Using rmarkdown and shiny with students

Summary

- rmarkdown is a tool for communicating data and analysis
  - produce html, pdf, word documents, presentations...
- I use it to:
  - help beginning scientists learn to visualize data and communicate results
  - present to students complex ideas in chemistry and biochemistry

<u>Problem I am working on</u>: Making student data reports consistent

- rmarkdown templates
- I am combining rmarkdown with r shiny to create educational apps
  - show students biochemical data and how to interpret results

https://github.com/CEBerndsen/R4DS_Mar_2018

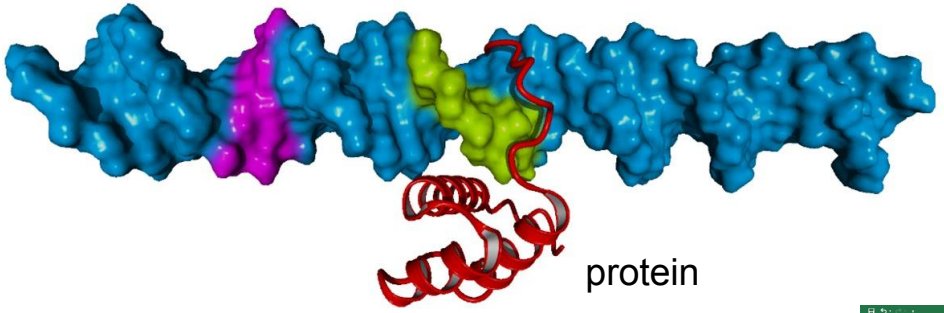
Context

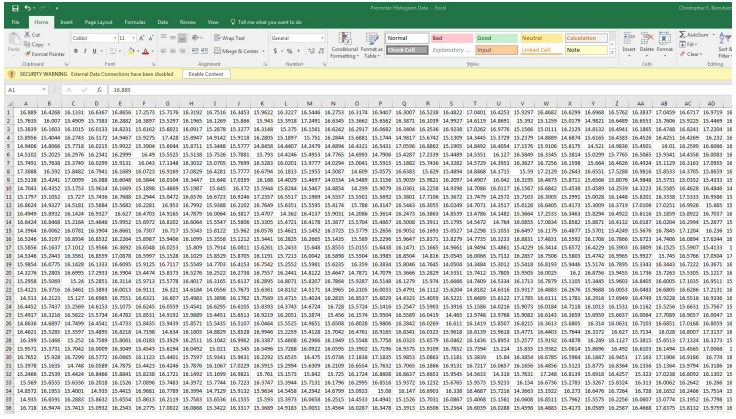
- I work in a Chemistry and Biochemistry department at a mid-sized, public university
  - Biochemist by training
  - I research topics related to protein function and structure with a lab of undergraduates
  - Teach undergraduate Biochemistry lecture and lab

I use R for:

- Visualizing data sets
- Communicating data and ideas to colleagues and students
- To help students learn to create data figures and communicate results

