

DMAC Training Modules

Kyle Roell, Lauren Koval, Julia Rager

2021-06-04

Contents

1	Introduction	5
2	Setting Up Your R Environment	7
2.1	R and RStudio	7
2.2	Scripting Basics	13
3	The Basics for Data Organization	17
3.1	Basic Data Manipulation	17
4	Finding and Visualizing Data Trends	19
4.1	Heat maps	19
4.2	Clustering	19
4.3	Data reduction (PCA)	20
4.4	Basic Statistical Tests and Visualizations of Data	20
5	Multi-Omics Analyses for Environmental Health	21
5.1	Exposomics	21
5.2	Transcriptomics	21
5.3	Genome-wide MicroRNA	21
5.4	Genome-wide DNA Methylation	21
5.5	Proteomics	22
6	Mixtures Analyses for Environmental Health	23
6.1	Sufficient Similarity	23
6.2	Mixtures Modeling through qgcomp	23

7	Environmental Health Databases	25
7.1	Comparative Toxicogenomics Database (CTD)	25
7.2	Gene Expression Omnibus (GEO)	25
7.3	NHANES	25

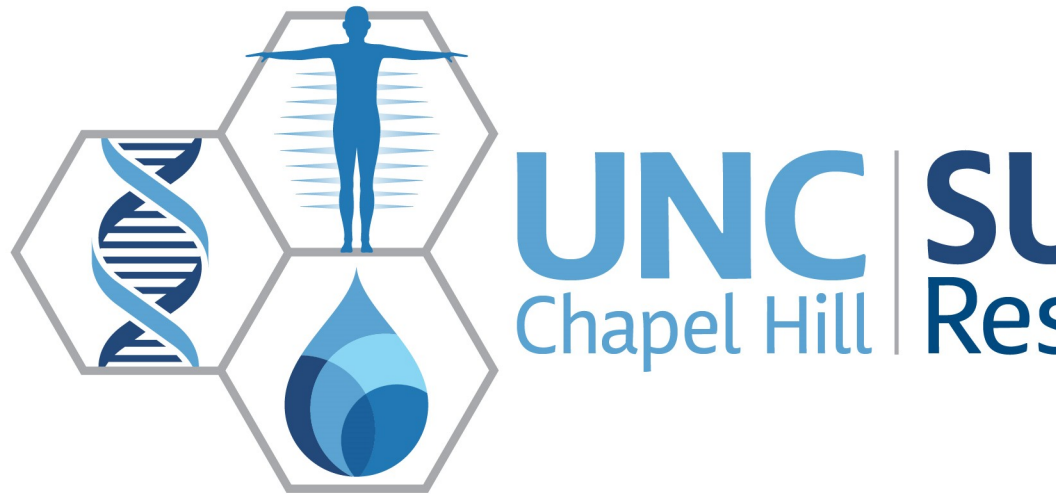
Chapter 1

Introduction

The UNC-Superfund Research Program (SRP) seeks to develop new solutions for reducing exposure to inorganic arsenic and prevent arsenic-induced diabetes through mechanistic and translational research.

The Data Analysis and Management Core (DMAC) provide the UNC-Superfund Research Program with critical expertise in bioinformatics, statistics, data management and data integration. Our goal is to support the data management, integration, and analysis needs of the researchers to reveal multi-factorial determinants of inorganic arsenic-induced metabolic dysfunction/diabetes.

All code for these modules can be found at the [UNC-SRP Github Page](#).



Chapter 2

Setting Up Your R Environment

Before learning about data manipulation and statistical methods for analyzing environmental health datasets, we will provide a brief introduction to R, RStudio, and setting up an R environment and simple scripts.

