

# Emacs: One text editor to rule them all

Stephen A. Sefick

2016-08-25

# Outline

- 1 Introduction
- 2 Using emacs to get work done
- 3 Integrated Development Environments (IDE; Getting data analysis done!)
- 4 Conclusions
- 5 Exercise
- 6 Test Slides

- 1 Introduction
- 2 Using emacs to get work done
- 3 Integrated Development Environments (IDE; Getting data analysis done!)
- 4 Conclusions
- 5 Exercise
- 6 Test Slides

# What is emacs???

- Emacs is a **TEXT** editor



- History
  - First written 1972 MIT AI Lab
  - GNU Emacs written 1984
  - <https://www.emacswiki.org/emacs/EmacsHistory>

# Motivation for using Emacs

- Many files represented as text

# Informatic text files

- Data
  - most data is text
- Software
  - Shell
  - R
  - Perl/Python
  - Markdown (github)
  - Many others

# Motivation for using Emacs

- Many files represented as text
- No focus change

# No focus change

- Less clicking more work!
  - (remember the hand?)
- Most work in 1 program
  - same shortcuts etc.



# Motivation for using Emacs

- Many file represented as text
- No focus change
- Highly Customizable
  - package manager
  - .emacs

# Motivation for using Emacs

- Many file represented as text
- No focus change
- Highly Customizable
  - package manager
  - .emacs
- 1 set keyboard shortcuts

# Motivation for using Emacs

- Many file represented as text
- No focus change
- Highly Customizable
  - package manager
  - .emacs
- 1 set keyboard shortcuts
- **Cross Platform**

- 1 Introduction
- 2 Using emacs to get work done
- 3 Integrated Development Environments (IDE; Getting data analysis done!)
- 4 Conclusions
- 5 Exercise
- 6 Test Slides

# Using emacs to get work done

- Keyboard shortcuts

# Keyboard shortcuts

- Disadvantages:
  - Lots of them!!!
- Benefits:
  - Hands do not leave the keyboard
  - Work similarly in all emacs modes
- Caveat: Productivity  $\neq$  Number of Shortcuts you know
- Lots of cheatsheets for most major modes
  - [Google](#)

# Keyboard shortcuts

## Example

C-x C-c hold Ctrl press x then hold Ctrl press c

M-x hold Alt key press x

## Very Useful Sortcuts

C-g - stop command \*\*\*STOP EMACS IN ITS TRACKS\*\*\*

C-x C-c - exits emacs

C-x C-s - saves file

C-x C-f - opens file

C-x k - kills a buffer

C-x 1 - single window

C-x 2 - horizontal 2 pane

C-x 3 - vertical 2 pane

M-x - run command

## Text Editing

C-f - forward 1 character

C-b - backward 1 character

C-n - down 1 line

C-p - up 1 line

C-a - beginning of line

C-e - end of line

C-space - set mark; select area

C-w - cut

M-w - copy

C-y - paste

- **MANY MANY MORE**
- **NOT NECESSARY** TO KNOW THEM ALL TO BE **PRODUCTIVE**



# Using emacs to get work done

- Keyboard shortcuts
- .emacs

- Linux/Unix **Text** configuration file

## .emacs

```
;;-----;;  
;;ESS R - assign <- to :  
(setq ess-smart-S-assign-key ";")  
;;-----;;  
  
;;-----;;  
;;org-mode  
(require 'org)  
(define-key global-map "\C-cl" 'org-store-link)  
(define-key global-map "\C-ca" 'org-agenda)  
(setq org-log-done t)  
;;-----;;
```

# Using emacs to get work done

- Keyboard shortcuts
- .emacs
- Package Manager

# Package manager

- MELPA **MELPA Package Manager**
- M-x list-packages

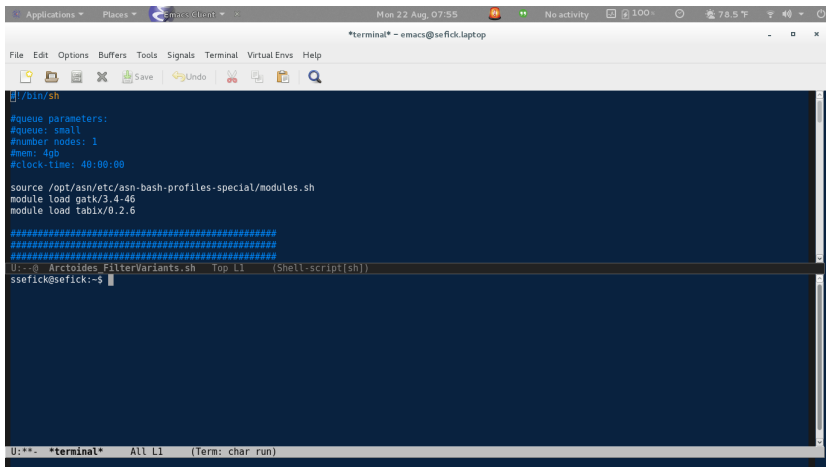
.emacs

```
;;-----;;  
;;Added Sefick 20160525  
(require 'package) ;; You might already have this line  
(add-to-list 'package-archives  
  '("melpa-stable" . "https://stable.melpa.org/packages/") t)  
(when (< emacs-major-version 24)  
  ;; For important compatibility libraries like cl-lib  
  (add-to-list 'package-archives '("gnu" .  
    "http://elpa.gnu.org/packages/")))  
(package-initialize) ;; You might already have this line  
;;-----;;
```