Initial steps -- visualizing DNA groove width data



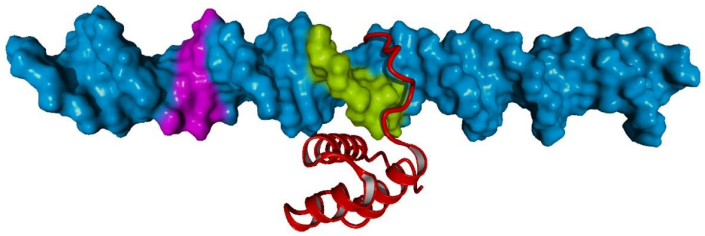


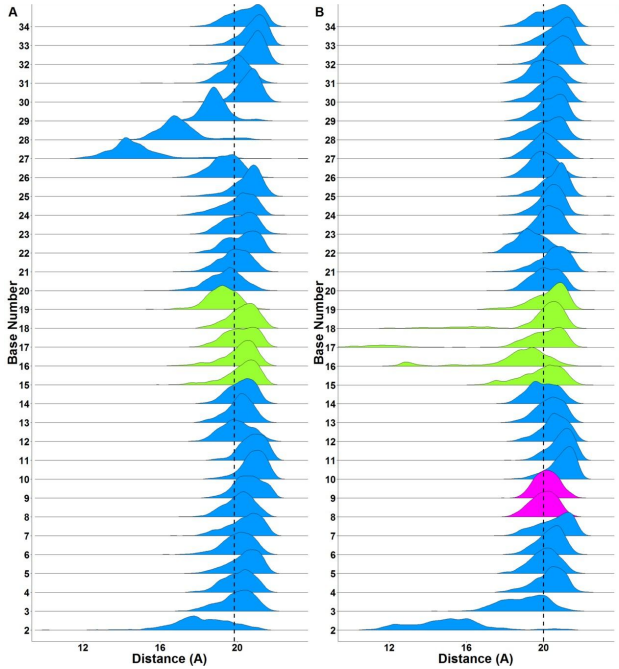
- Simulate all atom dynamics
  - 4 distinct simulations
- Measure width of DNA grooves
  - ~1000 measurements of ~40 positions
- Get a massive (for me) .tab file of numbers

The first success

1. Imported .xlsx (pg. 145) of ~40,000 measurements per plot
2. Tidied data with gather and filter (pg. 152-154, pg. 48)
3. Added new values for coloring using mutate (pg. 54)
4. Calculated data statistics with group_by and summarise (pg. 66)
5. Plotted using ggplot2 (pg. 3), forcats (pg. 223) and ggridges (geom_density_ridges2)
6. Arranged plots using cowplot

https://cran.r-project.org/web/packages/ggridges/index.html
https://github.com/wilkelab/cowplot





Hossain, Dunham, Enke and Berndsen, 2018, *Mol. Vis.*, in press

Things I learned with the first figure

- Wrangling data with R
  - Importing
  - Tidying
  - Transforming
  - Visualizing
- Communicating data
- Usefulness of R for making figure appearance consistent

rmarkdown

Document or a notebook

- Include R code chunks or in line code
- Include plots and interactivity
- Process the documents into other file formats including pdf or html or slides for a presentation

https://www.rstudio.com/resources/webinars/introducing-blogdown/
https://rmarkdown.rstudio.com/

Some basics of a rmarkdown document (HTML example)

```
---
title: "Data Dictionary"
author: "Berndsen, Roy, and Sutton"
date: "Developed January 26, 2018"
output:
  html_document: default
  pdf_document: default
---
```

YAML Header pg. 435

```
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)

library(tidyverse)
library(ggrepel)
library(reshape2)
library(readr)
library(readxl)
library(broom)
library(cowplot)
library(kableExtra)

```
```

R code chunk pg. 428

```
### REVISED: `r Sys.Date()`

## analysis.tab
```

Text + In line R code pg. 427 and pg. 434

```
```{r message=FALSE, warning=FALSE,fig.height=5, fig.width=5}
#Load Data
data <- read_table("Clean_Tetherin_analysis.tab")

#rename data columns
colnames(data) <- c("Time[ps]", "Energy[kJ/mol]", "Bond", "Angle", "Dihedral", "Planarity", "Coulomb", "VdW", "RMSDs[A]:CA", "Backbone",
"HeavyAtoms")

#Split table in two chunks for easier visualizing, do not do this during analysis
working <- data[1:6, 1:6]
workingtwo <- data[1:6, 7:11]

#Show data organization
knitr::kable(working, "html") %>% kable_styling(bootstrap_options = c("striped", "condensed"), full_width = FALSE)

knitr::kable(workingtwo, "html") %>% kable_styling(bootstrap_options = c("striped", "condensed"), full_width = FALSE)

```
```

R code chunk

https://github.com/CEBerndsen/R4DS_Mar_2018

HTML output of rmarkdown

analysis.tab

```
#Load Data
data <- read_table("Clean_Tetherin_analysis.tab")

#rename data columns
colnames(data) <- c("Time[ps]", "Energy[kJ/mol]", "Bond", "Angle", "Dihedral", "Planarity", "Coulomb", "VdW", "RMSDs[A]:CA",
 "Backbone", "HeavyAtoms")

#Split table in two chunks for easier visualizing, do not do this during analysis
working <- data[1:6, 1:6]
workingtwo <- data[1:6, 7:11]

