HBC
Harvard Chan Bioinformatics Core

Shell for Bioinformatics

https://tinyurl.com/hbc-shell-online

Harvard Chan Bioinformatics Core

in collaboration with

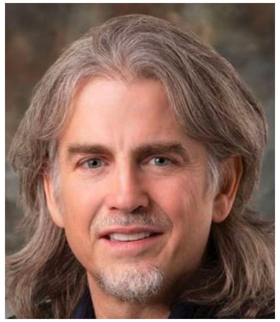
HMS Research Computing

Introductions!

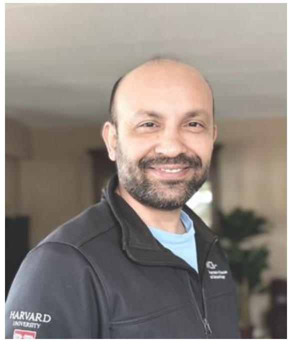
Shannan Ho Sui
Director

Meeta Mistry
Associate Director

Lorena Pantano
Director of Bioinformatics Platform



John Quackenbush
Faculty Advisor



Upen Bhattarai

Heather Wick

Will Gammerdinger

Noor Sohail

Alex Bartlett

Elizabeth Partan

Emma Berdan

James Billingsley

Zhu Zhuo

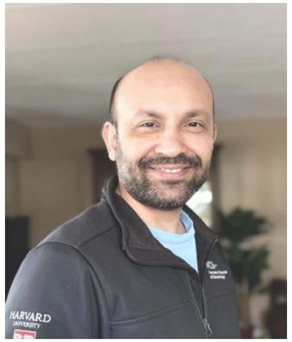
Maria Simoneau

Shannan Ho Sui
Director

Meeta Mistry
Associate Director

Lorena Pantano
Director of Bioinformatics Platform

John Quackenbush
Faculty Advisor



Upen Bhattarai

Heather Wick

Will Gammerdinger

Noor Sohail

Alex Bartlett

Elizabeth Partan

Emma Berdan

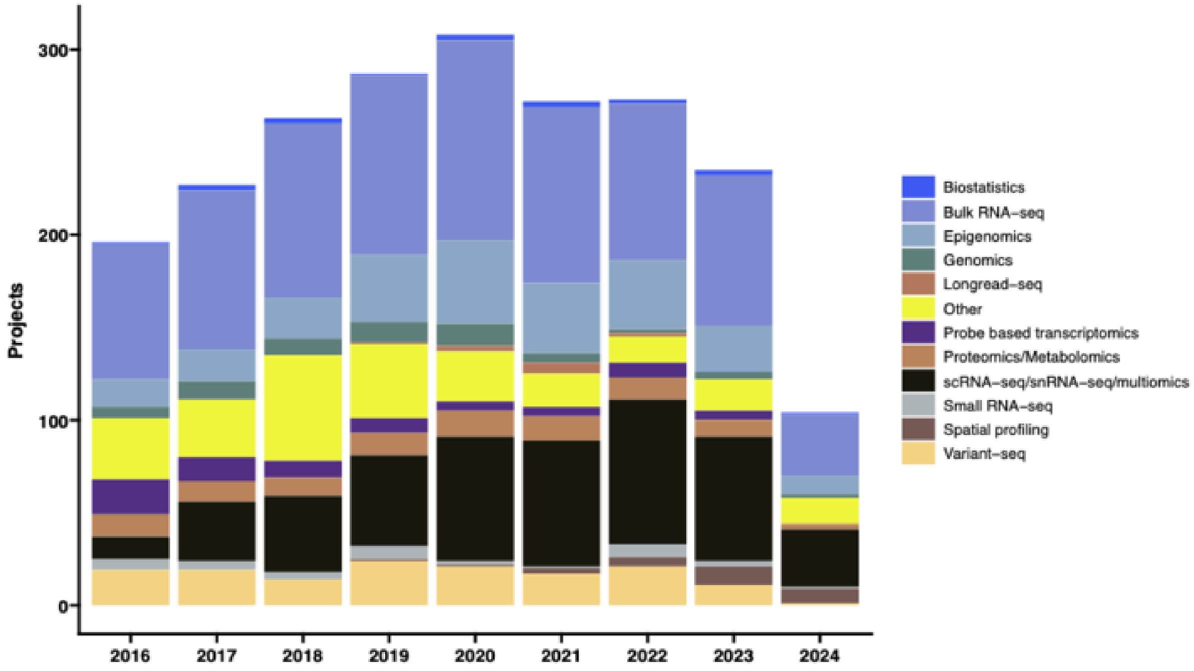
James Billingsley

Zhu Zhuo

Maria Simoneau

Consulting

- Transcriptomics: Bulk, single cell, small RNA
- Epigenomics: ChIP-seq, CUT&RUN, ATAC-seq, DNA methylation
- Variant discovery: WGS, resequencing, exome-seq and CNV
- Multiomics integration
- Spatial biology
- Experimental design and grant support



https://bioinformatics.sph.harvard.edu/services

Consulting

- Transcriptomics: Bulk, single cell, small RNA
- Epigenomics: ChIP-seq, CUT&RUN, ATAC-seq, DNA methylation
- Variant discovery: WGS, resequencing, exome-seq and CNV
- Multiomics integration
- Spatial biology
- Experimental design and grant support

CFAR
HARVARD UNIVERSITY
CENTER FOR AIDS RESEARCH

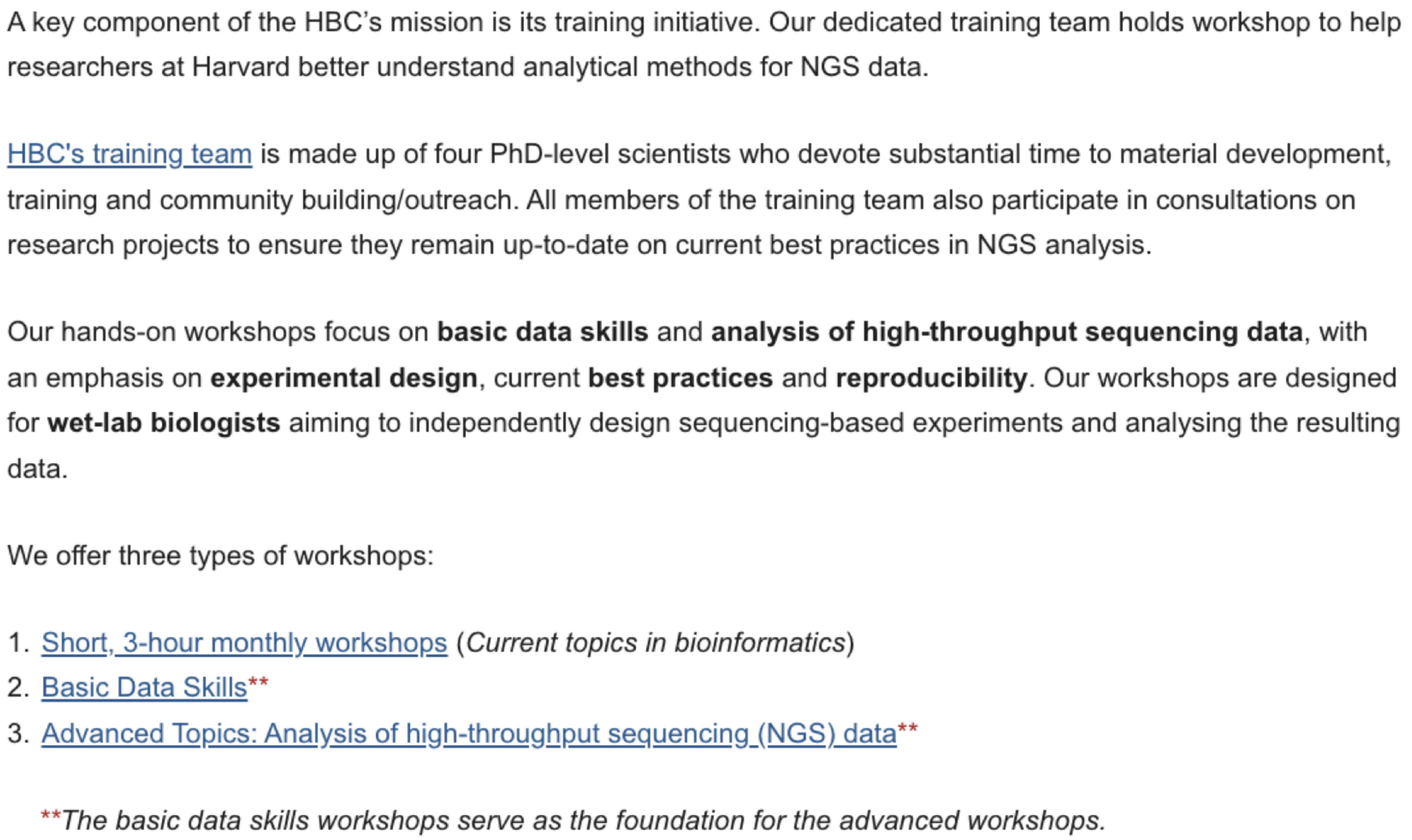
HARVARD
T.H. CHAN
SCHOOL OF PUBLIC HEALTH

NIEHS

HARVARD
CATALYST
THE HARVARD CLINICAL AND TRANSLATIONAL SCIENCE CENTER

HARVARD
MEDICAL SCHOOL

Training



A key component of the HBC's mission is its training initiative. Our dedicated training team holds workshop to help researchers at Harvard better understand analytical methods for NGS data.

