Biostatistics in Practice (BiP) 2014:

Data Sciences for the Life Sciences in a High Performance Computing Environment

Sponsored by the **Graduate Program in Biostatistics at UMass Amherst**, *the UMass Institute for Computational Biology, Biostatistics and Bioinformatics (ICB3) and the Massachusetts Green High Performance Computing Center (MGHPCC).*

Principles and Practice of Reproducible Research with R

Author: Andrea Foulkes, Gregory Matthews, Nicholas Reich

Biostatistics in Practice: Research Training in High-Performance Computing with R

This material is part of the **statsTeachR** *project*

Agenda

MODULE 1: 9:00AM –10:15AM

Principles of Reproducible Research

- Brief conceptual overview; Git/GitHub; RMarkdown

MODULE 2: 10:45AM – Noon

Simulation and Parallel Computing

- Simulation example with simple linear regression; Performing a permutation test; Running permutation tests in parallel

MODULE 3: 1:45PM – 3:00PM

Introduction to Cluster Computing

- VPN and ssh to MGHPCC; Data/code transfer; Parallel vs cluster computing; Submitting jobs (single and distributed)

MODULE 4: 3:30PM – 4:45 PM

Topics in Big Data Sciences

- Genome wide association data overview; Lab exercise applying workshop material

Special Thanks...

Individuals making this possible

- John Goodhue
- Claire Christopherson
- Karen Utgoff
- Ralph Zottola
- Al Ritacco & Chris Hull
- UMass OIT
  - Chris Misra
  - Jim Mileski
  - Jason Houghton
  - Genti Lagji

Special Thanks...

Support provided by:

- President's office S&T Initiative
- Vice Chancellor Research & Engagement
- UMass College of Natural Sciences
- UMass School of Public Health and Health Sciences
- MGHPCC
- UMass Graduate Program in Biostatistics

Introductions

Faculty:

- Dr. Nicholas Reich, Assistant Professor of Biostatistics, UMass Amherst
- Dr. Gregory Mathews, Lecturer in Biostatistics, UMass Amherst
- Dr. Andrea Foulkes, Associate Professor and Head of Biostatistics, UMass Amherst

Teaching Assistants:

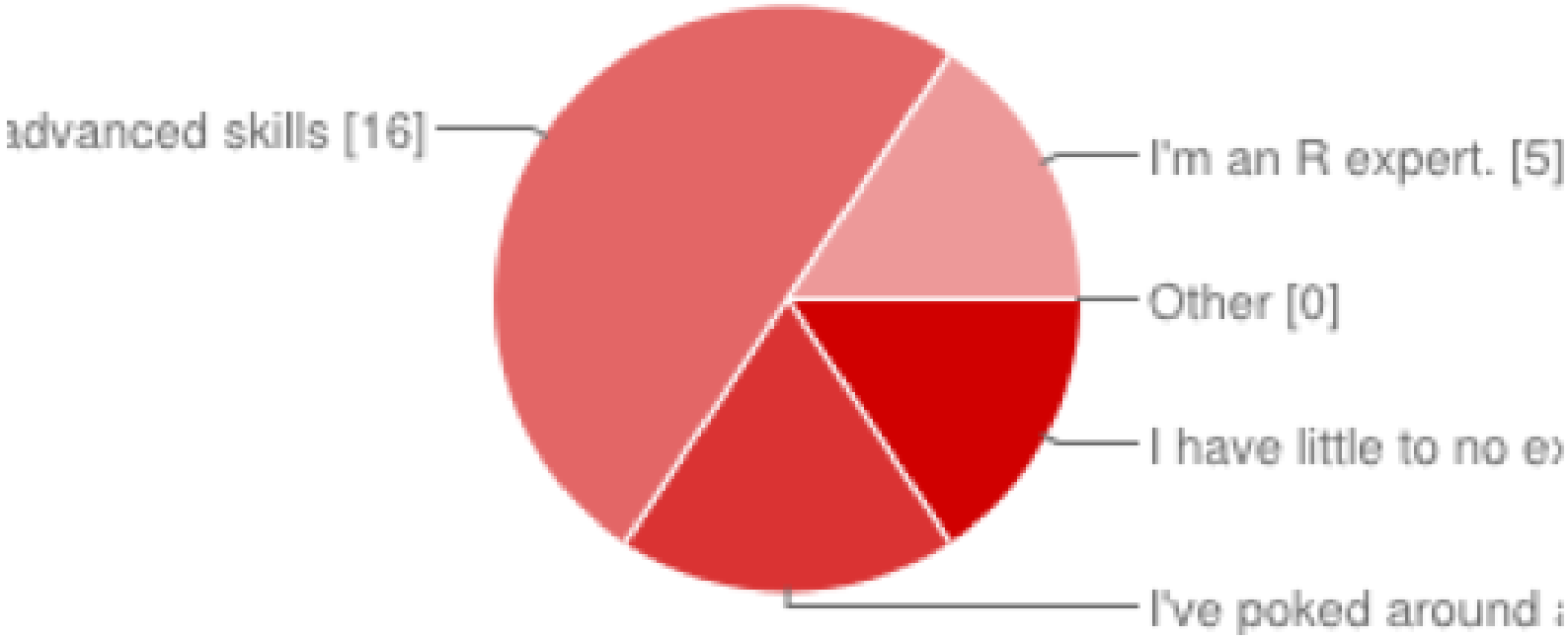
- Steven Calderbank
- Eric Cohen
- Stephen Lauer
- Sara Nuñez
- Emily Ramos
- Eric Reed