#Show data organization
knitr::kable(working, "html") %>% kable_styling(bootstrap_options = c("striped", "condensed"), full_width = FALSE)
```

Time[ps]	Energy[kJ/mol]	Bond	Angle	Dihedral	Planarity
0.000	-1742471	4312.702	24190.14	286614.0	175.960
25.000	-1371383	31645.499	100686.48	306050.1	803.322
50.000	-1368858	29765.646	100569.15	306820.7	784.134
75.000	-1372467	30430.012	100928.35	306327.5	845.116
100.000	-1372361	29939.837	100628.11	306787.3	890.528
125.000	-1373466	30101.563	100367.60	307323.5	776.737

PDF example output

Malachite Green Assay

Introduction

Pyrophosphatases and other phosphate metabolizing enzymes are of great interest to scientists because of the key role of phosphate and activated phosphates such as those in ATP in biology. There are a variety of assay methods for these enzymes including using radioactive ^{32}P labeled phosphate or coupling phosphate production to another enzyme such as the maltose phosphorylase-glucose oxidase-peroxidase system. However, these methods require specialized equipment or assay set ups. In this lab we will use one of the oldest methods for monitoring phosphate production, the formation of phosphomolybdate.

The general reaction scheme for the assay is as follows:

$$PO_4^{3-} + (NH_4)_2MoO_4 \rightarrow H_3PMo_{12}O_{40} + 2H^+ \quad \textit{Reaction 1}$$

$$H_3PMo_{12}O_{40} + \textit{malachite green} \rightarrow (MG^+)(H_2PMo_{12}O_{40}) \quad \textit{Reaction 2}$$

Malachite green on the left side of Reaction 2 absorbs light at ~450 nm while when it forms a complex with phosphomolbydate it turns green and absorbs around 640 nm. The absorbance of the malachite green-phosphomolybdate is directly proportional to the amount of phosphate produced. Using a standard curve of known phosphate concentrations absorbance can be converted to concentration of phosphate formed (see section on standards below).

Phosphate standard curve

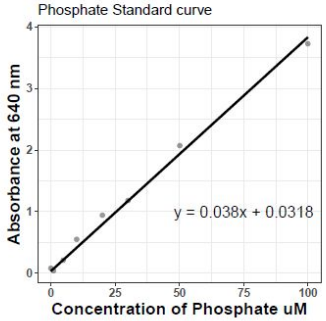
Dilute stock of phosphate in final volume of 100 μL to the concentrations listed in the table.

Standard number	Phosphate concentration (uM)	uL of Phosphate to add to 100 uL
1	0	
2	1	
3	5	
4	10	
5	20	
6	30	
7	50	
8	100	

After making standards, add 80 μL of standard to 20 μL of color reagent and let incubate for 10 minutes at room temperature. After 10 minutes, dilute sample to 1 mL in water in a cuvette and read absorbance at 640 nm.

Plotting the standard curve

Plot data as absorbance at 640 nm (y) vs. concentration of phosphate (x) like below and fit the data to a linear trendline as below.



Using the equation from the plot above, the absorbance measured for the enzyme reaction is inserted for y and then the equation is solved for x to get concentration of phosphate formed. This value is divided by 2 since pyrophosphate contains two phosphates. The rate can be determined by dividing concentration of phosphate by time.

3

Need a version of LaTex to make a pdf. Here I used TinyTex: https://yihui.name/tinytex/r/

How has it helped?

- Combines word processing and data visualization in one software
  - Previous workflow: Word + Excel + Inkscape

- Consistency in data reports

Course-based undergraduate research experience (CURE)

- Biochemistry lecture
  - 50 to 100 students per semester in a single class
    - ~70% students have no prior research experience
  - Analyze the effects of DNA mutation on protein structure and function