HBC's training team is made up of four PhD-level scientists who devote substantial time to material development, training and community building/outreach. All members of the training team also participate in consultations on research projects to ensure they remain up-to-date on current best practices in NGS analysis.

Our hands-on workshops focus on **basic data skills** and **analysis of high-throughput sequencing data**, with an emphasis on **experimental design**, current **best practices** and **reproducibility**. Our workshops are designed for **wet-lab biologists** aiming to independently design sequencing-based experiments and analysing the resulting data.

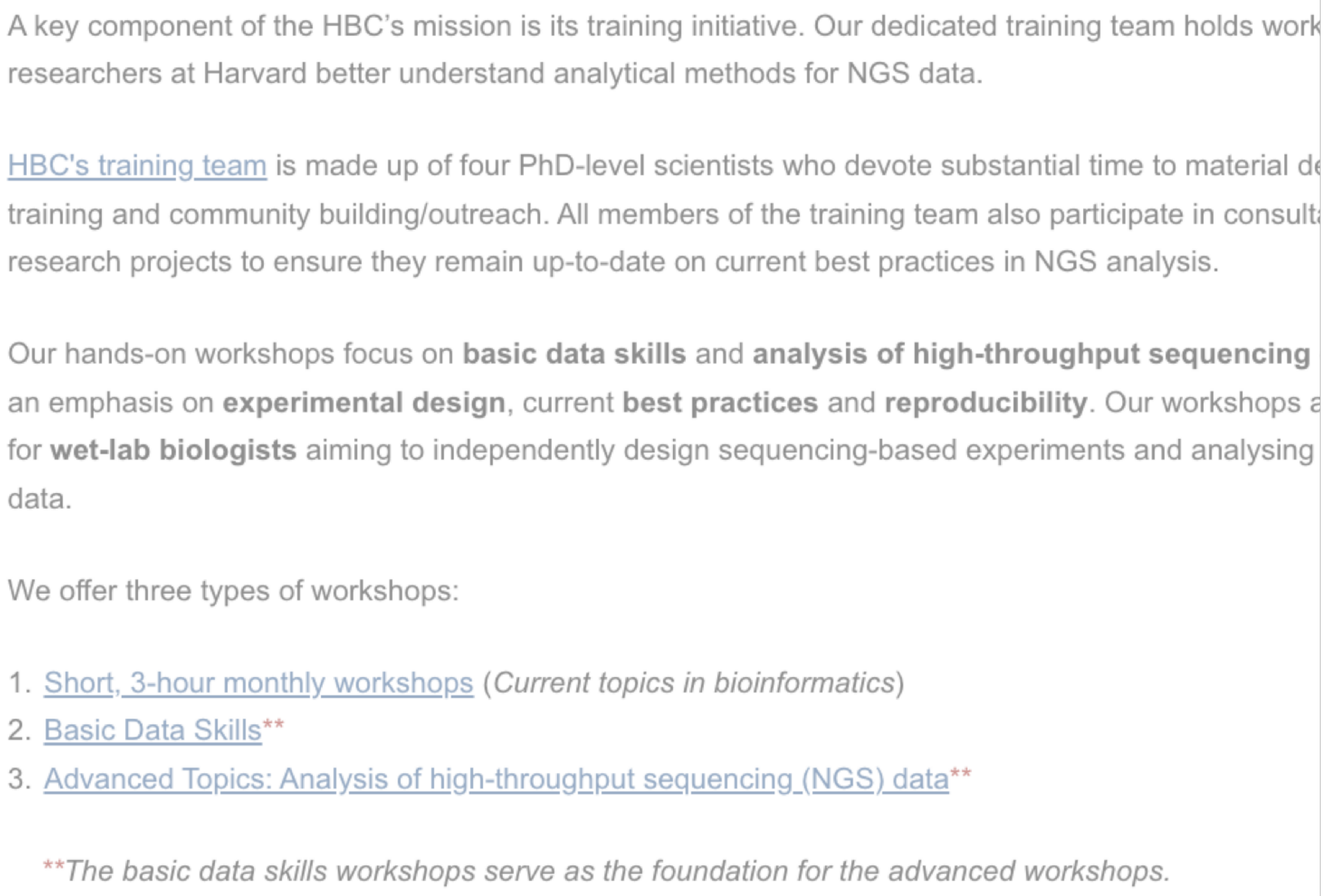
We offer three types of workshops:

1. Short, 3-hour monthly workshops (*Current topics in bioinformatics*)
2. Basic Data Skills**
3. Advanced Topics: Analysis of high-throughput sequencing (NGS) data**

***The basic data skills workshops serve as the foundation for the advanced workshops.*

https://bioinformatics.sph.harvard.edu/training

Training



A key component of the HBC's mission is its training initiative. Our dedicated training team holds work researchers at Harvard better understand analytical methods for NGS data.

HBC's training team is made up of four PhD-level scientists who devote substantial time to material de training and community building/outreach. All members of the training team also participate in consulta research projects to ensure they remain up-to-date on current best practices in NGS analysis.

Our hands-on workshops focus on **basic data skills** and **analysis of high-throughput sequencing** an emphasis on **experimental design**, current **best practices** and **reproducibility**. Our workshops a for **wet-lab biologists** aiming to independently design sequencing-based experiments and analysing data.

We offer three types of workshops:

1. Short, 3-hour monthly workshops (*Current topics in bioinformatics*)
2. Basic Data Skills**
3. Advanced Topics: Analysis of high-throughput sequencing (NGS) data**

***The basic data skills workshops serve as the foundation for the advanced workshops.*

https://bioinformatics.sph.harvard.edu/training

CFAR
HARVARD UNIVERSITY
CENTER FOR AIDS RESEARCH

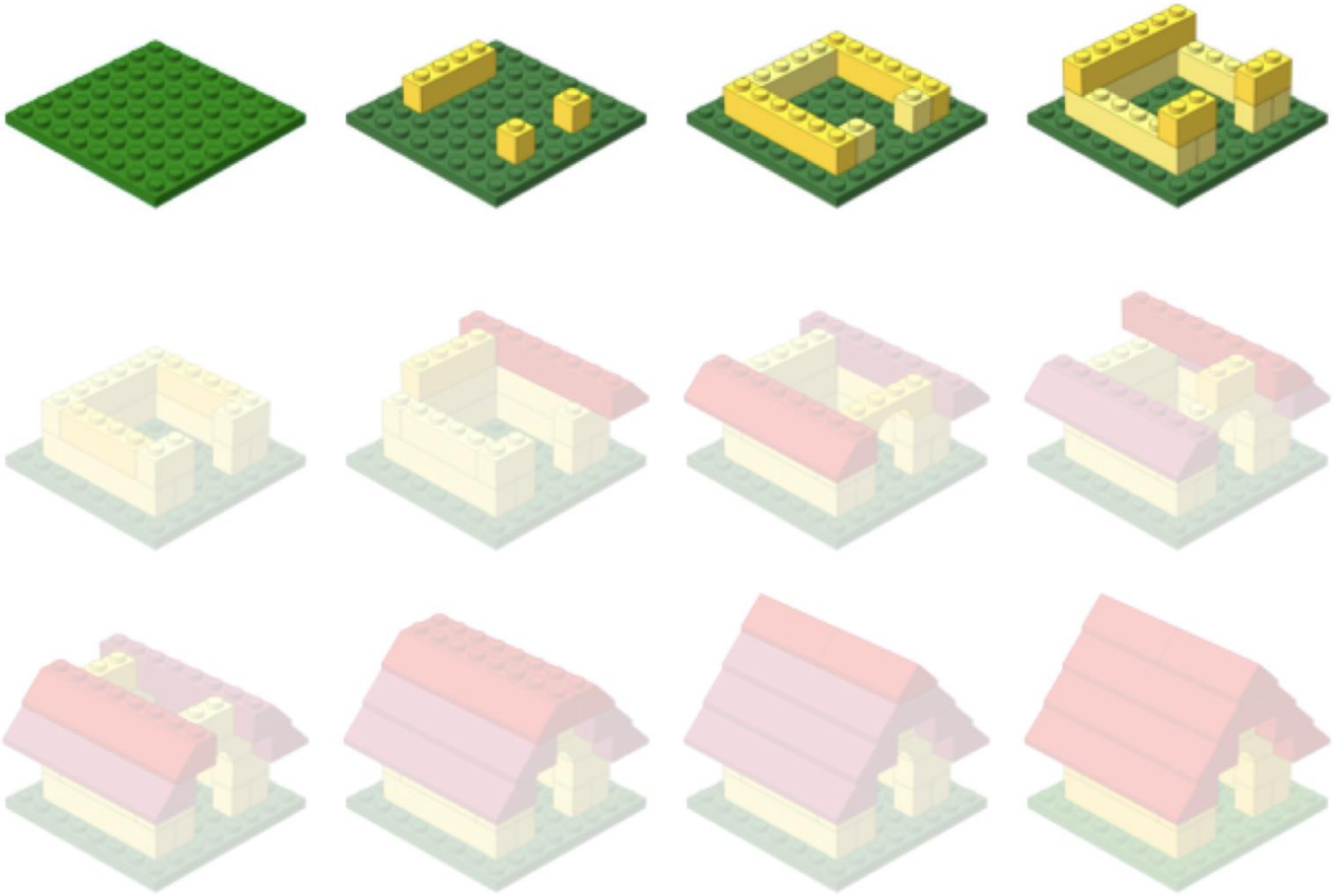
HARVARD
T.H. CHAN
SCHOOL OF PUBLIC HEALTH

