Keeping a Computational Lab Notebook & How to Get Programming Help

Brittany N. Lasseigne, PhD

HudsonAlpha Intstitute for Biotechnology

4 June 2018

@bnlasse blasseigne@hudsonalpha.org

What is a lab notebook?

What is a lab notebook?

- Complete record of procedures, reagents, data, and thoughts

What is a lab notebook?

- Complete record of procedures, reagents, data, and thoughts
- Explanation of why and how you are doing an experiment, as well as the results

What is a lab notebook?

- Complete record of procedures, reagents, data, and thoughts
- Explanation of why and how you are doing an experiment, as well as the results
- Legal document* to prove patents and defend against accusations of fraud (*requirements vary depending on situation)

What is a lab notebook?

- Complete record of procedures, reagents, data, and thoughts
- Explanation of why and how you are doing an experiment, as well as the results
- Legal document* to prove patents and defend against accusations of fraud (*requirements vary depending on situation)
- Lab legacy/institutional memory

Adapted from ‘Keeping a Lab Notebook’, Philip Ryan, Office of Intramural Training and Education (NIH)

Why do we need to keep computational laboratory notebooks?

‘Ten Simple Rules for a Computational Biologist’s Laboratory Notebook.’ S. Schnell, PLOS Computational Biology, 2015.

Why do we need to keep computational laboratory notebooks?

"One of the major hurdles I face as the head of a computational biology laboratory is convincing my research team—particularly those pursuing exclusively mathematical and computational modeling—that they need to keep a laboratory notebook. There seems to be a misconception in the computational biology community that a lab notebook is only useful for recording experimental protocols and their results. **A lab notebook is much more than that. It is an organizational tool and memory aid, which serves as the primary record of scientific research and activity for all scientists. It also serves as a legal record of ownership of the ideas and results obtained by a scientist." -S. Schnell**

'Ten Simple Rules for a Computational Biologist's Laboratory Notebook.' S. Schnell, PLOS Computational Biology, 2015.

10 Simple Rules for a CompBio Lab Notebook

Adapted from ‘Ten Simple Rules for a Computational Biologist’s Laboratory Notebook.’ S. Schnell, PLOS Computational Biology, 2015.

10 Simple Rules for a CompBio Lab Notebook

- Rule 1: Learn your institution's/lab's notebook policy

Adapted from 'Ten Simple Rules for a Computational Biologist's Laboratory Notebook.' S. Schnell, PLOS Computational Biology, 2015.

10 Simple Rules for a CompBio Lab Notebook

- Rule 1: Learn your institution's/lab's notebook policy
- Rule 2: Select the right medium for your lab notebook (e.g. physical notebook vs. electronic notebook, one notebook vs. many, version control, back-up notebook, etc.)

Adapted from 'Ten Simple Rules for a Computational Biologist's Laboratory Notebook.' S. Schnell, PLOS Computational Biology, 2015.

10 Simple Rules for a CompBio Lab Notebook

- Rule 1: Learn your institution's/lab's notebook policy
- Rule 2: Select the right medium for your lab notebook (e.g. physical notebook vs. electronic notebook, one notebook vs. many, version control, back-up notebook, etc.)
- Rule 3: Make the habit of keeping your lab notebook up to date by writing things down while you are working

Adapted from 'Ten Simple Rules for a Computational Biologist's Laboratory Notebook.' S. Schnell, PLOS Computational Biology, 2015.

10 Simple Rules for a CompBio Lab Notebook

- Rule 1: Learn your institution's/lab's notebook policy
- Rule 2: Select the right medium for your lab notebook (e.g. physical notebook vs. electronic notebook, one notebook vs. many, version control, back-up notebook, etc.)
- Rule 3: Make the habit of keeping your lab notebook up to date by writing things down while you are working
- Rule 4: Record all* scientific activities in your lab notebook —it is a chronological log of everything scholarly a scientist does (*more wiggle room on this rule)

Adapted from 'Ten Simple Rules for a Computational Biologist's Laboratory Notebook.' S. Schnell, PLOS Computational Biology, 2015.

10 Simple Rules for a CompBio Lab Notebook

- Rule 5: Every entry should be recorded with a date, subject, and protocol (explain your thought process—the ‘why’).

10 Simple Rules for a CompBio Lab Notebook

- Rule 5: Every entry should be recorded with a date, subject, and protocol (explain your thought process—the ‘why’).
  - If you make a mistake: strikethrough so still legible, enter correct information and new date

10 Simple Rules for a CompBio Lab Notebook

- Rule 5: Every entry should be recorded with a date, subject, and protocol (explain your thought process—the ‘why’).
  - If you make a mistake: strikethrough so still legible, enter correct information and new date
- Rule 6: Keep a record of how every result was produced—the gold standard is reproducibility (i.e. raw data to final model/figure/statistical analysis)

Adapted from ‘Ten Simple Rules for a Computational Biologist’s Laboratory Notebook.’ S. Schnell, PLOS Computational Biology, 2015.

10 Simple Rules for a CompBio Lab Notebook

- Rule 5: Every entry should be recorded with a date, subject, and protocol (explain your thought process—the ‘why’).
  - If you make a mistake: strikethrough so still legible, enter correct information and new date

- Rule 6: Keep a record of how every result was produced—the gold standard is reproducibility (i.e. raw data to final model/figure/statistical analysis)

- Rule 7: Use version control for models/algorithms/computer code (standardized names for models/scripts)

Adapted from ‘Ten Simple Rules for a Computational Biologist’s Laboratory Notebook.’ S. Schnell, PLOS Computational Biology, 2015.

10 Simple Rules for a CompBio Lab Notebook

- Rule 5: Every entry should be recorded with a date, subject, and protocol (explain your thought process—the ‘why’).
  - If you make a mistake: strikethrough so still legible, enter correct information and new date

- Rule 6: Keep a record of how every result was produced—the gold standard is reproducibility (i.e. raw data to final model/figure/statistical analysis)

- Rule 7: Use version control for models/algorithms/computer code (standardized names for models/scripts)

- Rule 8: Keep a lab notebook that can serve as a legal record of your work

Adapted from ‘Ten Simple Rules for a Computational Biologist’s Laboratory Notebook.’ S. Schnell, PLOS Computational Biology, 2015.

10 Simple Rules for a CompBio Lab Notebook

- Rule 5: Every entry should be recorded with a date, subject, and protocol (explain your thought process—the ‘why’).
  - If you make a mistake: strikethrough so still legible, enter correct information and new date

- Rule 6: Keep a record of how every result was produced—the gold standard is reproducibility (i.e. raw data to final model/figure/statistical analysis)

- Rule 7: Use version control for models/algorithms/computer code (standardized names for models/scripts)

- Rule 8: Keep a lab notebook that can serve as a legal record of your work

- Rule 9: Make a table of contents for your lab notebook

Adapted from ‘Ten Simple Rules for a Computational Biologist’s Laboratory Notebook.’ S. Schnell, PLOS Computational Biology, 2015.

