GENOME 569

Class 2: Single-cell RNA-seq & NGS read alignment

What is a bioinformatics workflow?

A bioinformatics workflow is the collection of **scripts**, **programs**, and **procedures** you use to transform your data into a paper.

Goals for the course

Learn how to write reproducible workflows to analyze your data.

Learn how to use some awesome software tools.

Become a better programmer.

A bit about me

- **High-school**: worked as student software engineer for US Army
- **College**: software engineer for stock, futures, & foreign currency broker
- **After college**: software engineer for a “Business intelligence” company
- **Grad school**: Computer science (supercomputing, then bioinformatics)
- **Postdoc**: RNA biology, stem cell biology, genomics tech dev
- **Now**: single-cell genomics tech dev, developmental bio

A bit about you?

What you will learn in this class

- How to automate tedious and computationally intensive jobs with Snakemake and UNIX tools
- How to explore a dataset with R
- How to make your data analysis and figure generation reproducible
- How to use AI coding tools to accelerate your work and learn about new tools

The Classes

- Before each class, I will give you a lot of stuff to read, but it will be fun
- I will present some slides on a couple of really awesome programming tools each class.
- Then you will do some in class exercises with them - these will build up a bioinformatics workflow over time.

The Projects

1. A pipeline to process raw single-cell RNA-seq reads
2. An electronic lab notebook that encapsulates your exploration of the dataset

A typical scenario in grad school

C. elegans

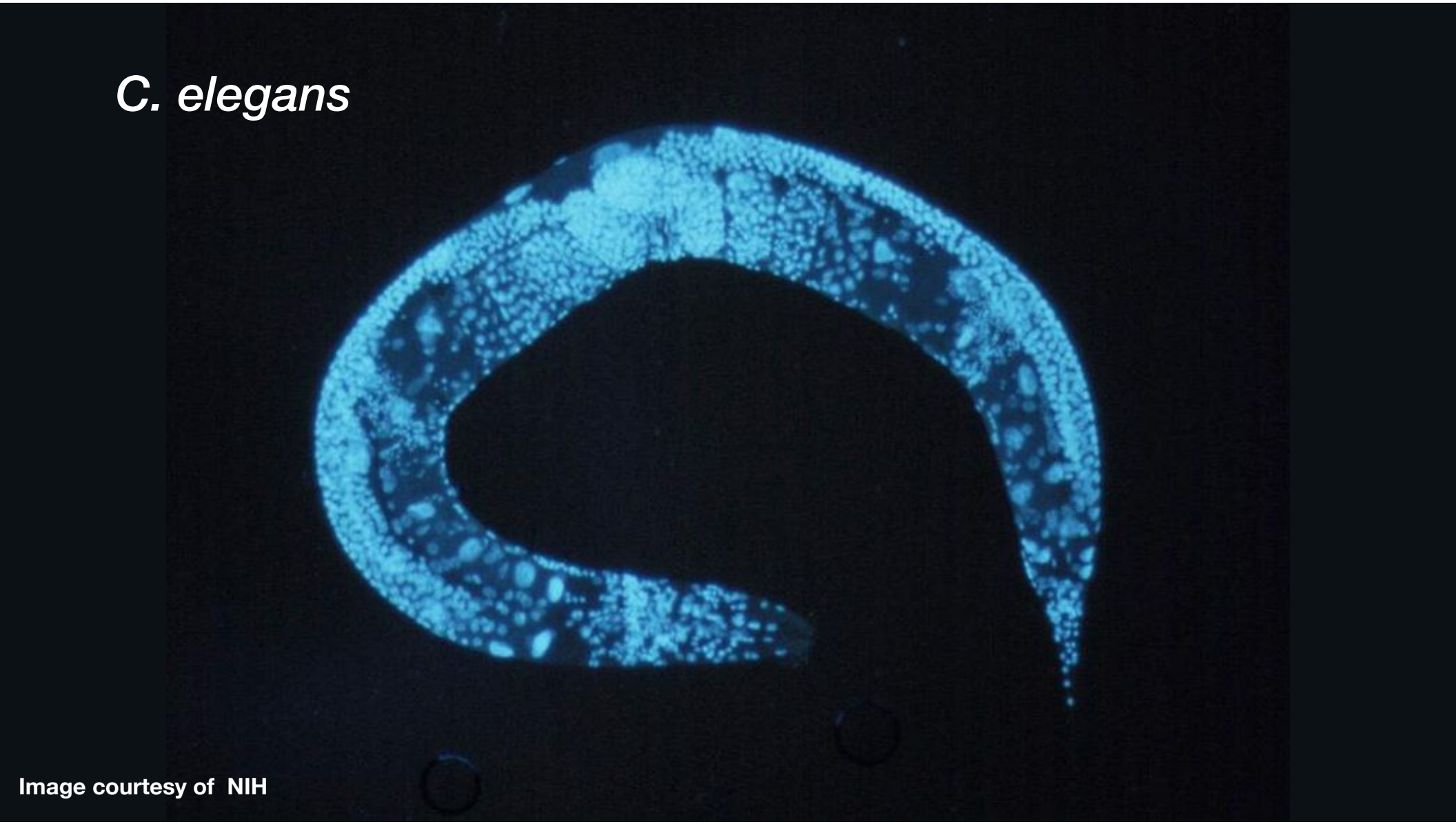
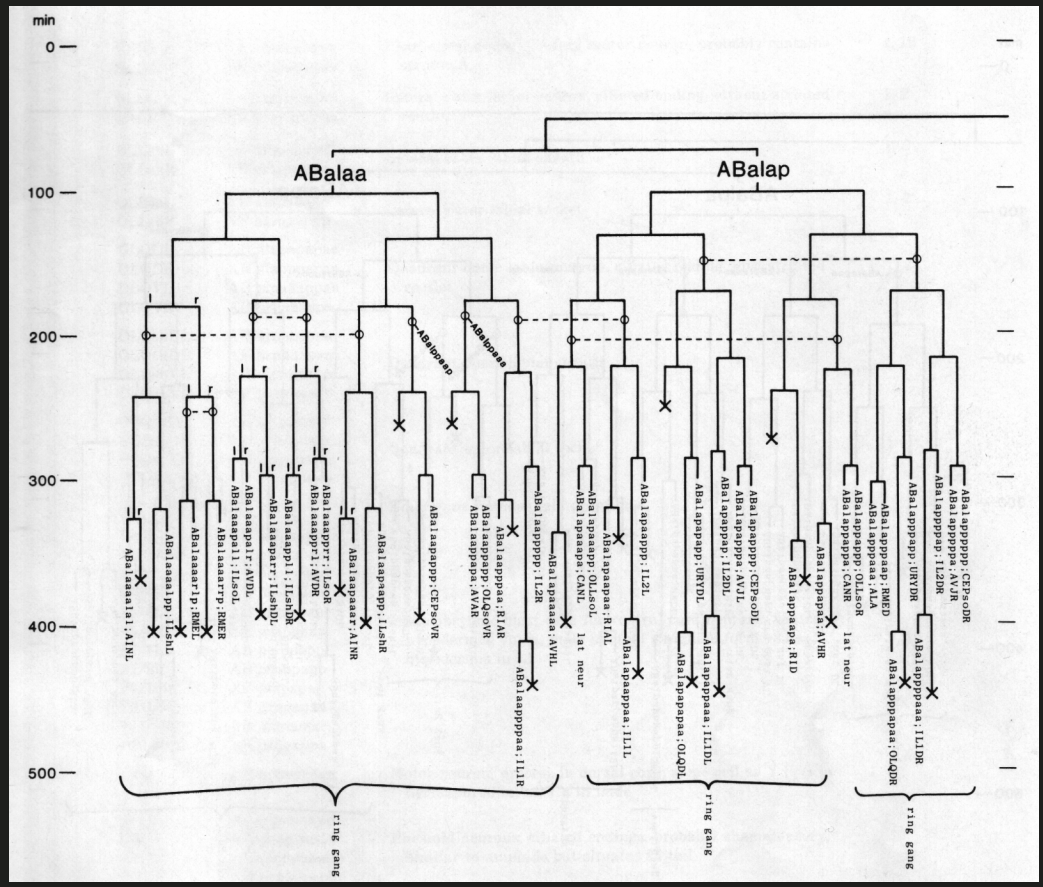


Image courtesy of NIH

The worm has an invariant cell lineage



Science

RESEARCH ARTICLES

Cite as: J. S. Packer *et al.*, *Science*
10.1126/science.aax1971 (2019).

