HuGen2080 book

Daniel E. Weeks

2024-01-06

Table of contents

Preface

This is a Quarto book created from markdown and executable code using Quarto within RStudio.

Book web site: https://danieleweeks.github.io/HuGen2080/

Book source code: https://github.com/DanielEWeeks/HuGen2080

Created by Daniel E. Weeks

Websites:

https://www.sph.pitt.edu/directory/daniel-weeks

To learn more about Quarto books visit https://quarto.org/docs/books.

1 Introduction

This is a Quarto book created from markdown and executable code using Quarto within RStudio.

Book web site: https://danieleweeks.github.io/HuGen2080/

Book source code: https://github.com/DanielEWeeks/HuGen2080

Created by Daniel E. Weeks

Websites:

https://www.sph.pitt.edu/directory/daniel-weeks

2 Logistics

2.1 GitHub: Set up an account

Please go to https://github.com and set up a GitHub account.

Choose your GitHub user name carefully, as you may end up using it later in a professional context.

2.2 GitHub Classroom

As GitHub Classroom will be used to distribute course materials and to submit assignments, it would be best if you get git working on your own computer. The easiest way to do this is to install RStudio, R, and git on your computer.

Please follow the detailed instructions in https://github.com/jfiksel/github-classroom-for-students

In particular, see Step 5 re generating an ssh key so you don't need to login every time.

3 Readings

3.1 Course Preparation

3.1.1 Required Readings

Background reading: Ziegler and König - Chapter 1: Molecular Genetics Ziegler and König - Chapter 2: Formal Genetics

3.1.2 Henry Stewart Talk:

Genotyping algorithms for genome wide association studies/ Dr. Vincent Plagnol

3.2 Models, Maps, and Markers

3.2.1 Learning Objectives

- To review basic genetic models
- To learn about genetic markers

3.2.2 Required Readings

Ziegler and König - Chapter 3: Genetic Markers Ziegler and König - Chapter 5: Genetic Map Distances

3.2.3 Supplementary Readings

Elston RC (2000) Introduction and overview. Statistical methods in genetic epidemiology. Stat Methods Med Res 9:527-541

Chapters 1, 2, and 3: Laird NM, Lange C (2011) The Fundamentals of Modern Statistical Genetics. Springer.

3.2.4 Henry Stewart Talk:

Introductory genetics for statisticians/ Robert C. Elston.

3.3 Study Design Overview

3.3.1 Learning Objectives

- To learn the basic principles of study design for genetic studies
- To understand the vital importance of phenotype definition
- To understand the best sample selection strategies

3.3.2 Active Learning:

Intro to Unix & PLINK

3.3.3 Henry Stewart Talk:

Designing a genome-wide association study/ Dr. Chris Spencer

3.4 No class - Martin Luther King Day

3.5 Familial Aggregation: Recurrence Risk Ratios, Heritability

3.5.1 Learning Objectives

- To learn aggregation analysis
- To learn how to estimate recurrence risk ratios
- To review the concept of heritability

3.5.2 Active Learning:

Student Presentation

3.5.3 Required Readings

Ziegler and König - Chapter 6: Familiality, Heritability, and Segregation Analysis

3.5.4 Supplementary Readings

Chapter 4: Laird NM, Lange C (2011) The Fundamentals of Modern Statistical Genetics. Springer.

3.5.5 Henry Stewart Talk:

Heritability and its uses/ Doug Speed.