10 Simple Rules for a CompBio Lab Notebook

- Rule 5: Every entry should be recorded with a date, subject, and protocol (explain your thought process—the ‘why’).
  - If you make a mistake: strikethrough so still legible, enter correct information and new date
- Rule 6: Keep a record of how every result was produced—the gold standard is reproducibility (i.e. raw data to final model/figure/statistical analysis)
- Rule 7: Use version control for models/algorithms/computer code (standardized names for models/scripts)
- Rule 8: Keep a lab notebook that can serve as a legal record of your work
- Rule 9: Make a table of contents for your lab notebook
- Rule 10: Protect your lab notebook—the original belongs to your institution.

Adapted from ‘Ten Simple Rules for a Computational Biologist’s Laboratory Notebook.’ S. Schnell, PLOS Computational Biology, 2015.

Example Lab Notebook Entry:

180410

Goal: I'm trying to understand how many TCGA KIRC samples have each combination of data types

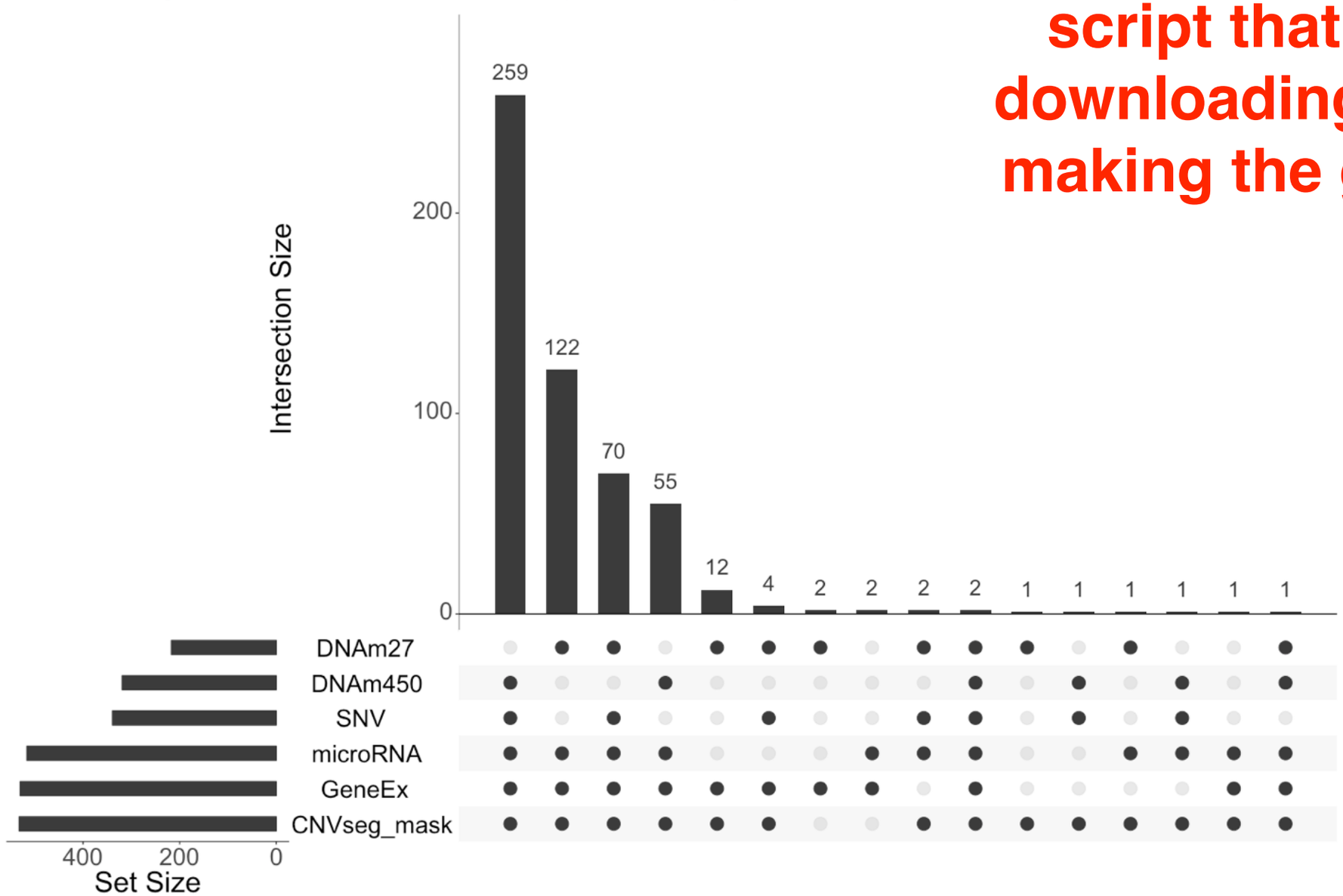
-Working scripts are here: /Data_nonGoogleDrive/TCGAKIRC_TCGAbioinks_180314/getTCGAKIRCdata.R

-I set up tcgabiolinks queries for each dataset (except for clinical) and recorded the number of T/N samples for each

-Analyzed the tumor overlap between datasets, including limiting datasets with Reduce(intersect, list()) notation

-Visually examined the results with UpSetR graph (vignette: https://cran.r-project.org/web/packages/UpSetR/vignettes/basic.usage.html)

/Data_nonGoogleDrive/TCGAKIRC_TCGAbiolinks_180314/KIRC_UpSetRPlot_180410.png



Conclusions: switch from m27 to m450 happened during this dataset so about 25% of tumors have m27k data and rest with m450k data

script that goes from downloading raw data to making the graph below

Example Lab Notebook Entry (continued):

(Partial for Illustrative Purposes) Code to Plot Graph on Previous Slide

```
##visualize intersections
library(UpSetR)
listInput<-list(GeneEx=substr(getResults(query.exp.hg38.KIRC, cols="cases"), 1, 12),
                DNAm450=substr(getResults(query.DNAm450.hg38.KIRC, cols="cases"), 1, 12),
                DNAm27=substr(getResults(query.DNAm27.hg38.KIRC, cols="cases"), 1, 12),
                microRNA=substr(getResults(query.micro.hg38.KIRC, cols="cases"), 1, 12),
                #CNVseg=substr(getResults(query.CNVseg.hg38.KIRC, cols="cases"), 1, 12),
                CNVseg_mask=substr(getResults(query.CNVsegmasked.hg38.KIRC, cols="cases"), 1, 12),
                SNV=substr(getResults(query.SNV.hg38.KIRC, cols="cases"), 1, 12))
#commented out CNVseg because masked has known germline snps removed

str(listInput)

upset(fromList(listInput), order.by=c("freq"), nsets=7, line.size=0, point.size=5, nintersects = NA,
text.scale=3)
#KIRC_UpSetRPlot_180410.png
#nsets shows all 7 data sets
#line.size=0 removes vertical lines in plot
#point.size changes circle sizes
#nintersects says number of intersects to plot
#text.scale changes text size
```

