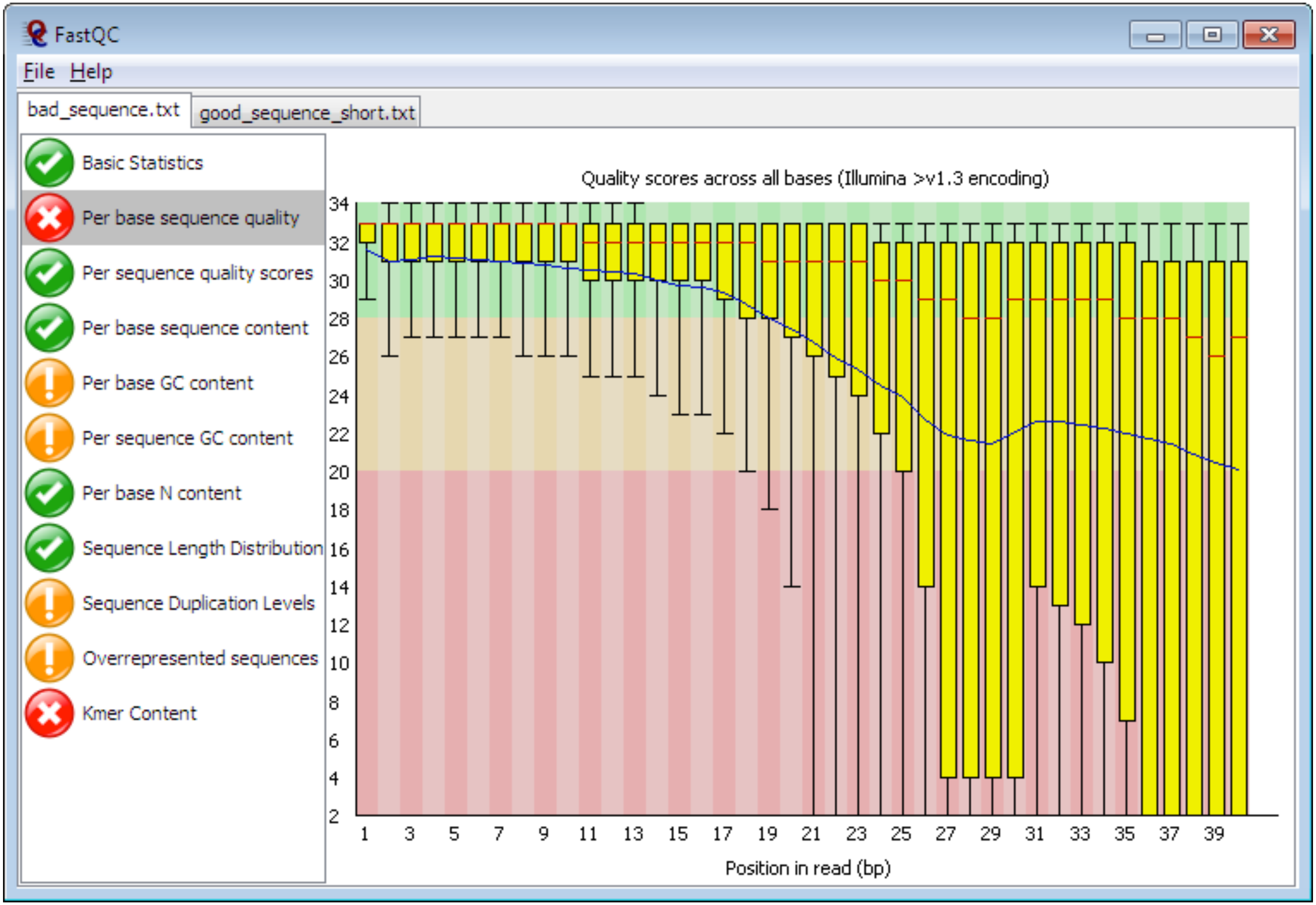




```
  SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....................................................
  ..........................XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......................
  ...............................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII......................
  .................................JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ......................
  LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL....................................................
  !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
  |                         |    |        |                              |                     |
 33                        59   64       73                            104                   126

S - Sanger        Phred+33,  raw reads typically (0, 40)
X - Solexa        Solexa+64, raw reads typically (-5, 40)
I - Illumina 1.3+ Phred+64,  raw reads typically (0, 40)
J - Illumina 1.5+ Phred+64,  raw reads typically (3, 40)
   with 0=unused, 1=unused, 2=Read Segment Quality Control Indicator (bold)
   (Note: See discussion above).
L - Illumina 1.8+ Phred+33,  raw reads typically (0, 41)
```