Inferring relatedness/ Prof. Emmanuelle Génin

3.6 Familial Aggregation: Segregation Analysis, Ascertainment

3.6.1 Learning Objectives

- To learn about segregation analysis
- To understand how to take ascertainment into account in the segregation models
- To formulate testable hypotheses about genetic models

3.6.2 Active Learning:

SOLAR heritability computer lab

3.6.3 Required Readings

Ziegler and König - Chapter 6: Familiality, Heritability, and Segregation Analysis

3.7 LOD scores: Model-based Linkage Analysis

3.7.1 Learning Objectives

- To learn how to compute LOD scores
- To learn about different map functions, and the distinction between genetic and physical maps
- To formulate testable hypotheses about linkage

3.7.2 Active Learning:

Student Presentation

3.7.3 Required Readings

Ziegler and König - Chapter 7: Model-based Linkage Analysis

3.7.4 Supplementary Readings

Chapters 5 and 6: Laird NM, Lange C (2011) The Fundamentals of Modern Statistical Genetics. Springer.

3.7.5 Henry Stewart Talk:

Linkage and sequence analysis in families/ Christopher Amos.

3.8 Non-parametric methods

3.8.1 Learning Objectives

- To learn how to carry out non-parametric linkage analyses
- To understand the motivation behind non-parametric linkage analysis approaches

3.8.2 Active Learning:

Merlin computer lab

3.8.3 Required Readings

Ziegler and König - Chapter 8: Model-free Linkage Analysis for Dichotomous Traits

3.8.4 Supplementary Readings

Shih MC, Whittemore AS (2001) Allele-sharing among affected relatives: non-parametric methods for identifying genes. Stat Methods Med Res 10:27-55

3.9 Association: Case/Control & Quantitative Traits

3.9.1 Learning Objectives

- To formulate testable hypotheses about association
- To understand and apply various case/control association tests
- To understand allele-based and genotype-based association tests, and trend tests.

3.9.2 Active Learning:

Student Presentation

3.9.3 Required Readings

Ziegler and König - Chapter 10: Fundamental Concepts of Association Analysis Ziegler and König - Chapter 11: Association Analysis with Unrelated Individuals

3.9.4 Supplementary Readings

Chapter 7: Laird NM, Lange C (2011) The Fundamentals of Modern Statistical Genetics. Springer.

Cardon LR, Bell JI (2001) Association study designs for complex diseases. Nat Rev Genet 2:91-99.

Balding DJ (2006) A tutorial on statistical methods for population association studies. Nat Rev Genet 7:781-791

3.9.5 Henry Stewart Talk:

Introduction to genetic association analysis/ Jenny Barrett.

Statistical tests for association/ Dr. Andrew Morris

3.10 Association: Family-based and Haplotype-based

3.10.1 Learning Objectives

- To learn how to analyze family data for association
- To learn how to test haplotypes for association
- To understand sparsity issues involved in haplotyped-based tests

3.10.2 Active Learning:

PLINK computer lab

3.10.3 Required Readings

Ziegler and König - Chapter 12: Association Analysis in Families

3.10.4 Supplementary Readings

Chapter 9 & Chapter 10, Section 2: Laird NM, Lange C (2011) The Fundamentals of Modern Statistical Genetics. Springer.

Ott J, Kamatani Y, Lathrop M (2011) Family-based designs for genome-wide association studies. Nat Rev Genet 12:465-474

3.11 Multiple testing

3.11.1 Learning Objectives

- To understand how to adjust for multiple testing

3.11.2 Active Learning:

Student Presentation

3.11.3 Required Readings

Ziegler and König - Chapter 14, Section 14.4: Multiple Testing

3.11.4 Supplementary Readings

Chapter 10, Section 1: Laird NM, Lange C (2011) The Fundamentals of Modern Statistical Genetics. Springer.

3.11.5 Henry Stewart Talk:

Assessing significance in genome-wide studies/ Dr. David Evans

3.12 Power to detect Association: Linkage vs. Association

3.12.1 Learning Objectives

- To learn how to compute power for detecting association
- To compare and contrast linkage and association
- To understand the relative strengths and weaknesses of linkage and association tests

3.12.2 Active Learning:

Student Presentation

3.12.3 Supplementary Readings

Clerget-Darpoux F, Elston RC (2007) Are linkage analysis and the collection of family data dead? Prospects for family studies in the age of genome-wide association. Hum Hered 64:91-96

3.13 Rare variants

3.13.1 Learning Objectives

- To learn how to test rare variants for association
- To learn about burden tests, collapsing or grouping tests, weighted sum tests, and variable threshold tests.

