

Introduction to RStudio Server

The CCDL

The following two analysis examples are both microarray differential expression analyses.

Differential Analysis Example 1: “100% Up to Date Analysis”

Methods Documentation: ~200 words describe the general methods in a publication. The step-by-step recount of how these data came to be were handwritten, with one copy located in a lab closet.

Data availability: The original data are on a flashdrive, in a desk drawer, in a lab. The already processed version is on [GEO](#).

Project organization: Many files with various nebulous terms like “MasterSheet” and “Sorted” and “Edit” and “100% Up to Date”.

Software versions: No idea what software package(s), regardless of version.

IncrNAMicroarrayMasterSheet11-4-14CS								
View		Zoom		Add Category		Collaborate		
Insert		Table		Chart		Text		Shape
+	IncrNATargetStatus	IncrNATargetLLMicroarrayData	SummaryLLData	OriginalLLMicroarray	qPCRSummary	qPCRRunsDetails	IncrNAIlluminaData	TRAFluorimCelli
1	ProbeName	Sequence	GeneName	SystematicName	Description	RName	RSavedName	F
881	CHIP_JH_SE3_19777	GCTTGTATGACGCCAG CHIP_JH_SE3_19777		CHIP_JH_SE3_19777	Unknown	{ 880 }	{ 880 }_AK056136->CHIP_JH_SE3_19777 < Unknown .	
26316	CHIP_JH_SE3_19780	CTAGACAGAAACAGCACT CHIP_JH_SE3_19780		CHIP_JH_SE3_19780	Unknown	{ 26315 }	{ 26315 }_AK056136->CHIP_JH_SE3_19780 < Unknown .	
27565	CHIP_CON_ALB_196	CTAGATGAAAATAAAAGTC CHIP_CON_ALB_196		CHIP_CON_ALB_196	Unknown	{ 27564 }	{ 27564 }_NM_014178->CHIP_CON_ALB_196 < Unknown .	
28964	CHIP_CON_ALB_196	CTAGATGAAAATAAAAGTC CHIP_CON_ALB_196		CHIP_CON_ALB_196	Unknown	{ 28963 }	{ 28963 }_NM_014178->CHIP_CON_ALB_196 < Unknown .	
19195	CHIP_CON_ALB_193	GTTCGATTTTTACATTTCCC CHIP_CON_ALB_193		CHIP_CON_ALB_193	Unknown	{ 19194 }	{ 19194 }_NM_014178->CHIP_CON_ALB_193 < Unknown .	
32499	CHIP_CON_ALB_193	GTTCGATTTTTACATTTCCC CHIP_CON_ALB_193		CHIP_CON_ALB_193	Unknown	{ 32499 }	{ 32499 }_NM_014178->CHIP_CON_ALB_193 < Unknown .	
6932	CHIP_CON_ALB_196	CTAGATGAAAATAAAAGTC CHIP_CON_ALB_196		CHIP_CON_ALB_196	Unknown	{ 6937 }	{ 6937 }_NM_014178->CHIP_CON_ALB_196 < Unknown .	
773	CHIP_CON_ALB_196	CTAGATGAAAATAAAAGTC CHIP_CON_ALB_196		CHIP_CON_ALB_196	Unknown	{ 772 }	{ 772 }_NM_014178->CHIP_CON_ALB_196 < Unknown .	
33242	CHIP_JH_SE3_19779	TTATGAGAATATGGCTCAC CHIP_JH_SE3_19779		CHIP_JH_SE3_19779	Unknown	{ 33243 }	{ 33243 }_AK056136->CHIP_JH_SE3_19779 < Unknown .	
9632	CHIP_CON_ALB_193	GTTCGATTTTTACATTTCCC CHIP_CON_ALB_193		CHIP_CON_ALB_193	Unknown	{ 9631 }	{ 9631 }_NM_014178->CHIP_CON_ALB_193 < Unknown .	
9658	CHIP_JH_SE3_13875	TGTTGGATGTGTCTTCAAG CHIP_JH_SE3_13875		CHIP_JH_SE3_13875	Unknown	{ 9657 }	{ 9657 }_AK056431->CHIP_JH_SE3_13875 < Unknown .	
10134	CHIP_JH_SE3_19778	AAAGAGAAATGATATCTGTG1 CHIP_JH_SE3_19778		CHIP_JH_SE3_19778	Unknown	{ 10133 }	{ 10133 }_AK056136->CHIP_JH_SE3_19778 < Unknown .	
791	CHIP_JH_SE3_23399	AGAGCTAACACACTCTAACAGA CHIP_JH_SE3_23399		CHIP_JH_SE3_23399	Unknown	{ 790 }	{ 790 }_AK074776->CHIP_JH_SE3_23399 < Unknown .	
2421	CHIP_JH_SE3_16148	TCTTAACCCTTCTCAAA CHIP_JH_SE3_16148		CHIP_JH_SE3_16148	Unknown	{ 2420 }	{ 2420 }_AL83245->CHIP_JH_SE3_16148 < Unknown .	
35094	CHIP_JH_SE3_34118	CAGCTTCTGTATTACTTCA CHIP_JH_SE3_34118		CHIP_JH_SE3_34118	Unknown	{ 35093 }	{ 35093 }_CR598488->CHIP_JH_SE3_34118 < Unknown .	
9840	CONTROL_EPILEPSY_SE3_3	TGTTTTGGGGATGAGAGGG CONTROL_EPILEPSY_SE3_382		CONTROL_EPILEPSY_SE3_Unknown	Unknown	{ 9839 }	{ 9839 }_NM_004233->CONTROL_EPILEPSY_SE3_382 < Unknown .	
12692	CONTROL_EPILEPSY_SE3_3	TGTTTTGGGGATGAGAGGG CONTROL_EPILEPSY_SE3_382		CONTROL_EPILEPSY_SE3_Unknown	Unknown	{ 12691 }	{ 12691 }_NM_004233->CONTROL_EPILEPSY_SE3_382 < Unknown .	
34255	CONTROL_EPILEPSY_SE3_3	TGTTTTGGGGATGAGAGGG CONTROL_EPILEPSY_SE3_382		CONTROL_EPILEPSY_SE3_Unknown	Unknown	{ 34254 }	{ 34254 }_NM_004233->CONTROL_EPILEPSY_SE3_382 < Unknown .	
2994	CHIP_CON_ALB_194	GAATGAAATTTGGACCA CHIP_CON_ALB_194		CHIP_CON_ALB_194	Unknown	{ 29945 }	{ 29945 }_NM_014178->CHIP_CON_ALB_194 < Unknown .	
26892	CHIP_JH_SE3_23396	CTAACACACACAAAAAG CHIP_JH_SE3_23396		CHIP_JH_SE3_23396	Unknown	{ 26891 }	{ 26891 }_AK074776->CHIP_JH_SE3_23396 < Unknown .	
40335	CHIP_JH_SE3_23398	GAGCTAACACACCTCAAAG CHIP_JH_SE3_23398		CHIP_JH_SE3_23398	Unknown	{ 40334 }	{ 40334 }_AK074776->CHIP_JH_SE3_23398 < Unknown .	
31145	CHIP_JH_SE3_23400	GAGACGACTACACACCTCAAAG CHIP_JH_SE3_23400		CHIP_JH_SE3_23400	Unknown	{ 31144 }	{ 31144 }_AK074776->CHIP_JH_SE3_23400 < Unknown .	
3994	CHIP_JH_SE3_969	TTGTGAATGTGTGTGA CHIP_JH_SE3_969		CHIP_JH_SE3_969	Unknown	{ 3993 }	{ 3993 }_AK130883->CHIP_JH_SE3_969 < Unknown .	
14335	CONTROL_EPILEPSY_SE3_3	TGTTTTGGGGATGAGAGGG CONTROL_EPILEPSY_SE3_382		CONTROL_EPILEPSY_SE3_Unknown	Unknown	{ 14334 }	{ 14334 }_NM_004233->CONTROL_EPILEPSY_SE3_382 < Unknown .	
13000	CONTROL_EPILEPSY_SE3_2	TAGAAGCTAACACAAACCC CONTROL_EPILEPSY_SE3_275		CONTROL_EPILEPSY_SE3_Unknown	Unknown	{ 12999 }	{ 12999 }_NM_032895->CONTROL_EPILEPSY_SE3_275 < Unknown .	
32032	CONTROL_EPILEPSY_SE3_3	AGGTAGTACCAAGGACCTT CONTROL_EPILEPSY_SE3_381		CONTROL_EPILEPSY_SE3_Unknown	Unknown	{ 32031 }	{ 32031 }_NM_004233->CONTROL_EPILEPSY_SE3_381 < Unknown .	
25796	CHIP_JH_SE3_972	GGTAGATGTGCAAAGTATC CHIP_JH_SE3_972		CHIP_JH_SE3_972	Unknown	{ 25795 }	{ 25795 }_AK130883->CHIP_JH_SE3_972 < Unknown .	