2.1 R and RStudio

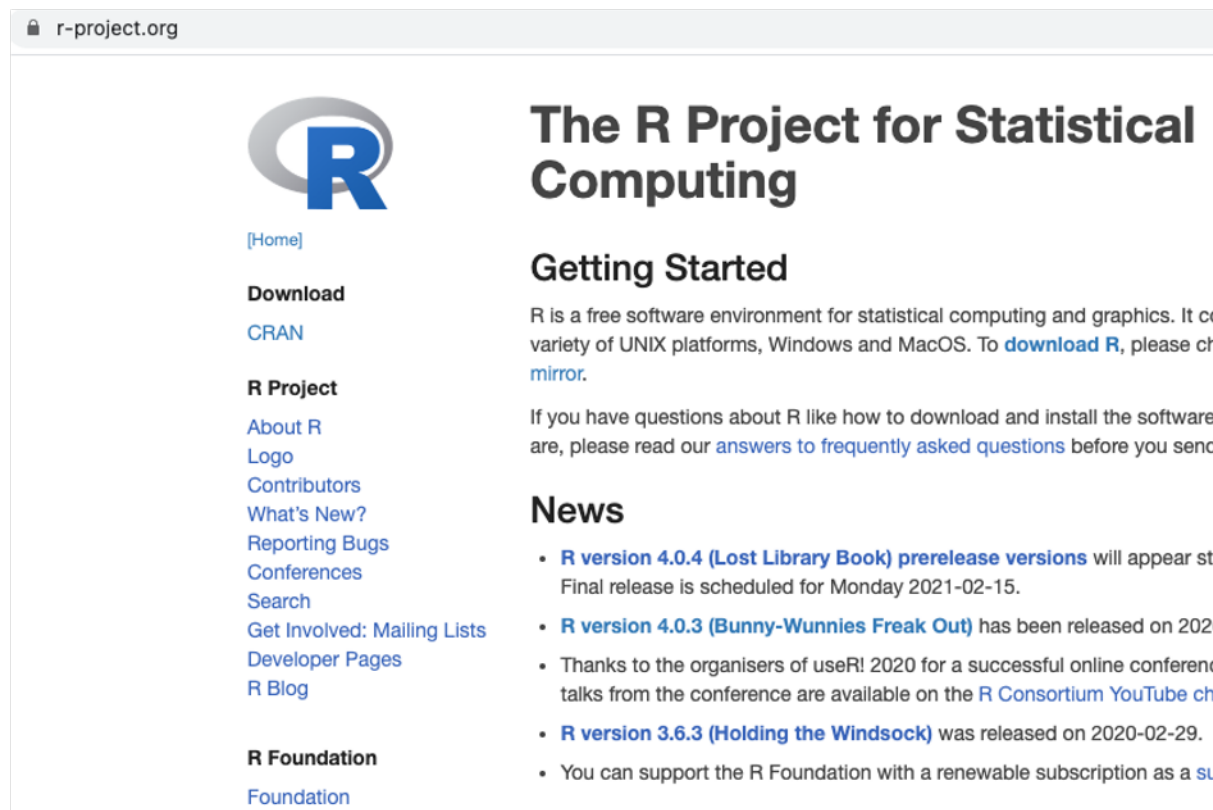
R is a free, open source programming language for statistical computing and graphics that anyone can download and use. It doesn't require a license and is good for reproducible analyses. There exists a large, diverse collection of packages and very comprehensive documentation.

It is easy to download, install, and setup R. Additionally, RStudio is an open source integrated development environment for R. RStudio makes programming in R and using R scripts and features more user friendly. R should be downloaded prior to downloading RStudio.

2.1.1 Downloading R and RStudio

The following is a walkthrough on how to download R and RStudio.

1. **Navigate to R Website**
2. **Select the appropriate CRAN mirror (Duke is fastest if at UNC)**
3. **Select the appropriate R distribution**
4. **Download R**
5. **Navigate to RStudio website and download RStudio (free edition)**

Figure 2.1: R Website, <https://www.r-project.org>

CRAN Mirrors

The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. Some statistics on the status of the mirrors can be found here: [main page](#).

If you want to host a new mirror at your institution, please have a look at the [CRAN Mirror HOWTO](#).

0-Cloud	https://cloud.r-project.org/	Automatic redirection to servers worldwide, currently sponsored by Rstudio
Algeria	https://cran.usthb.dz/	University of Science and Technology Houari Boumediene
Argentina	http://mirror.fcaglp.unlp.edu.ar/CRAN/	Universidad Nacional de La Plata
Australia	https://cran.csiro.au/ https://mirror.aarnet.edu.au/pub/CRAN/ https://cran.ms.unimelb.edu.au/ https://cran.curtin.edu.au/	CSIRO AARNET School of Mathematics and Statistics, University of Melbourne Curtin University
Austria	https://cran.wu.ac.at/	Wirtschaftsuniversität Wien
Belgium	https://www.freeststatistics.org/cran/ https://lib.ugent.be/CRAN/	Patrick Wessa Ghent University Library
Brazil	https://nbcgib.uesc.br/mirrors/cran/ https://cran-rc3sl.ufpr.br/ https://cran.fiocruz.br/ https://vps.fmvz.usp.br/CRAN/ https://briegeer.esalq.usp.br/CRAN/	Computational Biology Center at Universidade Estadual de Santa Cruz Universidade Federal do Parana Oswaldo Cruz Foundation, Rio de Janeiro University of Sao Paulo, Sao Paulo University of Sao Paulo, Piracicaba