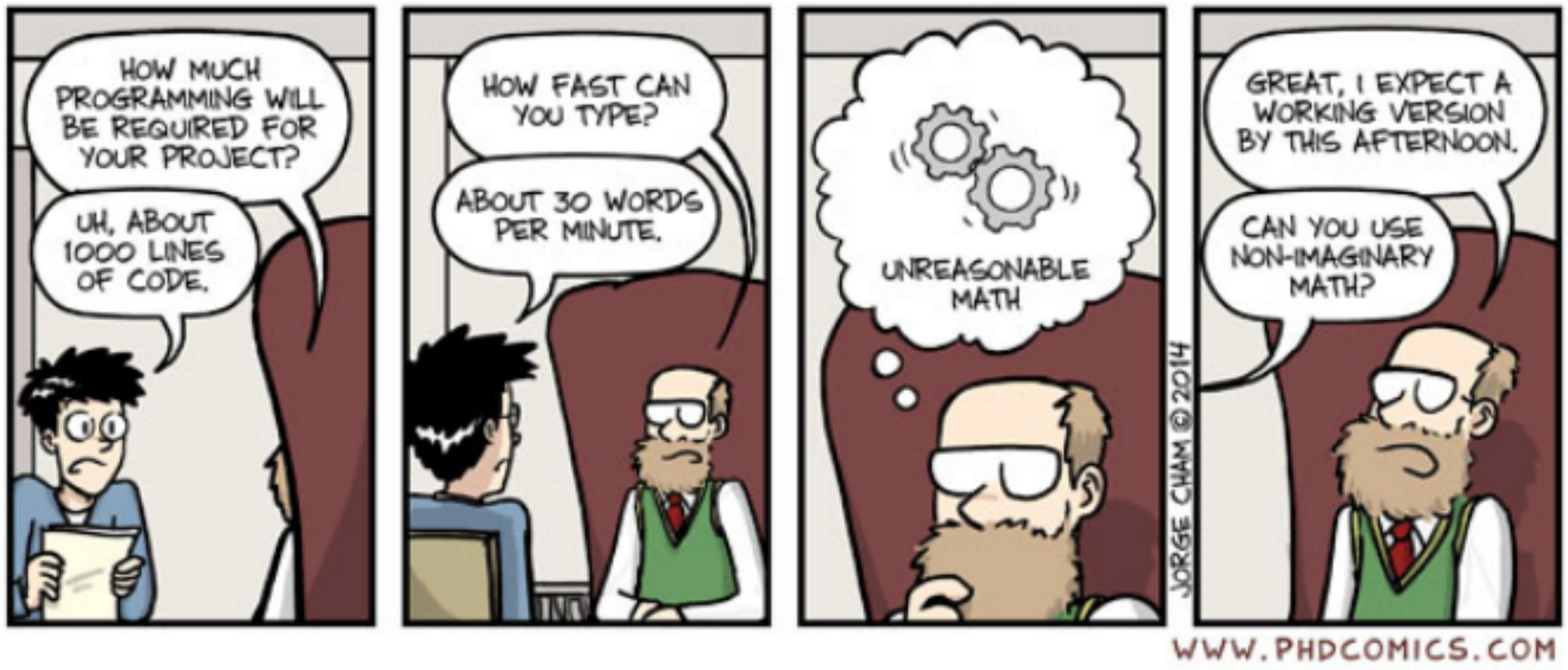
Example Lab Notebook Entry (continued):

(Partial for Illustrative Purposes) Code to Plot Graph on Previous Slide

```
##visualize intersections
library(UpSetR)
listInput<-list(GeneEx=substr(getResults(query.exp.hg38.KIRC, cols="cases"), 1, 12),
                DNAm450=substr(getResults(query.DNAm450.hg38.KIRC, cols="cases"), 1, 12),
                DNAm27=substr(getResults(query.DNAm27.hg38.KIRC, cols="cases"), 1, 12),
                microRNA=substr(getResults(query.micro.hg38.KIRC, cols="cases"), 1, 12),
                #CNVseg=substr(getResults(query.CNVseg.hg38.KIRC, cols="cases"), 1, 12),
                CNVseg_mask=substr(getResults(query.CNVsegmasked.hg38.KIRC, cols="cases"), 1, 12),
                SNV=substr(getResults(query.SNV.hg38.KIRC, cols="cases"), 1, 12))
#commented out CNVseg because masked has known germline snps removed

str(listInput)

upset(fromList(listInput), order.by=c("freq"), nsets=7, line.size=0, point.size=5, nintersects = NA,
text.scale=3)
#KIRC_UpSetRPlot_180410.png
#nsets shows all 7 data sets
#line.size=0 removes vertical lines in plot
#point.size changes circle sizes
#nintersects says number of intersects to plot
#text.scale changes text size
```

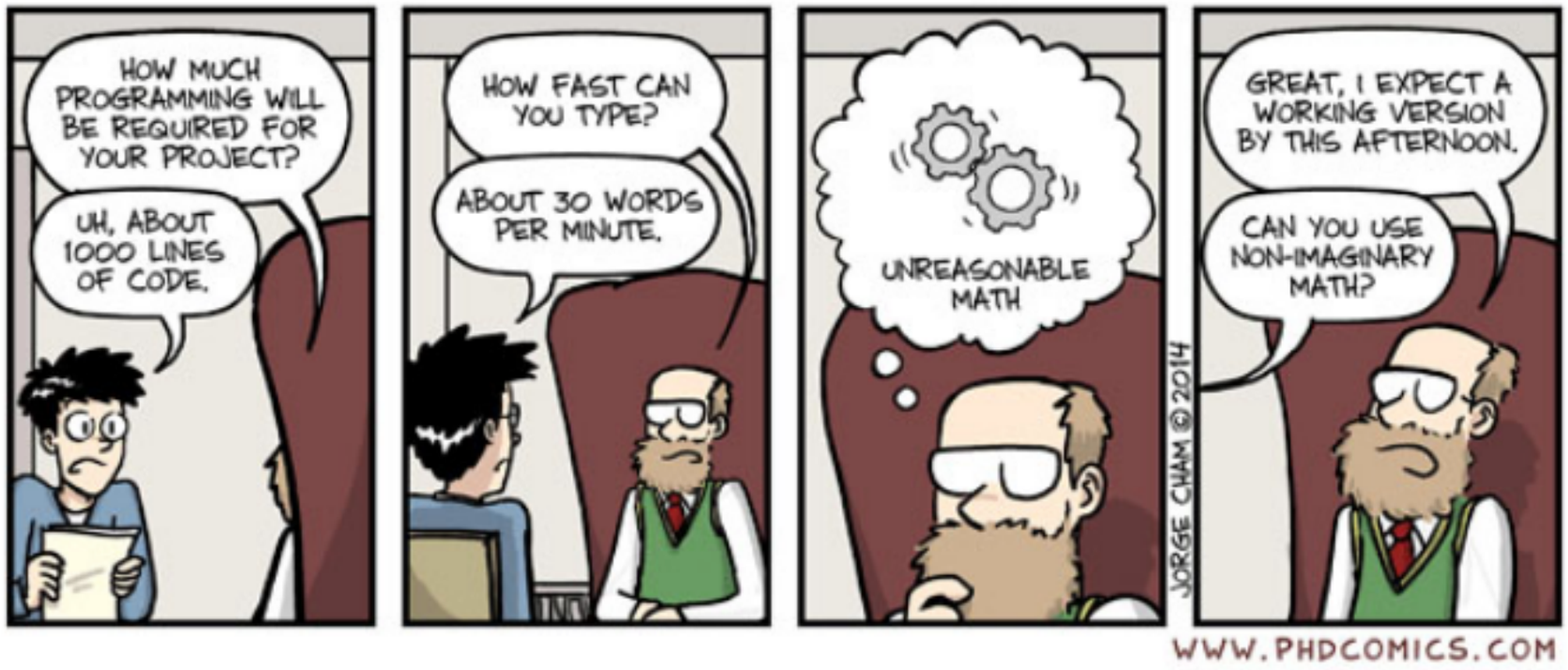
- ✓ consistent naming
- ✓ explain thought process
- ✓ reproducible

