Principles and Applications of Modern DNA Sequencing

EEEB GU4055

Session 10: Genome Assembly

Today's topics

1. Continued: long read technologies.
2. New: Genome assembly
3. Assignment: Kmers and graphs

Review of course topics

1. Intro to Jupyter/Python and history of genomics.
2. Python bootcamp I and genome structure.
3. Python bootcamp II and genome annotation.
4. Scientific Python and Homology.
5. Scientific Python and APIs/BLAST.
6. Recombination and Meiosis.
7. Inheritance and pedigrees.
8. Intro to Illumina and read mapping.
9. Intro to long-read technologies and read mapping.
10. **Intro to Genome Assembly: Kmers and graphs**
11. **Genome Assembly: Hands-on.**
12. **The Coalescent and Genetic Diversity.**
13. **Phylogenetics and Phylogenomics.**
14. **Phylogenomics Continued and Midterm Review.**

Where we left off: API queries

```
# search term
term = "FOXP2[GENE] AND Mammalia[ORGN] AND phylogenetic study[PROP]"

# make a request to esearch
res = requests.get(
    url="https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi",
    params={
        "db": "nucleotide",
        "term": term,
        "sort": "Organism Name",
        "retmode": "text",
        "retmax": "20",
        "tool": "genomics-course",
        "email": "student@columbia.edu",
    },
)
```

Where we left off: API queries

```
# the request object generates a URL to query the API
print(res.url)
```

```
'https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=nucleotide&term=F
```

Data regarding the SARS-CoV-2 (Severe acute respiratory syndrome coronavirus 2, 2019-nCoV) outbreak sequences can be found in GenBank/SRA, the NCBI Virus resource, and a specialized BLAST page that searches Betacoronavirus sequences.

Last

GenBank Overview

What is GenBank?

GenBank ® is the NIH genetic sequence database, an annotated collection of all publicly available DNA sequences (*Nucleic Acids Research*, 2013 Jan;41(D1):D36-42). GenBank is part of the International Nucleotide Sequence Database Collaboration, which comprises the DNA DataBank of Japan (DDBJ), the European Nucleotide Archive (ENA), and GenBank at NCBI. These three organizations exchange data on a daily basis.

A GenBank release occurs every two months and is available from the ftp site. The release notes for the current version of GenBank provide detailed information about the release and notifications of upcoming changes to GenBank. Release notes for previous GenBank releases are also available. GenBank growth statistics for both the traditional GenBank divisions and the WGS division are available from each release. GenBank growth statistics for both the traditional GenBank divisions and the WGS division are available from each release.

An annotated sample GenBank record for a *Saccharomyces cerevisiae* gene demonstrates many of the features of the GenBank flat file format.

Access to GenBank

There are several ways to search and retrieve data from GenBank.

- Search GenBank for sequence identifiers and annotations with Entrez Nucleotide.
- Search and align GenBank sequences to a query sequence using BLAST (Basic Local Alignment Search Tool). BLAST searches CoreNucleotide, dbEST, and dbGSS independently; see BLAST info for more information about the numerous BLAST databases.
- Search, link, and download sequences programatically using NCBI e-utilities.
- The ASN.1 and flatfile formats are available at NCBI's anonymous FTP server: ftp://ftp.ncbi.nlm.nih.gov/ncbi-asn1 and ftp://ftp.ncbi.nlm.nih.gov/genbank.

GenBank Data Usage

The GenBank database is designed to provide and encourage access within the scientific community to the most up-to-date and comprehensive DNA sequence information. Therefore, NCBI places no restrictions on the use or distribution of the GenBank data. However, some submitters may claim patent, copyright, or other intellectual property rights in all or a portion of the data they have submitted. NCBI is not in a position to assess the validity of such claims, and therefore cannot provide comment or unrestricted permission concerning the use, copying, or distribution of the information contained in GenBank.