100% Up to Date Log 3-3-15 CS.xlsx	
	CAGE Data
	CorrelationREV3LTRAf.spv
	FANTOMCAGEData3-4-15CS.xlsx
	PRKCQ-AS1CAGEData.txt
	PRKCQCAGEData.txt
	REV3LTRAf3IP2-AS1PromoterDataFANTOM.txt
	TRAf3IP2 Promoter Data.txt
	TRAf3IP2 Promoter Data.txt
	TRAFREVTRAf3IP2PromoterData3-4-15CS.sav
	ListofMicroRNAs3-5-15CS.xlsx
	IncRNAMicroarrayMasterSheet11-4-14CS.xlsx
	IncRNASortedDataEdit3-28-14.xlsx
	Mice LncRNAs
	Conservation_DataTable.csv
	GenomeCoordinateOutput.txt
	GenomeCoordinateOutput2.txt
	LncRNAs_NFKB Mouse Gutman 2-27-15 CS.xls
	lncRNA in Mouse Brain.txt
	LncRNA Lists 2-26-15 CS.xlsx
	lncRNAs enriched in Mouse and Human 2-24-15 CS.xls
	mmcl1 (1).pdf
	mouse_long_lncRNAs_with_orthologous_human_transcripts.xls
	pnas_0706729105_06729Table5.xls
	pnas_0706729105_06729Table7.xls
	New list of LncRNAs of Interest 1-20-15 CS.xls
	NFKB Signaling.xlsx
	NPP Format Tables and Figures IncRNA Paper
	Redone P2 P3 westerns
	Related Journal Articles
	Tox Data Analysis
	DrugsbyMonth.xls
	NMS-ToxData2014Original
	NMSData2-3-15CS.sav
	NMSData2-5-15CS.sav
	NMSDatabyMonth2-6-15CS (2).sav
	NMSDatabyMonth2-6-15CS.sav
	NMSDatazeros2-6-15CS.sav
	NMSTOAccess.accdb
	ToxDataAnalysisSPSSoutput.xlsx
	WCME-NMSDataSort2-3-15CS.xlsx
	WCME-NMSDataSort2-5-15CS.xlsx

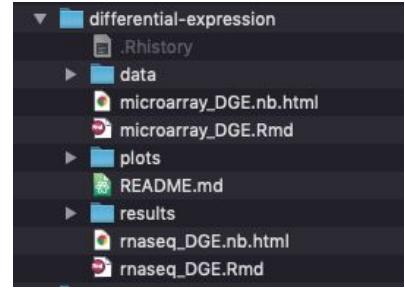
Data Analysis Example 2: “GitHubbed Analysis”

Methods Documentation: Publicly available, online notebooks show step-by-step process that can be re-run.