DF/HCC
DANA-FARBER / HARVARD CANCER CENTER

HARVARD CATALYST
THE HARVARD CLINICAL AND TRANSLATIONAL SCIENCE CENTER

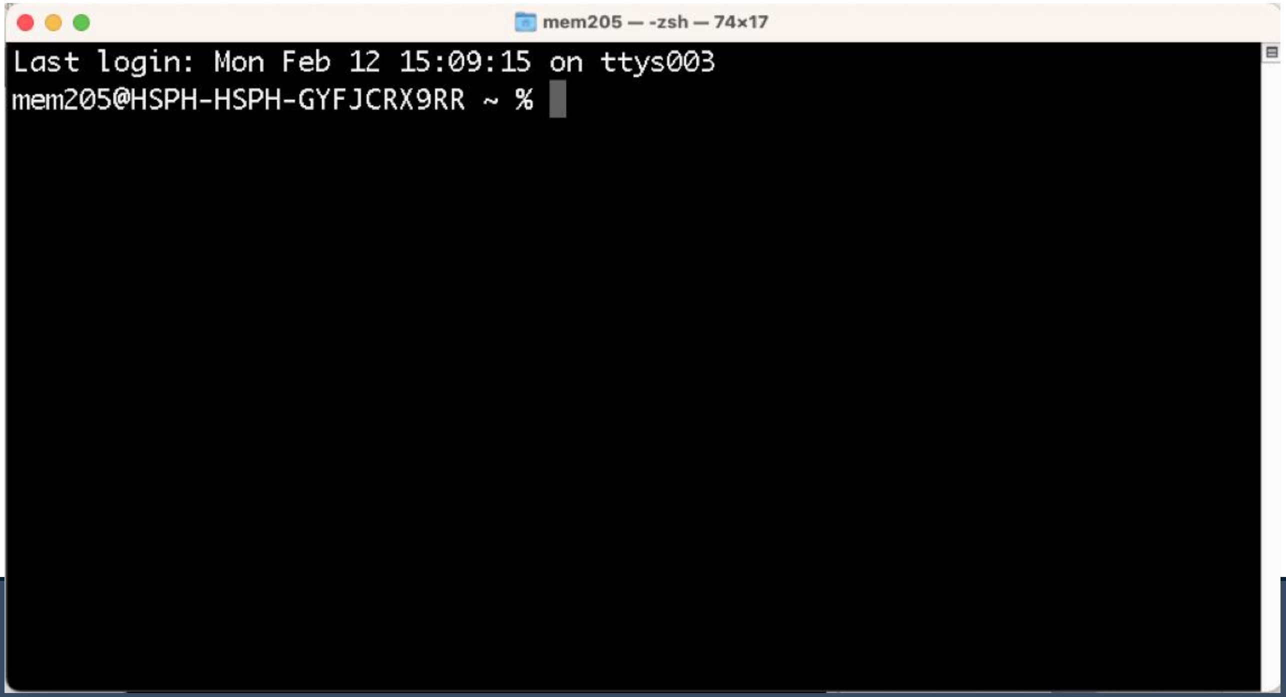
HARVARD
MEDICAL SCHOOL

Workshop scope



Learning Bioinformatics

What is shell?



```
Last login: Mon Feb 12 15:09:15 on ttys003
mem205@HSPH-HSPH-GYFJCRX9RR ~ %
```

- Shell is a program that allows users to control Unix/Linux OS with text commands

Terminology

- **Unix/Linux** - The operating systems of High Performance Computers (HPC)

Terminology

- **Unix/Linux** - The operating systems of High Performance Computers (HPC)
- **Shell** - A program that allows users to control Unix/Linux OS with text commands

Terminology

- **Unix/Linux** - The operating systems of High Performance Computers (HPC)
- **Shell** - A program that allows users to control Unix/Linux OS with text commands
- **Bash** - The most prevalent kind of shell

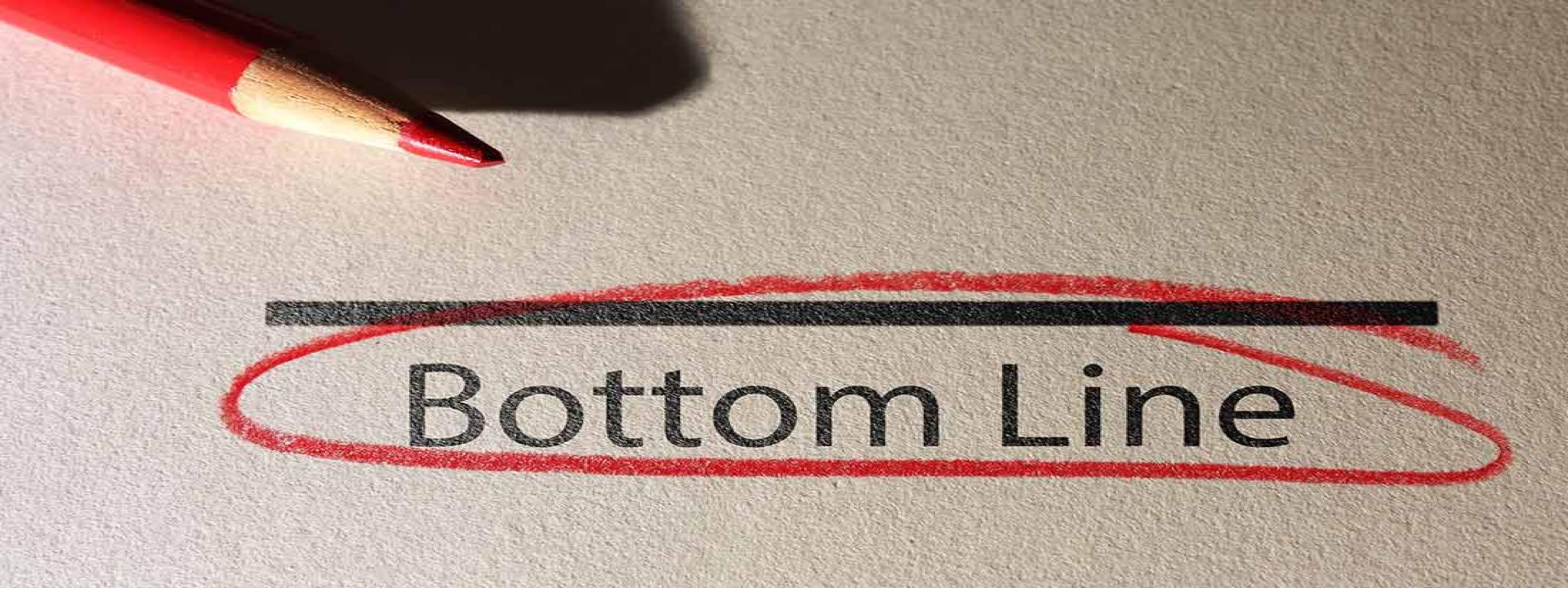


Image source: Balboa Capital Blog

If you plan to process raw high throughput sequencing data yourself, you will need to learn shell.

1. You need more resources than what is available on your laptop

- Sequence data files are LARGE
- Processing these data require increased CPU and memory
- High performance compute clusters have the necessary resources!