Confidentiality

Some authors are concerned that the appearance of their data in GenBank prior to publication will compromise their work. GenBank will, upon request, withhold release of new submissions for a specified period of time. However, if the accession number or sequence data appears in print or online prior to the specified date, your sequence will be released. In order to prevent the delay in the appearance of published sequence data, we urge authors to inform us of the appearance of the published data. As soon as it is available, please send the full publication data--all authors, title, journal, volume, pages and date--to the following address: update@ncbi.nlm.nih.gov

Privacy

If you are submitting human sequences to GenBank, do not include any data that could reveal the personal identity of the source. GenBank assumes that the submitter has received any necessary informed consent authorizations required prior to submitting sequences.

Disclaimer

Privacy statement

GenBank Resources

GenBank Home

Submission Types

Submission Tools

Search GenBank

Update GenBank Records

Where we left off: API queries

```
# search term
term = "SARS-CoV-2[ORGN] complete genome"

# make a request to esearch
res = requests.get(
    url="https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi",
    params={
        "db": "nucleotide",
        "term": term,
        "sort": "Organism Name",
        "retmode": "text",
        "retmax": "20",
        "tool": "genomics-course",
        "email": "student@columbia.edu",
    },
)
```

Download data from NCBI: Coronavirus

```
# parse the fasta data and print only headers
fna = [i for i in fastas.strip().split("\n\n")]
for seq in fna:
    print(seq.split("\n")[0][:90], '...')
```

```
>MN908947.3 Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, cor
>NC_045512.2 Wuhan seafood market pneumonia virus isolate Wuhan-Hu-1, complete geno
>MN938384.1 Severe acute respiratory syndrome coronavirus 2 isolate 2019-nCoV_HKU-S
>MN975262.1 Severe acute respiratory syndrome coronavirus 2 isolate 2019-nCoV_HKU-S
>MN985325.1 Severe acute respiratory syndrome coronavirus 2 isolate 2019-nCoV/USA-W
>MN988668.1 Severe acute respiratory syndrome coronavirus 2 isolate 2019-nCoV WHU01
>MN988669.1 Severe acute respiratory syndrome coronavirus 2 isolate 2019-nCoV WHU02
>MN988713.1 Severe acute respiratory syndrome coronavirus 2 isolate 2019-nCoV/USA-I
...
```

Long Read Technologies

PacBio and **Oxford Nanopore** currently offer two distinct technologies for generating long sequenced reads that are now widely used in genomics, particularly for the assembly of reference genomes, but also for other purposes as well.

Your last assignment and reading introduced you to long read data.

Long Read Technologies

PacBio has greater accuracy but is currently has an upper limit on read lengths (~20Kb). Nanopore reads have more errors but can provide contiguous information up to the size of physical DNA fragments (e.g., >1Mb).

Long Read Technologies

PacBio has greater accuracy but is currently has an upper limit on read lengths (~20Kb). Nanopore reads have more errors but can provide contiguous information up to the size of physical DNA fragments (e.g., >1Mb).

RESEARCH ARTICLE SUMMARY

NONHUMAN GENOMICS

Long-read sequence assembly of the gorilla genome

David Gordon,* John Huddleston,* Mark J. P. Chaisson,* Christopher M. Hill,* Zev N. Kronenberg,* Katherine M. Munson, Maika Malig, Archana Raja, Ian Fiddes, LaDeana W. Hillier, Christopher Dunn, Carl Baker, Joel Armstrong, Mark Diekhans, Benedict Paten, Jay Shendure, Richard K. Wilson, David Haussler, Chen-Shan Chin, Evan E. Eichler†

INTRODUCTION: The accurate sequence and assembly of genomes is critical to our understanding of evolution and genetic variation. Despite advances in short-read sequencing technology that have decreased cost and increased throughput, whole-genome assembly of mammalian genomes remains problematic because of the presence of repetitive DNA.