Data availability: Data that was originally used is [available for download online](#).

Project organization: Folders labeled, most recent is present, with [prior versions tracked on GitHub](#).

Software versions: Specific requirements and version numbers shown in notebook and [README](#).



- 1) Install libraries
- 2) Import and set up data
- 3) Set up design matrix
- 4) Apply linear model
- 5) Explore fitness of model
- 6) Write statistics to output results file

Differential Expression Analysis: Microarray

ALSF CCDEL - Candace Savonen

Purpose: This notebook takes data and metadata from refine.bio and identifies differentially expressed genes. This script is generally applicable to microarray data.

1) Install libraries

This script uses the bioconductor R package limma to identify differentially expressed genes. The full guide on [limma](#) shows examples of limma functions. Citation: Ritchie ME, Phipson B, Wu D, Hu Y, Law CW, Shi W, Smyth GK (2015). "limma powers differential expression analyses for RNA-sequencing and microarray studies." Nucleic Acids Research, 43(7), e47.

```
if (!("limma" %in% installed.packages())) {  
  # Install limma  
  BiocManager::install("limma", update = FALSE)  
}
```

Attach the `limma` library:

```
# Magrittr pipe  
`%>%` <- dplyr::`%>%`  
# Attach library  
library(limma)
```

Create output folders.

```
# Create the results folder if it doesn't exist  
if (!dir.exists("results")) {  
  dir.create("results")  
}  
# Create the plots folder if it doesn't exist  
if (!dir.exists("plots")) {  
  dir.create("plots")  
}
```

refine.bio Example Workflow: Differential expression

refine.bio includes both microarray and RNA-seq experiments. The methods for performing differential gene expression (DGE) analysis for each technology differ. In this module, we include simple two-group comparison examples for each.

Contents

- **Microarray notebook:** takes microarray data and metadata from refine.bio and identifies genes that are differentially expressed between two groups.
- **RNA-seq notebook:** takes RNA-seq data without quantile normalization from refine.bio and identifies genes that are differentially expressed between two groups. Read more about skipping quantile normalization [here](#) in our documentation.
- **GenePattern differential expression analysis:** GenePattern modules can be run via a GUI. To use refine.bio data with GenePattern, you will need to change the format as described in this section.

Requirements and usage

This module requires you to install the following software to run examples yourself:

- **R**
- **RStudio** for working with [R Notebooks](#).
- **Bioconductor**
- **tidyverse**

These requirements can be installed by following the instructions at the links above. The example R Notebooks are designed to check if additional required packages are installed and will install them if they are not.

“100% Up to Date Analysis” OR “GitHubbed Analysis”

Which analysis would you...

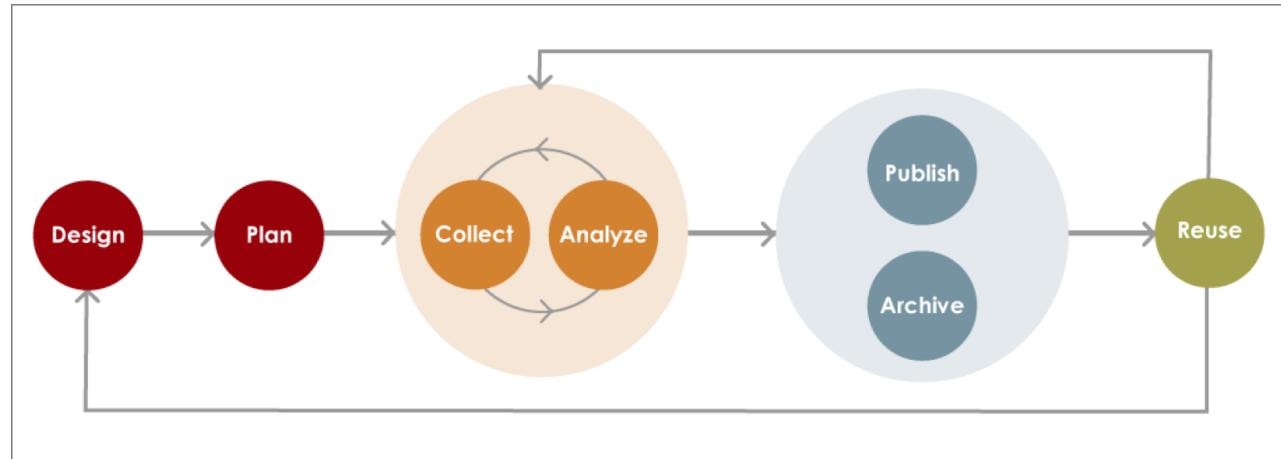
...be more inclined to borrow the methods from?

...trust the methods of more?

...feel would be *easier to reproduce*?

A word about research reproducibility

- Thoughtful **data management** and **documentation** throughout the research life cycle:
 - Project organization, file naming conventions, file back up (3+2+1)
 - Document all steps, metadata, and computational environment
 - Version control and interoperability
 - Make data, protocol and code accessible, reusable and citable



More resources: <https://guides.library.cmu.edu/reproducibility> or join our relevant workshops

Command line vs GUI (graphics user interface)

- An interface is how you interact with a program
- GUI's have buttons you can *click* to do things,
- Command-line interfaces have you *type* out things to do them

RStudio Server: A basic guide

The screenshot shows the RStudio Server interface running in a web browser. The top navigation bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help, and Sessions. The user is identified as "trainee". The Project dropdown shows "(None)" and the R version is 3.6.1.