http://en.wikipedia.org/wiki/FASTQ_format

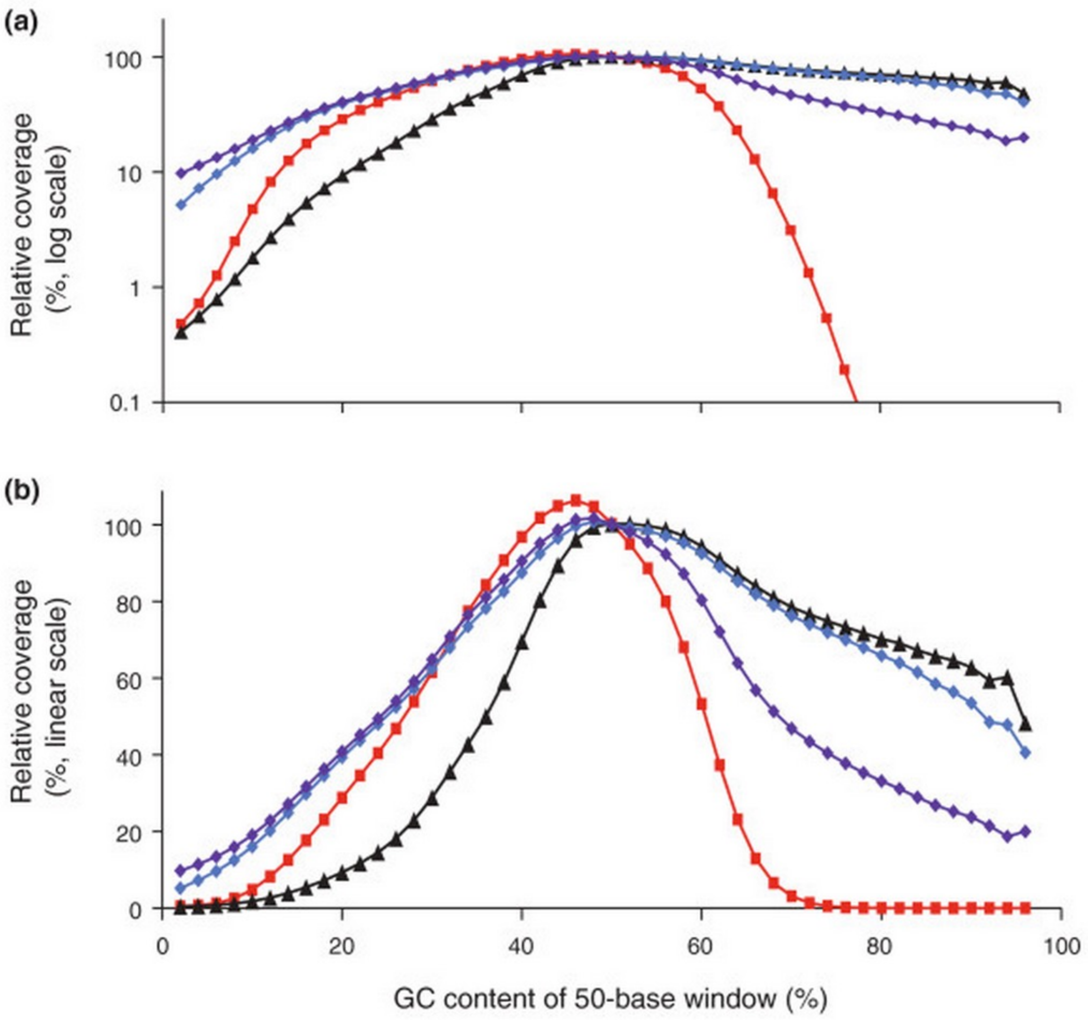
FASTQC: Is my data any good?



✓ Basic Statistics
✗ Per base sequence quality
✓ Per sequence quality scores
✓ Per base sequence content
! Per base GC content
! Per sequence GC content
✓ Per base N content
✓ Sequence Length Distribution
! Sequence Duplication Levels
! Overrepresented sequences
✗ Kmer Content