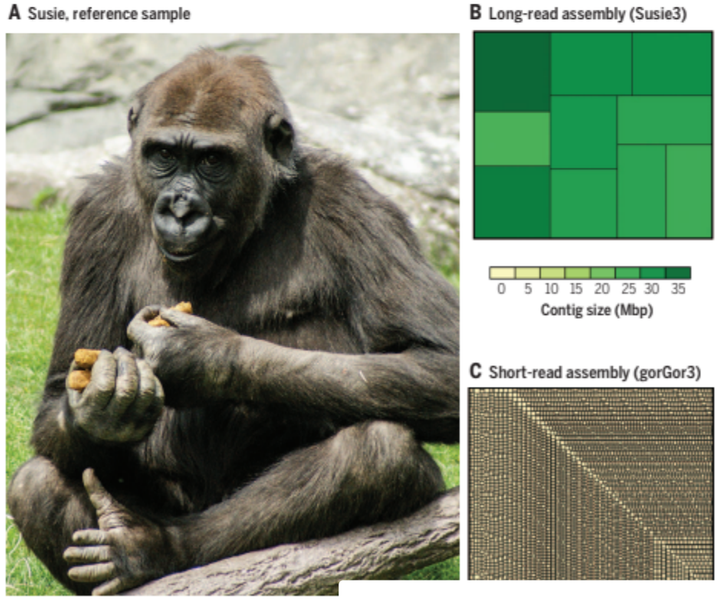
RATIONALE: The goal of this study was to sequence and assemble the genome of the western lowland gorilla by using primarily single-molecule, real-time (SMRT) sequencing technology and a novel assembly algorithm that takes advantage of long (>10 kbp) sequence reads. We specifically compare the properties of this assembly to gorilla genome assemblies that were generated by using more routine short sequence read approaches in order to determine the value and biological impact of a long-read genome assembly.

RESULTS: We generated 74.8-fold SMRT whole-genome shotgun sequence from peripheral blood DNA isolated from a western lowland gorilla (*Gorilla gorilla gorilla*) named Susie. We applied a string graph assembly algorithm, Falcon, and consensus algorithm, Quiver, to generate a 3.1-Gbp assembly with a contig N50 of 9.6 Mbp. Short-read sequence data from an additional six gorilla genomes was mapped so as to reduce indel errors and improve the accuracy of the final assembly. We estimate that 98.9% of the gorilla euchromatin has been assembled into 1854 sequence contigs. The assembly represents an improvement in contiguity: >800-fold with respect to the published gorilla genome assembly and >180-fold with respect to a more recently released upgrade of the gorilla assembly. Most of the sequence gaps are now closed, considerably increasing the yield of complete gene models. We estimate that 87% of the missing exons and 94% of the incomplete genes are recovered. We find that the sequence of most full-length common repeats is resolved, with the most significant gains occurring for the longest and most G+C–rich retrotransposons. Although complex regions such as the major histocompatibility locus are accurately sequenced and assembled, both heterochromatin and large, high-identity segmental duplications are not because read lengths are insufficiently long to traverse these repetitive structures. The long-read assembly produces a much finer map of structural variation down to 50 bp in length, facilitating the discovery of thousands of lineage-specific structural variant differences that have occurred since divergence from the human and chimpanzee lineages. This includes the disruption of specific genes and loss of predicted regulatory regions between the two species. We show that use of the new gorilla genome assembly changes estimates of divergence and diversity, resulting in subtle but substantial effects on previous population genetic inferences, such as the timing of species bottlenecks and changes in the effective population size over the course of evolution.

ON OUR WEBSITE

Read the full article at org/10.1126/science.aae0344

CONCLUSION: The genome assembly that results from using the long-read data provides a more complete picture of gene content, structural variation, and repeat biology, improving population genetic and evolutionary inferences. Long-read sequencing technology now makes it practical for individual laboratories to generate high-quality reference genomes for complex mammalian genomes. ■



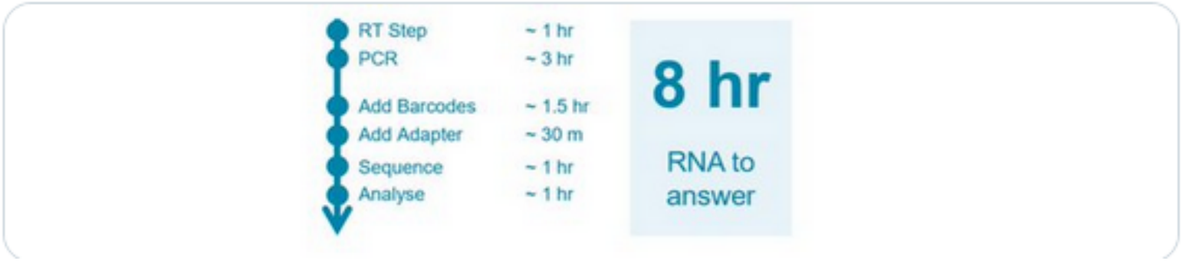
Long-read sequence assembly of the gori gorilla, was used as the reference sample for courtesy of Max Block]. (**B** and **C**) Treemaps representing the differences in fragmentation of the

he list of author affiliations is available in the full article online.
*These authors contributed equally to this work.