How to ask for programming help:



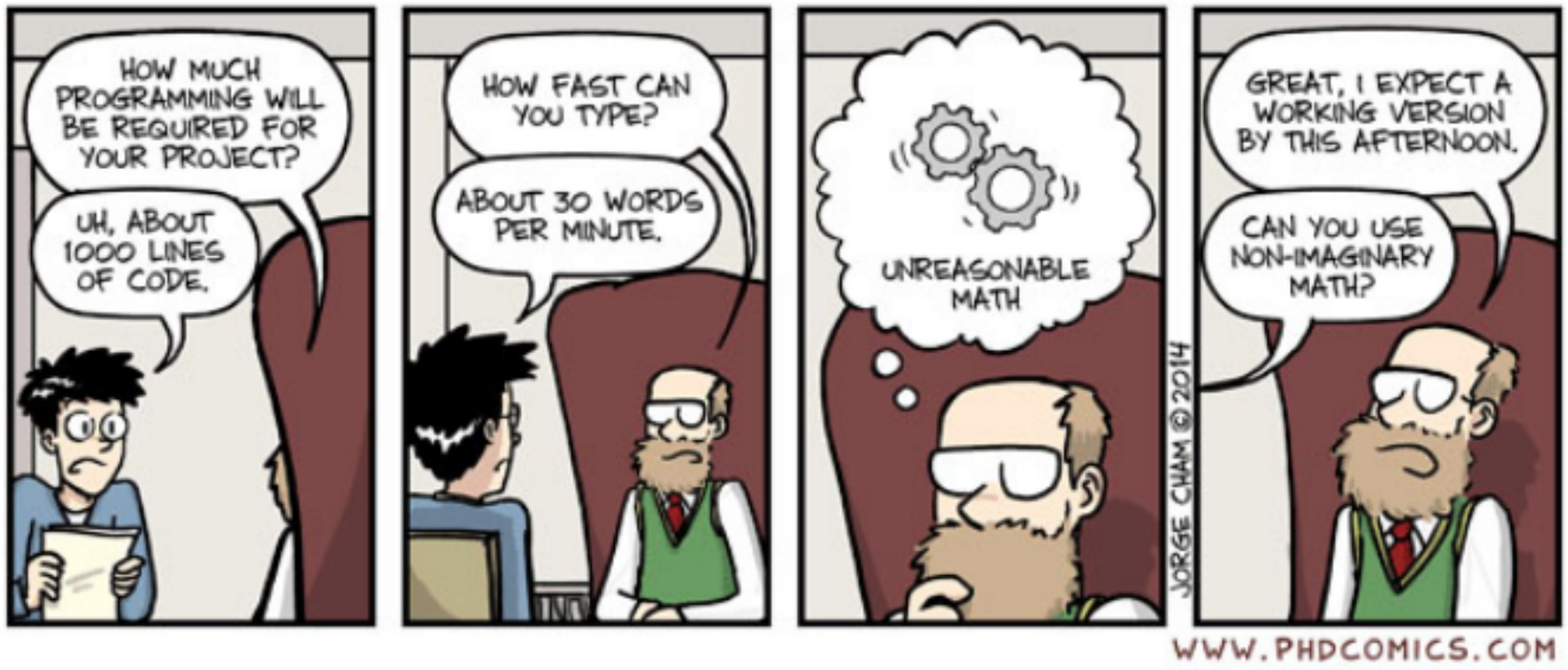
How to ask for programming help:

- Do not be afraid to ask!



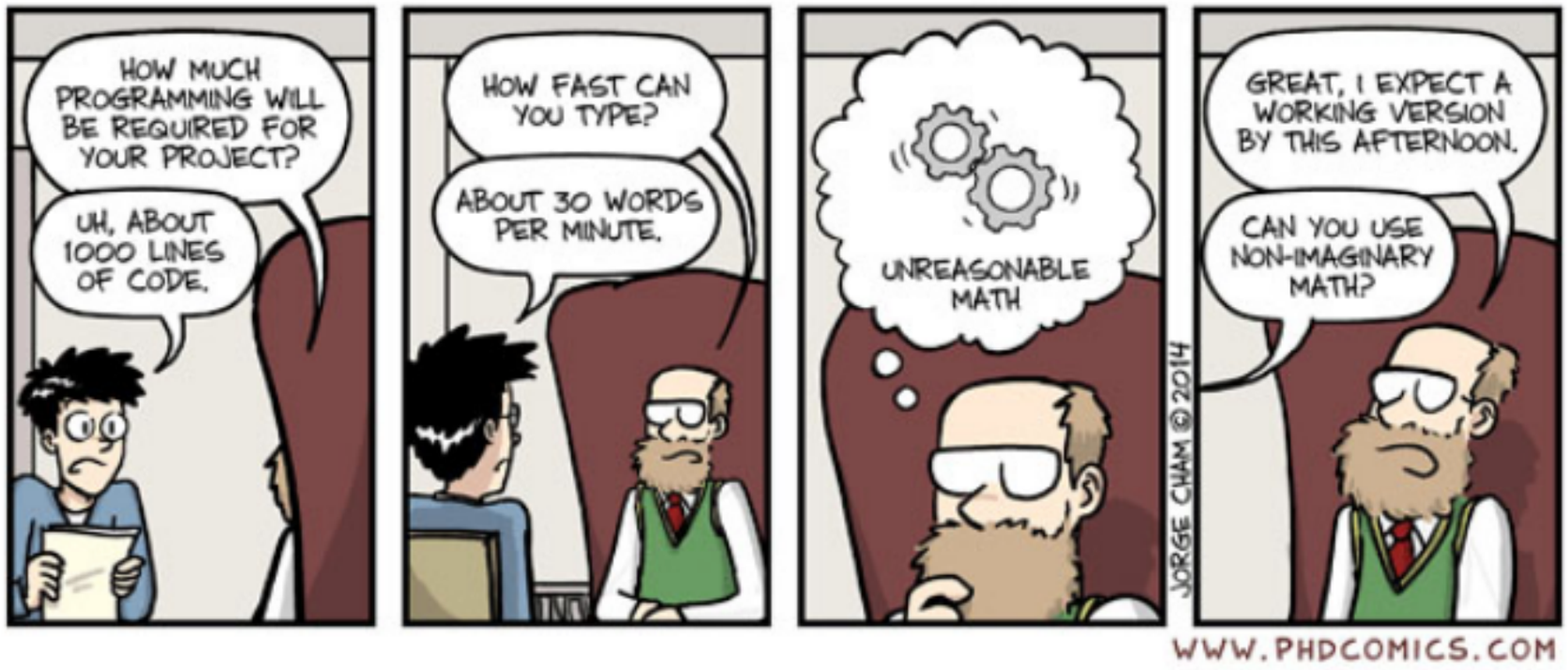
How to ask for programming help:

- Do not be afraid to ask!
- Try something first.