2. Many bioinformatics tools are only available as command-line tools

10XGenomics/
cellranger

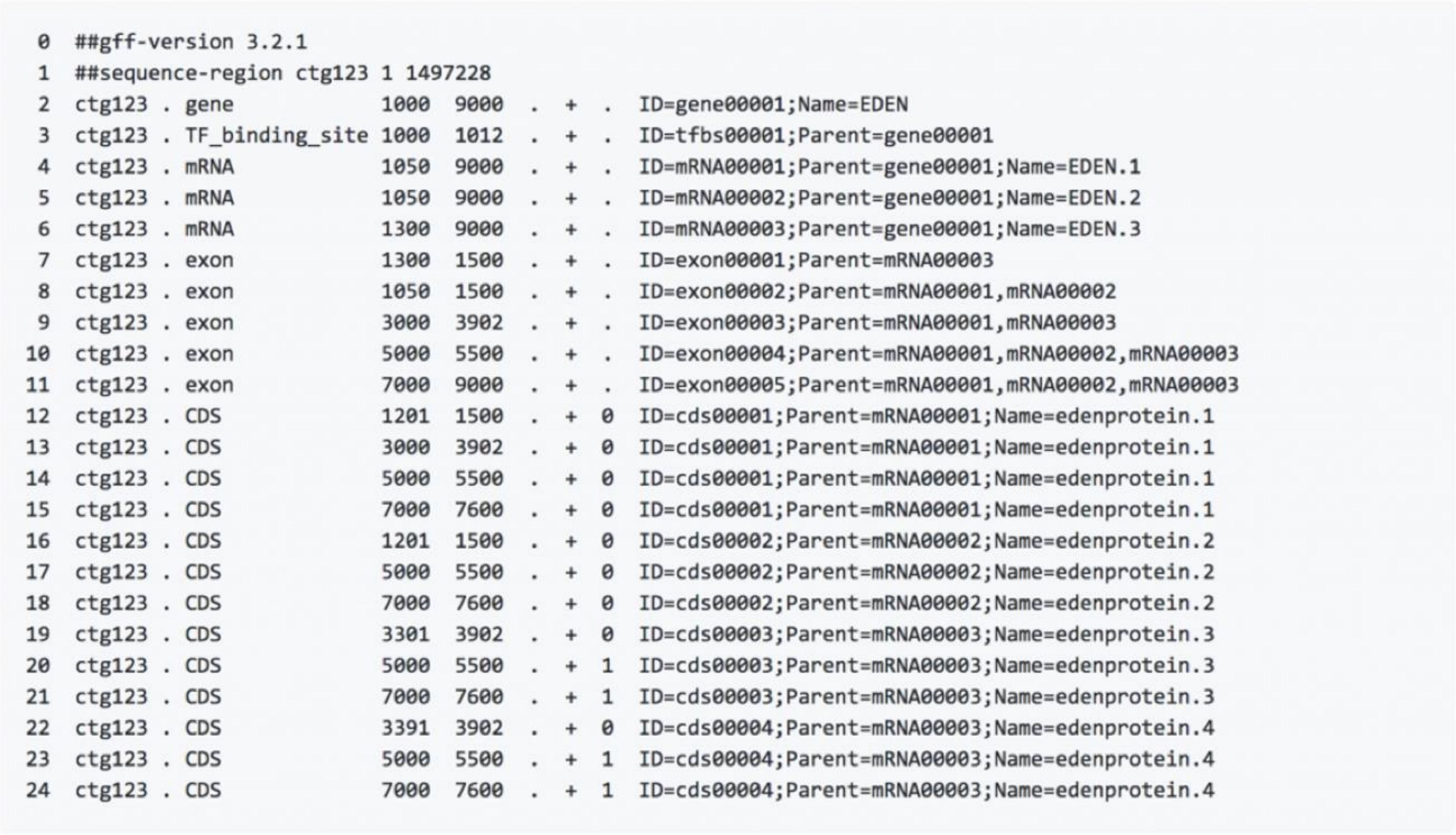
10x Genomics Single Cell Analysis

3. Many genomics filetypes are binary

- Binary files are not human readable
- Binary files need an interpreter

4. There are many useful commands that can help work with enormous data files

- Commands for easily viewing files: less, cat, head, tail



```
##gff-version 3.2.1
##sequence-region ctg123 1 1497228
ctg123 . gene            1000  9000  .  +  .  ID=gene00001;Name=EDEN
ctg123 . TF_binding_site 1000  1012  .  +  .  ID=tfbs00001;Parent=gene00001
ctg123 . mRNA            1050  9000  .  +  .  ID=mRNA00001;Parent=gene00001;Name=EDEN.1
ctg123 . mRNA            1050  9000  .  +  .  ID=mRNA00002;Parent=gene00001;Name=EDEN.2
ctg123 . mRNA            1300  9000  .  +  .  ID=mRNA00003;Parent=gene00001;Name=EDEN.3
ctg123 . exon            1300  1500  .  +  .  ID=exon00001;Parent=mRNA00003
ctg123 . exon            1050  1500  .  +  .  ID=exon00002;Parent=mRNA00001,mRNA00002
ctg123 . exon            3000  3902  .  +  .  ID=exon00003;Parent=mRNA00001,mRNA00003
ctg123 . exon            5000  5500  .  +  .  ID=exon00004;Parent=mRNA00001,mRNA00002,mRNA00003
ctg123 . exon            7000  9000  .  +  .  ID=exon00005;Parent=mRNA00001,mRNA00002,mRNA00003
ctg123 . CDS             1201  1500  .  +  0  ID=cds00001;Parent=mRNA00001;Name=edenprotein.1
ctg123 . CDS             3000  3902  .  +  0  ID=cds00001;Parent=mRNA00001;Name=edenprotein.1
ctg123 . CDS             5000  5500  .  +  0  ID=cds00001;Parent=mRNA00001;Name=edenprotein.1
ctg123 . CDS             7000  7600  .  +  0  ID=cds00001;Parent=mRNA00001;Name=edenprotein.1
ctg123 . CDS             1201  1500  .  +  0  ID=cds00002;Parent=mRNA00002;Name=edenprotein.2
ctg123 . CDS             5000  5500  .  +  0  ID=cds00002;Parent=mRNA00002;Name=edenprotein.2
ctg123 . CDS             7000  7600  .  +  0  ID=cds00002;Parent=mRNA00002;Name=edenprotein.2
ctg123 . CDS             3301  3902  .  +  0  ID=cds00003;Parent=mRNA00003;Name=edenprotein.3
ctg123 . CDS             5000  5500  .  +  1  ID=cds00003;Parent=mRNA00003;Name=edenprotein.3
ctg123 . CDS             7000  7600  .  +  1  ID=cds00003;Parent=mRNA00003;Name=edenprotein.3
ctg123 . CDS             3391  3902  .  +  0  ID=cds00004;Parent=mRNA00003;Name=edenprotein.4
ctg123 . CDS             5000  5500  .  +  1  ID=cds00004;Parent=mRNA00003;Name=edenprotein.4
ctg123 . CDS             7000  7600  .  +  1  ID=cds00004;Parent=mRNA00003;Name=edenprotein.4
```

5. Automation is the name of the game

- Launch many jobs with one command
- Code is used and reused to iterate tasks over multiple files
- Parallelization to complete tasks using multiple cores and increase speed!

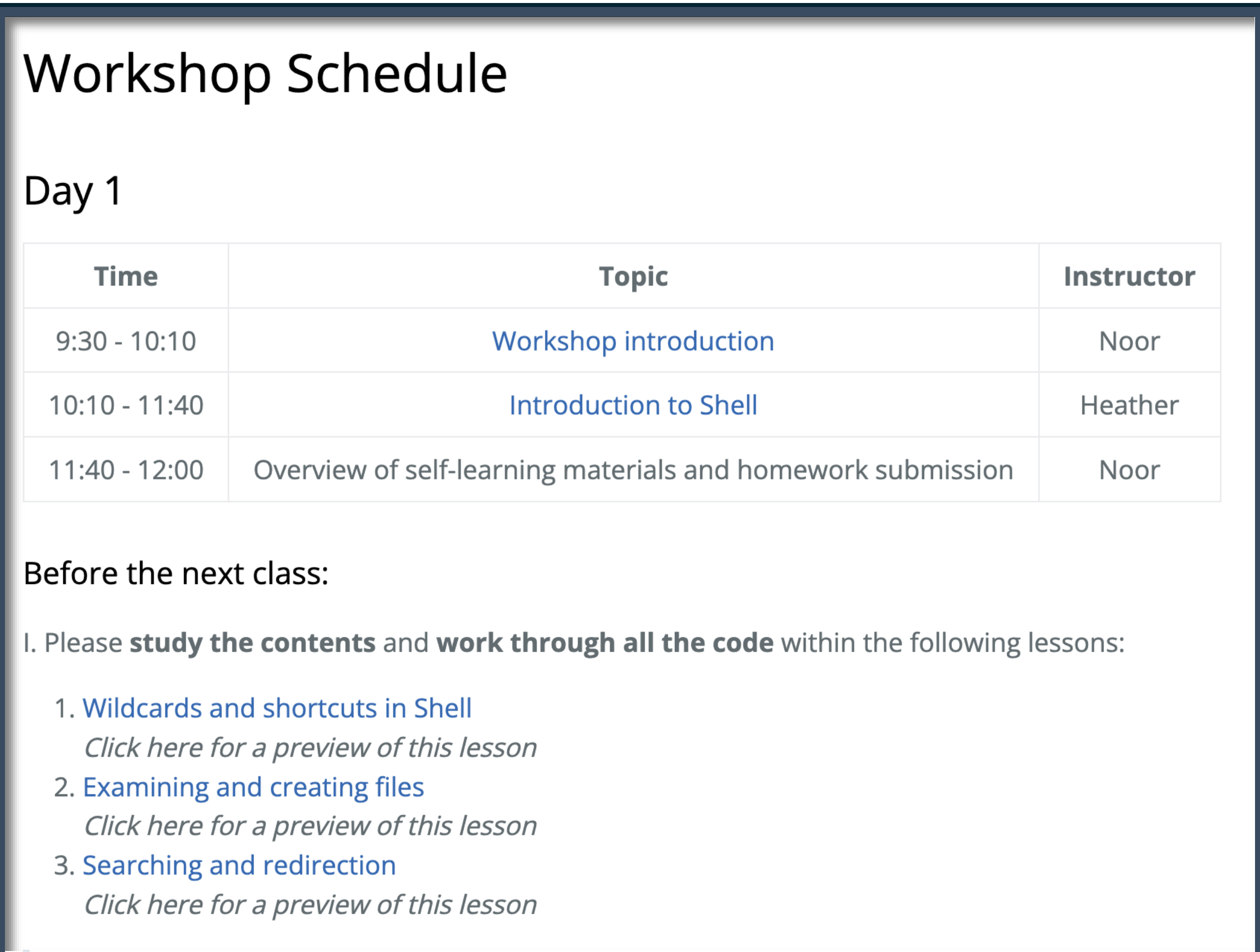
This could be you watching your analysis run!