Nanopore is fast and portable

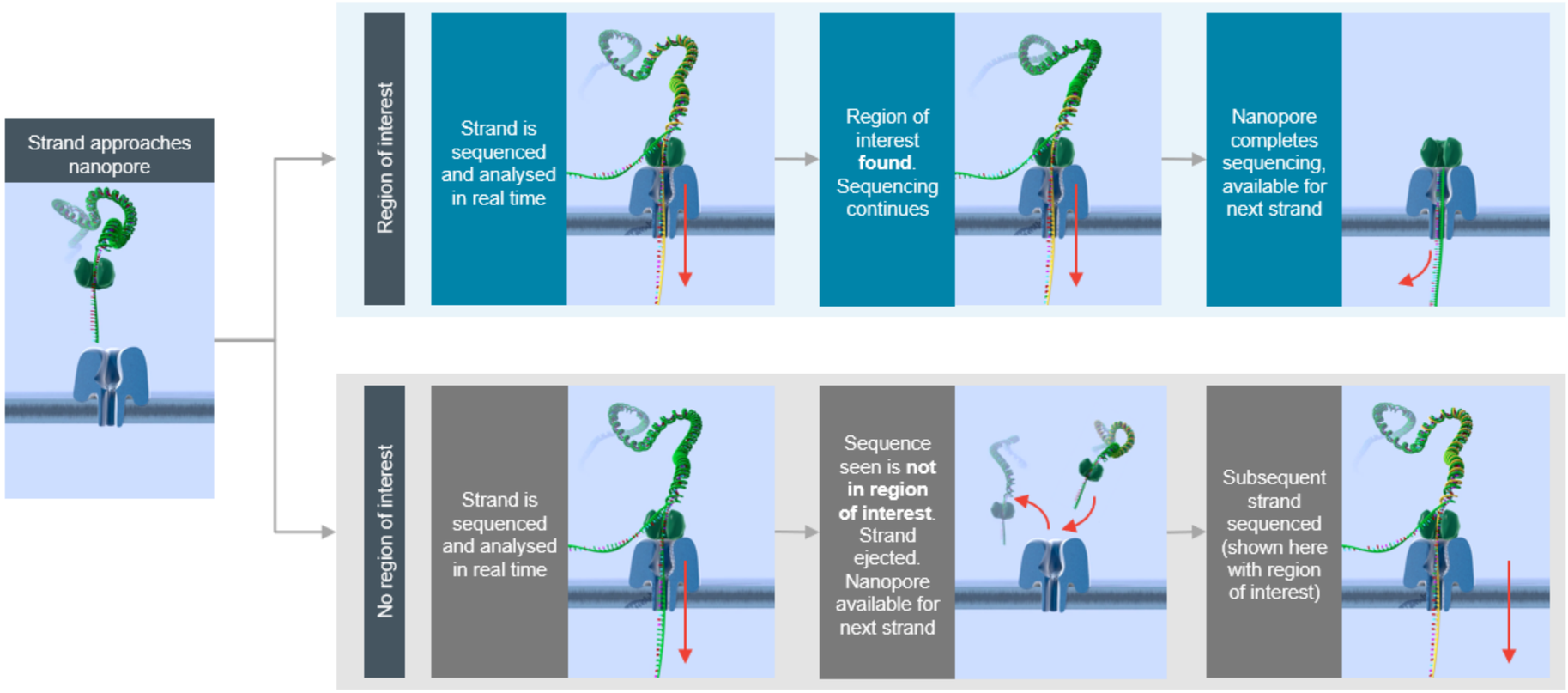
Oxford Nanopore
@nanopore

#COVID19 genomes sequenced using a protocol developed by the @NetworkArtic were released earlier this month. The methods shared by @NetworkArtic enable accurate sample-to-sequence on nanopore tech within 8 hours. Read more & access protocol here: bit.ly/2UKhGDO #ncov2019



Nanopore Read-Until Targeting

Two papers (link) recently described a method for *targeted sequencing* with nanopores. The analysis API rejects DNA fragments that do not match a desired signal (e.g., region from a reference genome) thus *enriching* coverage of the target.