USA	https://mirror.las.iastate.edu/CRAN/ http://ftp.usgs.iu.edu/CRAN/ https://rweb.crmda.ku.edu/cran/ https://repo.miserver.it.umich.edu/cran/ http://cran.wustl.edu/ http://archive.linux.duke.edu/cran/ https://cran.case.edu/ https://ftp.osuosl.org/pub/cran/ http://lib.stat.cmu.edu/R/CRAN/ http://cran.mirrors.hoobly.com/ https://mirrors.nics.utk.edu/cran/ https://cran.microsoft.com/	Iowa State University, Ames, IA Indiana University University of Kansas, Lawrence, KS MBNI, University of Michigan, Ann Arbor, MI Washington University, St. Louis, MO Duke University, Durham, NC Case Western Reserve University, Cleveland, OH Oregon State University Statlib, Carnegie Mellon University, Pittsburgh, PA Hoobly Classifieds, Pittsburgh, PA National Institute for Computational Sciences, Oak Ridge, TN Revolution Analytics, Dallas, TX
-----	--	---

Figure 2.2: CRAN Mirror, <https://cran.r-project.org/mirrors.html>

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want to download these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code, which must be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2020-10-10, Bunny-Wunnies Freak Out) [R-4.0.3.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features](#) and [bug reports](#) filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read the [frequently asked questions](#) before you send an email.

Figure 2.3: R Download Link, <http://archive.linux.duke.edu/cran/>

R for Mac OS X

This directory contains binaries for a base distribution and packages to run on Mac OS X (release 10.6 and above). Mac OS 8.6 to 9.2 (and Mac OS X 10.1) are no longer supported but you can find releases for these systems (which is R 1.7.1) [here](#). Releases for old Mac OS X systems (through Mac OS X 10.5) and PowerPC Macs can be found in the [old](#) directory.

Note: CRAN does not have Mac OS X systems and cannot check these binaries for viruses. Although we take precautions when assembling binaries, please use the normal precautions with any software you download from the Internet.

Package binaries for R versions older than 3.2.0 are only available from the [CRAN archive](#) so users of such versions should adjust the CRAN mirror setting (<https://cran.archive.r-project.org/>).

R 4.0.3 "Bunny-Wunnies Freak Out" released on 2020/10/10

Please check the MD5 checksum of the downloaded image to ensure that it has not been tampered with or corrupted during the mirroring process. For example type

```
openssl sha1 R-4.0.3.pkg
```

in the *Terminal* application to print the SHA1 checksum for the R-4.0.3.pkg image. On Mac OS X 10.7 and later you can also validate the signature using

```
pkgutil --check-signature R-4.0.3.pkg
```

R-4.0.3.pkg (notarized and signed)

SHA1-hash: 8402f586aef1fdb12c6c34c73b286697318db1bc
(ca. 85MB)

[NEWS](#) (for Mac GUI)

[Mac-GUI-1.73.tar.gz](#)

SHA1-hash: 764b1d050757ce78545bdeb9d178a69d13046aa1

Note: Previous R versions for El Capitan can be found in the [el-capitan/base](#) directory.

Latest release:

R 4.0.3 binary for macOS 10.13 (High Sierra) and higher, signed and notarized package. Contains R 4.0.3 framework, R.app for Intel Macs, Tcl/Tk 8.6.6 X11 libraries and Texinfo 6.7. The latter two components are optional and can be omitted when installing, they are only needed if you want to use the `tcltk` R package or build package documentation from sources.

Note: the use of X11 (including `tcltk`) requires [XQuartz](#) to be installed since it is no longer part of OS X. Always re-install XQuartz when upgrading your macOS to a new major version.

Important: this release uses Xcode 10.1 and GNU Fortran 8.2. If you wish to compile R packages from sources, you will need Xcode 10.1 and GNU Fortran 8.2 - see the [tools](#) directory.