A lineage-resolved molecular atlas of *C. elegans* embryogenesis at single-cell resolution

Jonathan S. Packer[1]*, Qin Zhu[2]*, Chau Huynh[1], Priya Sivaramakrishnan[3], Elicia Preston[3], Hannah Dueck[3]†, Derek Stefanik[4], Kai Tan[3,5,6,7], Cole Trapnell[1], Junhyong Kim[4]‡, Robert H. Waterston[1]‡, John I. Murray[3]‡

[1]Department of Genome Sciences, University of Washington, Seattle, WA, USA. [2]Genomics and Computational Biology Graduate Group, University of Pennsylvania, Philadelphia, PA, USA. [3]Department of Genetics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA. [4]Department of Biology, University of Pennsylvania, Philadelphia, PA, USA. [5]Division of Oncology and Center for Childhood Cancer Research, Children's Hospital of Philadelphia, Philadelphia, PA, USA. [6]Department of Pediatrics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA. [7]Department of Cell and Developmental Biology, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA.

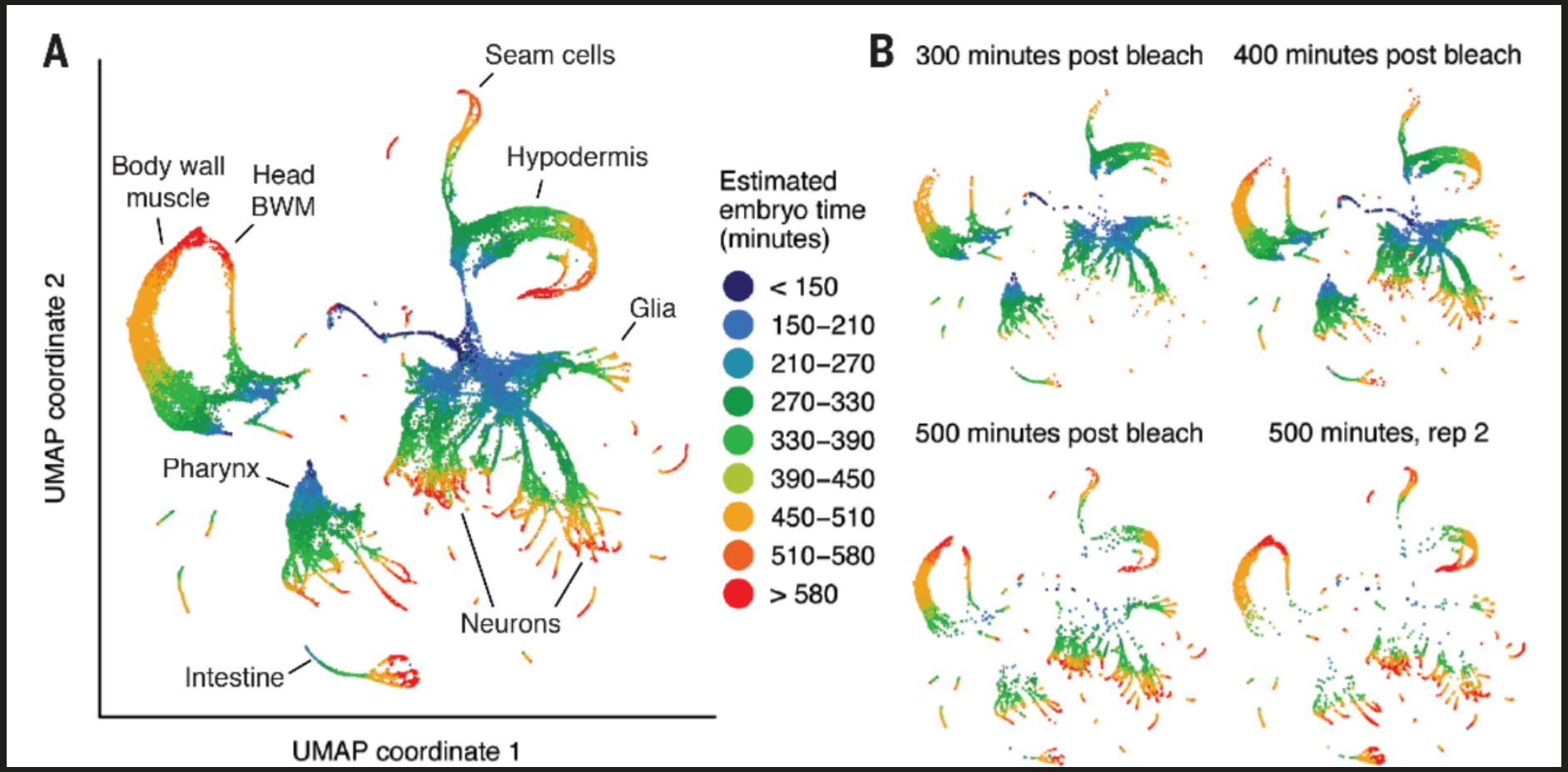
*These authors contributed equally to this work.

†Present address: National Cancer Institute, Bethesda, MD, USA.

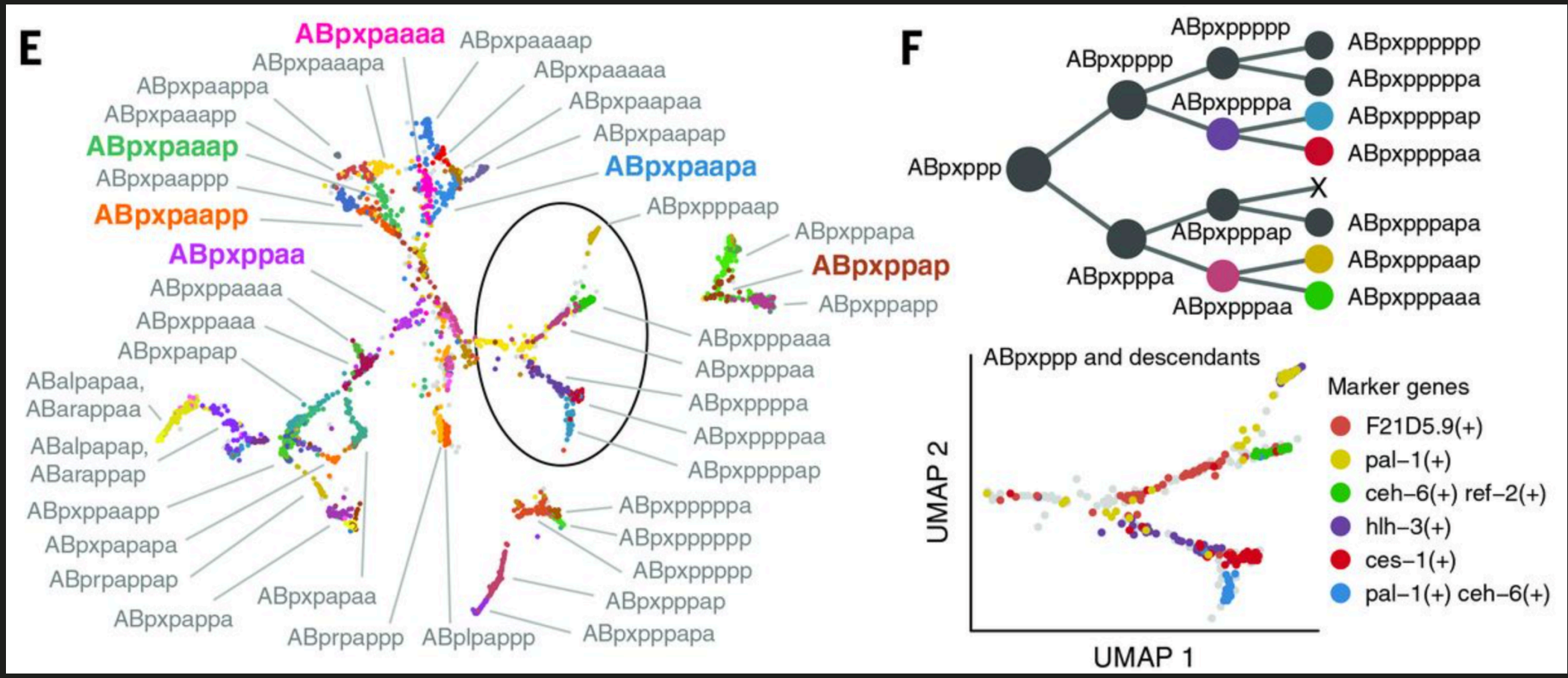
‡Corresponding author. Email: junhyong@sas.upenn.edu (J.K.); watersto@uw.edu (R.H.W.); jmurr@pennmedicine.upenn.edu (J.I.M.)