Long read assignment: functions revisited

```
def getLengthDistribution(thisFilePath):
    "Return a list of lengths of each sequence in a fasta file."
    lenList = []
    for record in SeqIO.parse(thisFilePath, "fasta"):
        lenList.append(len(record))
    return lenList


def makeLengthPlot(thisRunName, ax):
    "Plot the length plot for the raw sequence data"

    # set histogram bin size
    bins = range(500, 5000, 100)

    # plots a matplotlib histogram onto 'ax' axes
    sns.distplot(
        getLengthDistribution(file_path[thisRunName]),
```

SeqIO

Loading fasta files as SeqIO record objects.

```
# The SeqIO module is useful for working with Fasta files
from Bio import SeqIO

# load a Fasta file from a path with Bio
record = SeqIO.parse(path, format="fasta")

# the record object makes the seq data accessible (e.g. length)
len(record)
```

```
600
```

A dictionary to access long file names easier.

```
# Global dictionary of technology name to file paths.
FILE_PATH = {
    'Sanger': 'files/sanger.total.aftertrim.removeCT.min500bp.fasta',
    'PacBio': 'files/PB.Cell1and2.raw.fasta',
    'Nanopore': 'files/LejlaControl.2D.min500bp.fasta',
}

# Global dictionary of technology name to colors for plotting
COLORS = {
    'Sanger': '#4daf4a',
    'PacBio': '#377eb8',
    'Nanopore': '#984ea3',
}
```

A function to plot a histogram with matplotlib.

```
def makeLengthPlot(thisRunName, ax):
    "Plot histogram of read lengths. Gets filepath from global dict"
    # get readlengths
    filepath = FILE_PATH[thisRunName])          # <--- global
    lengths = getLengthDistribution(filepath)

    # make histogram
    sns.distplot(
        length,
        ax=ax,
        bins=range(500, 5000, 100),
        kde=False,
        color=COLORS[thisRunName],              # <--- global
        label=thisRunName,
        hist_kws={"alpha": 1})

    # plot styling: legend, xlimit, ylabel, ylimit
    ax.legend()
    ax.set_xlim([1, 5000])
    ax.set_ylabel('number of {} reads'.format(thisRunName))
    ax.set_ylim([0, 14000])

    # make unique y-axis label for each run
    if thisRunName == 'Sanger':
        ax.set_ylim([0, 800])
```

A function to plot a histogram with matplotlib.

```
# initalize plot with 3 rows and 1 column
fig, ax = plt.subplots(nrows=3, ncols=1, figsize=(8, 20))

# Plot Sanger, Pacbio, and Nanopore read length distributions
makeLengthPlot('Sanger', ax[0])
makeLengthPlot('PacBio', ax[1])
makeLengthPlot('Nanopore', ax[2])

# add label to x axis
ax[2].set_xlabel('size [bp]')

# Show the plot
plt.show()
```

Challenge 4 (2 points): Print the top ten longest reads and average read length:

```
# 1. get filepath for Nanopore data set
npath = file_path["Nanopore"]

# 2. call getLengthDistribution on this file
readlens = getLengthDistribution(npath)

# 3. get top ten longest reads
topten = sorted(readlens)[::-1][:10]
print(topten)

# 4. get average length (of all reads)
avglen = sum(readlens) / len(readlens)
print(avglen)
```

```
[298549, 108360, 72322, 67205, 61366, 60592, 45980, 45605, 41057, 39953]
1512.50977897917
```