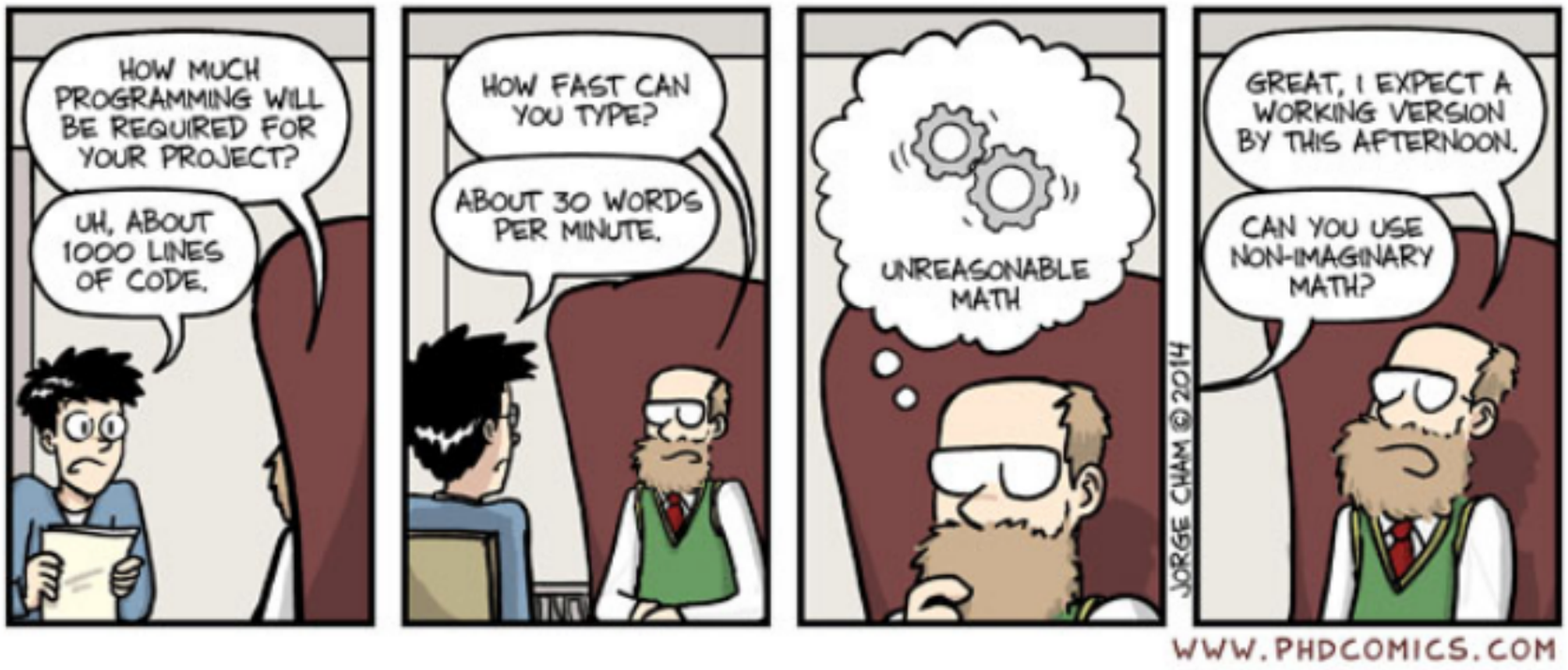
How to ask for programming help:

- Do not be afraid to ask!
- Try something first.
- Always check the documentation.



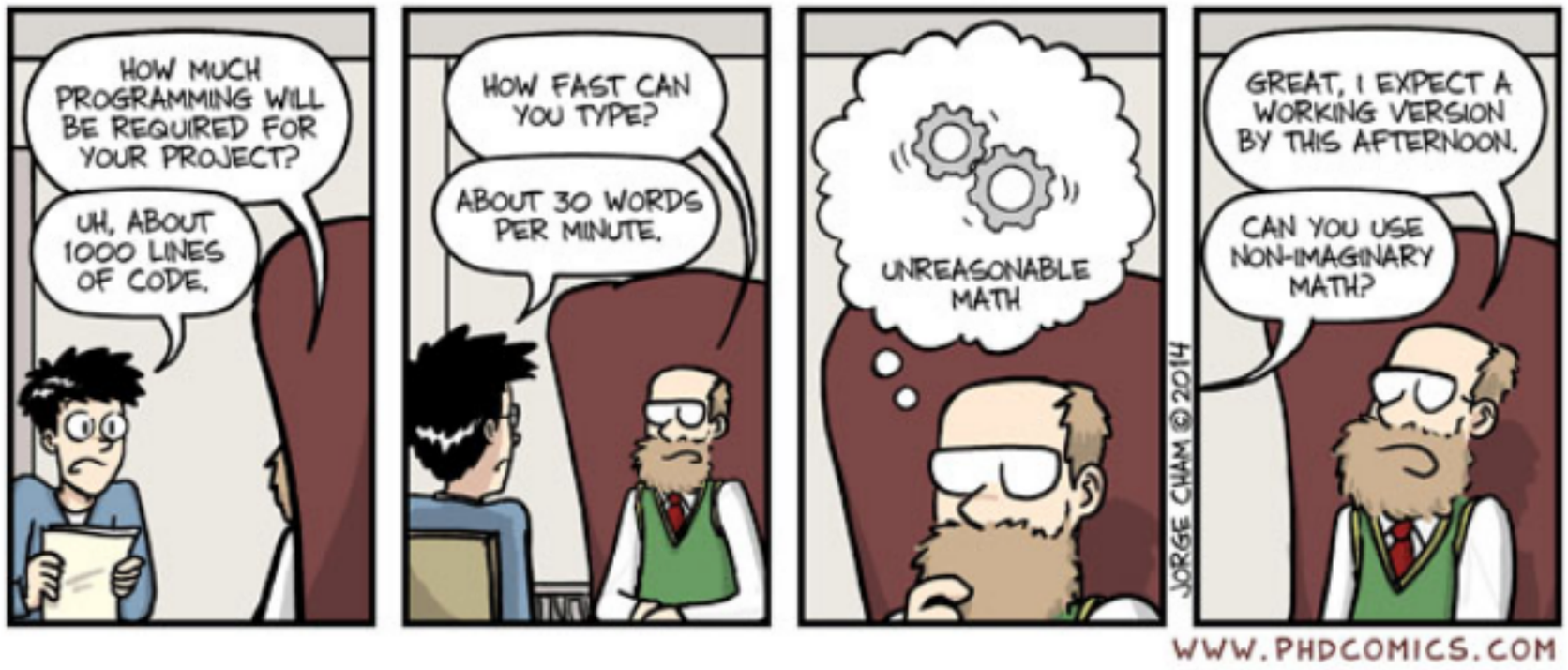
How to ask for programming help:

- Do not be afraid to ask!
- Try something first.
- Always check the documentation.
- Google your question/error message/etc.