Caenorhabditis elegans **is an animal with few cells, but a striking diversity of cell types. Here, we characterize the molecular basis for their specification by profiling the transcriptomes of 86,024 single embryonic cells. We identify 502 terminal and pre-terminal cell types, mapping most single-cell transcriptomes to their exact position in *C. elegans*' invariant lineage. Using these annotations, we find that: 1) the correlation between a cell's lineage and its transcriptome increases from mid to late gastrulation, then falls dramatically as cells in the nervous system and pharynx adopt their terminal fates; 2) multilineage priming contributes to the differentiation of sister cells at dozens of lineage branches; and 3) most distinct lineages that produce the same anatomical cell type converge to a homogenous transcriptomic state.**

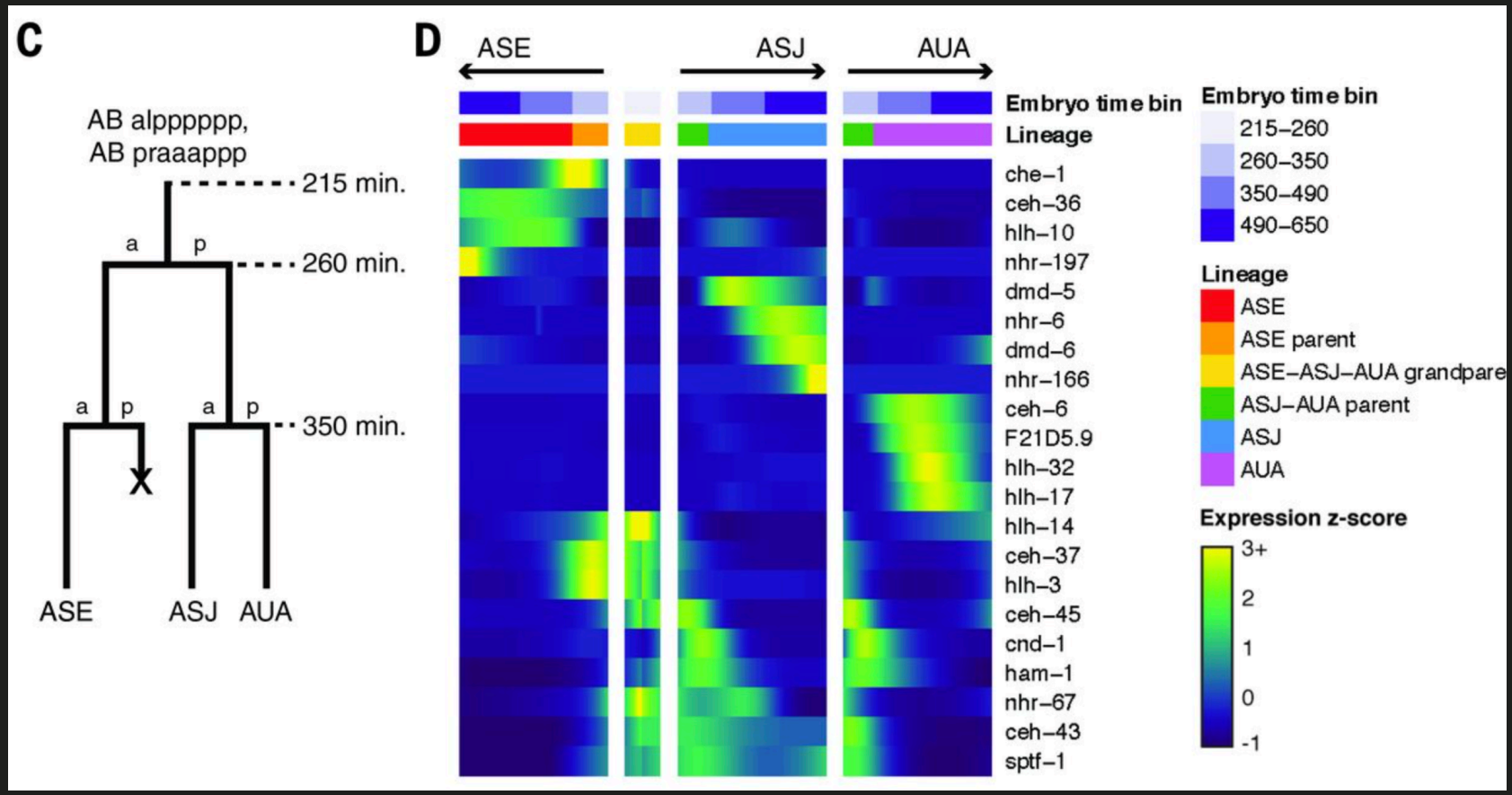
A single-cell atlas of the developing worm embryo



Single-cell RNA-seq captures fate decisions



Multi-lineage priming is pervasive in the worm



“The lineage of the ASE, ASJ, and AUA neurons (spanning embryo time ~215 to 650 min) serves as a representative example (Fig. 3C). Approximately three or four TFs are specific to each terminal neuron type in this lineage (Fig. 3D). Similar numbers of branch-specific TFs were observed for other lineage bifurcations (fig. S22)”

“Multilineage priming events are distributed throughout several generations of both the ectoderm and mesoderm (fig. S24), demonstrating that it is a common and pervasive mechanism of gene regulation.”