3.13.2 Active Learning:

Student Presentation

3.13.3 Supplementary Readings

Asimit J, Zeggini E (2010) Rare variant association analysis methods for complex traits. Annu Rev Genet 44:293-308

Bansal V, Libiger O, Torkamani A, Schork NJ (2010) Statistical analysis strategies for association studies involving rare variants. Nat Rev Genet 11:773-785

3.14 Methods for correlated data: LME, GEE, Score

3.14.1 Learning Objectives

- To learn about linear mixed effects models, generalized estimating equations, and score tests
- To learn how to properly model relatedness while testing genetic hypotheses

3.14.2 Active Learning:

Student Presentation

3.14.3 Required Readings

Sul JH, Martin LS, Eskin E. Population structure in genetic studies: Confounding factors and mixed models. PLoS Genet. 2018; 14(12):e1007309. doi:10.1371/journal.pgen.1007309

3.15 Bayesian Methods in Human Genetics

3.15.1 Learning Objectives

- To learn about Bayesian methods in human genetics
- To understand Bayesian principles

3.15.2 Active Learning:

Student Presentation

3.15.3 Required Readings

Stephens M, Balding DJ. Bayesian statistical methods for genetic association studies. Nat Rev Genet. 2009 Oct;10(10):681-90. doi: 10.1038/nrg2615. Review. PubMed PMID: 19763151.

3.15.4 Supplementary Readings

Wakefield J. Bayes factors for Genome-wide association studies: Comparison with P-values. Genetic Epidemiology. 2009;33(1):79–86. DOI: https://doi.org/10.1002/gepi.20359

3.16 Review for Mid-term exam

3.17 Mid-term exam

3.18 Gene x Gene interaction, vQTLs

3.18.1 Learning Objectives

- To learn how to test for gene x gene interaction
- To formulate testable hypotheses about gene x gene interaction

3.18.2 Required Readings

Cordell HJ (2009) Detecting gene-gene interactions that underlie human diseases. Nat Rev Genet 10:392-404

3.18.3 Supplementary Readings

Gilbert-Diamond D, Moore JH (2011) Analysis of gene-gene interactions. Curr Protoc Hum Genet Chapter 1:Unit1 14

3.19 Gene x Environment interaction

3.19.1 Learning Objectives

- To learn how to test for gene x environment interaction
- To formulate testable hypotheses about gene x environment interaction

3.19.2 Active Learning:

Student Presentation

3.19.3 Required Readings

Thomas D (2010) Gene–environment-wide association studies: emerging approaches. Nat Rev Genet 11:259-272

3.19.4 Supplementary Readings

Chapter 10, Section 3: Laird NM, Lange C (2011) The Fundamentals of Modern Statistical Genetics. Springer.

Ottman R (1990) An epidemiologic approach to gene-environment interaction. Genet Epidemiol 7:177-185

3.19.5 Henry Stewart Talk:

Statistical issues in epidemiologic studies of gene-environment interaction/ Peter Kraft, Donna Spiegelman.

GxE interactions in genome-wide association studies/ David V. Conti.

3.20 Spring Recess

3.21 Spring Recess

3.22 Special Topic Lecture by Chris McKennan

3.23 Fine mapping

3.23.1 Learning Objectives

- To learn how to carry out fine mapping
- To understand and apply conditional tests of association

3.23.2 Active Learning:

Student Presentation

3.23.3 Required Readings

Schaid DJ, Chen W, Larson NB. From genome-wide associations to candidate causal variants by statistical fine-mapping. Nature Reviews Genetics. Nature Publishing Group; 2018 Aug 29;19(8):491–504. DOI: https://doi.org/10.1038/s41576-018-0016-z

3.23.4 Supplementary Readings

Cano-Gamez E, Trynka G. From GWAS to Function: Using Functional Genomics to Identify the Mechanisms Underlying Complex Diseases. Front Genet. 2020;11:424. PMID: 32477401 PMCID: PMC7237642 DOI: https://doi.org/10.3389/fgene.2020.00424

3.24 Meta analysis

3.24.1 Learning Objectives

- To learn about the different types of meta analysis.
- To understand the assumptions made by meta analysis.