- 1 Introduction
- 2 Using emacs to get work done
- 3 Integrated Development Environments (IDE; Getting data analysis done!)
- 4 Conclusions
- 5 Exercise
- 6 Test Slides

- Program completion (generally)
- interactive program building
  - write in one window
  - evaluate in another
- Knows language conventions
  - autocomplete
  - TABS!!!
- Syntax highlighting

# Shell Scripts



The screenshot shows an Emacs Client window titled "Mon 22 Aug, 07:55". The active buffer is "\*terminal\* ~ emacs@sefick.laptop". The menu bar includes File, Edit, Options, Buffers, Tools, Signals, Terminal, Virtual Envs, and Help. The toolbar contains icons for opening files, saving, undo, redo, and search. The terminal content is as follows:

```
#!/bin/sh

#queue parameters:
#queue: small
#number nodes: 1
#mem: 4gb
#clock-time: 40:00:00

source /opt/asn/etc/asn-bash-profiles-special/modules.sh
module load gatk/3.4-46
module load tabix/0.2.6

#####
#####
#####
U:--@ Arcoides_FilterVariants.sh Top L1 (Shell-script[sh])
sefick@sefick:~$
```

The status bar at the bottom of the terminal window displays "U:-- \*terminal\* All L1 (Term: char run)".

# Emacs Speaks Statistics R (ESS)

```
#!/usr/bin/env Rscript

#####

require(argparse)

parser <- ArgumentParser()
# specify our desired options
# by default ArgumentParser will add an help option

parser$add_argument("-i", "--input_vcf", action="store", default="Table from VCF file", dest="input_vcf",
  help="Input %(default)s")

parser$add_argument("-o", "--output_pdf", action="store", default="out.pdf", dest="output_pdf",
  help="Output %(default)s")

#---- plot_stats_vcf CL ARGS.R Top L1 (ESS[S] [none])
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

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.

[1] "It's go time"
# options(STERM='iESS', str.dendrogram.last="", editor='emacsclient', show.error.locations=TRUE)

U:**- *R* Bot L22 (iESS [R db -]: run)
```



# Markdown: Emacs

# Markdown: Github

The screenshot shows a Google Chrome browser window with the address bar displaying `https://github.com/ssefick/UsefulBioinformaticScripts`. The page content is a README file for the repository. The title of the repository is `UsefulBioinformaticScripts`. The README text states: "I am posting Useful Bioinformatics scripts here that I find useful." Below this, there is a section titled `vcf_cutoff_stats.R`. The text describes this as an R script to calculate statistics related to depth and quality of `vcf_files`. It notes that while the R part is not the most efficient, the cat part is inefficient. It provides a typical use case and the commands to run the script.

```
#typical use case:
vcftools --site-depth --vcf your.vcf --out site_depth
vcftools --site-quality --vcf your.vcf --out site_quality

#outputs site_depth.ldepth and site_quality.lqual
#then pipe this output to the script

cat site_quality.lqual | vcf_cutoff_stats.R > vcf_quality_summary.txt
cat site_depth.ldepth | vcf_cutoff_stats.R > vcf_depth_summary.txt
```

# Many others

- Python
- Perl
- C
- Java
- etc., etc., etc.

- 1 Introduction
- 2 Using emacs to get work done
- 3 Integrated Development Environments (IDE; Getting data analysis done!)
- 4 Conclusions**
- 5 Exercise
- 6 Test Slides

# Text Editing Rules!

- 1 program/many uses
- No loss of focus
- More time with fingers on keyboard
- Minimal setup I would recomend
  - Markdown
  - ESS
  - Python/Perl IDE
- Further Help (Emacs was scary to me in June!!!)
  - Google
  - **Just start EXPERIMENTING!**

- 1 Introduction
- 2 Using emacs to get work done
- 3 Integrated Development Environments (IDE; Getting data analysis done!)
- 4 Conclusions
- 5 Exercise**
- 6 Test Slides

# Sort a bed file

- Bed files define genomic regions
- `sim.bed` is a bedfile generated from a VCF SNP file
- We need to sort it for downstream analysis
- Let's write a bash script to do this
- Things to remember about the terminal in Emacs
  - when in charmode (default) acts like a terminal
  - when in linemode acts like an emacs buffer

# Exercise

Point play around with emacs

Work together

Do not get bogged down with **sort** command

- 1 unzip Emacs\_exercise.zip
- 2 open emacs in Emacs\_exercise folder
  - emacs -nw
- 3 split buffer
- 4 move to other buffer and start terminal
- 5 while the focus is in the terminal change to linemode
- 6 Try to navigate with keyboard shortcuts
- 7 use ls and then copy sim.bed
- 8 move back to script window and paste
- 9 now write a script to sort the bed file



- 1 Introduction
- 2 Using emacs to get work done
- 3 Integrated Development Environments (IDE; Getting data analysis done!)
- 4 Conclusions
- 5 Exercise
- 6 Test Slides

# What are CNTs?

when? who?

- 1952 bla bla bla
- 1991 Dr. Sumio Iijima publishes “Helical microtubules of graphitic carbon”