Learning Objectives

- Navigate around the command line interface (bash/shell)
- Create and manipulate text files
- Submit jobs to a high-performance computing cluster

Logistics

Course schedule



Workshop Schedule

Day 1

Time	Topic	Instructor
9:30 - 10:10	Workshop introduction	Noor
10:10 - 11:40	Introduction to Shell	Heather
11:40 - 12:00	Overview of self-learning materials and homework submission	Noor

Before the next class:

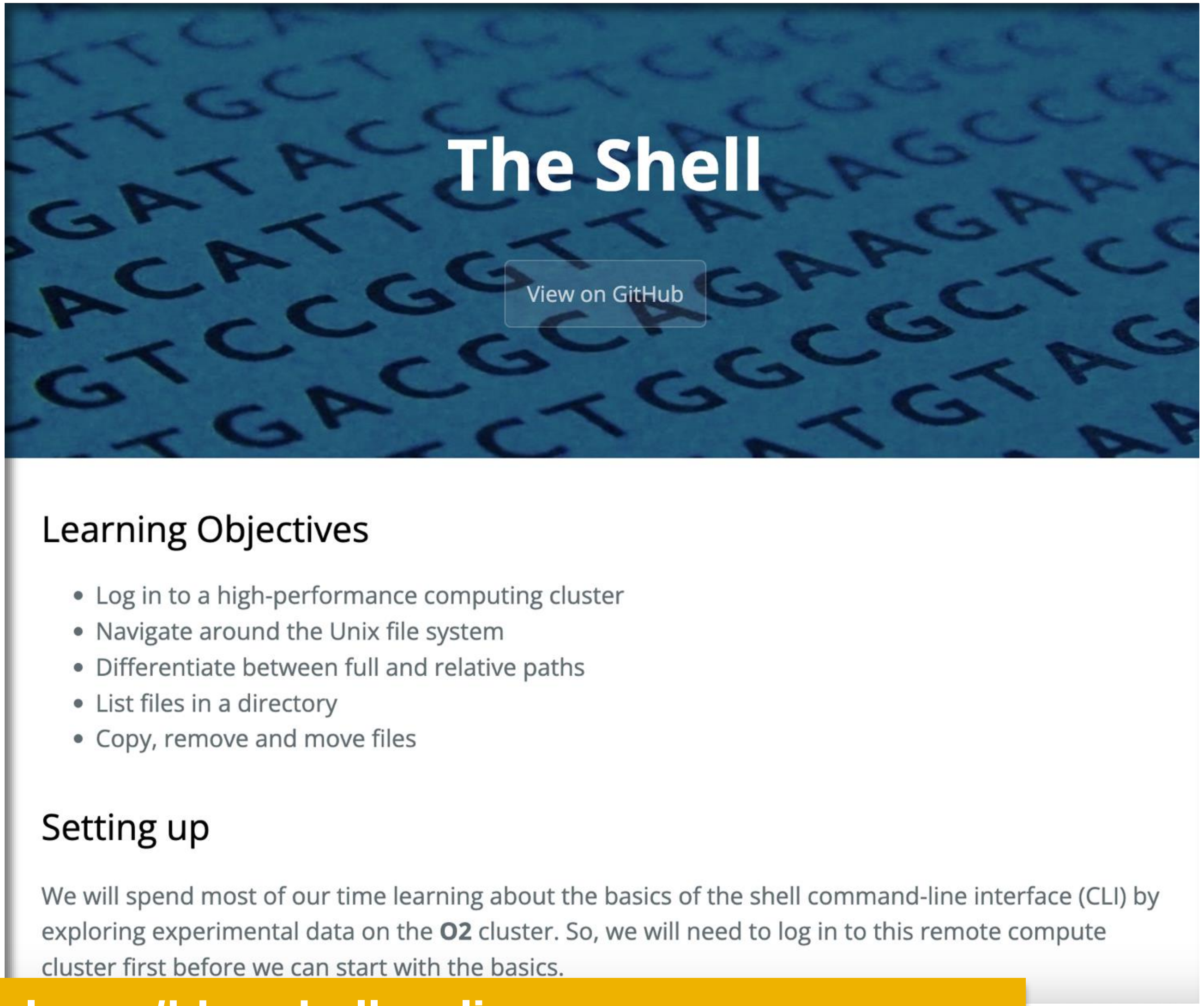
I. Please **study the contents** and **work through all the code** within the following lessons:

1. Wildcards and shortcuts in Shell
   Click here for a preview of this lesson
2. Examining and creating files
   Click here for a preview of this lesson
3. Searching and redirection
   Click here for a preview of this lesson

https://tinyurl.com/hbc-shell-online

Course materials

- We continuously update our materials to reflect changes in the field/software



The Shell

Learning Objectives

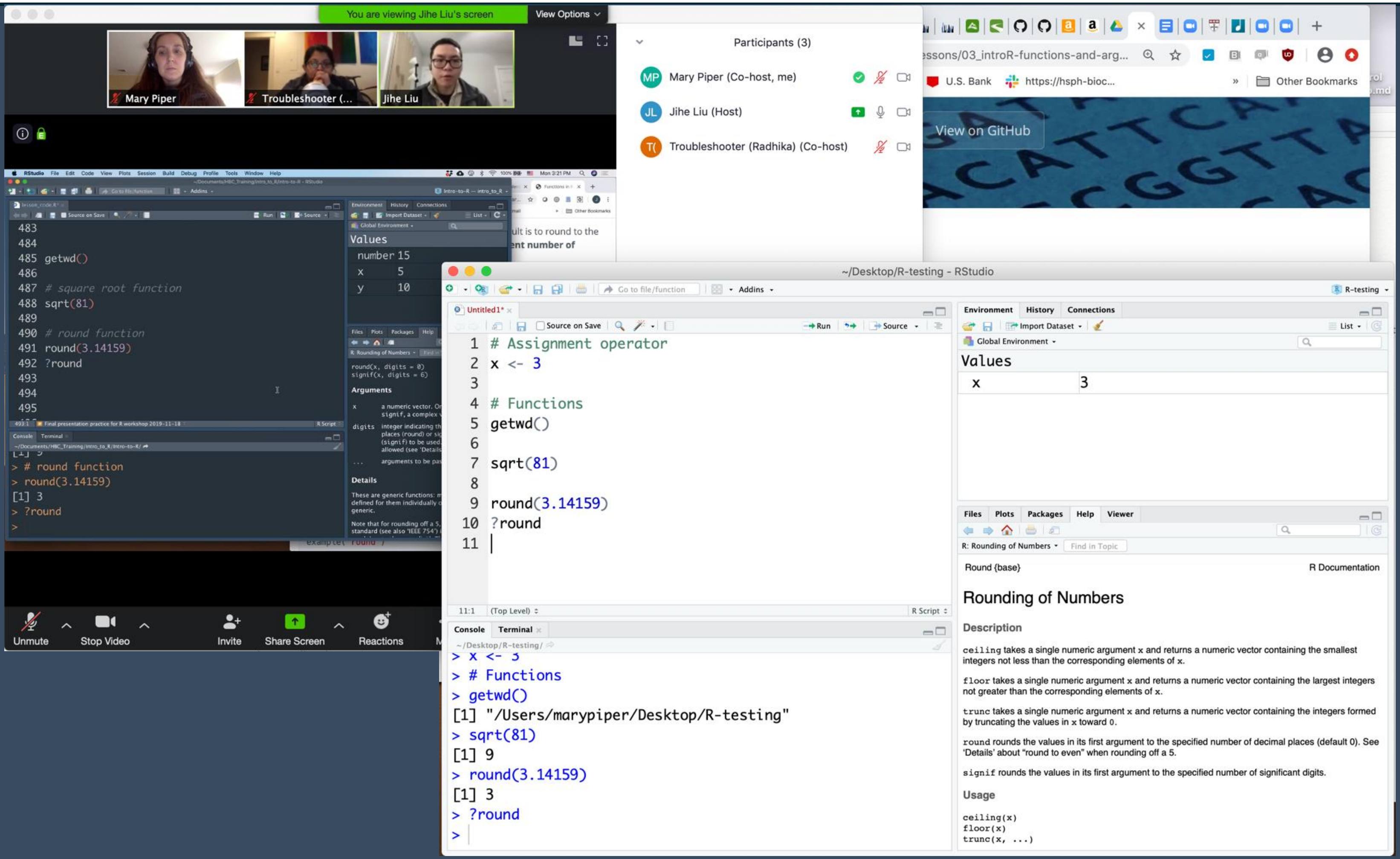
- Log in to a high-performance computing cluster
- Navigate around the Unix file system
- Differentiate between full and relative paths
- List files in a directory
- Copy, remove and move files