3.24.2 Required Readings

Ziegler and König - Chapter 14, Section 14.5: Accumulating Data from Genome-wide Association Studies

3.24.3 Henry Stewart Talk:

Winner's curse, replication and meta-analysis/ Frank Dudbridge.

Meta-analysis in genome-wide association studies: application to type 2 diabetes/ Dr. Eleftheria Zeggini

3.25 Methods for multivariate phenotypes

3.25.1 Learning Objectives

- To learn about methods for analyzing multivariate phenotypes
- To learn how to properly account for correlation among phenotypes

3.25.2 Active Learning:

Student Presentation

3.26 Heritability from GWAS

3.26.1 Learning Objectives

- To learn how to estimate heritability using unrelated samples.
- To understand polygenicity.

3.26.2 Active Learning:

Student Presentation

3.26.3 Henry Stewart Talk:

Heritability and its uses/ Doug Speed.

3.27 LDscore regression

3.27.1 Learning Objectives

- To understand the principles of LDscore regression
- To understand polygenicity.

3.27.2 Active Learning:

Student Presentation

3.28 Mendelian Randomization

3.28.1 Learning Objectives

- To understand the basic principles of Mendelian Randomization.
- To formulate testable hypotheses about causation using Mendelian Randomization approaches.

3.28.2 Active Learning:

Student Presentation

3.28.3 Henry Stewart Talk:

Causal inference in genetic epidemiology: Mendelian randomization and beyond / Krista Fischer.

3.29 Genetic Risk Scores & Polygenic Risk Scores & Genomic Prediction

3.29.1 Learning Objectives

- To understand how to construct and use genetic risk scores.
- To understand the limits of genomic predication.

3.29.2 Active Learning:

Student Presentation

3.30 Special Topic Lecture by Lacey Heinsberg

3.30.1 Required Readings

Heinze G, Wallisch C, Dunkler D. Variable selection – A review and recommendations for the practicing statistician. Biometrical Journal. 2018;60(3):431–449. DOI: https://doi.org/10.1002/bimj.201700067

3.31 Special Topic Lecture by Jenna Carlson

3.32 Review for final exam

3.33 Final exam

4 GitHub

4.1 GitHub Introduction lecture

Here's a recording of this lecture (32 minutes 8 seconds):

Recording

4.2 GitHub Introduction slides

PDF slide set

5 Git Commands

5.1 git - best practices

pull - work - commit - pull - push

- `git pull`
- Make changes
- `git commit` your changes to your local repository
- `git pull` the latest remote changes to your local repository
- `git push` your changes.

Pay attention to any error messages.

5.2 Outline of essential Git commands

Here's an outline of essential Git commands, initially created by ChatGPT:

5.2.1 Initialization and Configuration

- `git init`: Initializes a new Git repository in the current directory.
- `git config`: Configure Git settings.

5.2.2 Basic Workflow

- `git add`: Stage changes.
- `git commit -m "message"`: Commits staged changes with a descriptive message.

5.2.3 Remote Repositories

- `git clone`: Clones a remote repository to your local machine.
- `git push`: Send local changes to remote repository.
- `git pull`: Retrieve changes from remote.
- `git remote`: Manage remote repositories.

5.2.4 Status and Changes

- `git status`: Shows the current state of your working directory.
- `git diff`: Displays changes between working directory and the last commit.