http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Beware of GC Biases

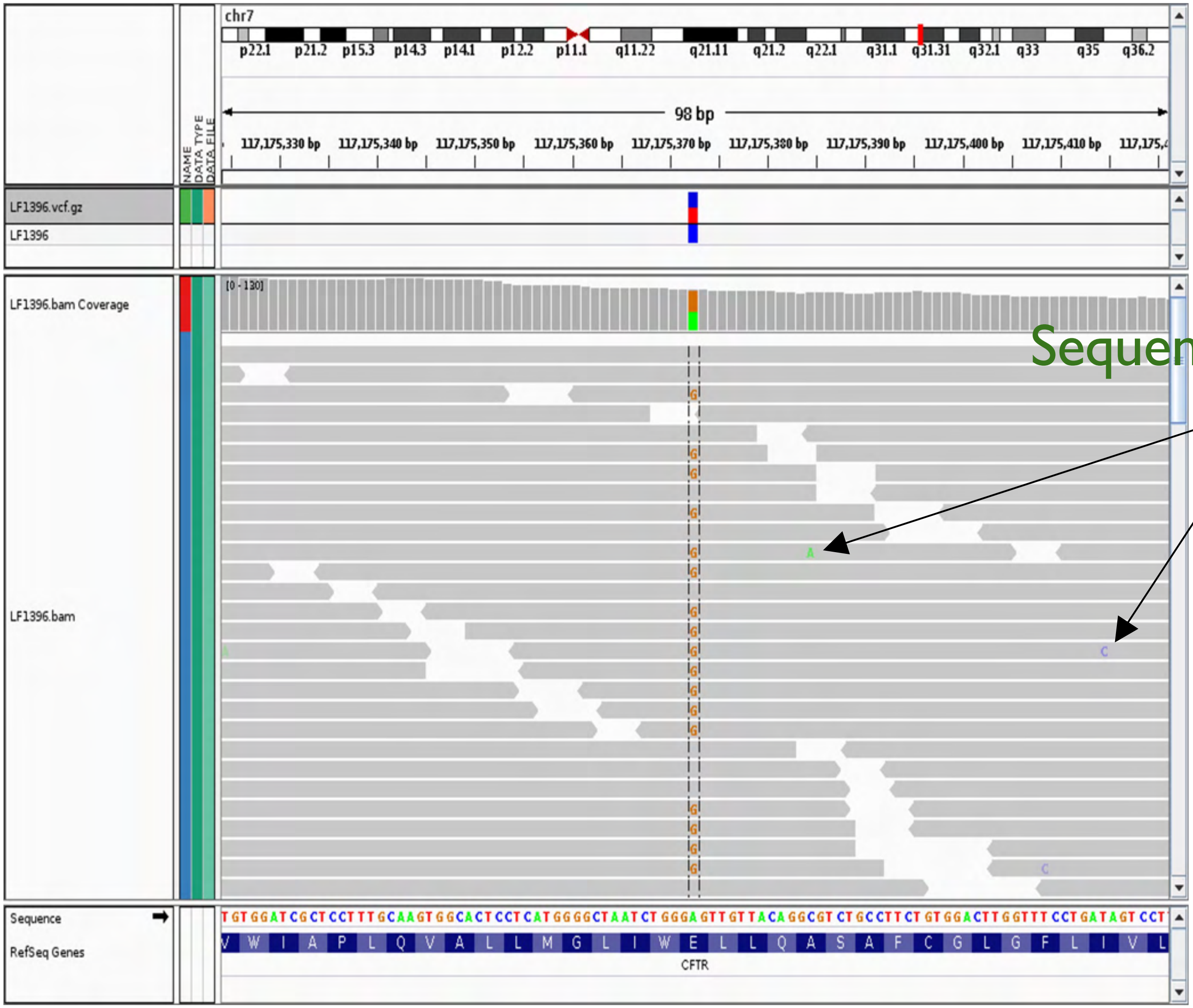


Illumina sequencing does not produce uniform coverage over the genome

- Coverage of extremely high or extremely low GC content will have reduced coverage in Illumina sequencing
- Biases primarily introduced during PCR; lower temperatures, slower heating, and fewer rounds minimize biases
- This makes it very difficult to identify variants (SNPs, CNVs, etc) in certain regions of the genome

Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries.
Aird et al. (2011) *Genome Biology.* 12:R18.