    - Open access mutation and protein structure databases
    - Molecular modeling using YASARA
    - Web-based modeling servers
  - Produce novel insights into the effects (or lack thereof) of human mutations on protein structure

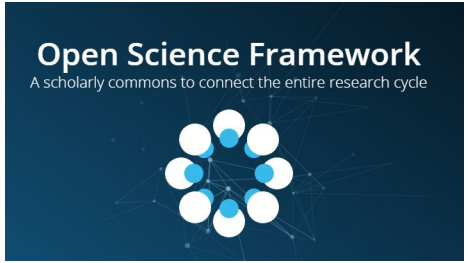
https://www.ensembl.org

https://www.rcsb.org/

→

Student data collection and analysis

→



https://osf.io/

How to make student reporting consistent?

Challenges

- Diverse approaches and data types
- Diverse backgrounds and skill levels
- Providing guided instruction without blocking student innovation
  - Challenge but do not frustrate students
- Limited instructor and student resources
- Mix of operating systems (Windows, Mac, Chromebook, iPad, etc.)

Solution(?) Shiny/rmarkdown app for teaching data interpretation and communication

When Excel failed in the middle of class...

A Simple Spreadsheet Program To Simulate and Analyze the Far-UV Circular Dichroism Spectra of Proteins

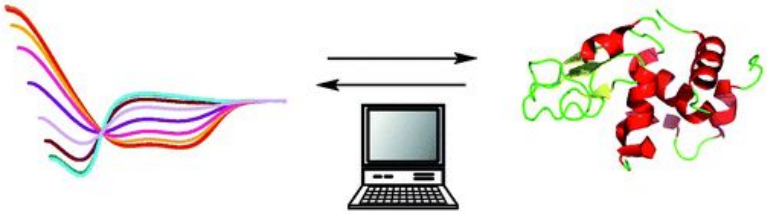
Luciano A. Abriata*
Instituto de Biología Molecular y Celular de Rosario (IBR-CONICET), Rosario, Argentina

J. Chem. Educ., **2011**, *88* (9), pp 1268–1273
DOI: 10.1021/ed200060t
Publication Date (Web): June 22, 2011

E-mail: abriata@ibr.gov.ar.

Cite this: *J. Chem. Educ. 88, 9, 1268-1273*

Abstract



rmarkdown in RStudio (CD.Rmd/CD.html)

Secondary structures fall into three main categories:

1. Helices
2. Beta strands
3. Random Coils

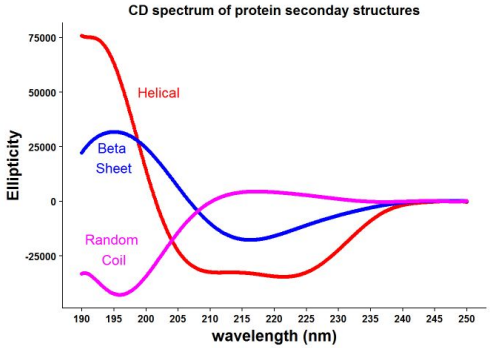
There are sub-categories within each of these three structures, however we will limit our discussion to these three for now. **Helices** are formed when amino acids form a hydrogen bond between the backbone carbonyl of one amino acid and the backbone amine of an amino acid 3 to 4 positions away. **Beta Strands** are a fully extended chain of amino acids, which typically form hydrogen bonds with other beta strands to form **beta sheets**. When a structure falls into neither of these categories we usually say the region is unstructured or in a **random coil**.

More information on secondary structure from NCBI.

Circular Dichroism

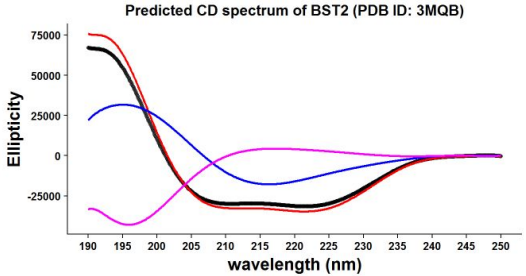
Most amino acids are **chiral** and therefore proteins are also chiral. The chirality of proteins means that they will preferentially absorb polarized light in one direction. The unequal absorbance of circularly polarized light is called **circular dichroism (CD)** and measuring the CD of proteins is the standard method for determining secondary structure. For more information and explanantions on CD spectroscopy of proteins see Using circular dichroism spectra to estimate protein secondary structure by Norma Greenfield.

The different structures of protein absorb circularly polarized light and therefore have distinct CD spectra.



secondary structure consistent with being folded or for comparing a mutated protein to the unmutated version, it is harder to definitively assign secondary structure. A few examples of proteins and their CD spectrum are shown below. The protein data is shown in black, while the helical, beta sheet, and random coil standard spectra are shown in red, blue, and magenta, respectively.

Mostly helical



92% helix, 8% random coil

rmarkdown + Shiny (CDexplainv2.Rmd)

- Shiny allows you to make interactive R code (pg. 476)
- Interactive elements in a rmarkdown file
  - Need to include shiny app code in a R chunk.

User interface