News features and changes in the R.app Mac GUI


Sources for the R.app GUI 1.73 for Mac OS X. This file is only needed if you want to join the development of the GUI, it is not needed by regular users. Read the INSTALL file for further instructions.

Figure 2.4: R Download, <http://archive.linux.duke.edu/cran/bin/macosx/>

RStudio Desktop 1.4.1103

[- Release Notes](#)


1. Install R. RStudio requires R 3.0.1+.
2. Download RStudio Desktop. Recommended for your system:



DOWNLOAD RSTUDIO FOR MAC

1.4.1103 | 152.77MB

Requires macOS 10.13+ (64-bit)



All Installers

Linux users may need to [import RStudio's public code-signing key](#) prior to installation, depending on the operating system's security policy. RStudio requires a 64-bit operating system. If you are on a 32 bit system, you can use an [older version of RStudio](#).

OS	Download	Size	SHA-256
Windows 10/8/7	RStudio-1.4.1103.exe	156.96 MB	c3384189
macOS 10.13+	RStudio-1.4.1103.dmg	152.77 MB	20148bd6
Ubuntu 16	rstudio-1.4.1103-amd64.deb	119.26 MB	f0857e27
Ubuntu 18/Debian 10	rstudio-1.4.1103-amd64.deb	120.30 MB	76864349
Fedora 19/Red Hat 7	rstudio-1.4.1103-x86_64.rpm	138.02 MB	8fcb2d29
Fedora 28/Red Hat 8	rstudio-1.4.1103-x86_64.rpm	138.01 MB	e2bf11e9
Debian 9	rstudio-1.4.1103-amd64.deb	120.45 MB	4a4d159c
OpenSUSE 15	rstudio-1.4.1103-x86_64.rpm	122.02 MB	fdc33f7a

Figure 2.5: RStudio Download, <https://rstudio.com/products/rstudio/download/>

2.1.2 Installing R and RStudio

Once R and RStudio have been downloaded, install R first and then RStudio, following the instructions of the installer.

2.1.3 Installing and Loading Packages

Packages in R are units of shareable code that contain functions, data, and documentation on how to use all of these resources. Because R is an open source programming language, packages are constantly being developed and updated. There are many R packages that exist spanning many topics such as graphics and plotting, machine learning, and data manipulation. R packages are often written by R users and submitted to the Comprehensive R Archive Network (CRAN), or another host such as BioConductor or GitHub.

Packages can be installed from the host, but need to be loaded into the workspace. Most of the time, you do not need to download anything from a website. Instead, you can install packages through running code in R or RStudio.

```
install.packages("ggplot2", repos = "https://cran.rstudio.com")
```

Once a package is installed, it needs to be loaded using the *library* function or explicitly referenced to use functions or datasets from that package.

```
library(ggplot2)
```

2.2 Scripting Basics

Before demonstrating the basics of writing R code and scripts, it is worth noting that a function can be queried in RStudio by typing a question mark before the name of the function (e.g. `?install.packages`). This will bring up documentation in the viewer window. Additionally, R will autofill function names, variable names, etc. by pressing tab while typing. If multiple matches are found, R will provide you with a drop down list to select from, which may be useful when searching through newly installed packages or trying to quickly type variable names in an R script.

R also allows for scripts to contain non-code elements, called comments, that will not be run or interpreted. To make a comment, simply use a `#` followed by the comment. A `#` only comments out a single line of code, i.e. only that line will not be run. Comments are useful to help make code more interpretable for others or to add reminders of what and why parts of code may have been written.

```
# This is an R comment!  
  
# Loading ggplot2 package  
library(ggplot2)
```

2.2.1 Setting Working Directory

When working in R, it can be helpful to set the working directory to a local directory where data are located or output files will be saved. The current working directory can also be displayed.

```
# Show current working directory  
getwd()
```

```
## [1] "/Users/kroell/Documents/IEHS/UNC-SRP/test1"
```

```
# Set working directory  
setwd("~/Documents/UNCSR/PA/Data/")
```

2.2.2 Importing and Exporting Files

After setting the working directory, importing and exporting files can be done using various functions based on the type of file being read or written. Often, it is easiest to import data into R that are in a comma separated values, comma delimited, (CSV) file or tab delimited file. Other datatypes such as SAS data files, large csv files, etc. may require different functions to be more efficiently read in and some of these file formats will be discussed in future modules.

```
# Read in CSV data  
csv.dataset = read.csv(file="~/Documents/UNCSR/PA/Data/example_data.csv")  
  
# Read in tab delimited data  
tab.dataset = read.table(file="~/Documents/UNCSR/PA/Data/example_data.txt")
```