- Packer et al

“We need to re-analyze this dataset!!!”
- Your PI

Project Phase I

Clone the starter project repo

Click on the link I post to slack: https://classroom.github.com/a/x_7YyFYs

`coletrapnell-teaching/packer-workflow-starter-<your_gihub_id>.git`

Now check out the README.md

Single-cell RNA-seq

Droplet-based Single-cell RNA-seq (10X)

ARTICLE

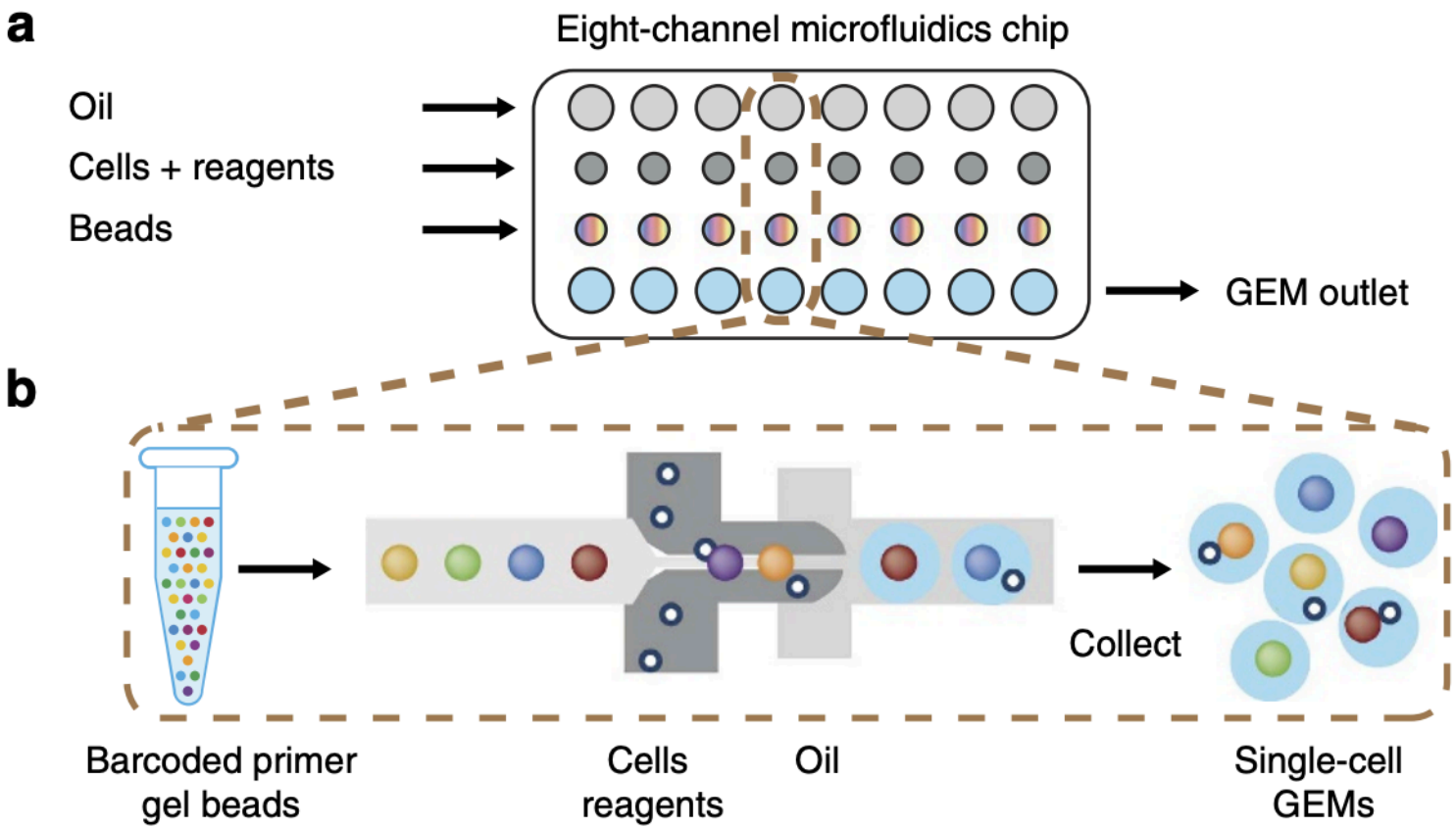
Received 20 Sep 2016 | Accepted 23 Nov 2016 | Published 16 Jan 2017

DOI: 10.1038/ncomms14049 OPEN

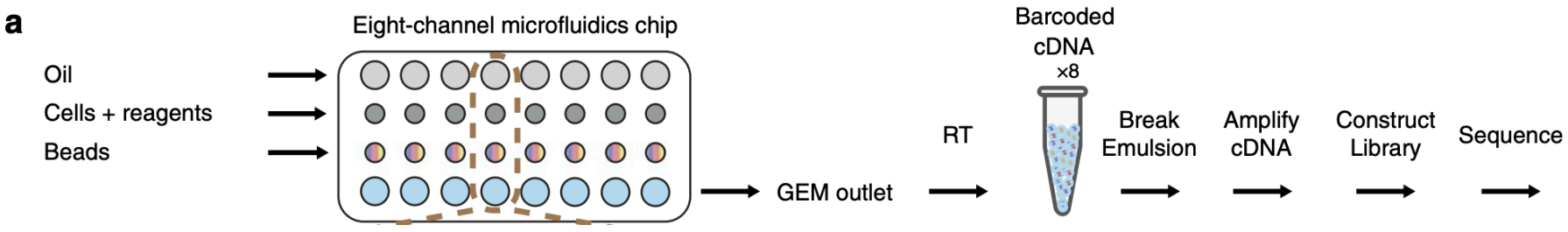
Massively parallel digital transcriptional profiling of single cells

Grace X.Y. Zheng[1], Jessica M. Terry[1], Phillip Belgrader[1], Paul Ryvkin[1], Zachary W. Bent[1], Ryan Wilson[1], Solongo B. Ziraldo[1], Tobias D. Wheeler[1], Geoff P. McDermott[1], Junjie Zhu[1], Mark T. Gregory[2], Joe Shuga Luz Montesclaros[1], Jason G. Underwood[1,3], Donald A. Masquelier[1], Stefanie Y. Nishimura[1], Michael Schnall-Levin[1], Paul W. Wyatt[1], Christopher M. Hindson[1], Rajiv Bharadwaj[1], Alexander Wong[1], Kevin D. Ness[1], Lan W. Beppu[4], H. Joachim Deeg[4], Christopher McFarland[5], Keith R. Loeb[4,6], William J. Valente[2,7,8], Nolan G. Ericson[2], Emily A. Stevens[4], Jerald P. Radich[4], Tarjei S. Mikkelsen[1], Benjamin J. Hindson[1] & Jason H. Bielas[2,6,8,9]

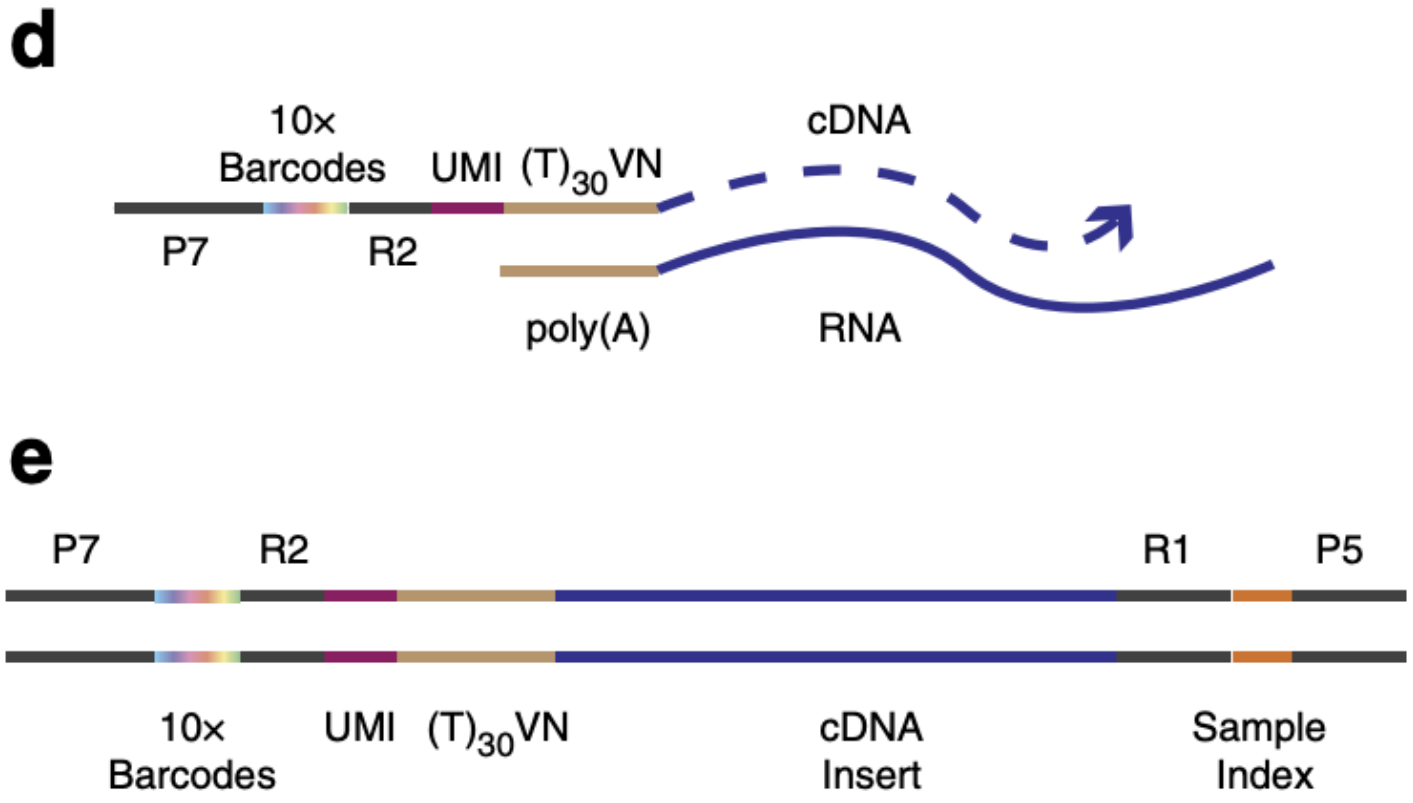
Droplet-based Single-cell RNA-seq (10X)