```
inputPanel(
  selectInput("mode",
              "Which mode do you want to use?",
              choices = list("Prediction" = "predict",
                             "Display" = "display")
        )
        ,

      mainPanel(
          #conditional panel
          conditionalPanel(
            condition = "input.mode == 'predict'",
            checkboxInput("guides",
                          "Show secondary structure guides?",
                          value = FALSE,
                          width = NULL)
            ,
            numericInput("helix",
                         "% alpha helix",
                         min = 0,
                         max = 100,
                         step = 5,
                         value = 50)
            ,
            numericInput("sheet",
                         "% beta sheet",
                         min = 0,
                         max = 100,
                         step = 5,
                         value = 50)
            ,
            numericInput("coil",
                         "% random coil",
                         min = 0,
                         max = 100,
                         step = 5,
                         value = 0)
            )
          ,
          conditionalPanel(
            condition = "input.mode == 'display'",
            selectInput("protein",
                        "Pick a protein to show predicted data for",
                        choices = list("Lysozyme (1LYD)" = "lyso",
                                       "Ubiquitin (1UBQ)" = "ub",
                                       "BST2 (3MQB)" = "bst",
                                       "Hemoglobin (2HHB)" = "hemo",
                                       "Antibody (1IGT)" = "ab")
            )
            ,
            "Simulated numbers generated in YASARA using the PDB IDs indicated"
            )
          )
)
```

R function

```
  mutate(coil = 1*10^8 * (-580939.072386969*lambda^0 +
                            25845.2673351998*lambda^1 +
                            -516.713088253122*lambda^2 +
                            6.1134023680003*lambda^3 +
                            -4.74021175198809E-02*lambda^4 +
                            2.51692531821056E-04*lambda^5 +
                            -9.26824208397782E-07*lambda^6 +
                            2.33714935193268E-09*lambda^7 +
                            -3.86247107852678E-12*lambda^8 +
                            3.77764956561175E-15*lambda^9 +
                            -1.66039998403172E-18*lambda^10))

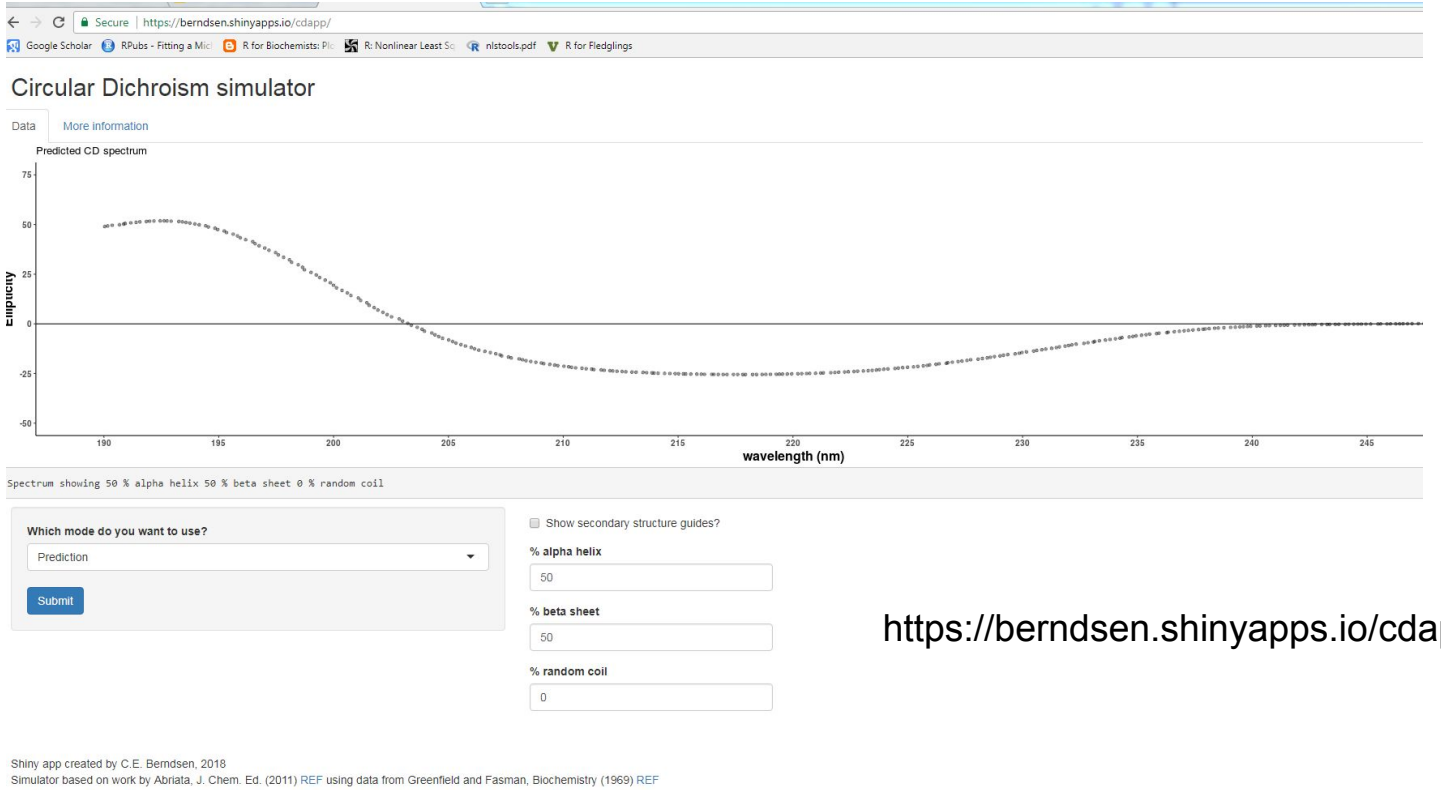
  #Predict spectrum based on user input
  CDdat <- CDdat %>% mutate(prediction = (input$helix)/100*helix + (input$sheet/100)*beta + (input$coil/100)*coil)