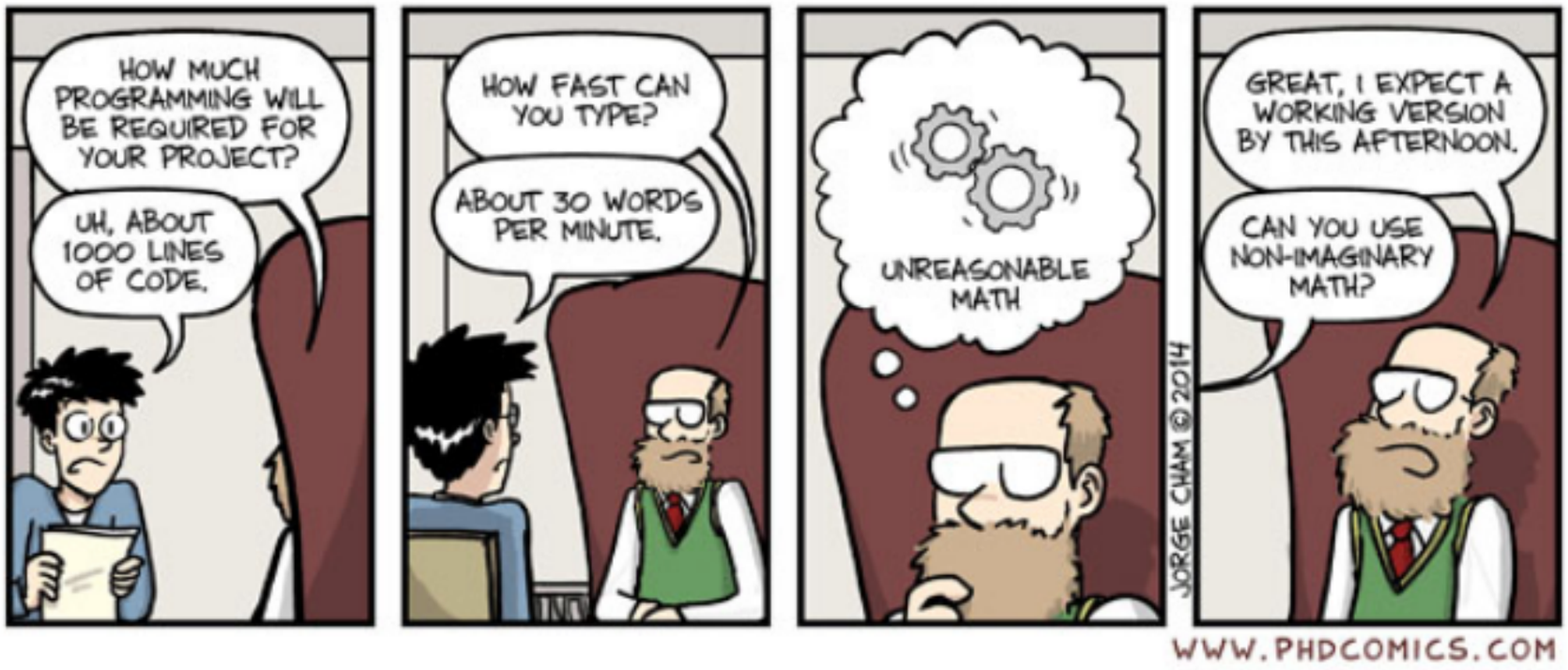
How to ask for programming help:

- Do not be afraid to ask!
- Try something first.
- Always check the documentation.
- Google your question/error message/etc.
- Narrow down the problem. Try to create a miniature version of the problem if you can.



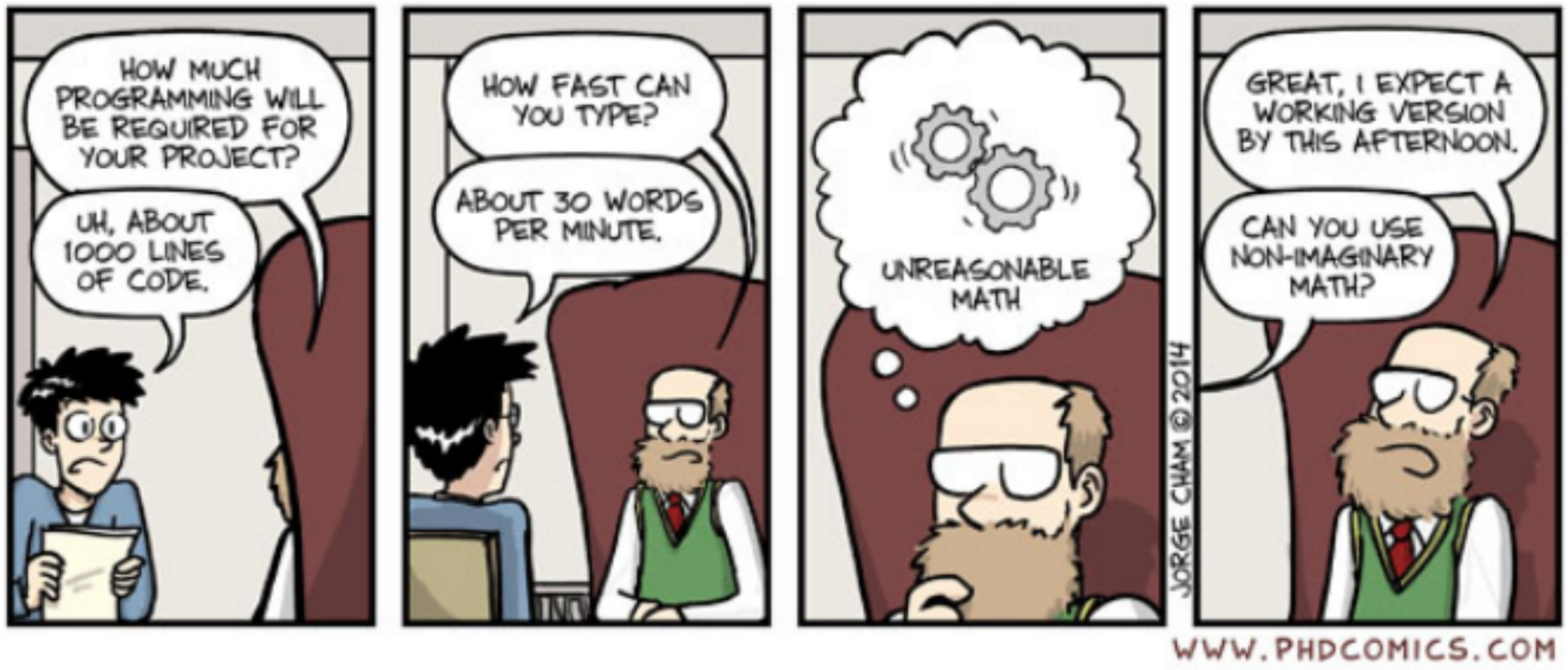
How to ask for programming help:

- Do not be afraid to ask!
- Try something first.
- Always check the documentation.
- Google your question/error message/etc.
- Narrow down the problem. Try to create a miniature version of the problem if you can.
- But remember to share what your overall goal is.