5.2.5 History and Logs

- `git log`: View commit history.
- `git log --oneline`: Compact commit history.

5.2.6 Ignoring Files

- Create `.gitignore` file.

5.2.7 Branching

- `git branch`: List/create branches.
- `git checkout`: Switch branches.
- `git merge`: Merge branches.

5.2.8 Undoing Changes

- `git reset`: Unstage or reset changes.
- `git revert`: Create undoing commits.

5.2.9 Tagging

- `git tag`: Create and manage tags.

5.2.10 Stashing

- `git stash`: Temporarily store changes.

6 Basic Shell Commands

6.1 Acknowledgment and License

This chapter is a derivative of the Basic Shell Commands cheat sheet from the DEPRECATED-boot-camps/shell/shell_cheatsheet.md file created by Software Carpentry and is used under the Creative Commons - Attribution license CC BY 3.0

Minor section numbering and formatting changes were made here.

This chapter is licensed under the CC BY 3.0 license by Daniel E. Weeks.

6.2 Shell Basics:

Command	Definition
`.`	a single period refers to the current directory
`..`	a double period refers to the directory immediately above the current directory
`~`	refers to your home directory. *Note:* this command does NOT work on Windows machines (Mac and Linux are okay)
`cd`	changes the current directory to the directory `dirname`
`./dirname`	
`ls -F`	tells you what files and directories are in the current directory
`pwd`	tells you what directory you are in (`pwd` stands for *p*rint *w*orking *d*irectory)
`history`	lists previous commands you have entered. `history | less` lets you page through the list.
`man` *cmd*	displays the *man*ual page for a command.

6.3 Creating Things:

6.3.1 How to create new files and directories..

Command	Definition
`mkdir ./dirname`	makes a new directory called dirname below the current directory. *Note:* Windows users will need to use `\` instead of `/` for the path separator
`nano filename`	if `filename` does not exist, `nano` creates it and opens the `nano` text editor. If the file exists, `nano` opens it. *Note: (i)* You can use a different text editor if you like. In gnome Linux, `gedit` works really well too. *(ii)* `nano` (or `gedit`) create text files. It doesn't matter what the file extension is (or if there is one)

6.3.2 How to delete files and directories...

6.3.2.1 *Remember that deleting is forever. There is NO going back*

Command	Definition
`rm ./filename`	deletes a file called `filename` from the current directory
`rmdir ./dirname`	deletes the directory `dirname` from the current directory. *Note:* `dirname` must be empty for `rmdir` to run.

6.3.3 How to copy and rename files and directories...

Command	Definition
`mv tmp/filename .`	moves the file `filename` from the directory `tmp` to the current directory. *Note: (i)* the original `filename` in `tmp` is deleted. *(ii)* `mv` can also be used to rename files (e.g., `mv filename newname`
`cp tmp/filename .`	copies the file `filename` from the directory `tmp` to the current directory. *Note: (i)* the original file is still there

6.4 Pipes and Filters

6.4.1 How to use wildcards to match filenames...

Wildcards are a shell feature that makes the command line much more powerful than any GUI file managers. Wildcards are particularly useful when you are looking for directories, files, or file content that can vary along a given dimension. These wildcards can be used with any command that accepts file names or text strings as arguments.

6.4.1.1 Table of commonly used wildcards

Wildcard	Matches
`*`	zero or more characters
`?`	exactly one character
`[abcde]`	exactly one of the characters listed
`[a-e]`	exactly one character in the given range
`[!abcde]`	any character not listed
`[!a-e]`	any character that is not in the given range
`{software,carpentry}`	exactly one entire word from the options given

See the cheatsheet on regular expressions on the second page of this PDF cheatsheet for more "wildcard" shortcuts.

6.4.2 How to redirect to a file and get input from a file ...

Redirection operators can be used to redirect the output from a program from the display screen to a file where it is saved (or many other places too, like your printer or to another program where it can be used as input).

