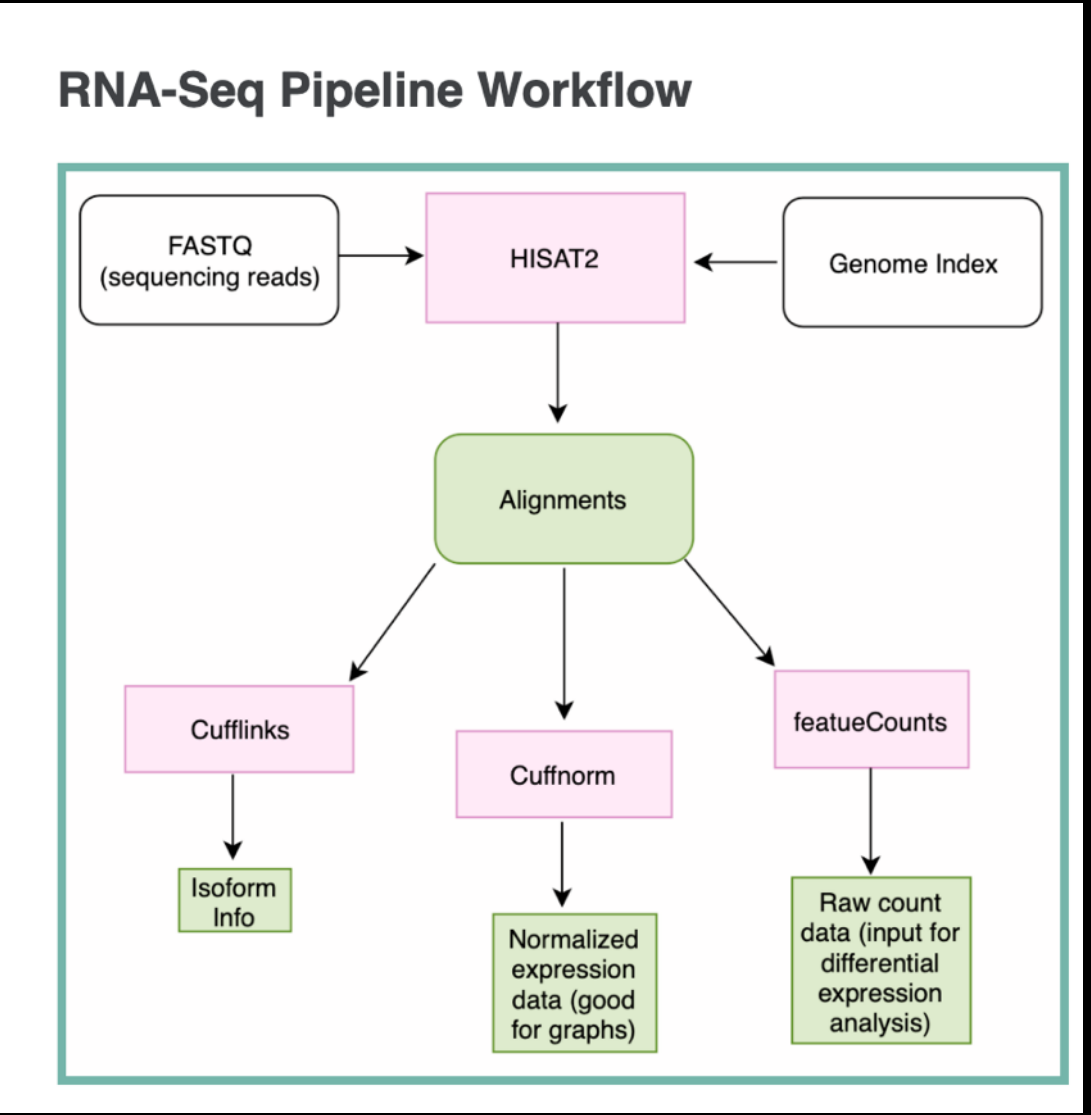
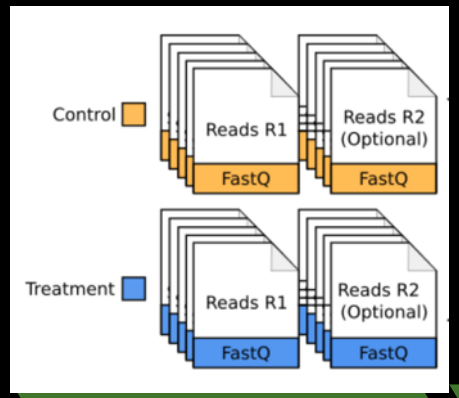
Intro to R for biologists: concepts and practice