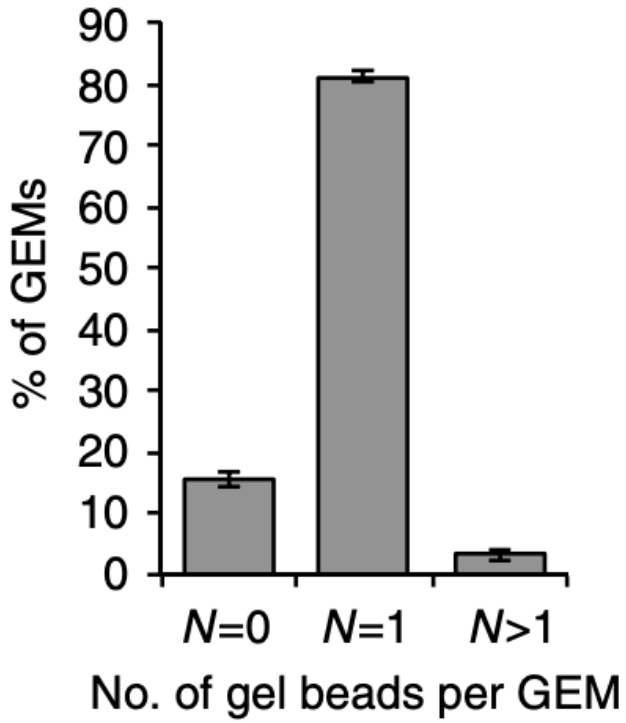
Droplet-based Single-cell RNA-seq (10X)



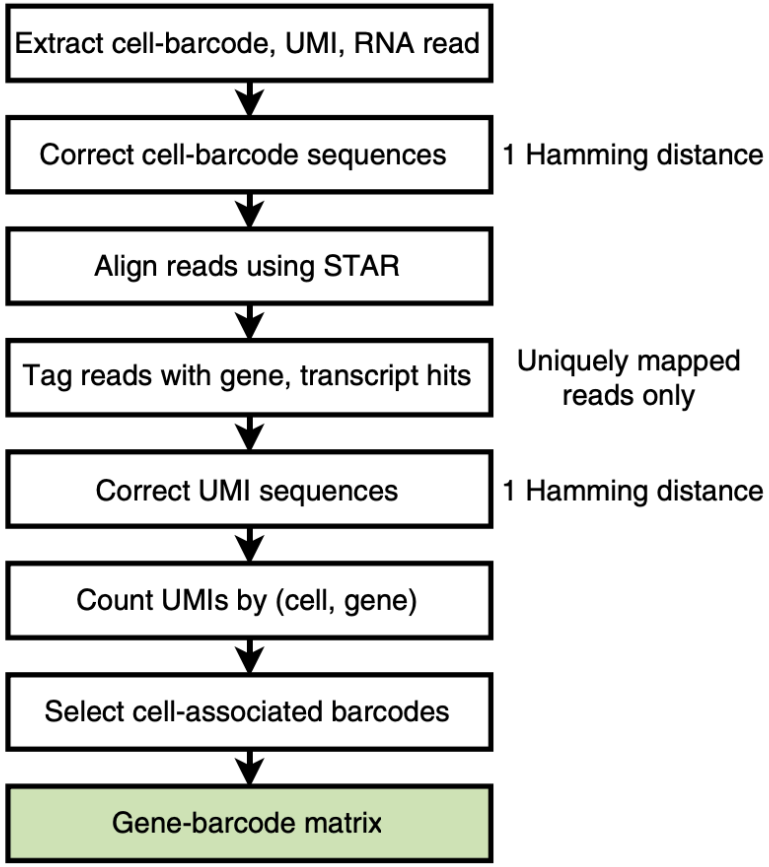
10X RNA-seq read structure (v1)

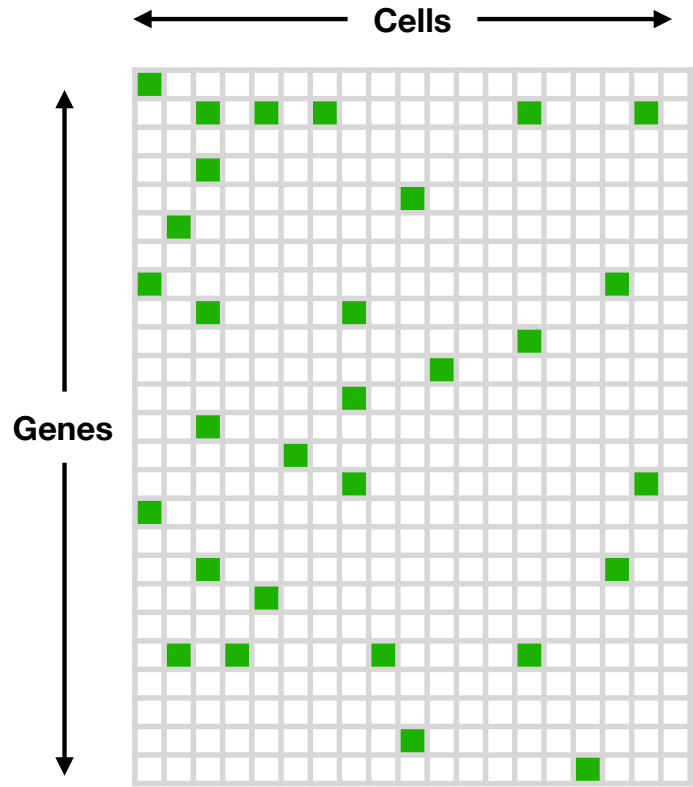


Droplets contain ~1 cell (“Poisson loading”)



The 10X bioinformatics workflow





Most genes detected in few cells - matrix is very sparse!