Oxford nanopore sequence mapping

```
# align nanopore reads (reads.fasta) to reference (ref.fa)
minimap2 -ax map-ont \
    /home/codio/workspace/files/ref.fa \
    /home/codio/workspace/files/reads.fasta \
    > /home/codio/workspace/files/aligned.sam
```

```
[M::mm_idx_gen::0.187*0.88] collected minimizers
[M::mm_idx_gen::0.220*1.19] sorted minimizers
[M::main::0.221*1.19] loaded/built the index for 4319 target sequence(s)
[M::mm_mapopt_update::0.232*1.18] mid_occ = 11
[M::mm_idx_stat] kmer size: 15; skip: 10; is_hpc: 0; #seq: 4319
[M::mm_idx_stat::0.241*1.17] distinct minimizers: 731433 (98.40% are singletons); a
[M::worker_pipeline::0.373*1.78] mapped 112 sequences
[M::main] Version: 2.15-r915-dirty
[M::main] CMD: ./minimap2/minimap2 -ax map-ont /home/codio/workspace/files/ref.fa /
[M::main] Real time: 0.381 sec; CPU: 0.671 sec; Peak RSS: 0.053 GB
```

Challenge 8 (2 points): Look at the last alignment in the SAM file, how many mismatches or gaps?

```
%%bash
# show the last 1 line
tail -n 1 files/aligned.sam
```

```
4b09492e-1e14-4c2a-9719-4cd4f1434703    2048    gi|545778205|gb|U00096.3|:c728821-
```

Challenge 8 (2 points): Look at the last alignment in the SAM file, how many mismatches or gaps?

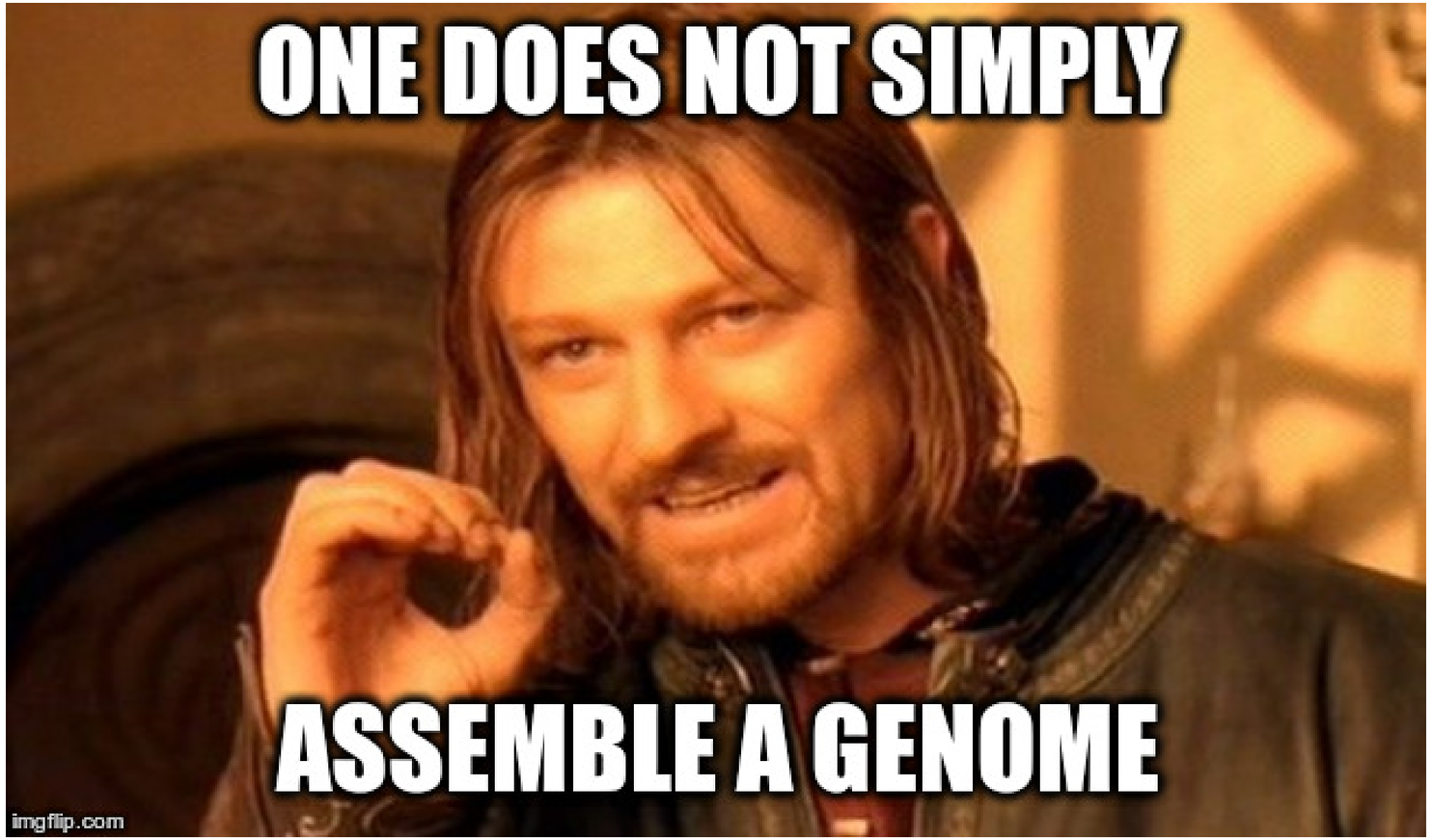
```
%%bash
# show the last 5 lines, and only the 12th delimited column
tail -n 5 files/aligned.sam | cut -f 12
```

```
NM:i:164
NM:i:171
NM:i:109
NM:i:21
NM:i:9
```