Console Tab:

```
R version 3.6.1 (2019-07-05) -- "Action of the Toes"  
Copyright (C) 2019 The R Foundation for Statistical Computing  
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.
```

Environment Tab:

Global Environment

Environment is empty

Files Tab:

Name	Size	Modified
.R	28 B	Apr 23, 2020
.Rhistory		
shared-data		
training-modules		

Terminal Tab:

```
> |
```

The screenshot shows the RStudio interface running on a Mac OS X system. The title bar indicates the session is named "trainee" and the version is "R 3.6.1".

Console Pane: This pane is highlighted with a red border. It displays the standard R startup message, including the version number (3.6.1), copyright information, and license details. It also shows the R environment as empty.

```
R version 3.6.1 (2019-07-05) -- "Action of the Toes"  
Copyright (C) 2019 The R Foundation for Statistical Computing  
Platform: x86_64-pc-linux-gnu (64-bit)  
  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.  
  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.
```

Environment Pane: This pane shows the Global Environment, which is currently empty.

Files Pane: This pane shows the directory structure under "Home".

	Name	Size	Modified
<input type="checkbox"/>	.R	28 B	Apr 23, 2020
<input type="checkbox"/>	.Rhistory		
<input type="checkbox"/>	shared-data		
<input type="checkbox"/>	training-modules		

The screenshot shows the RStudio interface with several panes open:

- Top Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help, trainee, Sessions, Project: (None), R 3.6.1.
- Toolbar:** Includes tabs for Console, Terminal, Go to file/function, Addins, and Project: (None).
- Console Tab:** Shows "Terminal 1" with the path "/home/may2020/trainee". A yellow box highlights the tabs, and a blue arrow points to the "Terminal" tab with the text "tabs to switch".
- Terminal Content:** Displays the command "trainee:~\$".
- Environment Tab:** Shows the Global Environment with the message "Environment is empty".
- Files Tab:** Shows the contents of the current directory:

	Name	Size	Modified
<input type="checkbox"/>	.R		
<input type="checkbox"/>	.Rhistory	28 B	Apr 23, 2020
<input type="checkbox"/>	shared-data		
<input type="checkbox"/>	training-modules		
- Plots Tab:** Not visible in the screenshot.

The Terminal:
Where you tell the computer (outside R) what to do through command line instructions

The screenshot shows the RStudio interface with a red box highlighting the Terminal and Environment panes.

Terminal Pane:

- Shows the command prompt: `trainee:~$`.
- Two arrows point from the text "These indicate what **directory** you are **currently** carrying out a command in" to the tilde (~) and the dollar sign (\$).
- The path `/home/may2020/trainee` is displayed above the prompt.

Environment Pane:

- Shows the Global Environment tab.
- Message: "Environment is empty".

File Browser:

- Shows the contents of the "Home" directory.
- Table of contents:

Name	Size	Modified
.R	28 B	Apr 23, 2020
.Rhistory		
shared-data		
training-modules		

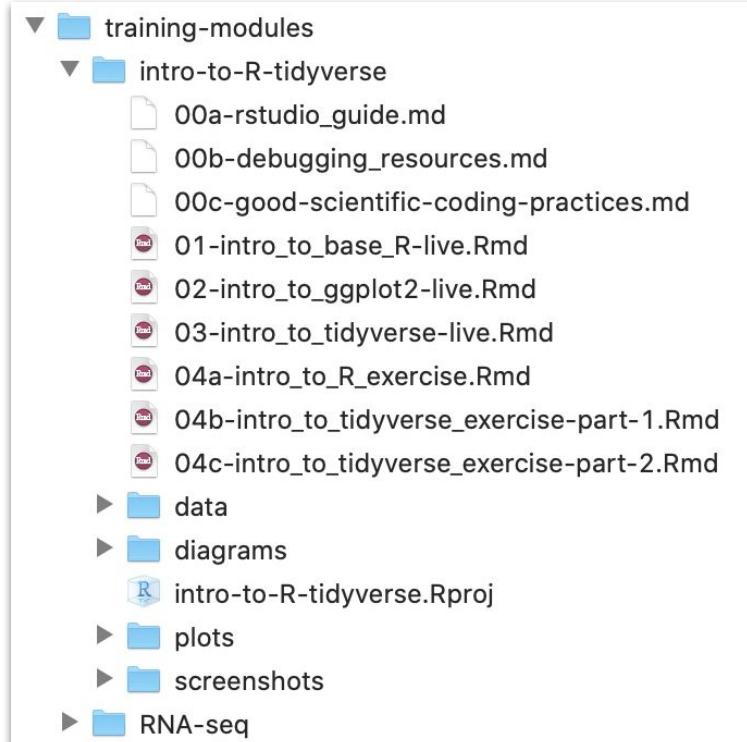
These indicate what **directory** you
are **currently** carrying out a command in

This is called your "**current directory**"

"**~**" is a shortcut for your "**Home**"
directory, so these mean the same thing.

Directories = Folders

When we are doing things through command line, we have to look at where the files we are using are being kept.



The screenshot shows the RStudio interface running on a Mac OS X system. The top bar includes standard OS X window controls (red, yellow, green, close, minimize, maximize) and the URL bar showing `rstudio.ccdatalab.org`. The main menu bar has options: File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help. The top right corner shows the user name "trainee", session status, and R version "R 3.6.1".