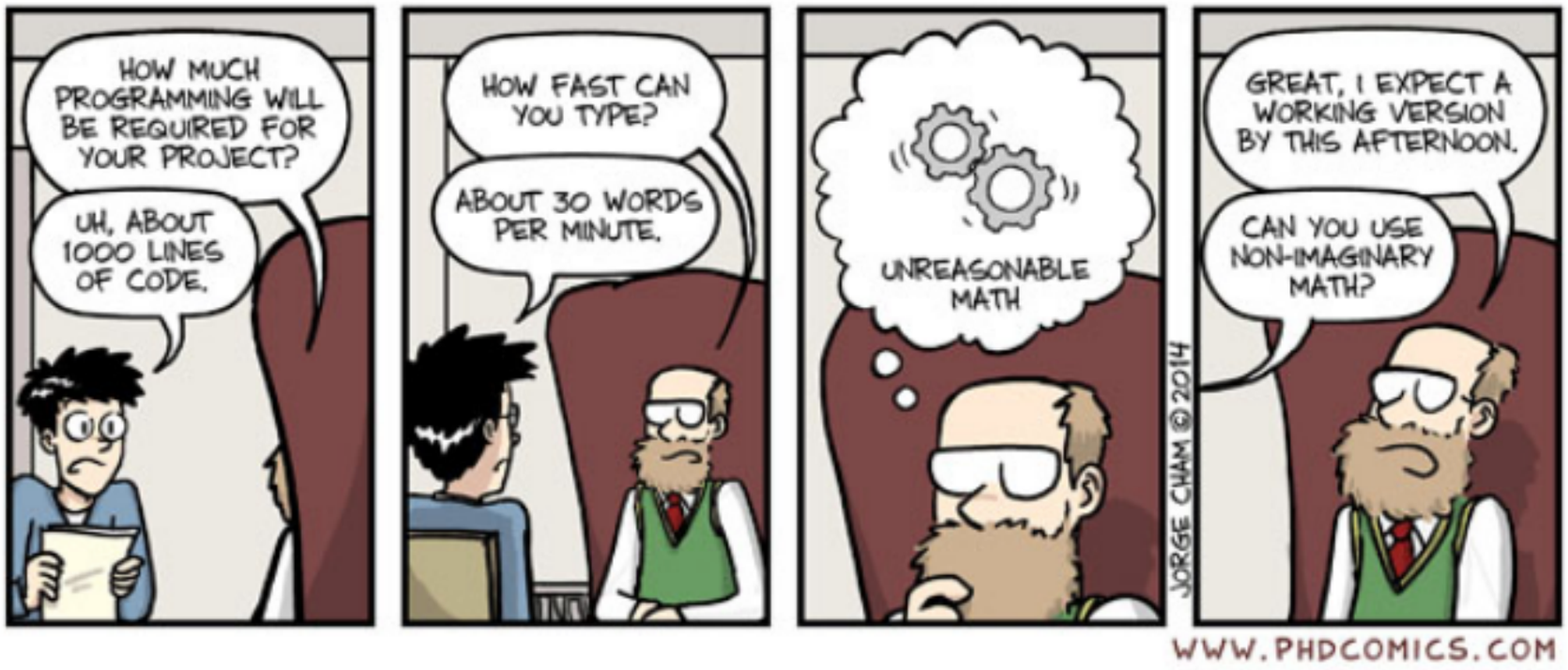
How to ask for programming help:

- Do not be afraid to ask!
- Try something first.
- Always check the documentation.
- Google your question/error message/etc.
- Narrow down the problem. Try to create a miniature version of the problem if you can.
- But remember to share what your overall goal is.
- Be patient and courteous.



How to ask for programming help:

- Do not be afraid to ask!
- Try something first.
- Always check the documentation.
- Google your question/error message/etc.
- Narrow down the problem. Try to create a miniature version of the problem if you can.
- But remember to share what your overall goal is.
- Be patient and courteous.
- Record your troubleshooting in your lab notebook!