Setting up

We will spend most of our time learning about the basics of the shell command-line interface (CLI) by exploring experimental data on the **O2** cluster. So, we will need to log in to this remote compute cluster first before we can start with the basics.

https://tinyurl.com/hbc-shell-online

Single Screen & 3 Windows



Mary Piper (Co-host, me)

Jihe Liu (Host)

Troubleshooter (Radhika) (Co-host)

```
getwd()

# square root function
sqrt(81)

# round function
round(3.14159)
?round
```

```
> # round function
> round(3.14159)
[1] 3
> ?round
>
```

```
Values
number 15
x      5
y      10
```

```
# Assignment operator
x <- 3

# Functions
getwd()

sqrt(81)

round(3.14159)
?round
```

```
> x <- 3
> # Functions
> getwd()
[1] "/Users/marypiper/Desktop/R-testing"
> sqrt(81)
[1] 9
> round(3.14159)
[1] 3
> ?round
>
```

```
Values
x      3
```

Round {base} R Documentation

Rounding of Numbers

Description

`ceiling` takes a single numeric argument `x` and returns a numeric vector containing the smallest integers not less than the corresponding elements of `x`.

`floor` takes a single numeric argument `x` and returns a numeric vector containing the largest integers not greater than the corresponding elements of `x`.

`trunc` takes a single numeric argument `x` and returns a numeric vector containing the integers formed by truncating the values in `x` toward `0`.

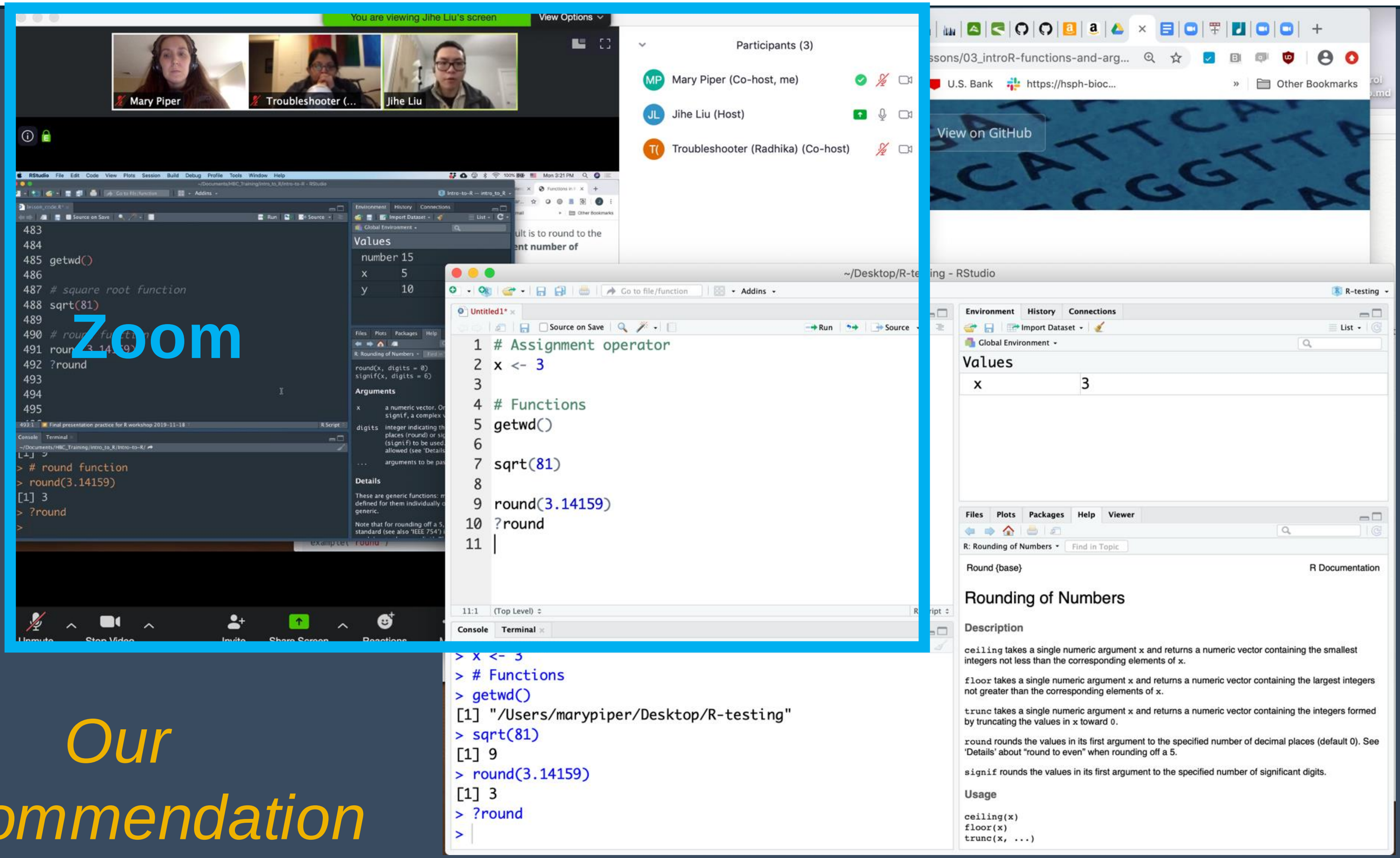
`round` rounds the values in its first argument to the specified number of decimal places (default 0). See 'Details' about "round to even" when rounding off a 5.

`signif` rounds the values in its first argument to the specified number of significant digits.

Usage

```
ceiling(x)
floor(x)
trunc(x, ...)
```

Single Screen & 3 Windows



```
getwd()

# square root function
sqrt(81)

# round function
round(3.14159)
?round
```

```
> # round function
> round(3.14159)
[1] 3
> ?round
>
```

Values

number	15
x	5
y	10

```
# Assignment operator
x <- 3

# Functions
getwd()

sqrt(81)

round(3.14159)
?round
```

Values

x	3

```
> x <- 3
> # Functions
> getwd()
[1] "/Users/marypiper/Desktop/R-testing"
> sqrt(81)
[1] 9
> round(3.14159)
[1] 3
> ?round
>
```

Round {base} R Documentation

Rounding of Numbers

Description

`ceiling` takes a single numeric argument `x` and returns a numeric vector containing the smallest integers not less than the corresponding elements of `x`.

`floor` takes a single numeric argument `x` and returns a numeric vector containing the largest integers not greater than the corresponding elements of `x`.

`trunc` takes a single numeric argument `x` and returns a numeric vector containing the integers formed by truncating the values in `x` toward `0`.

`round` rounds the values in its first argument to the specified number of decimal places (default 0). See 'Details' about "round to even" when rounding off a 5.