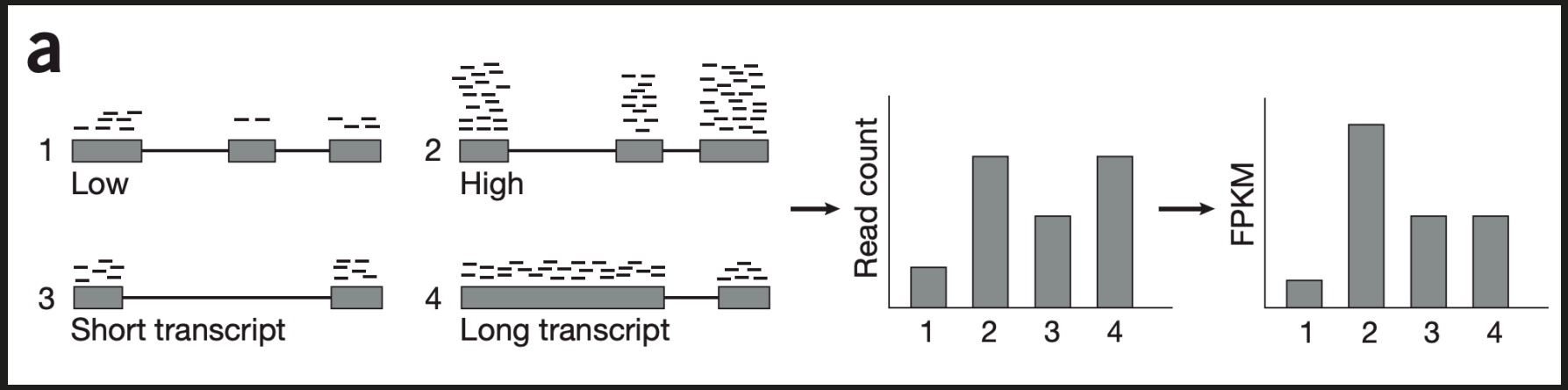
New Results

STARsolo: accurate, fast and versatile mapping/quantification of single-cell and single-nucleus RNA-seq data

Benjamin Kaminow, Dinar Yunusov, Alexander Dobin

doi: https://doi.org/10.1101/2021.05.05.442755

Measuring gene expression with NGS



The number of reads from a transcript is proportional to its abundance.

With random RT primers, you also need to correct for transcript length.

Garber, Grabherr, Guttman, & Trapnell, *Nature Methods* 2011

How to map billions of short reads onto genomes

Cole Trapnell & Steven L Salzberg

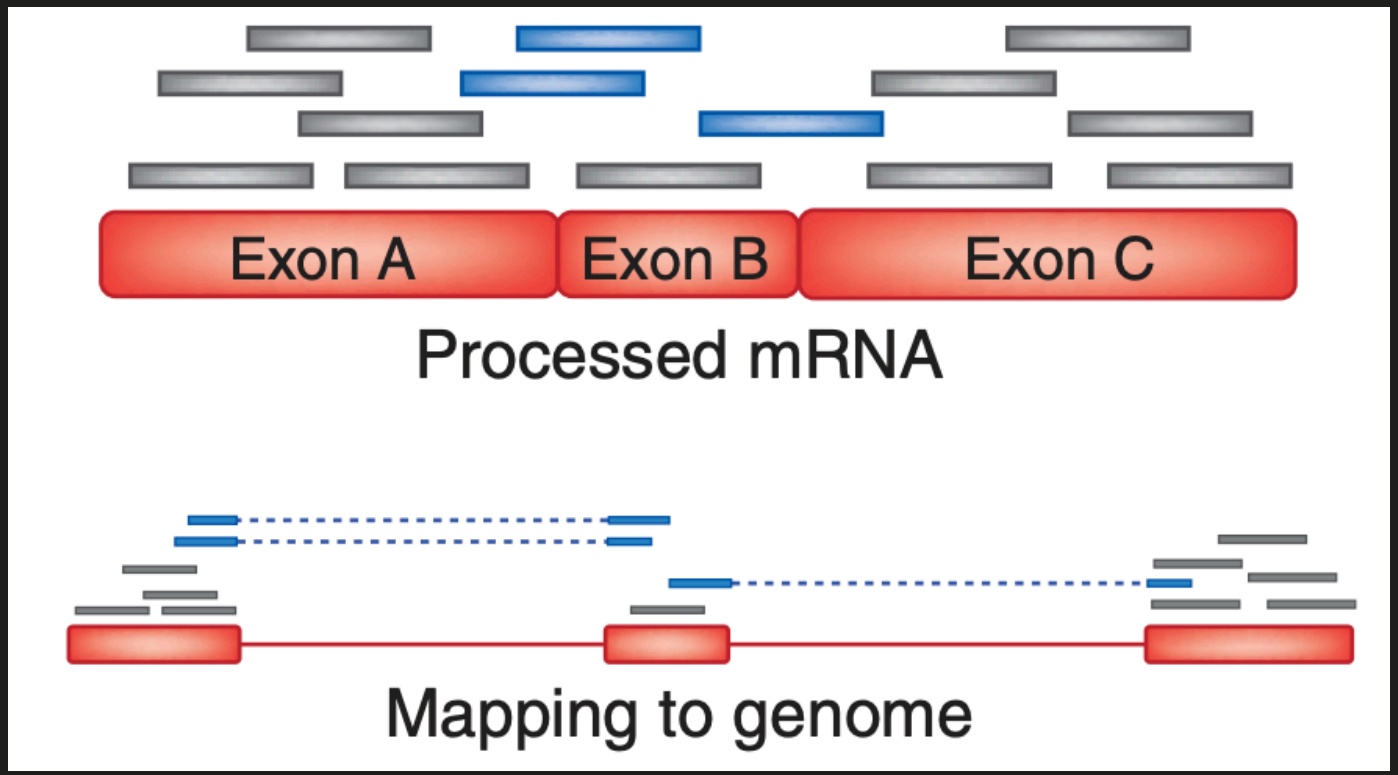
Mapping the vast quantities of short sequence fragments produced by next-generation sequencing platforms is a challenge. What programs are available and how do they work?

A new generation of DNA sequencers that can rapidly and inexpensively sequence billions of bases is transforming genomic science. These new machines are quickly becoming the technology of choice for whole-genome sequencing and for a variety of sequencing-based assays, including gene expression, DNA-protein interaction, human resequencing and RNA splicing studies[1–3]. For example, the RNA-Seq protocol, in which processed mRNA is converted to cDNA and sequenced, is enabling the identification of previously unknown genes and alter-

Table 1 A selection of short-read analysis software

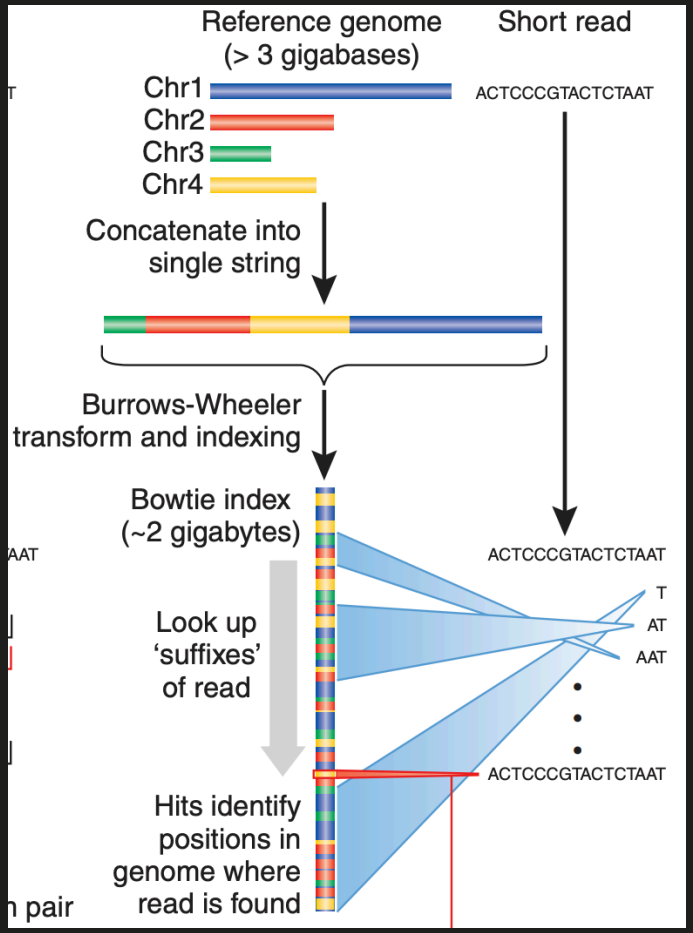
Program	Website	Open source?	Handles ABI color space?	Maximum read length
Bowtie	http://bowtie.cbcb.umd.edu	Yes	No	None
BWA	http://maq.sourceforge.net/bwa-man.shtml	Yes	Yes	None
Maq	http://maq.sourceforge.net	Yes	Yes	127
Mosaik	http://bioinformatics.bc.edu/marthlab/Mosaik	No	Yes	None
Novoalign	http://www.novocraft.com	No	No	None
SOAP2	http://soap.genomics.org.cn	No	No	60
ZOOM	http://www.bioinfor.com	No	Yes	240