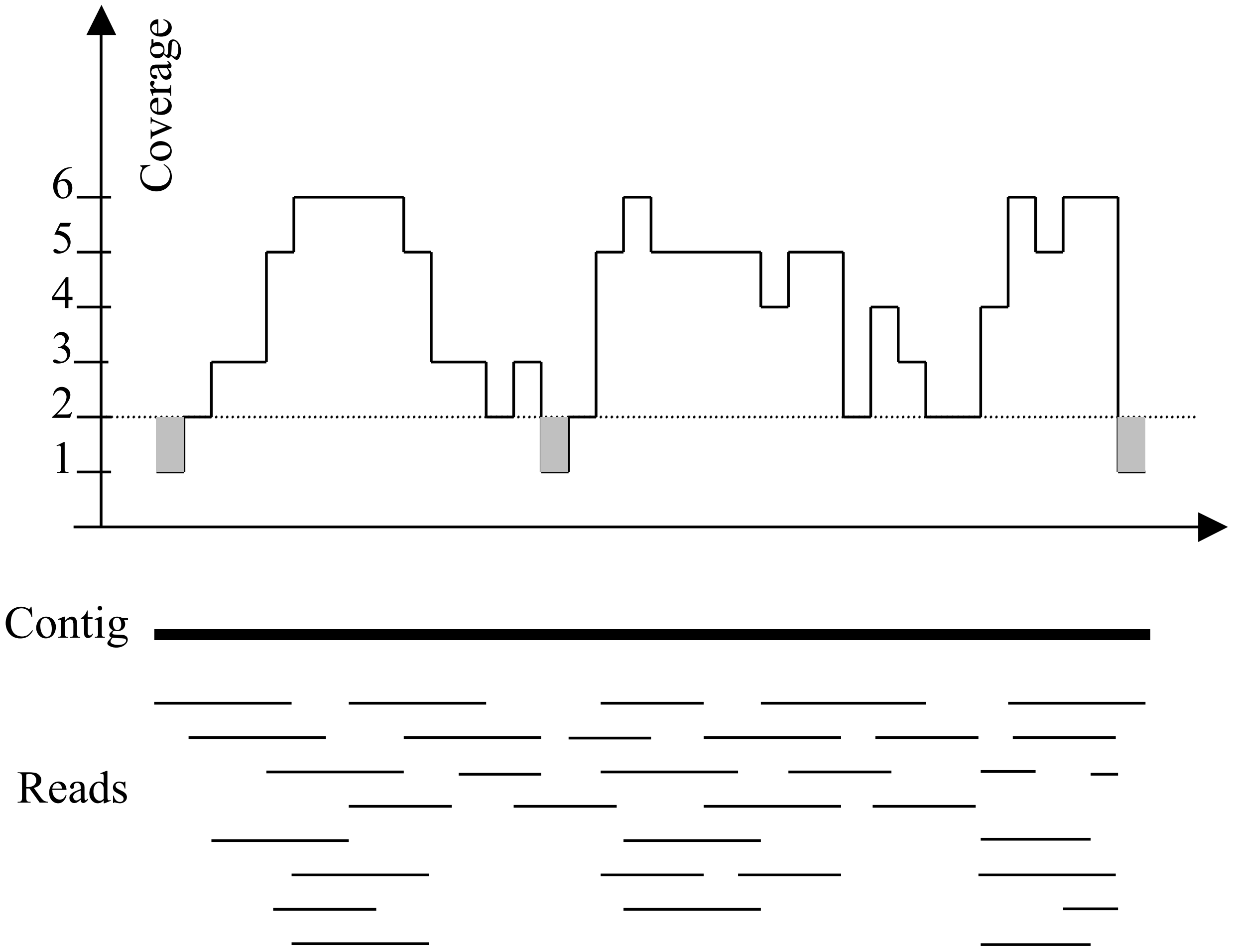
Sequencing errors fall out as noise (most of the time)


chr7
p22.1 p21.2 p15.3 p14.3 p14.1 p12.2 p11.1 q11.22 q21.11 q21.2 q22.1 q31.1 q31.31 q32.1 q33 q35 q36.2
98 bp
117,175,330 bp 117,175,340 bp 117,175,350 bp 117,175,360 bp 117,175,370 bp 117,175,380 bp 117,175,390 bp 117,175,400 bp 117,175,410 bp
NAME
DATA TYPE
DATA FILE
LF1396.vcf.gz
LF1396
LF1396.bam Coverage
[0 - 130]
LF1396.bam
G
A
C
Sequence
RefSeq Genes
TGTGGATCGCTCCTTTGCAAGTGGCACTCCTCATGGGGCTAATCTGGGAGTTGTTACAGGCGTCTGCCTTCTGTGGACTTGGTTTCCTGATAGTCCT
V W I A P L Q V A L L M G L I W E L L Q A S A F C G L G F L I V L
CFTR