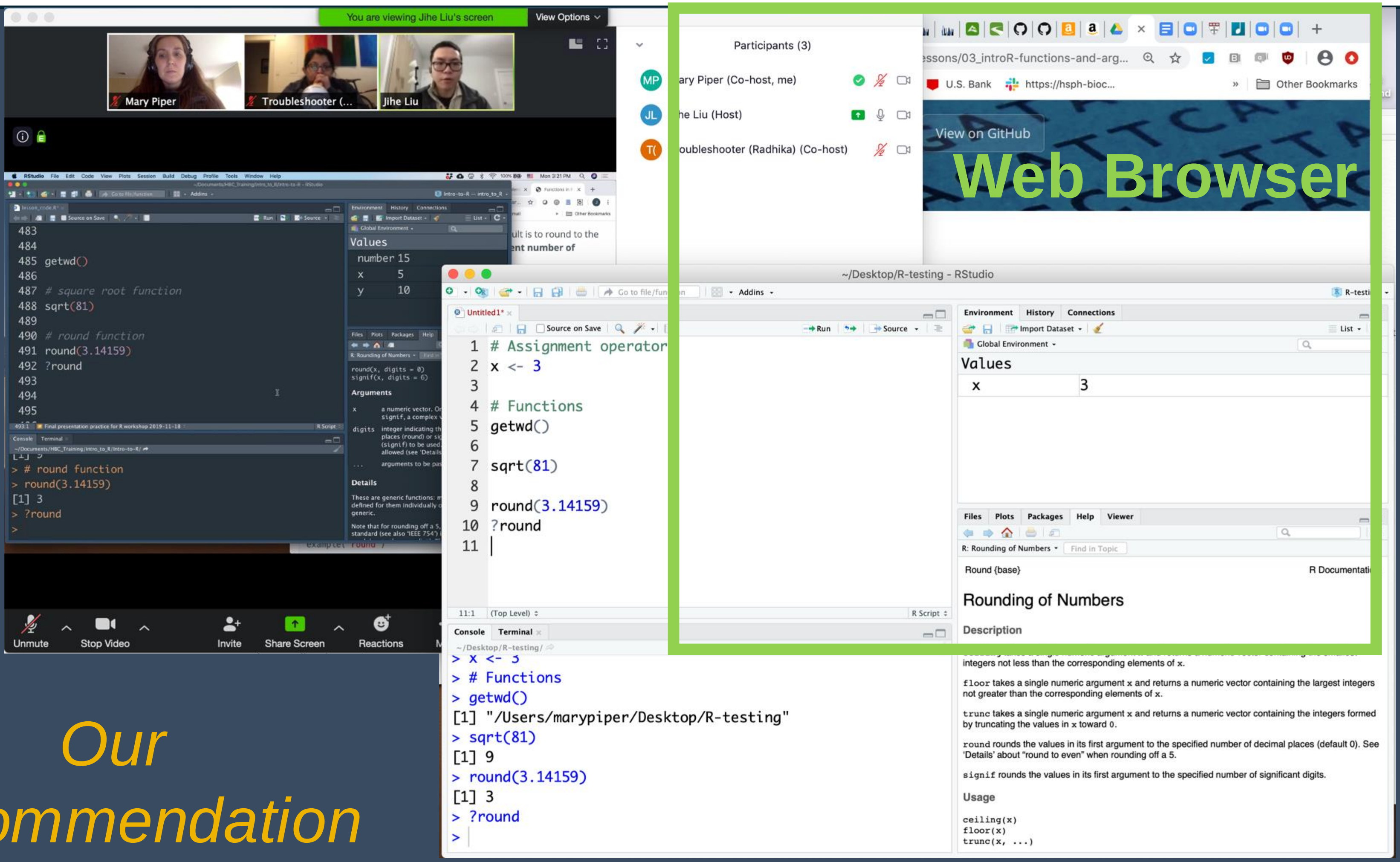
`signif` rounds the values in its first argument to the specified number of significant digits.

Usage

```
ceiling(x)
floor(x)
trunc(x, ...)
```

Our Recommendation

Single Screen & 3 Windows



```
getwd()

# square root function
sqrt(81)

# round function
round(3.14159)
?round
```

Values

number	15
x	5
y	10

```
> # round function
> round(3.14159)
[1] 3
> ?round
>
```

```
# Assignment operator
x <- 3

# Functions
getwd()

sqrt(81)

round(3.14159)
?round
```

Values

x	3

```
> x <- 3
> # Functions
> getwd()
[1] "/Users/marypiper/Desktop/R-testing"
> sqrt(81)
[1] 9
> round(3.14159)
[1] 3
> ?round
>
```

Round {base} R Documentation

Rounding of Numbers

Description

integers not less than the corresponding elements of x.

floor takes a single numeric argument x and returns a numeric vector containing the largest integers not greater than the corresponding elements of x.

trunc takes a single numeric argument x and returns a numeric vector containing the integers formed by truncating the values in x toward 0.

round rounds the values in its first argument to the specified number of decimal places (default 0). See 'Details' about "round to even" when rounding off a 5.

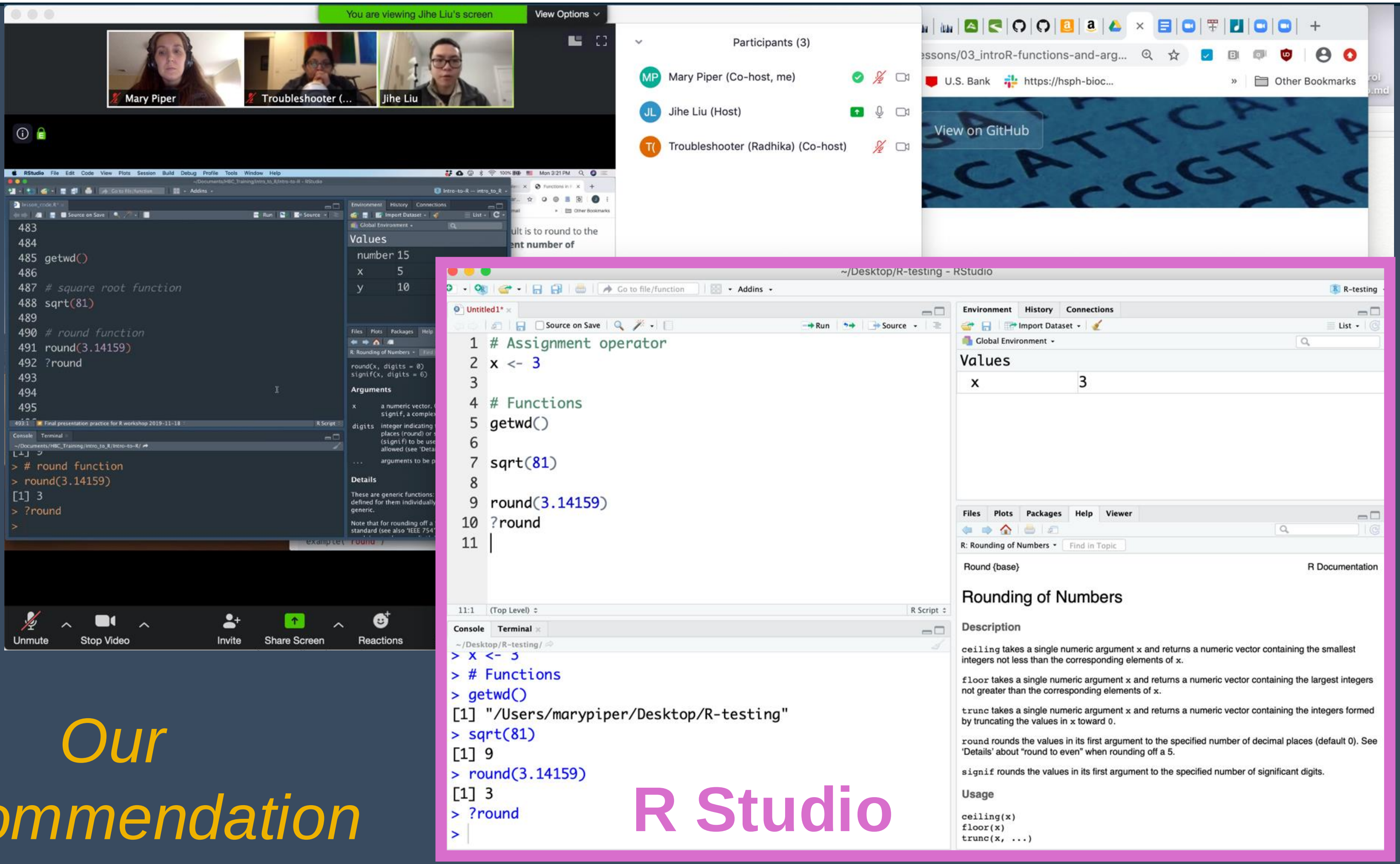
signif rounds the values in its first argument to the specified number of significant digits.

Usage

```
ceiling(x)
floor(x)
trunc(x, ...)
```

Our Recommendation

Single Screen & 3 Windows



```
getwd()

# square root function
sqrt(81)

# round function
round(3.14159)
?round
```

```
> # round function
> round(3.14159)
[1] 3
> ?round
>
```

```
# Assignment operator
x <- 3

# Functions
getwd()

sqrt(81)

round(3.14159)
?round
```

Values

x	3

```
~/Desktop/R-testing/
> x <- 3
> # Functions
> getwd()
[1] "/Users/marypiper/Desktop/R-testing"
> sqrt(81)
[1] 9
> round(3.14159)
[1] 3
> ?round
>
```

Round {base} R Documentation

Rounding of Numbers

Description

`ceiling` takes a single numeric argument `x` and returns a numeric vector containing the smallest integers not less than the corresponding elements of `x`.