Console pane: Shows the terminal session. The user runs `ls` to list files in the current directory (`/home/may2020/trainee`) and then changes the directory to `training-modules`.

```
trainee:~$ ls
shared-data  training-modules
trainee:~$ cd training-modules
```

Environment pane: Shows the global environment. It displays a message: "Environment is empty".

Files pane: Shows the contents of the current directory. It lists four items: ".R", ".Rhistory", "shared-data", and "training-modules".

	Name	Size	Modified
<input type="checkbox"/>	.R		
<input type="checkbox"/>	.Rhistory	28 B	Apr 23, 2020
<input type="checkbox"/>	shared-data		
<input type="checkbox"/>	training-modules		

Plots pane: This pane is currently empty.

trainee:~\$ ls

shared-data training-modules

trainee:~\$ cd training-modules

trainee:~/training-modules\$ ls

RNA-seq intro-to-R-tidyverse

trainee:~/training-modules\$

Note that the words before where our cursor is has changed to reflect that we are in the "training-modules" directory

ls - how to tell the computer to **list** the files and folders in a directory
(files that start with a '.' are not shown by default)

cd - how to **change** directories

Environment is empty

Files Plots Packages Help Viewer

New Folder Upload Delete Rename More

Home

	Name	Size	Modified
<input type="checkbox"/>	.R		
<input type="checkbox"/>	.Rhistory	28 B	Apr 23, 2020
<input type="checkbox"/>	shared-data		
<input type="checkbox"/>	training-modules		

The screenshot shows the RStudio interface on a Mac OS X system. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help, and a user-specific section for 'trainee'. The top right corner shows session details ('trainee' at home), a refresh icon, and a plus sign for new sessions.

The main workspace is divided into several panes:

- Console:** Displays a terminal session:

```
trainee:~$ ls
shared-data training-modules
trainee:~$ cd training-modules
trainee:~/training-modules$ ls
RNA-seq intro-to-R-tidyverse
trainee:~/training-modules$
```
- Environment:** Shows the global environment, which is currently empty.
- Files:** Shows the contents of the current directory:

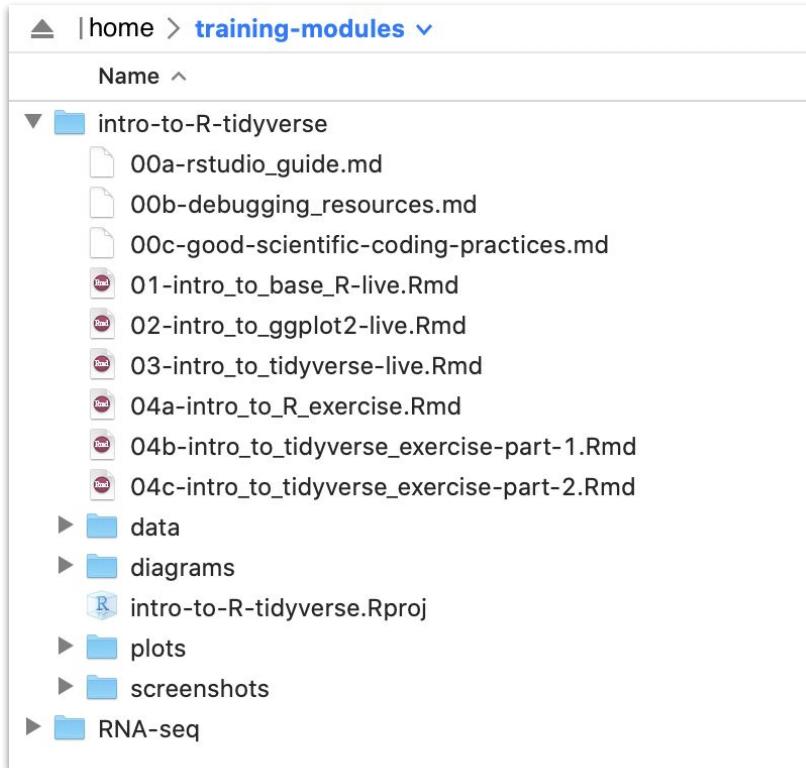
	Name	Size	Modified
<input type="checkbox"/>	.R		
<input type="checkbox"/>	.Rhistory	28 B	Apr 23, 2020
<input type="checkbox"/>	shared-data		
<input type="checkbox"/>	training-modules		

A large blue text overlay in the bottom right corner reads:

The files tab over here
does NOT reflect your current
directory or any changes within it

File paths: Directions to a file or folder

Let's say we want access to “**01-intro_to_base_R-live.Rmd**”

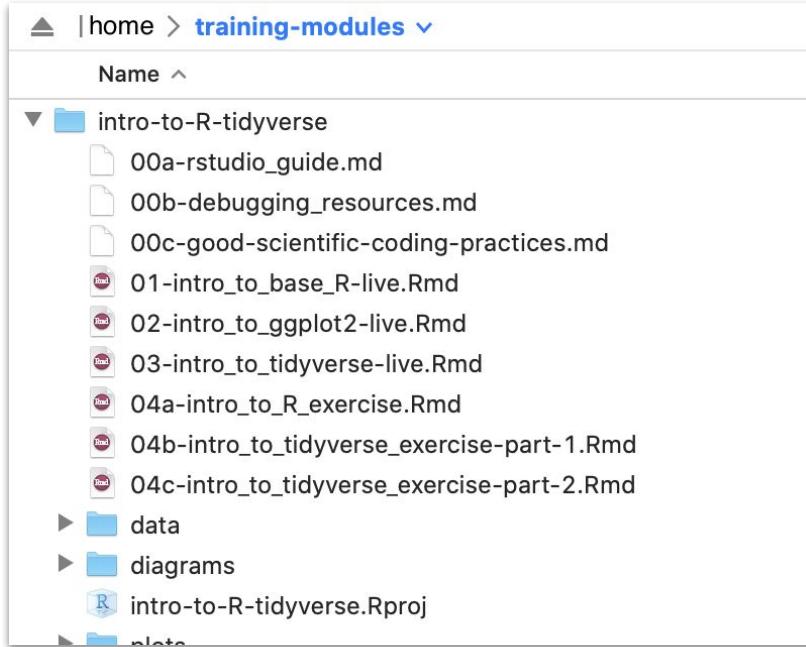


Current directory = “**training-modules**”