 # draw the plot
  if(input$guides == FALSE) {
    ggplot(CDdat, aes(x = lambda, y = prediction/1000), color = "red") +
      geom_jitter(alpha = 0.4) +
      scale_x_continuous(breaks = seq(190, 250, by = 5)) +
      labs(x = "wavelength (nm)", y = "Ellipticity", title = "Predicted CD spectrum") +
      geom_hline(yintercept = 0) +
      ylim(-50, 75) +
      theme_classic() +
      theme(axis.text = element_text(size = 10, face = "bold"), axis.title = element_text(size = 16, face = "bold"))
  }
  else {
    ggplot() +
      geom_jitter(data = CDdat, aes(x = lambda, y = prediction/1000), fill = "red", alpha = 0.4) +
      geom_line(data = CDdat, aes(x = lambda, y = helix/1000), color = "red") +
      geom_line(data = CDdat, aes(x = lambda, y = beta/1000), color = "green") +
      geom_line(data = CDdat, aes(x = lambda, y = coil/1000), color = "purple") +
      geom_hline(yintercept = 0) +
      scale_x_continuous(breaks = seq(190, 250, by = 5)) +
      labs(x = "wavelength (nm)", y = "Ellipticity", title = "Predicted CD spectrum") +
      annotate("text", x = 235, y = 25, label = "Helix", color = "red", size = 5) +
      annotate("text", x = 235, y = 20, label = "Beta Sheet", color = "green", size = 5) +
      annotate("text", x = 235, y = 15, label = "Random Coil", color = "purple", size = 5) +
      ylim(-50, 75) +
      theme_classic() +
      theme(axis.text = element_text(size = 10, face = "bold"), axis.title = element_text(size = 16, face = "bold"))

  }
  }
else {
  #Generate the wavelength values
  CDdat <- data.frame(lambda = seq(190, 250, by = 0.2))