Challenge 9: Make a sorted BAM file from the SAM.

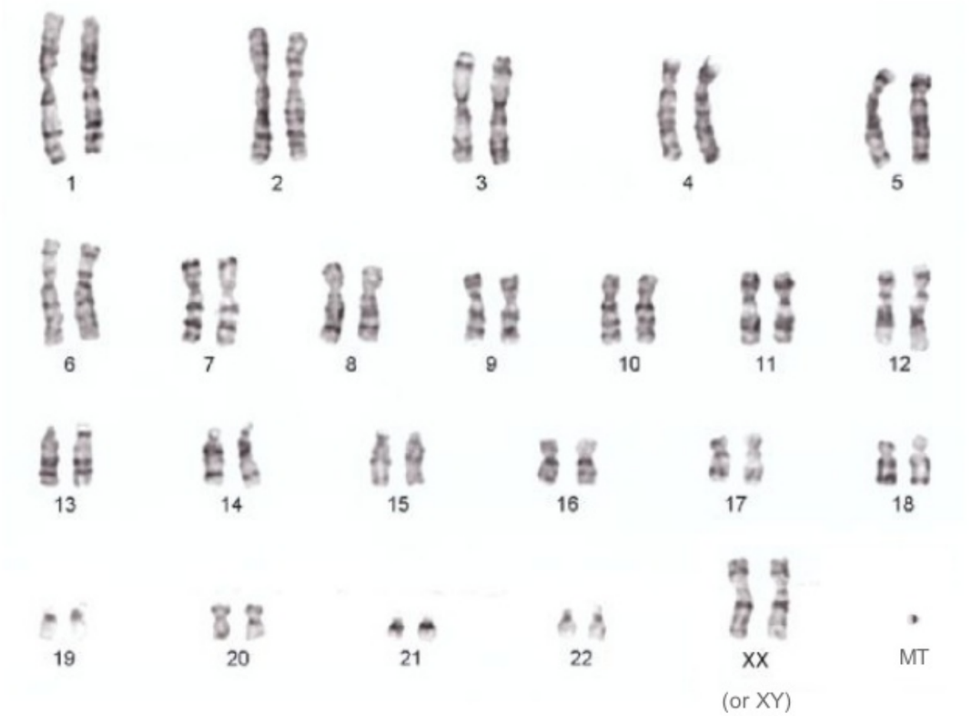
```
%%bash
# view as binary and direct to file (SAM -> BAM)
samtools view -b aligned.sam > aligned.bam

# sort and direct to file (BAM -> SORTED.BAM)
samtools sort aligned.bam > aligned.sorted.bam

# create index file (BAM -> BAM.BAI)
samtools index aligned.sorted.bam

# what are the file sizes?
du aligned.sam
du aligned.bam
```