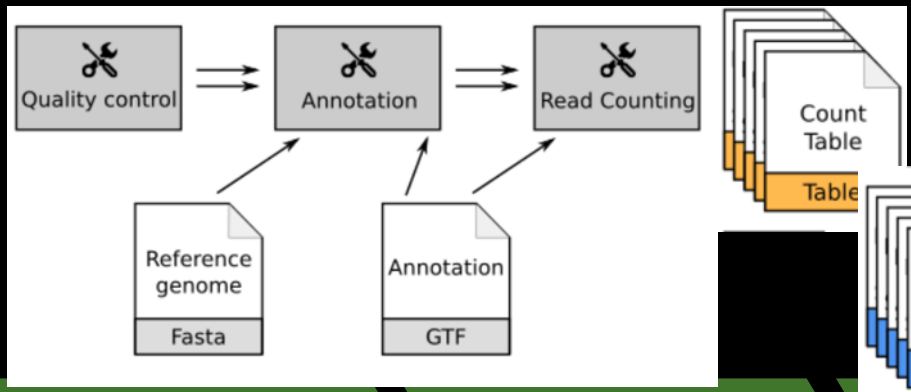
Jenna Oberstaller, PhD

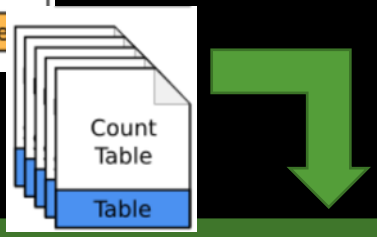
REVIEW


RNA-Seq Pipeline Workflow
FASTQ (sequencing reads)
HISAT2
Genome Index
Alignments
Cufflinks
Cuffnorm
featueCounts
Isoform Info
Normalized expression data (good for graphs)
Raw count data (input for differential expression analysis)

RNAseq skill-set recap







	Step 1	Step 2	Step 3	Step 4	NEXT
	Next-Gen sequencing at the bench: Illumina workflow	FastQ files: Quality assessment, control and processing	Mapping your reads to a reference genome `01_build_hisat_index.sh`	Generating read-counts tables (Justin's other scripts)	Differential expression analysis
Tools	BaseSpace	FastQC MultiQC	hisat2 samtools	featureCounts	deSeq2 ggplot2
Platform	Web	UNIX (student cluster)	UNIX (student cluster)	UNIX (student cluster)	RStudio (local)
Output	fastq.gz	fastq.gz	.bam .sam	.txt (table)	.txt (table) plots

What is R?

An open-source statistical programming language

- How is it different from what we did yesterday?

We used UNIX on the cluster to process our data down to transcript-counts; now we need stats and plots to analyze it!

- Why do we use it?