File path = “**intro-to-R-tidyverse/01-intro_to_base_R-live.Rmd**”

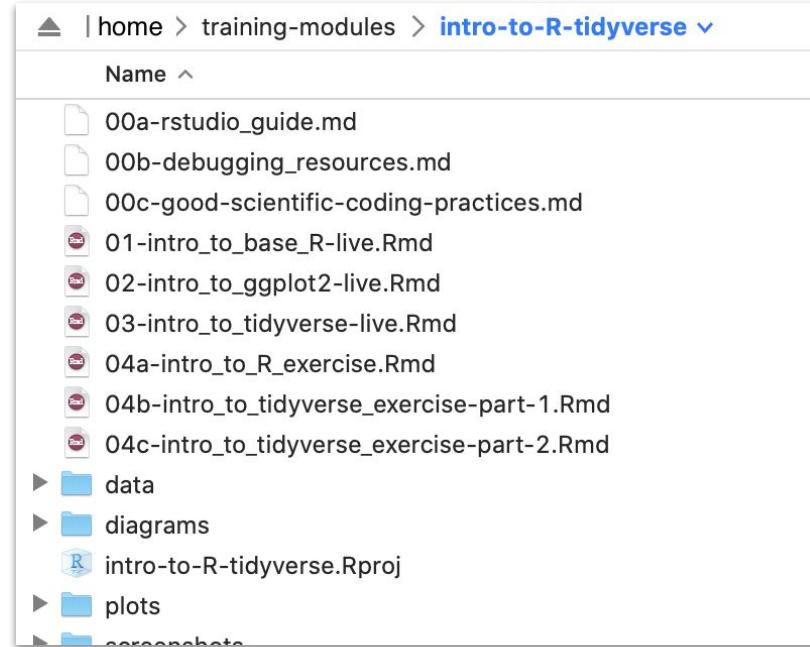
File Paths can be *relative*

Let's say we want access to “01-intro_to_base_R-live.Rmd”



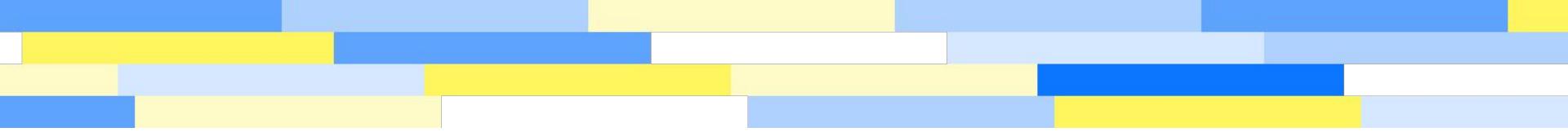
Current directory = “**training-modules**”

Relative file path =
“**intro-to-R-tidyverse/01-intro_to_base_R-live.Rmd**”



Current directory = “**training-modules/intro-to-R-tidyverse**”

Relative file path = “**01-intro_to_base_R-live.Rmd**”



Introduction to R

The CCDL

R programming

Programming: making executable scripts for accomplishing a task
(in this case, data analysis is our task)

Scripts allow others to see, step-by-step, what you did.

Why we use R:

- It's free
- People make cool packages that do stuff for us
- Many researchers in genomics use it (as well as Python)

One in five genetics papers contains errors thanks to Microsoft Excel

By Jessica Boddy | Aug. 29, 2016 , 1:45 PM

What you type	What you see	How Excel stores it
MARCH1	1-MAR	42430
SEPT2	2-SEP	42615

<https://www.sciencemag.org/news/2016/08/one-five-genetics-papers-contains-errors-thanks-microsoft-excel>

Ziemann et al. *Genome Biology* (2016) 17:177 DOI 10.1186/s13059-016-1044-7

R Notebooks

Use the "Files" tab to open: [training-modules/intro-to-R-tidyverse/01-intro_to_base_R-live.Rmd](#)

The screenshot shows the RStudio interface with the following details:

- Editor Tab:** The main window displays the content of the file `01-intro_to_base_R-live.Rmd`. The code includes YAML front matter and a list of objectives.
- Environment Tab:** Shows the Global Environment, which is currently empty.
- Files Tab:** The "Files" tab is active, showing the project structure. A red box highlights the path `Home > training-modules > intro-to-R-tidyverse`, and another red box highlights the file `01-intro_to_base_R-live.Rmd` in the list of files.
- File List:** The list of files in the current directory includes:

Name	Size	Modified
00a-rstudio_guide.md	3.4 KB	Apr 29, 2020
00b-debugging_resources.md	13.6 KB	Apr 29, 2020
00c-good-scientific-coding-pr...	3.9 KB	Apr 29, 2020
01-intro_to_base_R-live.Rmd	16 KB	Apr 29, 2020
02-intro_to_ggplot2-live.Rmd	9.9 KB	Apr 29, 2020
03-intro_to_tidyverse-live.Rmd	16.4 KB	Apr 29, 2020
04a-intro_to_R-exercise.Rmd	4.1 KB	Apr 29, 2020

R Notebooks

- R Notebooks allow you to have files that show both your code and results

Output from above code chunk

Executable code chunk

Can click here to run a code chunk

The screenshot shows the RStudio interface with an R Notebook open. The notebook contains the following text:

```
82 For example, we can do some simple multiplication like
83 When you execute code within the notebook, the results
84 appear beneath the code.
85 Try executing this chunk by clicking the *Run* button
86 within the chunk or by
87 placing your cursor inside it and pressing
88 *Cmd+Shift+Enter*.
89
90 ````{r}
91 5 * 6
92 ````
```

The code chunk at lines 87-92 is highlighted with a red rectangle. The output of the code, '[1] 30', is highlighted with a blue rectangle and has a green arrow pointing to the 'Run' button in the toolbar above the code chunk.