Genome Assembly



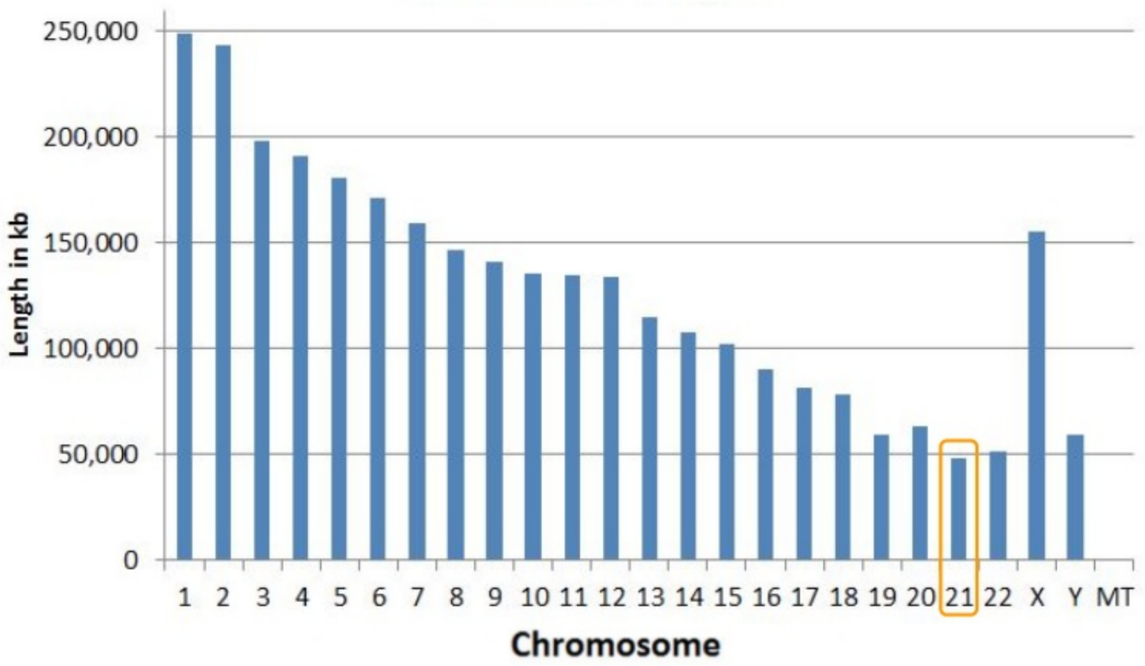
Genome Assembly

denovo genome assembly is computationally demanding. Requires reads that cover the full genome many times (e.g., 50X). The end goal is to assemble scaffolds that match to chromosomes -- the real *bits* of the genome.



Genome Assembly

Even the smallest chromosome of the human genome is 48Mbp!



Total haploid size

3,200,000,000 bp
(3.2 Gbp)

Total diploid size

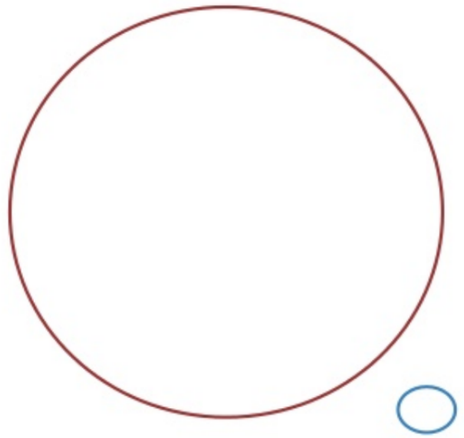
6,400,000,000 bp
(6.4 Gbp)

$\sum$ = 6,400,000,000 bp

Genome Assembly

Sometimes intermediate/draft genomes are good enough to answer many questions. For many tasks, though, they are not. e.g., genome annotation. Trade off in costs and time.

250 bp - Illumina - $250



12,000 bp - Pacbio - $2500

Why do we need (complete) reference genomes?

(1) To study genome structural variation; (2) FAST mapping of sequenced reads to the reference to study variation in sequence or abundance (e.g., RNA); (3) Spatial genetic information is useful for association studies (e.g., GWAS; mapping traits) based on how variants segregate among offspring (i.e., genetic linkage);