Free; particularly great for data-analysis and powerful visualizations; lots of ‘omics tools available

- How do we use it?

with RStudio: an **I**ntegrated **D**evelopment **E**nvironment for R

R vs. RStudio

R: Engine

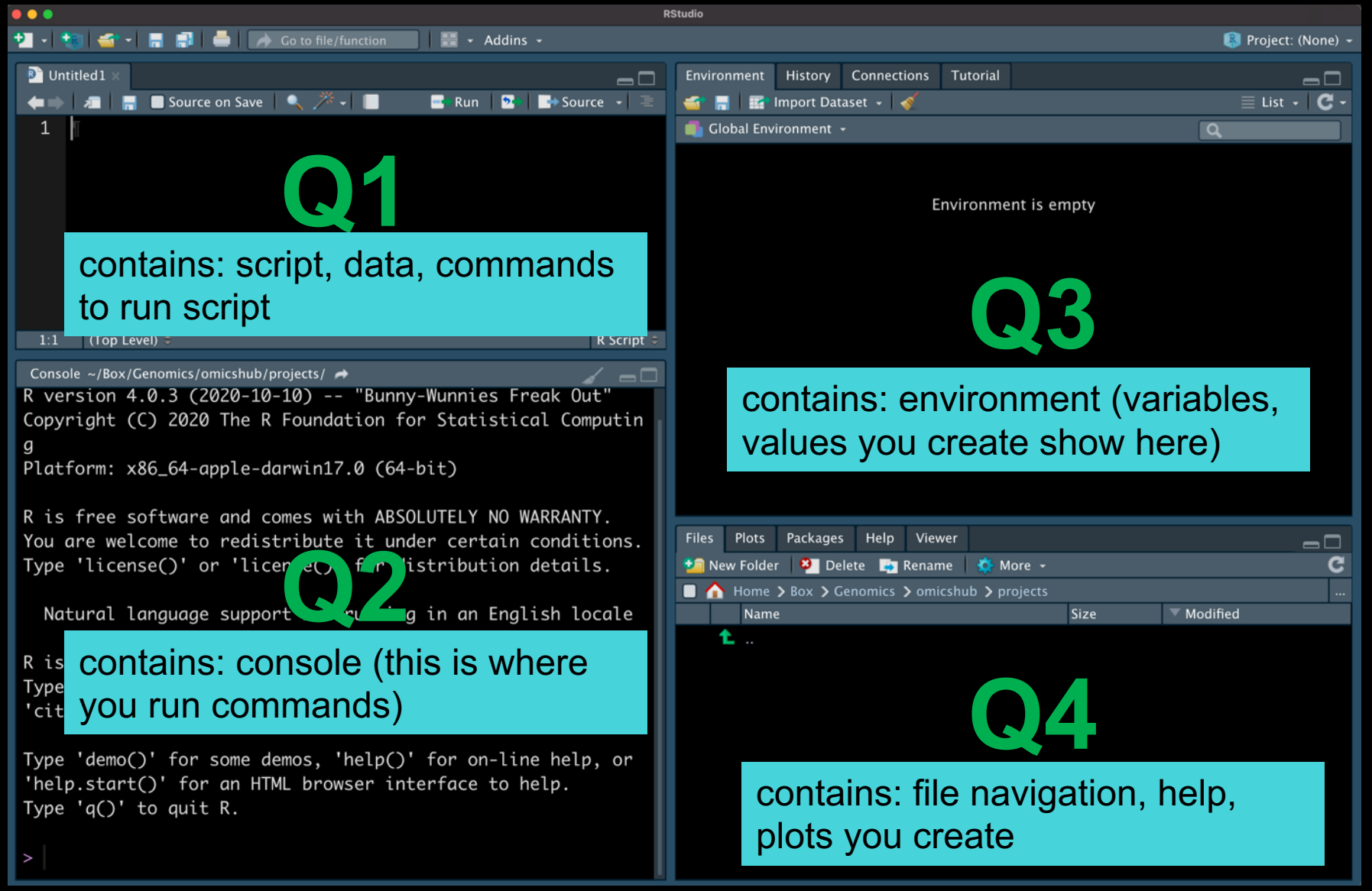
R: Do not open this

RStudio: Dashboard

RStudio: Open this

Credit: ModernDive

Looking around Rstudio: Basics



Bioconductor

https://www.bioconductor.org/about/

MISSION:

- To promote the statistical analysis and comprehension of current and emerging high-throughput biological assays.