Aligning RNA-seq reads



Trapnell & Salzberg, *Nature Biotechnology* 2009

Aligning RNA-seq reads



Software for mapping NGS reads

Tool	DNA/RNA	Mapping strategy
bwa	DNA	Burroughs Wheeler
Bowtie	DNA	Burroughs Wheeler
TopHat	RNA	Burroughs Wheeler + linkage
HISAT	RNA	Burroughs Wheeler
STAR	RNA	Suffix arrays

Many others (some highly specialized), but these are the most popular

STAR

Problem: map NGS reads to a genome

Solution:

```
STAR \
    --runThreadN 8 \
    --genomeDir $INDEX \
    --genomeLoad LoadAndKeep \
    --readFilesIn $INPUT/$FILE \
    --readFilesCommand zcat \
    --outFileNamePrefix $OUTPUT/$SAMPLE. \
    --outSAMtype BAM Unsorted \
    --outSAMstrandField intronMotif
```

Sample problem 1

$ module load STAR/latest

Download “small_reads.fastq”

Use STAR to map these reads to the worm genome:

```
wget https://ctrapnell.github.io/genome569/example_files/small_reads.fastq

STAR --genomeDir ~coletrap/teaching/genome569/reference \
     --readFilesIn small_reads.fastq \
     --outFileNamePrefix example/ \
     --outSAMtype BAM Unsorted \
     --outSAMstrandField intronMotif
```

SAM

Problem: store alignments in a standard format

Solution:

Sequence Alignment/Map Format Specification

The SAM/BAM Format Specification Working Group

5 Feb 2020

The master version of this document can be found at `https://github.com/samtools/hts-specs`.
This printing is version dfc3e48 from that repository, last modified on the date shown above.

1 The SAM Format Specification

SAM stands for Sequence Alignment/Map format. It is a TAB-delimited text format consisting of a header section, which is optional, and an alignment section. If present, the header must be prior to the alignments. Header lines start with '`@`', while alignment lines do not. Each alignment line has 11 mandatory fields for essential alignment information such as mapping position, and variable number of optional fields for flexible or aligner specific information.

Example alignment

```
Coor     12345678901234  5678901234567890123456789012345
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

+r001/1        TTAGATAAAGGATA*CTG
+r002         aaaAGATAA*GGATA
+r003       gcctaAGCTAA
+r004                       ATAGCT..............TCAGC
-r003                              ttagctTAGGC
-r001/2                                        CAGCGGCAT
```

```
@HD VN:1.6 SO:coordinate
@SQ SN:ref LN:45
r001   99 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002    0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003    0 ref  9 30 5S6M       *  0   0 GCCTAAGCTAA       * SA:Z:ref,29,-,6H5M,17,0;
r004    0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003 2064 ref 29 17 6H5M       *  0   0 TAGGC             * SA:Z:ref,9,+,5S6M,30,1;
r001  147 ref 37 30 9M         =  7 -39 CAGCGGCAT         * NM:i:1
```

Example alignment

```
Coor      12345678901234  5678901234567890123456789012345
ref       AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

+r001/1         TTAGATAAAGGATA*CTG
+r002         aaaAGATAA*GGATA
+r003       gcctaAGCTAA
+r004                       ATAGCT..............TCAGC
-r003                             ttagctTAGGC
-r001/2                                        CAGCGGCAT
```

Read name | **Where the Read maps** | **What's different** | **The original Read sequence** | **Additional metadata**

```
@HD VN:1.6 SO:coordinate
@SQ SN:ref LN:45
r001   99 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002    0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003    0 ref  9 30 5S6M       *  0   0 GCCTAAGCTAA       * SA:Z:ref,29,-,6H5M,17,0;
r004    0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003 2064 ref 29 17 6H5M       *  0   0 TAGGC             * SA:Z:ref,9,+,5S6M,30,1;
r001  147 ref 37 30 9M         =  7 -39 CAGCGGCAT         * NM:i:1
```

Key features of SAM

Widely adopted. Nearly every read aligner uses it, many analysis tools accept it as input

“Lossless” - SAM files include all the information in the raw reads (even those that don’t map to the genome)

Can be stored in a binary format and heavily compressed

Can be indexed for fast lookup. You can easily extract all the alignments for a specific locus.

samtools

Problem: extract alignments in a given locus

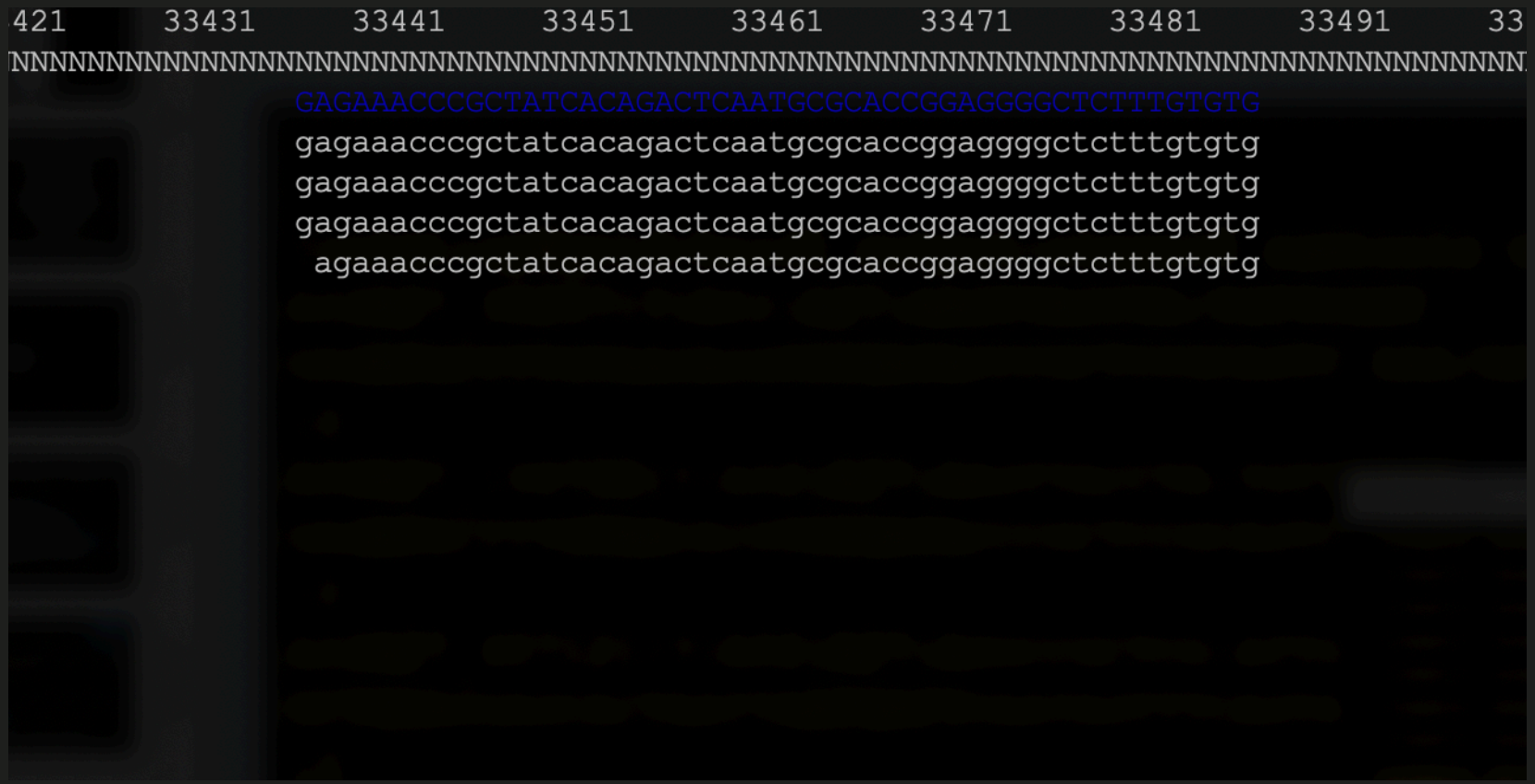
Solution: `samtools view input.bam <region>`