Sequencing errors

https://jchoigt.files.wordpress.com/2012/07/igv_e217g_snapshot.png

Typical sequencing coverage

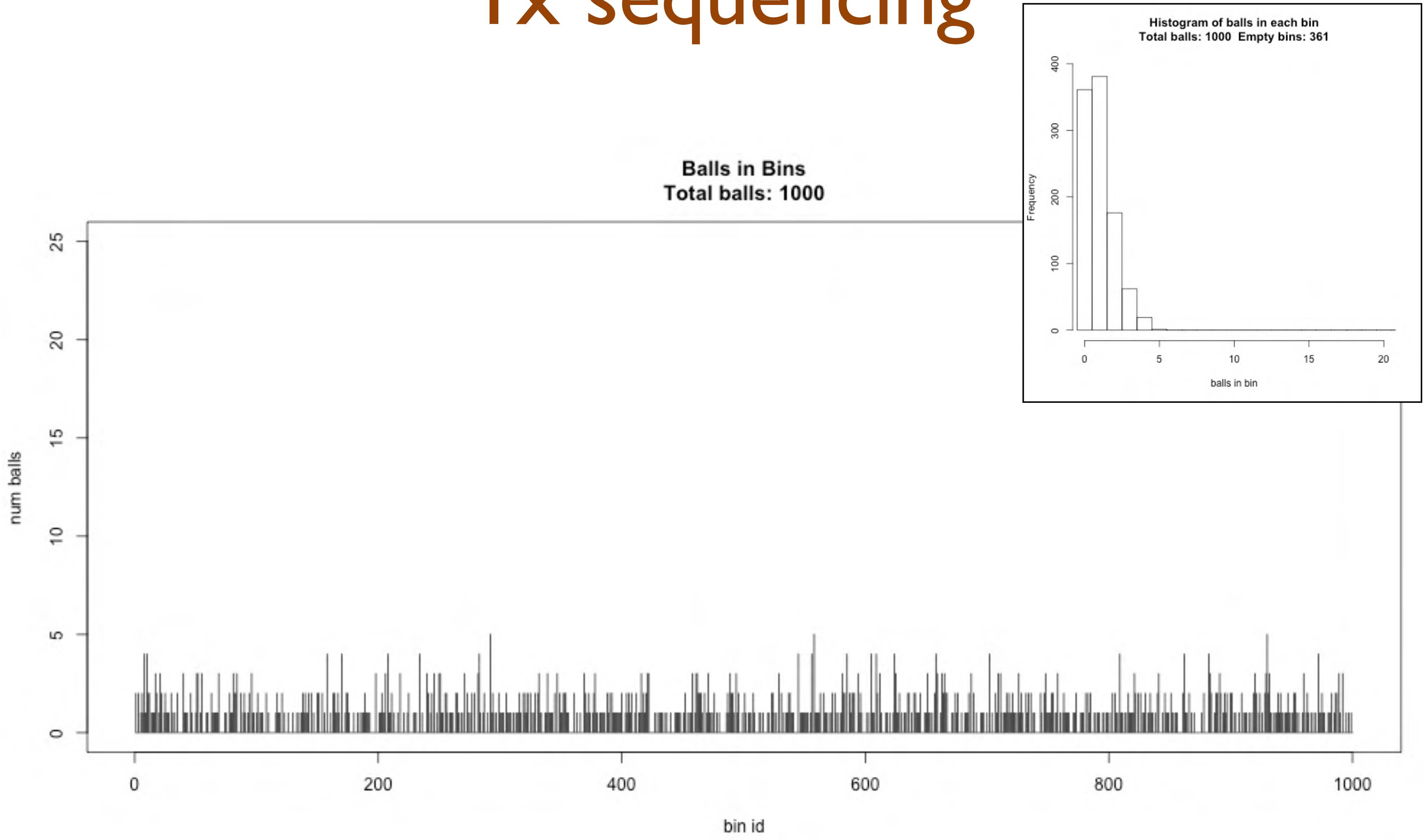


Imagine raindrops on a sidewalk

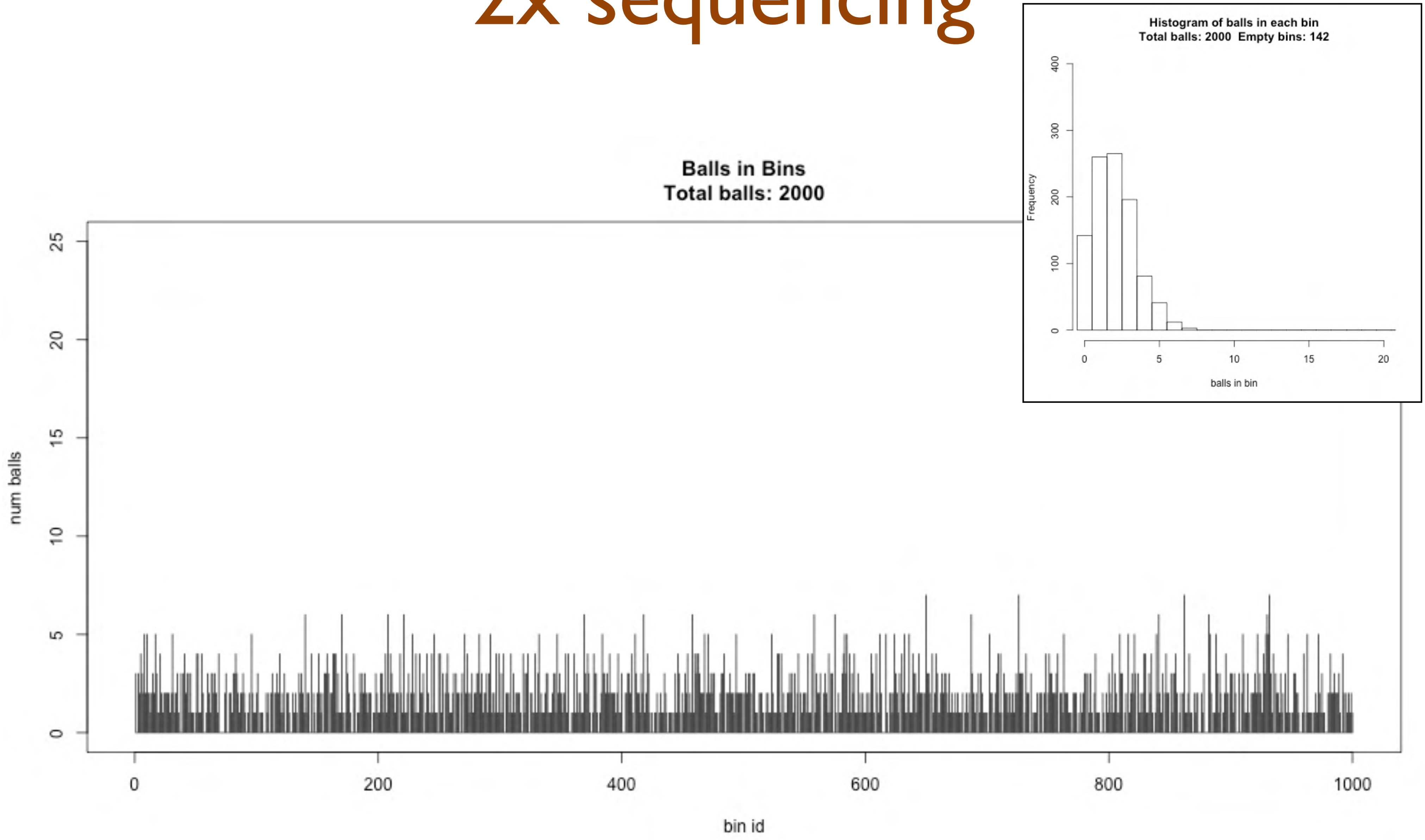
We want to cover the entire sidewalk but each drop costs $1

If the genome is 10 Mbp, should we sequence 100k 100bp reads?