Most *Bioconductor* components are distributed as R packages. The functional scope of Bioconductor packages includes the analysis of DNA microarray, sequence, flow, SNP, and other data.

- Bioconductor is committed to open source, collaborative, distributed software development and literate, reproducible research.

https://www.bioconductor.org/about/

PROJECT GOALS (highlights)

- To provide widespread access to a broad range of **powerful statistical and graphical methods** for the analysis of genomic data.
- To further scientific understanding by producing **high-quality documentation** and **reproducible research.**
- To train researchers on computational and statistical methods for the analysis of genomic data.

https://www.bioconductor.org/about/

PROJECT GOALS (highlights)

- To provide widespread access to a broad range of **powerful statistical and graphical methods** for the analysis of genomic data.

- Each Bioconductor package contains one or more vignettes to show you how to use them
    - SOME are useful

Further resources

- The amazing world of R-plots: inspiration and code

  https://www.r-graph-gallery.com/

- The “swirl” package: interactive R tutorials

- RStudio has its own tutorials built in too

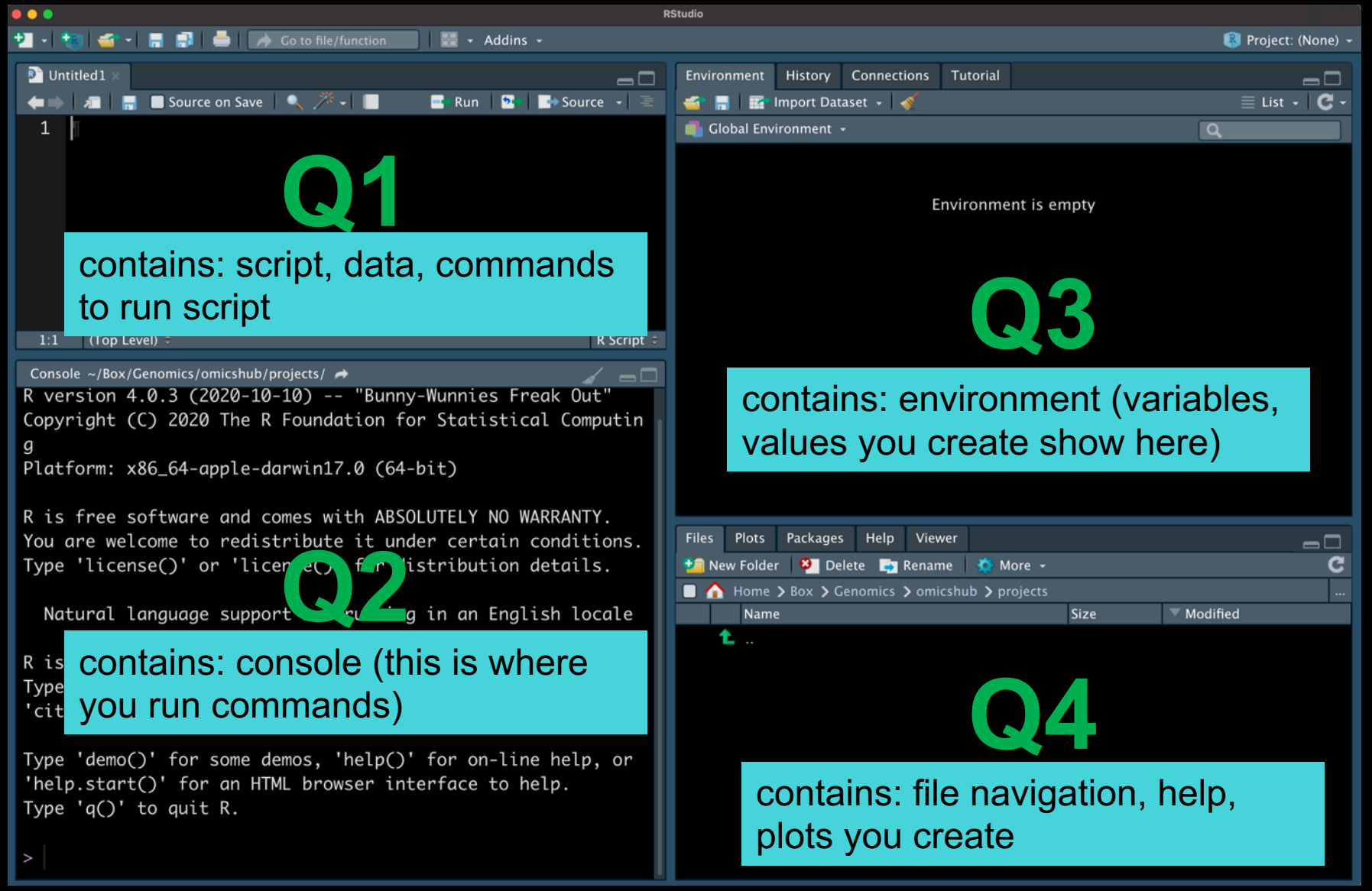
- SO MANY RESOURCES out there

Hands-on

PART 1 Exercise Objectives

- Know your way around RStudio
  - Know how to create a Project file and a script in RStudio
  - Understand the idea of a working directory and where your files live
  - Understand what happens in the “four quadrants” of RStudio

Looking around Rstudio: Basics



R language basics: (some) terminology

- objects
- operators
  - basic math (+, -, *, /)
  - assignment
- **functions**, aka **commands**

Syntax in R

- assign values to objects like this:

variable <- 1 + 2

PART 2 Installing and loading packages

- Objectives:
  - Understand what a package is
  - Understand how to install packages from the console (quadrant 2).
  - Be able to use the library function to load packages at the top of your script
  - Know how to find useful information about how to use a particular package when you are trying something new

Packages