The RStudio environment pane shows the Global Environment is empty. The file browser pane shows a folder structure under 'Home > training-modules > intro-to-R-tic'. The bottom pane shows the 'Console' tab with the command '# Introduction to R and RStudio' entered.

R Notebooks

- Code that runs in R Notebooks uses wherever the file is saved as its current directory

The screenshot shows the RStudio interface with an R Notebook open. The notebook file is titled "01-intro_to_base_R-live.Rmd". The code chunk at line 87 contains the expression `5 * 6`, which is evaluated in the console at line 1:1 and returns the result `[1] 30`. The RStudio environment pane shows that the global environment is empty. The files pane displays a folder structure: Home > training-modules > intro-to-R-tidyverse. A red box highlights the path "Home > training-modules > intro-to-R-tidyverse" in the files list.

```
82 For example, we can do some simple multiplication like
83 this.
84 When you execute code within the notebook, the results
85 appear beneath the code.
86 Try executing this chunk by clicking the *Run* button
87 within the chunk or by
88 placing your cursor inside it and pressing
89 *Cmd+Shift+Enter*.

[1] 30

90
91 Use the console to calculate other expressions. Standard
92 order of operations applies (mostly), and you can use
93 parentheses `()` as you might expect (but not brackets `[]` or
94 braces `{}`), which have special meanings). Note however,
95 that you must always specify multiplication with `*`;
96 implicit multiplication such as `10(3 + 4)` or `10x` will
97 not work as expected.

# Introduction to R and RStudio
```

File Edit Code View Plots Session Build Debug Profile Tools Help trainee Sessions Project: (None) R 3.6.1

01-intro_to_base_R-live.Rmd*

82 For example, we can do some simple multiplication like this.

83 When you execute code within the notebook, the results appear beneath the code.

84 Try executing this chunk by clicking the ***Run*** button within the chunk or by

85 placing your cursor inside it and pressing ***Cmd+Shift+Enter***.

86

87 ````{r}`

88 `5 * 6`

89 `````

[1] 30

90

91 Use the console to calculate other expressions. Standard order of operations applies (mostly), and you can use parentheses `()` as you might expect (but not brackets `[]` or braces `{}`, which have special meanings). Note however, that you must ****always**** specify multiplication with `*`; implicit multiplication such as `10(3 + 4)` or `10x` will

1:1 # Introduction to R and RStudio

R Markdown

Console

Environment History Connections

Import Dataset

Global Environment

Environment is empty

Files Plots Packages Help Viewer

New Folder Upload Delete Rename More

Home > training-modules > intro-to-R-tidyverse

Name	Size	Modified
00a-rstudio_guide.md	3.4 KB	Apr 29, 2020
03-intro_to_tidyverse-live.Rmd	16.4 KB	Apr 29, 2020
04a-intro_to_R_exercise.Rmd	4.1 KB	Apr 29, 2020
..		
3.6 KB		
9 KB		
6 KB		
9 KB		

Click here to show the Console

The screenshot shows the RStudio interface with the following components:

- Top Bar:** Contains the RStudio logo, file menu (File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help), user information (trainee), session management (Sessions), and project settings (Project: (None), R 3.6.1).
- Left Panel:** Shows the code editor for a file named "01-intro_to_base_R-live.Rmd". The code includes a block of R code (lines 113-118) and its output ([1] 5.5). Below this, text explains variable modification (lines 119-122). The file tab also lists "# Introduction to R and RStudio" and "R Markdown".
- Environment Pane:** Displays the current environment variables. It shows "x" with a value of "5.5".
- Files Pane:** Shows the directory structure under "Home > training-modules > intro-to-R-tidyverse". The files listed are:

Name	Size	Modified
00a-rstudio_guide.md	3.4 KB	Apr 29, 2020
00b-debugging_resources.md	13.6 KB	Apr 29, 2020
00c-good-scientific-coding-pr... 01-intro_to_base_R-live.Rmd	3.9 KB	Apr 29, 2020
02-intro_to_ggplot2-live.Rmd	16 KB	Apr 29, 2020
03-intro_to_tidyverse-live.Rmd	9.9 KB	Apr 29, 2020
04a-intro_to_R_exercise.Rmd	16.4 KB	Apr 29, 2020
04b-intro_to_tidyverse_exercise.Rmd	4.1 KB	Apr 29, 2020
- Console Pane:** Contains the R console history:

```
> x <- 5.5
> R Console:
> x
What you are actually telling R to do
[1] 5.5
>
```

The "Console" tab is highlighted with a red box.

The screenshot shows the RStudio interface with several panes:

- Code/Script Editor:** The main editor pane displays an R Markdown script titled "01-intro_to_base_R-live.Rmd". The code includes a block of R code and a narrative text block. A red box highlights the entire editor area.
- Environment Pane:** Located on the right, it shows the current state of the R environment. It includes tabs for "Environment", "History", and "Connections". Under "Values", there is a table:

	x	5.5
--	---	-----
- Files Pane:** Also on the right, it shows the file structure under "Home > training-modules > intro-to-R-tidyverse". A red box highlights this pane.