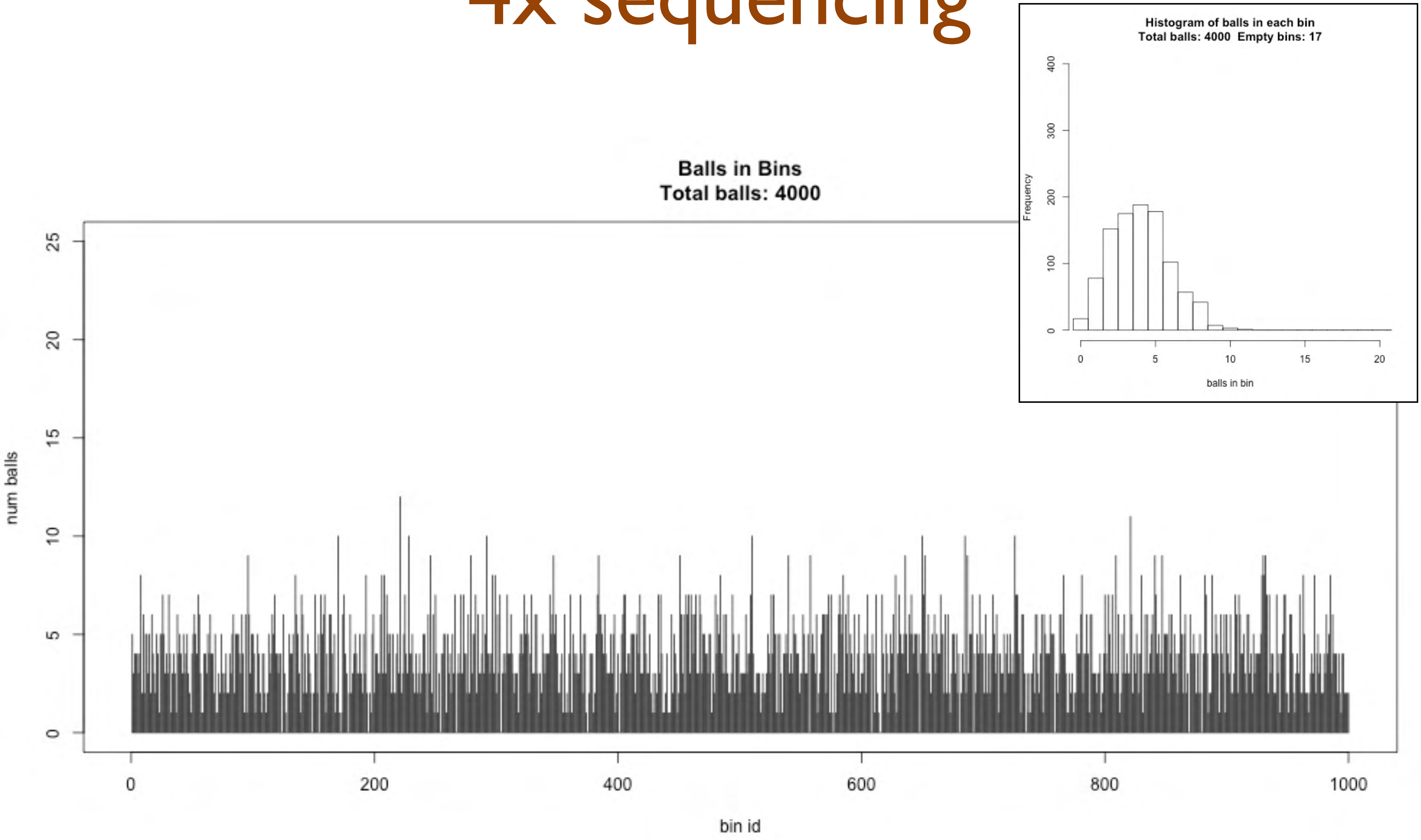
1x sequencing



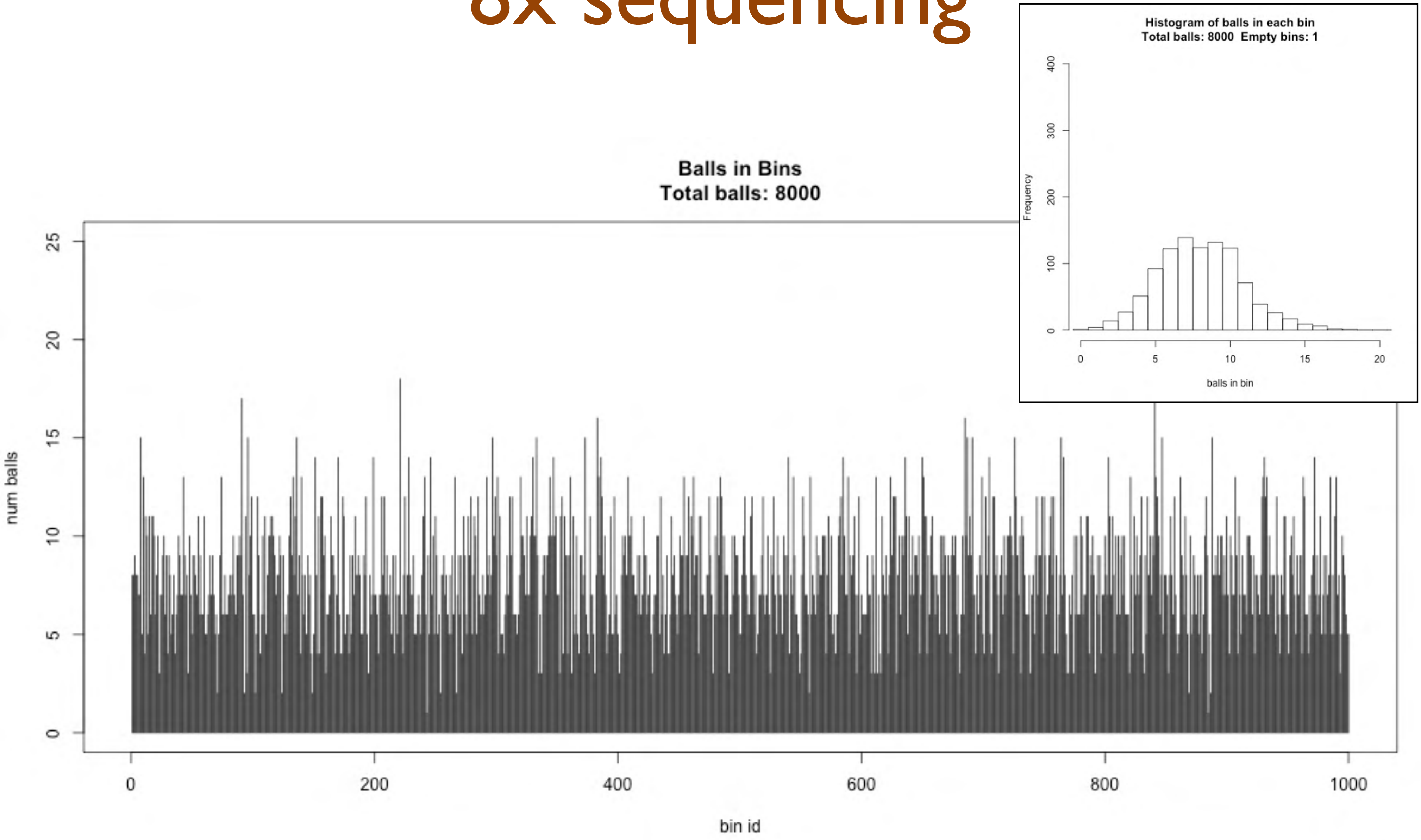
2x sequencing



4x sequencing



8x sequencing



Poisson Distribution

The probability of a given number of events occurring in a fixed interval of time and/or space if these events occur with a known average rate and independently of the time since the last event.