- “Code a generous person has written, tested, bundled and gave away FOR FREE because they are so terribly nice”

- Mostly designed to solve a specific problem
  - pull together functions to solve that problem
    - e.g.: general packages for “data-wrangling”, or visualization, etc.
    - or more specific packages designed already for someone trying to do exactly what you’re trying to do, such as differential gene-expression analysis

- You could write all your own functions, and hopefully one day you will
  - BUT TODAY IS NOT THAT DAY

practice

install.packages(“swirl”)

library(swirl)

GETTING HELP FOR ANY FUNCTION
?swirl

PART 3 some R-language basics

- Objectives:
  - Complete the interactive swirl exercise
    - Know how to assign values to variables
    - Know how to create a vector

PART 4
Getting data into RStudio

- Objectives:
  - Have an example data-file saved in your “Rdata” folder
  - Know how to use the read_csv function to read your data into RStudio
  - Know how to view your data, and how to use summary functions to explore it

Example data

- Sydney beaches:

https://raw.githubusercontent.com/rladiessydney/RYouWithMe/master/sydneybeaches.csv

install.packages("skimr")

functions

function-calls tend to have this sort of structure:
function(argument1, argument2, options)

read.delim()

view()

dim()

str()

summary()

skim()

Beyond local CSV files

- But what if my data isn't in .csv format?? No problem, there is a package for that! R can read in data from almost anywhere and any format. For example, you can learn more about reading data from…
- **Excel files**: install.packages("readxl") link
- **SPSS/Stata/SAS files**: install.packages("haven") link1 link2
- **Google sheets**: install.packages("googlesheets") link
- **Copy and paste using datapasta**: install.packages("datapasta") link
- … but R has tools for working with a very wide range of possible data formats and so there are packages out there for all sorts of things.

You can refer to ‘R Tutorial.pdf’ for more R basics

- data types
  - numeric
  - characters
    - “text”
  - logicals
    - TRUE or FALSE
- data structures
  - vectors (a series of values)
  - data frames
  - lists