`floor` takes a single numeric argument `x` and returns a numeric vector containing the largest integers not greater than the corresponding elements of `x`.

`trunc` takes a single numeric argument `x` and returns a numeric vector containing the integers formed by truncating the values in `x` toward `0`.

`round` rounds the values in its first argument to the specified number of decimal places (default 0). See 'Details' about "round to even" when rounding off a 5.

`signif` rounds the values in its first argument to the specified number of significant digits.

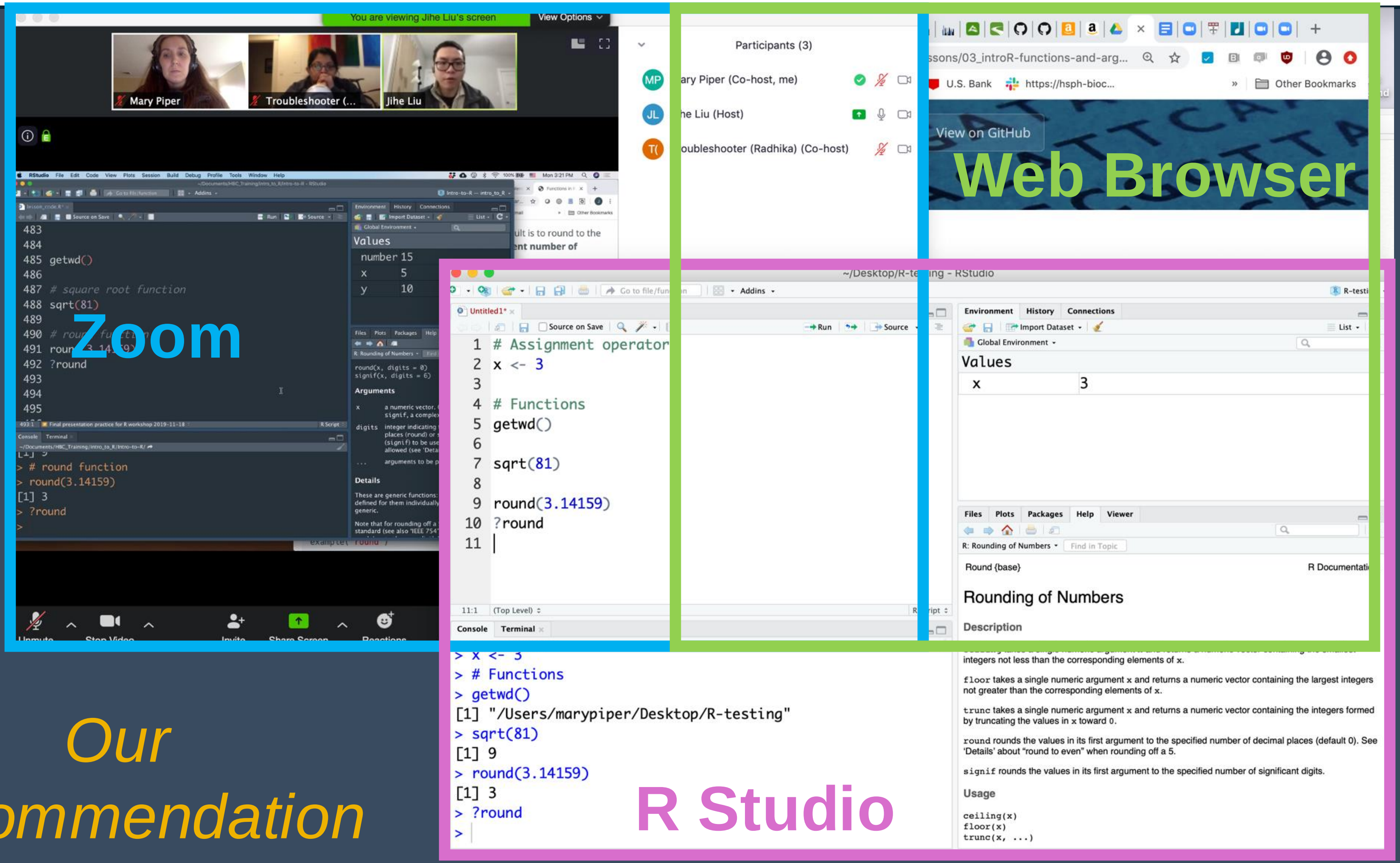
Usage

```
ceiling(x)
floor(x)
trunc(x, ...)
```

Our Recommendation

R Studio

Single Screen & 3 Windows



```
# Assignment operator
x <- 3

# Functions
getwd()

sqrt(81)

round(3.14159)
?round
```

Values

x	3

```
> x <- 3
> # Functions
> getwd()
[1] "/Users/marypiper/Desktop/R-testing"
> sqrt(81)
[1] 9
> round(3.14159)
[1] 3
> ?round
>
```

Round {base} R Documentation

Rounding of Numbers

Description

integers not less than the corresponding elements of x.

floor takes a single numeric argument x and returns a numeric vector containing the largest integers not greater than the corresponding elements of x.

trunc takes a single numeric argument x and returns a numeric vector containing the integers formed by truncating the values in x toward 0.

round rounds the values in its first argument to the specified number of decimal places (default 0). See 'Details' about "round to even" when rounding off a 5.

signif rounds the values in its first argument to the specified number of significant digits.

Usage

```
ceiling(x)
floor(x)
trunc(x, ...)
```

Our Recommendation

Course participation

- Mandatory review of self-learning lessons and assignments
- Attendance required for all classes
- Your questions and active participation drive learning
- **We look forward to all of your questions!**

Course participation

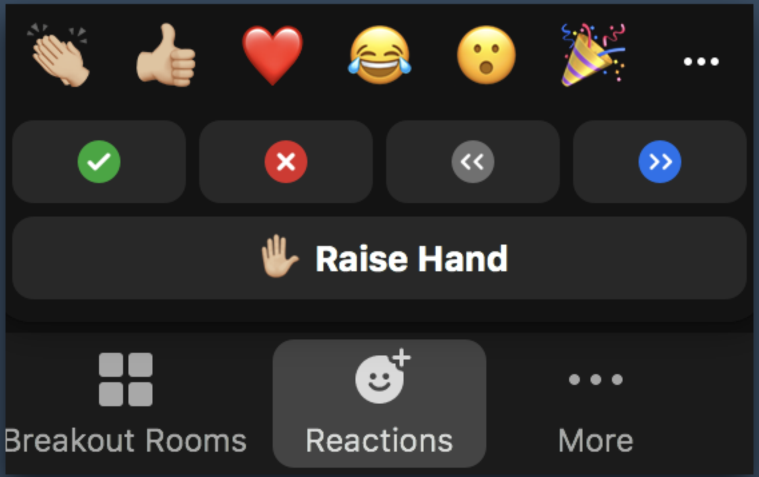
- At-home lessons and exercises after each session
- Cover material not previously discussed
- Provides us feedback to help pace the course appropriately
- 3-5 hours to complete
- Homework load is heavier in the beginning of this workshop series and tapers off

Using AI for Assignments

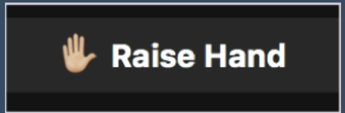
- Do
  - Try to resolve error messages with it
  - Test code written by AI on a dataset where you have expected results
  - Take the time to review the generated code line-by-line
- Don’t
  - Implement it in replacement to learning
  - Write code that you don’t understand
  - Assume the output from an AI process is correct

Odds & Ends

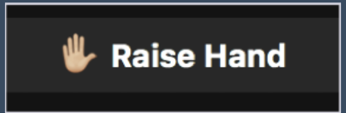
- Quit/minimize all applications that are not required for class
- Are you all set?
  - ✓ = "agree", "I'm all set"
  - ✗ = "disagree", "I need help"



Odds & Ends

- Questions for the presenter?
  - Post the question in the Chat window OR
  -  when the presenter asks for questions
  - Let the Troubleshooter know

Odds & Ends

- Questions for the presenter?
  - Post the question in the Chat window OR
  -  when the presenter asks for questions
  - Let the Troubleshooter know
- Technical difficulties with software?
  - Start a private chat with the Troubleshooter with a description of the problem

Thanks!

- Kathleen Chappell and Andy Bergman from HMS-RC
- Data Carpentry

Contact Us

- *HBC training team:* hbctraining@hsph.harvard.edu
- *HBC consulting:* bioinformatics@hsph.harvard.edu
- *O2 (HMS-RC):* rchelp@hms.harvard.edu