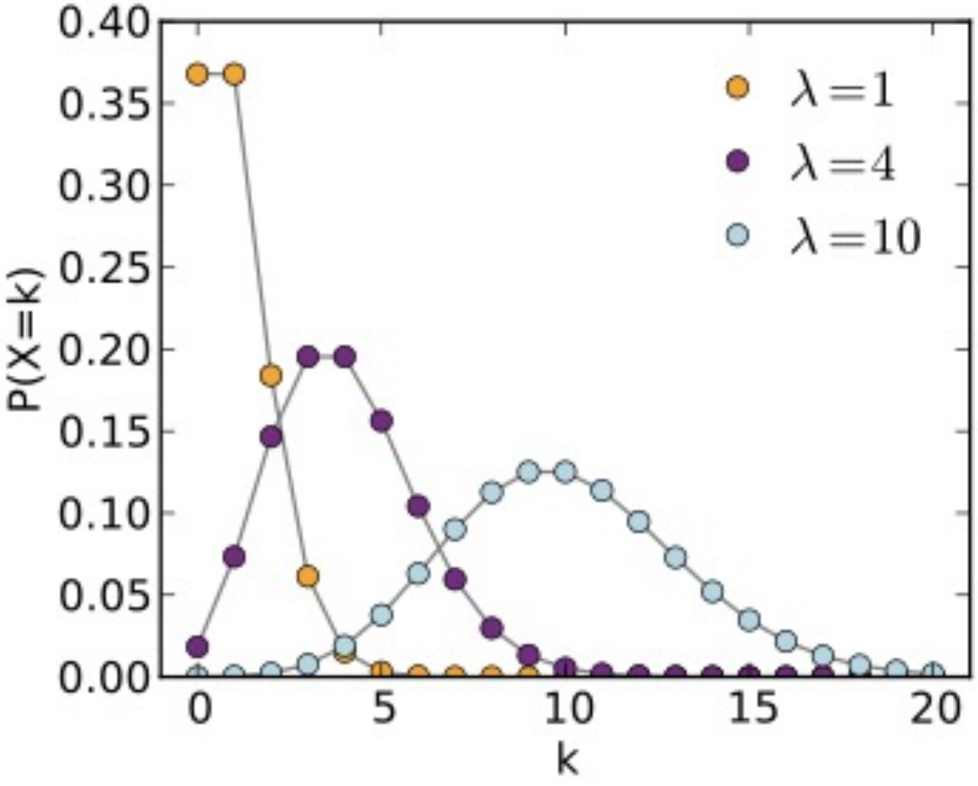
$$P(k) = \frac{\lambda^k}{k!} e^{-\lambda}$$

Formulation comes from the limit of the binomial equation

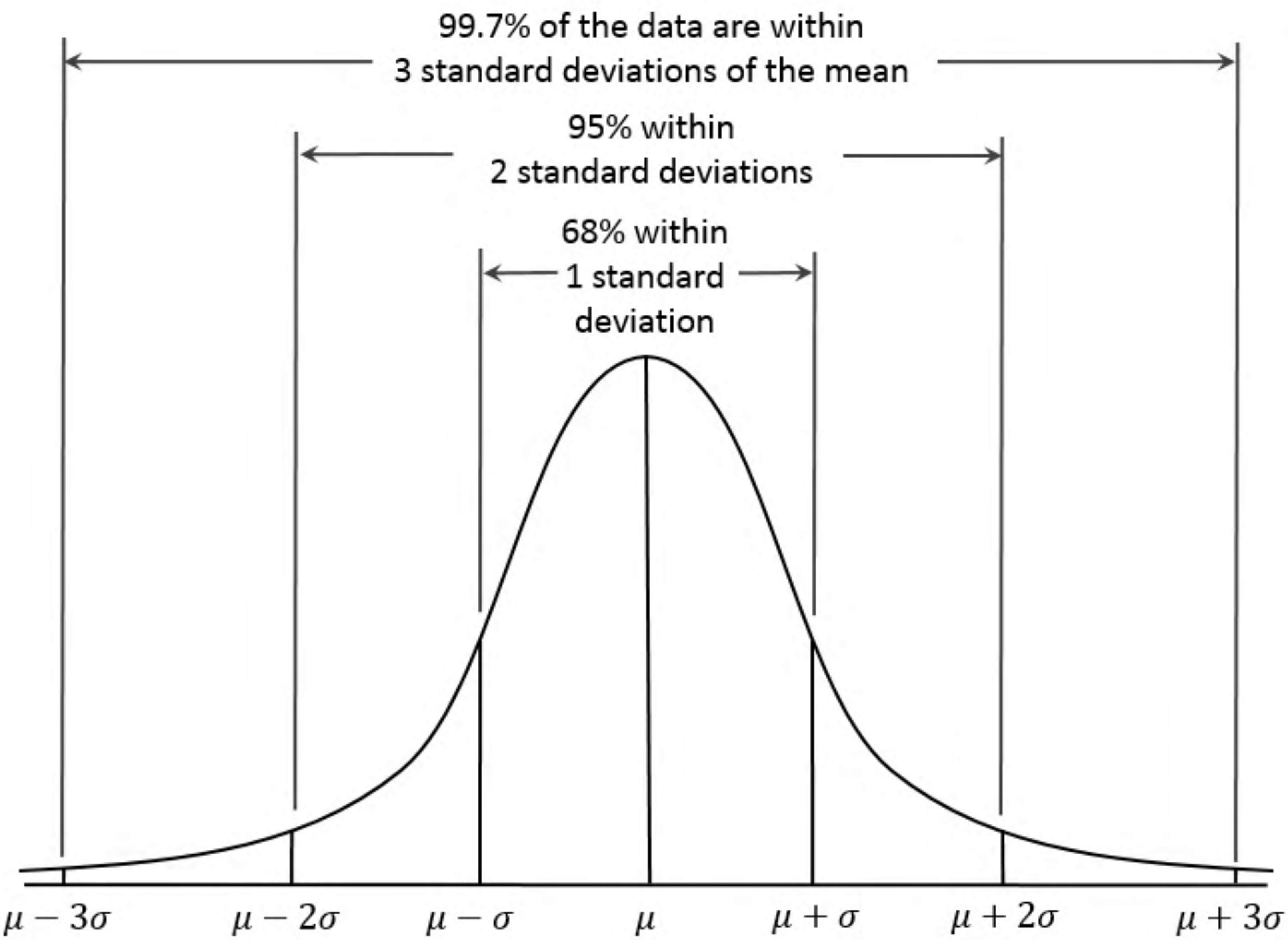
Resembles a normal distribution, but over the positive values, and with only a single parameter.