There are many ways to export data in R. Data can be written out into a CSV file, tab delimited file, RData file, etc. There are also many functions within packages that write out specific datasets generated by that package.

```
# Write out to a CSV file  
write.csv(csv.output, file="~/Documents/UNCSR/PA/Output/csv_output.csv")  
  
# Write out to a tab delimited file  
write.table(tab.output, file="~/Documents/UNCSR/PA/Output/tsv_output.txt", sep="\t")
```

R also allows objects to be saved in RData files. These files can be read into R, as well, and will load the object into the current workspace. Entire workspaces are also able to be saved.

```
# Read in saved single R data object
r.obj = readRDS(file=~ /Documents/UNCSR/Output/data.rds")

# Write single R object to file
saveRDS(object, file=~ /Documents/UNCSR/Output/single_object.rds")

# Read in multiple saved R objects
load(file=~ /Documents/UNCSR/Output/multiple_data.RData")

# Save multiple R objects
save(object1, object2, file=~ /Documents/UNCSR/Output/multiple_objects.RData")

# Save entire workspace
save.image(file=~ /Documents/UNCSR/Output/entire_workspace.RData")

# Load entire workspace
load(file=~ /Documents/UNCSR/Output/entire_workspace.RData")
```

2.2.3 Viewing Data

After data has been loaded into R, or created within R, it is good to inspect it. Datasets can be viewed in their entirety or subset to quickly look at part of the data.

```
# View first 5 rows of the previously loaded dataset
csv.dataset[1:5,]
```

```
##      Sample Var1 Var2 Var3
## 1 sample1     1    2    1
## 2 sample2     2    4    4
## 3 sample3     3    6    9
## 4 sample4     4    8   16
## 5 sample5     5   10   25
```

```
# View the entire dataset in RStudio
View(csv.dataset)
```


Chapter 3

The Basics for Data Organization

3.1 Basic Data Manipulation

3.1.1 Merging

3.1.2 Merging processed data with metadata file?

3.1.3 Cast

3.1.4 Melt

3.1.5 Filtering & subsetting

3.1.6 Tidyverse stuff (pivots)

Chapter 4

Finding and Visualizing Data Trends

4.1 Heat maps

4.1.1 pheatmap

4.1.2 heatmap2

4.1.3 superheat

4.2 Clustering

Examples with genomics: Rager et al. 2014

4.2.1 Hierarchical

4.2.2 K-means

4.3 Data reduction (PCA)

4.3.1 Visualize PCA Plot

4.3.2 Identify % of variance captured

4.4 Basic Statistical Tests and Visualizations of Data

Need an example dataset – maybe ELGAN shuffled/deidentified, with made-up environmental exposure column?

4.4.1 Normality

Kruskal wallis? The other one that I always forget? Shapiro wilks? - histogram

4.4.2 T-tests – column charts

4.4.3 Regression: linear regression and logistic regression

4.4.4 ANOVA + delicate commentary

4.4.5 Chi-squared test – box plots

4.4.6 Fisher's exact test

Chapter 5

Multi-Omics Analyses for Environmental Health

5.1 Exposomics

5.1.1 Placenta Exposome

about to be submitted to EI Dust NTA data

5.2 Transcriptomics

5.2.1 DESeq2 / RNAseq

Wildfire dataset, available through GEO

5.3 Genome-wide MicroRNA

Rager et al. 2014 miRNAs

5.4 Genome-wide DNA Methylation

5.4.1 Illumina array data

<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE58499>

<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE28368>

5.5 Proteomics

Bailey et al arsenic dataset maybe?

Chapter 6

Mixtures Analyses for Environmental Health

6.1 Sufficient Similarity

Botanicals example with chemistry and tox profiling – Julia has dataset

6.2 Mixtures Modeling through qgcomp

Could use published wildfire analysis here (Rager et al. 2021, STOTEN), or online example provide through Alex Keil's studies

Chapter 7

Environmental Health Databases

7.1 Comparative Toxicogenomics Database
(CTD)

7.2 Gene Expression Omnibus (GEO)

7.3 NHANES