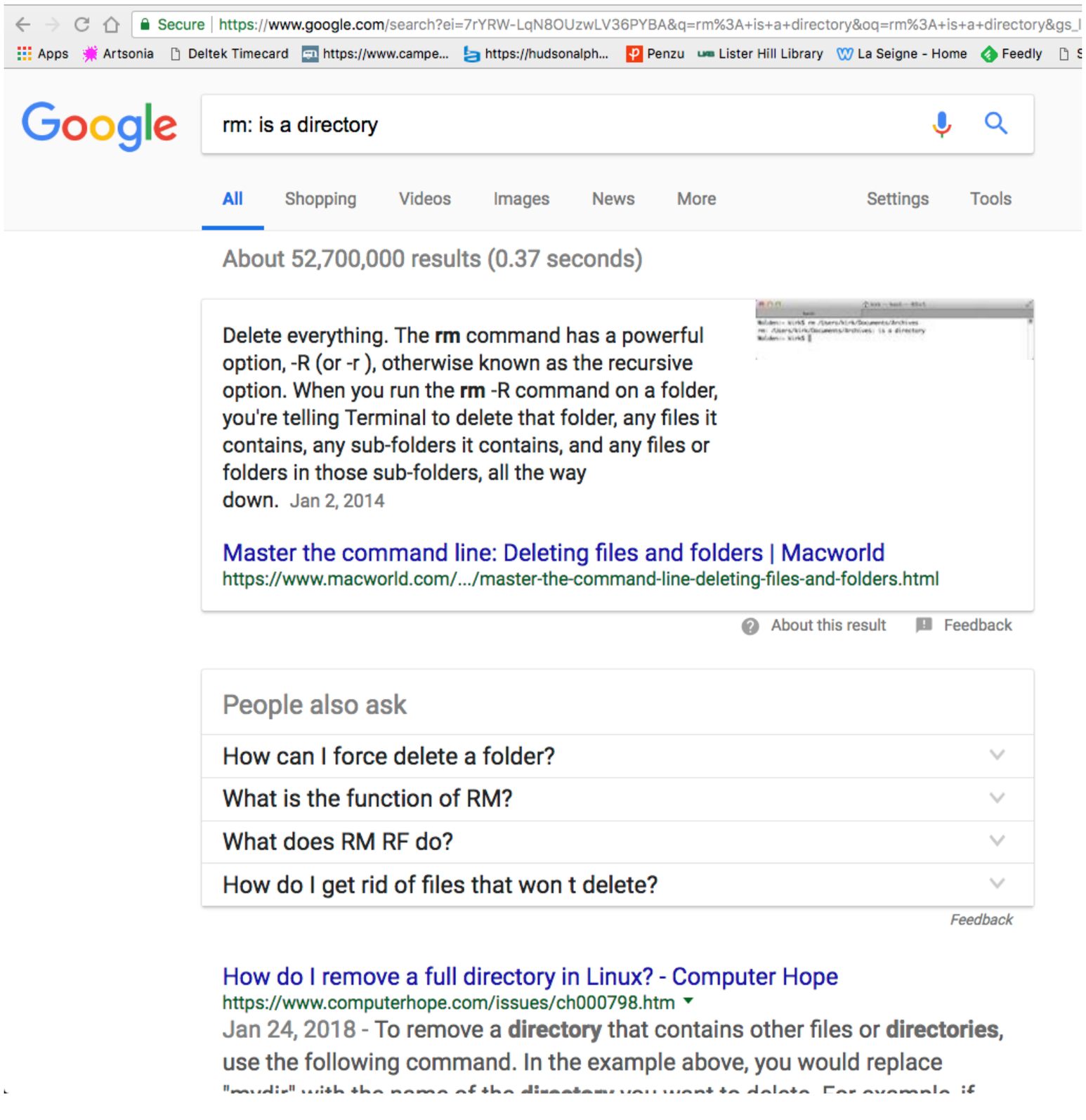


Example 1:
want to delete ‘demo1’ directory

DemoDir — -bash — 80×24

```
Brittany-Lasseigne:DemoDir brittany$ pwd
/Users/brittany/GoogleDrive/Teaching/DemoDir
Brittany-Lasseigne:DemoDir brittany$ ls
demo1
Brittany-Lasseigne:DemoDir brittany$ rm demo1
rm: demo1: is a directory
Brittany-Lasseigne:DemoDir brittany$ ls
demo1
Brittany-Lasseigne:DemoDir brittany$
```

Google the error message:



Check the manual: man rm

DemoDir — less ◂ man rm — 80×29

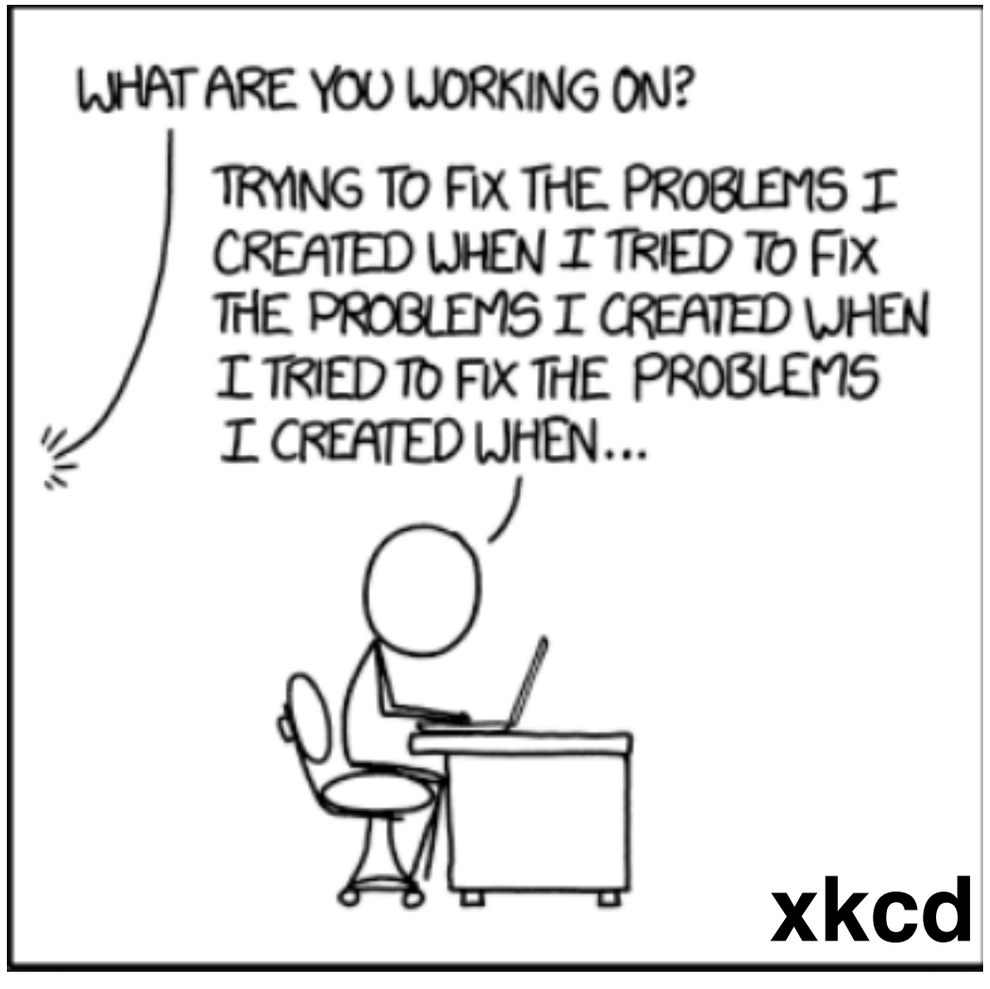
```
RM(1)                    BSD General Commands Manual                    RM(1)

NAME
     rm, unlink -- remove directory entries

SYNOPSIS
     rm [-dfiPRrvW] file ...
     unlink file

DESCRIPTION
     The rm utility attempts to remove the non-directory type files specified
     on the command line.  If the permissions of the file do not permit writ-
     ing, and the standard input device is a terminal, the user is prompted
     (on the standard error output) for confirmation.

     The options are as follows:

     -R          Attempt to remove the file hierarchy rooted in each file
                 argument.  The -R option implies the -d option.  If the -i
                 option is specified, the user is prompted for confirmation
                 before each directory's contents are processed (as well as
                 before the attempt is made to remove the directory).  If the
                 user does not respond affirmatively, the file hierarchy
                 rooted in that directory is skipped.

     -r          Equivalent to -R.
```

It works!

DemoDir — -bash — 80×24

```
Brittany-Lasseigne:DemoDir brittany$ pwd
/Users/brittany/GoogleDrive/Teaching/DemoDir
Brittany-Lasseigne:DemoDir brittany$ ls
demo1
Brittany-Lasseigne:DemoDir brittany$ rm demo1
rm: demo1: is a directory
Brittany-Lasseigne:DemoDir brittany$ ls
demo1
Brittany-Lasseigne:DemoDir brittany$ rm -R demo1
Brittany-Lasseigne:DemoDir brittany$ ls
Brittany-Lasseigne:DemoDir brittany$
```

Thanks! Slides available at https://www.lasseigne.org/post/2018-06-04-biotraincompbioworkshop2018/



Brittany N. Lasseigne, PhD

@bnlasse **blasseigne@hudsonalpha.org**