Name	Size	Modified
00a-rstudio_guide.md	3.4 KB	Apr 29, 2020
00b-debugging_resources.md	13.6 KB	Apr 29, 2020
00c-good-scientific-coding-pr... 01-intro_to_base_R-live.Rmd	3.9 KB 16 KB	Apr 29, 2020
02-intro_to_ggplot2-live.Rmd	9.9 KB	Apr 29, 2020
03-intro_to_tidyverse-live.Rmd	16.4 KB	Apr 29, 2020
04a-intro_to_R_exercise.Rmd	4.1 KB	Apr 29, 2020
- Console:** At the bottom left, it shows the command-line history:

```
>  
> x <- 5.5  
>  
> x  
[1] 5.5  
>
```
- Top Bar:** Includes tabs for File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help, and Addins. It also shows the user "trainee", the project "Project: (None)", and the version "R 3.6.1".

The screenshot shows the RStudio interface with a red box highlighting the "Environment" tab in the top navigation bar. The "Global Environment" pane is open, displaying the variable "x" with the value "5.5". Below this, a red box contains the text "R environment: What R knows and remembers for you".

File Edit Code View Plots Session Build Debug Profile Tools Help trainee Sessions Addins Go to file/function Project: (None) R 3.6.1

01-intro_to_base_R-live.Rmd*

```
113
114 ``{r}
115 x <- 5.5
116
117 x
118 ```

[1] 5.5
```

119
120 We can modify an existing variable by reassigning it to its same name.
121 Here we'll add `2` to `x` and reassign the result back to `x`.
122

Introduction to R and RStudio R Markdown

Console Terminal Jobs

```
>
> x <- 5.5
>
> x
[1] 5.5
>
```

Environment History Connections

Import Dataset

Global Environment

Values

x	5.5
---	-----

R environment:
What R knows and remembers for you

Files Plots Packages Help Viewer

New Folder Upload Delete Rename More

Home > training-modules > intro-to-R-tidyverse

Name	Size	Modified
00a-rstudio_guide.md	3.4 KB	Apr 29, 2020
00b-debugging_resources.md	13.6 KB	Apr 29, 2020
00c-good-scientific-coding-pr... 01-intro_to_base_R-live.Rmd	3.9 KB	Apr 29, 2020
02-intro_to_ggplot2-live.Rmd	16 KB	Apr 29, 2020
03-intro_to_tidyverse-live.Rmd	9.9 KB	Apr 29, 2020
04a-intro_to_R_exercise.Rmd	16.4 KB	Apr 29, 2020
	4.1 KB	Apr 29, 2020

The screenshot shows the RStudio interface with the following components:

- Top Bar:** Contains the R logo, File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help, and a session identifier "trainee".
- Toolbar:** Includes icons for file operations like Open, Save, Print, and Addins.
- Code Editor:** Displays the file "01-intro_to_base_R-live.Rmd" with the following content:

```
113
114 ``{r}
115 x <- 5.5
116
117 x
118
```

[1] 5.5

119
120 We can modify an existing variable by reassigning it to its same name.
121 Here we'll add `2` to `x` and reassign the result back to `x`.
122
- Console:** Shows the command `> x <- 5.5` and its output [1] 5.5.
- Environment Tab:** Shows the variable `x` with the value `5.5`.
- Files Tab:** A red box highlights this tab, showing the file structure:

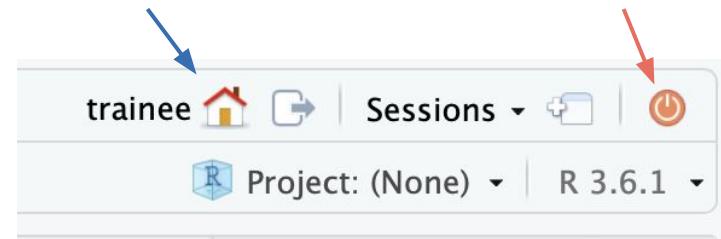
Name	Size	Modified
00a-rstudio_guide.md	3.4 KB	Apr 29, 2020
00b-debugging_resources.md	13.6 KB	Apr 29, 2020
00c-good-scientific-coding-pr... 01-intro_to_base_R-live.Rmd	3.9 KB	Apr 29, 2020
02-intro_to_ggplot2-live.Rmd	16 KB	Apr 29, 2020
03-intro_to_tidyverse-live.Rmd	9.9 KB	Apr 29, 2020
04a-intro_to_R_exercise.Rmd	16.4 KB	Apr 29, 2020
	4.1 KB	Apr 29, 2020

Other Assistance Tabs:
Things that help you in your coding

RStudio Sessions

- On the server, R run is running many times at once
 - Each user has their own “Session” running, with its own memory and processes
 - It is possible for a user to have more than one session at a time
- We will usually want to start new sessions between notebooks to keep the environment clean

Go to the sessions page



End the current session

Session Page

The 'Session Page' interface for RStudio Server Pro. At the top, it says 'R Studio Server Pro' and shows 'trainee Logout'. Below that is a navigation bar with 'Sessions' (selected), '+ New Session', '(Home)', 'Projects', '+ New Project', and 'Open Project'. The main area is titled 'Currently running sessions' and shows a single session entry:

	(Home)	• IDLE	R 3.6.1
		(Home)	CREATED: 12:59:54 PM
		LAST USED: 12:59:55 PM	

Below this, there are 'Info', 'Suspend', and 'Quit' buttons.