  #Generate the basis set from Abriata, L., J. Chem. Educ., 2011, 88 (9), pp 1268–1273 and Davidson, B. and Fasman, G. D., Biochemistry 1967 6 (6) 1616-1629
  CDdat <- CDdat %>% mutate(helix = 1*10^8 * (2230060.04151075*lambda^0 +
                                              -100548.516559741*lambda^1 +
                                              2037.18080475746*lambda^2 +
                                              -24.4244919907991*lambda^3 +
                                              0.191902430159954*lambda^4 +
                                              -0.00103245782924168*lambda^5 +
                                              0.000003852118890912 52*lambda^6 +
                                              -9.84175959744622E-09*lambda^7 +
```

Shiny + rmarkdown (app.R)



https://berndsen.shinyapps.io/cdapp/

Future

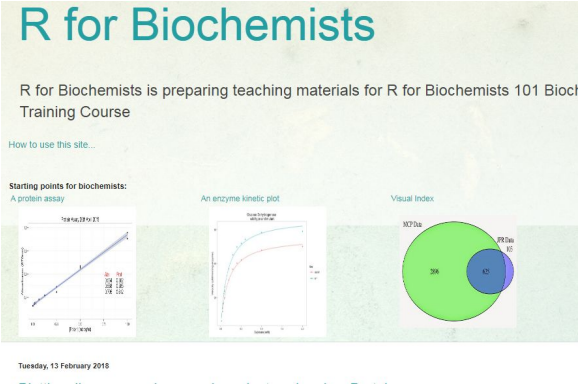
<u>Problem I am working on</u>: Making student data reports consistent *for a large class size*

Solution (?): Make functions out of data report template sections → R package → interactive Rmd/Shiny app for guiding students to make figures

- Refine in-class project
  - Challenge students but not frustrate them
  - Promote data and visualization literacy
- Build more simulators based on rmarkdown/shiny
  - Show students biochemical data examples and let them explore with data

Thank you!

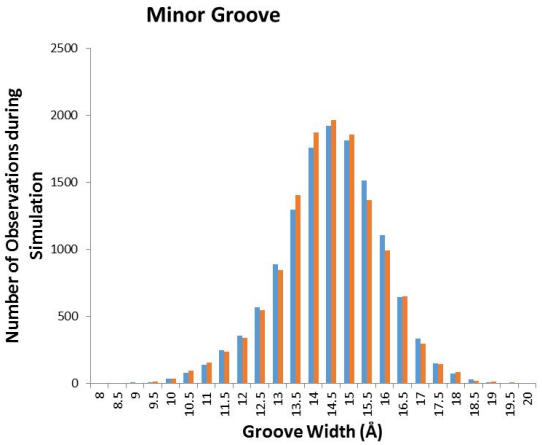
- The AMAZING JMU STUDENTS!
- Colleagues in the Dept of Chemistry and Biochemistry at JMU
- Paul Brennan who writes

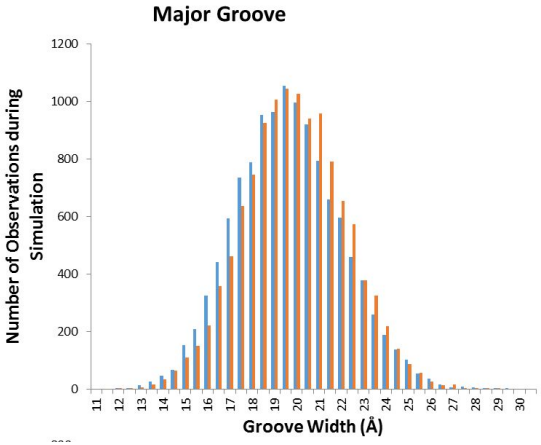


- Jesse and the R4DS community

Processing and visualizing in Excel

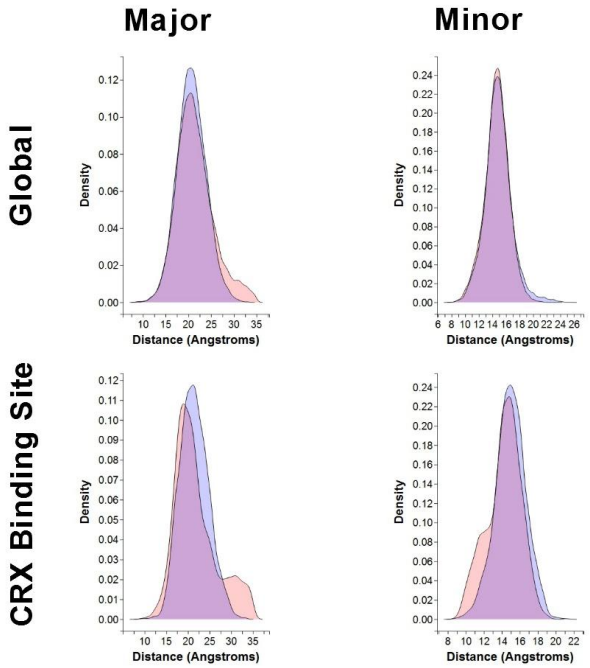
- Tedious and slow
- Lacked some information about measurement position
- Binning values
- Was a pain to do statistics





Process in Excel, visualize in ggplot2

- Tedious and slow
- Easier to make plots
  - No binning issues using a density plot
- Still lacking information on position



Open Science Framework

- Public access to data
- Forking for further development
- Want students to begin to develop a portfolio of their work

Using a consistent report format allows for easier comparison and consumption of the data by others.

Students still are involved in analysis of data.