Regions look like this:

`chr1:19200776-19220776`

```
[coletrap@grid-head1 output-clean]$ samtools view aligned-reads-filtered-sorted/A02A27.bam | head
A02A27_90257|?|A02|A27|AAGTTACCTA|TGTTCGGT      0       I       6247    255     51M     *       0       0       GAATGGCATTAA
GAGATGCTTTGTAACACGTCCGATACCCGCTCCGCAGTC AAA/AEEEEEEEEEE/EEEE/EEEEEEAEEEEEEAE/EEEEEEEEEEEEE     NH:i:1  HI:i:1  AS:i:50 nM:i
:0
A02A27_198730|?|A02|A27|AAGTTACCTA|TGTTCGGT     0       I       6247    255     51M     *       0       0       GAATGGCATTAA
GAGATGCTTTGTAACACGTCCGATACCCGCTCCGCAGTC AAAAAEEEEEEEEEEEEEEEEEEEAEEEE6EEEEEEEEEEEAEEEEEEEEE     NH:i:1  HI:i:1  AS:i:50 nM:i
:0
A02A27_407121|?|A02|A27|AGGCCGCTCG|ATTGCTAC     0       I       30051   255     51M     *       0       0       GAACTATATGTT
GCTGCAGGAGCACTAATTAATGATTCAATAGTTTCAGTA AAAAA6AAEEEEEEEEE/EEE6EEEAEEAEEEEEEEEEEEEEEEEEEEE     NH:i:1  HI:i:1  AS:i:50 nM:i
:0
A02A27_75625|?|A02|A27|AGGCCGCTCG|ATTGCTAC      0       I       30051   255     51M     *       0       0       GAACTATATGTT
GCTGCAGGAGCACTAATTAATGATTCAATAGTTTCAGTA AAAAAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE     NH:i:1  HI:i:1  AS:i:50 nM:i
:0
A02A27_511761|?|A02|A27|ACGGAGAATA|GCTTCAAC     0       I       73612   255     51M     *       0       0       GCCTACGACAAT
GGAATTGTAAGAATGAGAGCTAGCGGATGTGACGTGGCT AAAAAEEEEAEEEEEAEEEEEEEEEEEEAEEEEEEEEEEEEEE/EEEEEEE     NH:i:1  HI:i:1  AS:i:50 nM:i
:0
A02A27_456432|?|A02|A27|AGGCCGCTCG|CTGCTGAC     0       I       84306   255     20M     *       0       0       GCTCAGGGGTGC
GATTTTTG        /AAAAA//EE/EAEE/EEAA    NH:i:1  HI:i:1  AS:i:19 nM:i:0
A02A27_526097|?|A02|A27|GTATACCGAA|TACAATGT     0       I       84924   255     51M     *       0       0       TCAGACTACCTA
TCAGACGATTCAATGGATTGTTGATAGAGATCGATCGGT AAAA/EE/EEEEEEEE/EAAEEEEEEEEEAEEEEEEEAEEEEEEEAEEEAE     NH:i:1  HI:i:1  AS:i:50 nM:i
:0
A02A27_531142|?|A02|A27|GGCTATGACT|CTTGGTTT     0       I       86836   255     44M     *       0       0       CAGATATGTCGA
TTATTGTGGCCAGCCTCACGCAAATGTGCTGT        66A6/E/EA/E//E//EA/EEE//EA//E/AE//A/EEEE//E6    NH:i:1  HI:i:1  AS:i:43 nM:i:0
A02A27_364043|?|A02|A27|GGCTATGACT|CTTGGTTA     0       I       86836   255     51M     *       0       0       CAGATATGTCGA
TTATTGTGGACAGCCACACGCAAATGTGCTGTGATCCAT AA6AAEA/AA/6AEAEA/E/A/E//AA/EE///EEEA<A//A/E/EE/EEA     NH:i:1  HI:i:1  AS:i:46 nM:i
:2
A02A27_411978|?|A02|A27|GGCTATGACT|CTTGGTTT     0       I       86836   255     51M     *       0       0       CAGATATGTCGA
TTATTGTGGCCAGCCTCACGCAAATGTGCTGTGATCCAT AAAAAEEEEEEEEEEEEEEEEEEEEAEEEEEEAEEEEEEEEEEEEEEEEE     NH:i:1  HI:i:1  AS:i:50 nM:i
:0
```

samtools tview input.bam



BED

BED

Browser **E**xtensible **D**ata format introduced by UCSC genome browser

Widely used to annotate genomes with intervals of interest

Simple, tab-delimited text file

Not very extensible, so used for very simple features (e.g. enhancers)

FYI gene models typically stored in GFF or GTF format, which is much more complex.

WS260.rRNA.genes.bed

Chromosome	Start	Stop	Feature name	"Score"	Strand
I	15062083	15063836	WBGene00004512	255	+
I	15064301	15064453	WBGene00004567	255	+
I	15064838	15068346	WBGene00004622	255	+
I	15069280	15071033	WBGene00004513	255	+
MtDNA	898	1593	WBGene00014454	255	+
MtDNA	10403	11354	WBGene00014472	255	+
V	17115903	17116021	WBGene00077465	255	+
V	17117863	17117981	WBGene00077466	255	+
V	17118837	17118935	WBGene00077467	255	+

bedtools

Problem: compute overlap between BED files

Solution: `bedtools intersect -a reads.bed -b genes.bed`

This command computes the number of bases in file “B” that are covered by intervals in file “A”.

Bedtools has many utilities

Command	Function
bamtobed	Convert a BAM file to a BED file
closest	For each interval in one file, find the closest in another
overlap	Compute the overlap between intervals
merge	Merge intervals that overlap
subtract	Remove the overlapping regions from intervals

And many, many, many more functions.
Many with multiple modes of operation.

Sample problem 2

Download “genes.bed”

Use bedtools to count the number of reads you just mapped that hit each gene.

```
samtools sort example/Aligned.out.bam > example/sorted.bam

bedtools intersect -a genes.bed -b example/sorted.bam -wa -c
```

Sample problem 2

```
I       11947001        11953126        WBGene00011060  255     +       1
I       11953512        11961984        WBGene00002004  255     -       15
I       11971179        11971797        WBGene00044805  255     -       0
I       11979401        11985612        WBGene00013135  255     +       1
```

Note that the output is a BED file! The “score” field I told you to ignore earlier has the results.

THE END

For now