Command	Description
`>`	write `stdout` to a new file; overwrites any file with that name (e.g., `ls *.md > mardkownfiles.txt`)
`>>`	append `stdout` to a previously existing file; if the file does not exist, it is created (e.g., `ls *.md >> markdownfiles.txt`)
`<`	assigns the information in a file to a variable, loop, etc (e.g., `n < markdownfiles.md`)

6.4.2.1 How to use the output of one command as the input to another with a pipe...

A special kind of redirection is called a pipe and is denoted by `|`.

Command	Description
`|`	Output from one command line program can be used as input to another one (e.g. ls *.md | head gives you the first 5 `*.md` files in your directory)

6.4.2.1.1 Example:

```
ls *.md | head | sed -i `s/markdown/software/g`
```

changes all the instances of the word `markdown` to `software` in the first 5 `*.md` files in your current directory.

6.5 How to repeat operations using a loop...

Loops assign a value in a list or counter to a variable that takes on a different value each time through the loop. There are 2 primary kinds of loops: `for` loops and `while` loops.

6.5.1 For loop

For loops loop through variables in a list

```
for varname in list
do
    command1 $varname
    command2 $varname
done
```

where,

- `for`, `in`, `do`, and `done` are keywords
- `list` contains a list of values separated by spaces. e.g. `list` can be replaced by `1 2 3 4 5 6` or by `Bob Mary Sue Greg`. `list` can also be a variable:
- `varname` is assigned a value without using a `$` and the value is retrieved using `$varname`

–

```
list[0]=Sam
list[1]=Lynne
list[2]=Dhavide
list[3]=Trevor
.
.
.
list[n]=Mark
```

which is referenced in the loop by:

```
for varname in ${list[@]}
do
    command1 $varname
    command2 $varname
done
```

Note: Bash is zero indexed, so counting always starts at `0`, not `1`.

6.5.2 While Loop

While loops loop through the commands until a condition is met. For example

```
COUNTER=0
while [ ${COUNTER} -lt 10 ]; do
    command 1
    command 2
    COUNTER=`expr ${COUNTER} + 1`
done
```

continues the loop as long as the value in the variable COUNTER is less than 10 (incremented by 1 on each iteration of the loop).

- `while`, `do`, and `done` are keywords

6.5.2.1 Commonly used conditional operators

Operator	Definition
`-eq`	is equal to
`-ne`	is not equal to
`-gt`	greater than
`-ge`	greater than or equal to
`-lt`	less than
`-le`	less than or equal to

Use `man bash` or `man test` to learn about other operators you can use.

6.6 Finding Things

6.6.1 How to select lines matching patterns in text files...

To find information within files, you use a command called `grep`.

Example command	Description
`grep [options] day haiku.txt`	finds every instance of the string `day` in the file haiku.txt and pipes it to standard output

6.6.1.1 Commonly used `grep` options

	`grep` options
`-E`	tells grep you will be using a regular expression. Enclose the regular expression in quotes. *Note:* the power of `grep` comes from using regular expressions. Please see the regular expressions sheet for examples
`-i`	makes matching case-insensitive
`-n`	limits the number of lines that match to the first n matches
`-v`	shows lines that do not match the pattern (inverts the match)
`-w`	outputs instances where the pattern is a whole word

6.6.2 How to find files with certain properties...

To find file and directory names, you use a command called `find`

Example command	Description
`find . -type d`	`find` recursively descends the directory tree for each path listed to match the expression given in the command line with file or directory names in the search path

6.6.2.1 Commonly used `find` options

	`find` options
`-type [df]`	`d` lists directories; `f` lists files
`-maxdepth n`	`find` automatically searches subdirectories. If you don't want that, specify the number of levels below the working directory you would like to search
`-mindepth n`	starts `find`'s search `n` levels below the working directory

7 Summary

In summary, this book is a work in progress.

References