Key properties:

- ***The standard deviation is the square root of the mean.***
- ***For mean > 5, well approximated by a normal distribution***



Normal Approximation



Can estimate Poisson distribution as a normal distribution when λ > 10

K-mers and K-mer counting

GATTACATACACATTGGATG

K-mers and K-mer counting

GATTACATACACACATTGGATG

GAT ACA ACA ATT GAT

ATT CAT CAC TTG ATG

TTA ATA ACA TGG

TAC TAC CAT GGA

Kmers:

- Divide a string into substrings of length k
- Notice every position is covered k times
- Notice there are G – k + 1 kmers from a string of length G

K-mers and K-mer counting

GATTACATACACATTGGATG

GAT ACA ACA ATT GAT

ATT CAT CAC TTG ATG

TTA ATA ACA TGG

TAC TAC CAT GGA

GAT:2 CAT:2 ATG:1 TGG:1

ACA:3 CAC:1 TTA:1 TAC:2

ATT:2 TTG:1 ATA:1 GGA:1

K-mers and K-mer counting

GATTACATACACATTGGATG

GAT:2 CAT:2 ATG:1 TGG:1
ACA:3 CAC:1 TTA:1 TAC:2
ATT:2 TTG:1 ATA:1 GGA:1

1: 7 (ATG, TGG, …)
2: 4 (GAT, CAT, ATT, TAC)
3: 1 (ACA)

K-mers and K-mer counting

GATTACATACACATTGGATG

```
1: 7 (ATG, TGG, …)
2: 4 (GAT, CAT, ATT, TAC)
3: 1 (ACA)
```

How long should k be?

K-mers and K-mer counting

GATTACATACACATTGGATG

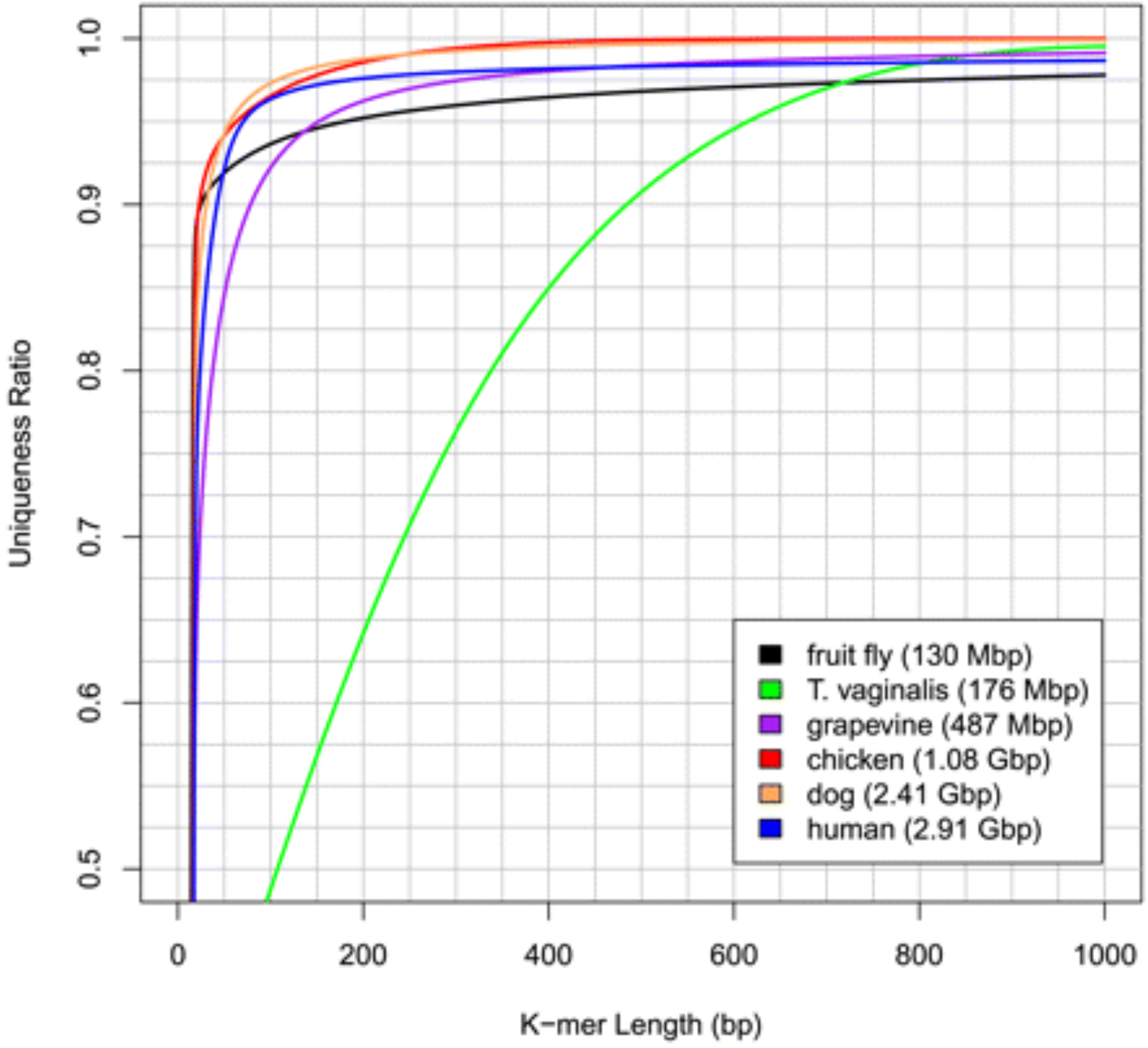
```
1: 7 (ATG, TGG, …)
2: 4 (GAT, CAT, ATT, TAC)
3: 1 (ACA)
```

How long should k be?

K=1 : Too short, every base is present
K=2 : Too short, every pair of bases will be present

Pick k so that G/(4^k) << 1
k = log_4 (G)
At least 15 for human, often a bit longer
But not too long or could loose resolution

K-mer Uniqueness



Assembly of large genomes using second-generation sequencing
Schatz et al. (2010) Genome Research. doi: 10.1101/gr.101360.109