Where we are coming from

- UMass-Amherst
  - Biostatistics
  - Mathematics and Statistics
  - Linguistics
  - Landscape Ecology
  - Veterinary and Animal Sciences Department
  - Political Science
  - Resource Economics
  - Biology
  - Organismic and Evolutionary Biology
  - Environmental Conservation
- UMass Medical School: Quantitative Health Sciences
- Babson College
- Northeastern University
- Framingham Heart Study
- Boston University

Diverse background in R

What is your experience with R?



I have little to no experience with R	**5**	16%
I've poked around a bit with R, but am still not too familiar	**6**	19%
I've used R quite a bit, but I'm looking for more advanced skills	**16**	50%
I'm an R expert.	**5**	16%
Other	**0**	0%

Principles of Reproducible Research – Definition

Reproducible research has been defined in the scientific community as published scientific work that can be recreated using code and data made available by the authors:

- Creating reproducible research requires authors to carefully document approaches used to process, manage, analyze, and visualize data.
- It also requires authors to have a foundational understanding of the uncertainty that underlies the statistical model they use to describe their data.

Principles of Reproducible Research – A Brief History

- Roots of reproducible research can be traced to the concept of literate programming heralded by Donald Knuth

  – Knuth, D. E. (1992). Literate Programming (1st ed.). Center for the Study of Language and Information.

- Concept operationalized in 2002 by Friederic Leisch with introduction of Sweave, a program that allows the user to weave together R code and natural language descriptions

  – Leisch, F. (2002a). Sweave. Dynamic generation of statistical reports using literate data analysis. SFB Adaptive Information Systems and Modelling in Economics and Management Science, WU Vienna University of Economics and Business; Leisch, F. (2002b). Sweave, part I: Mixing R and LaTeX. R News, 2/3, 2831.

- Importance of reproducibility discussed in vast array of fields, from econometrics, epidemiology and biostatistics, bioinformatics, and engineering

  – Koenker, R. (1996). Reproducible econometric research. Retrieved September 17, 2012, from: `http://www.econ.uiuc.edu/~roger/research/repro/repro.html`; Peng, R. D. (2009). Reproducible research and Biostatistics. Biostatistics, 10(3), 405408. doi:10.1093/biostatistics/kxp014; Gentleman, R. (2005). Reproducible research: a bioinformatics case study. Statistical applications in genetics and molecular biology, 4, Article2. doi:10.2202/1544-6115.1034; Vandewalle, P., Barrenetxea, G., Jovanovic, I., Ridolfi, A., & Vetterli, M. (2007). Experiences with Reproducible Research in Various Facets of Signal Processing Research. IEEE International Conference on Acoustics, Speech and Signal Processing. Proceedings, 4, IV1256. doi:10.1109/ICASSP.2007.367304)

Principles of Reproducible Research – Training

Foundational training in reproducible research includes rigorous instruction in:

1. Analysis and visualization of data through literate programming (Module 1 & 2);
2. Dissemination of open-source and extensible code, documentation, and, whenever possible, data (Modules 1 & 2);
3. Basic statistical literacy, including characterization of uncertainty using sound statistical and computational principles (Modules 2, 3 & 4).

Tools for reproducible data analysis with R

Topics

- R/RStudio
- Version control: git & GitHub.com
- ggplot2
- Dynamic documents: knitr, RMarkdown, Sweave

Introduction to R

What is R?

- R is a language and environment for statistical computing and graphics.
- Website: www.r-project.org

Introduction to R

Pros

- R is free.
- There are many packages for R. Chances are someone has written it already.
- Many cutting edge techniques are available very quickly.

Introduction to R

Cons

- R is free. This means there is no support for R.
- This also means that you use a package at your own risk and you trust the author wrote the code correctly.
- Takes a little while to learn.

RStudio

What is Rstudio^{TM}?

- RStudio? is a free and open source integrated development environment (IDE) for R.
- rstudio.org

RStudio

Demo...

Tools for reproducible data analysis with R

Topics

- R/RStudio
- Version control: git & GitHub.com
- ggplot2
- Dynamic documents: knitr, RMarkdown, Sweave

Version control systems

Common VCS

- git
- subversion (svn)
- mercurial
- ...

Version control and reproducibility

Why version control?

- allows you to roll back to previous versions easily
- allows you to try things out without disrupting existing code
- versions can be flagged as “releases”

Version control systems

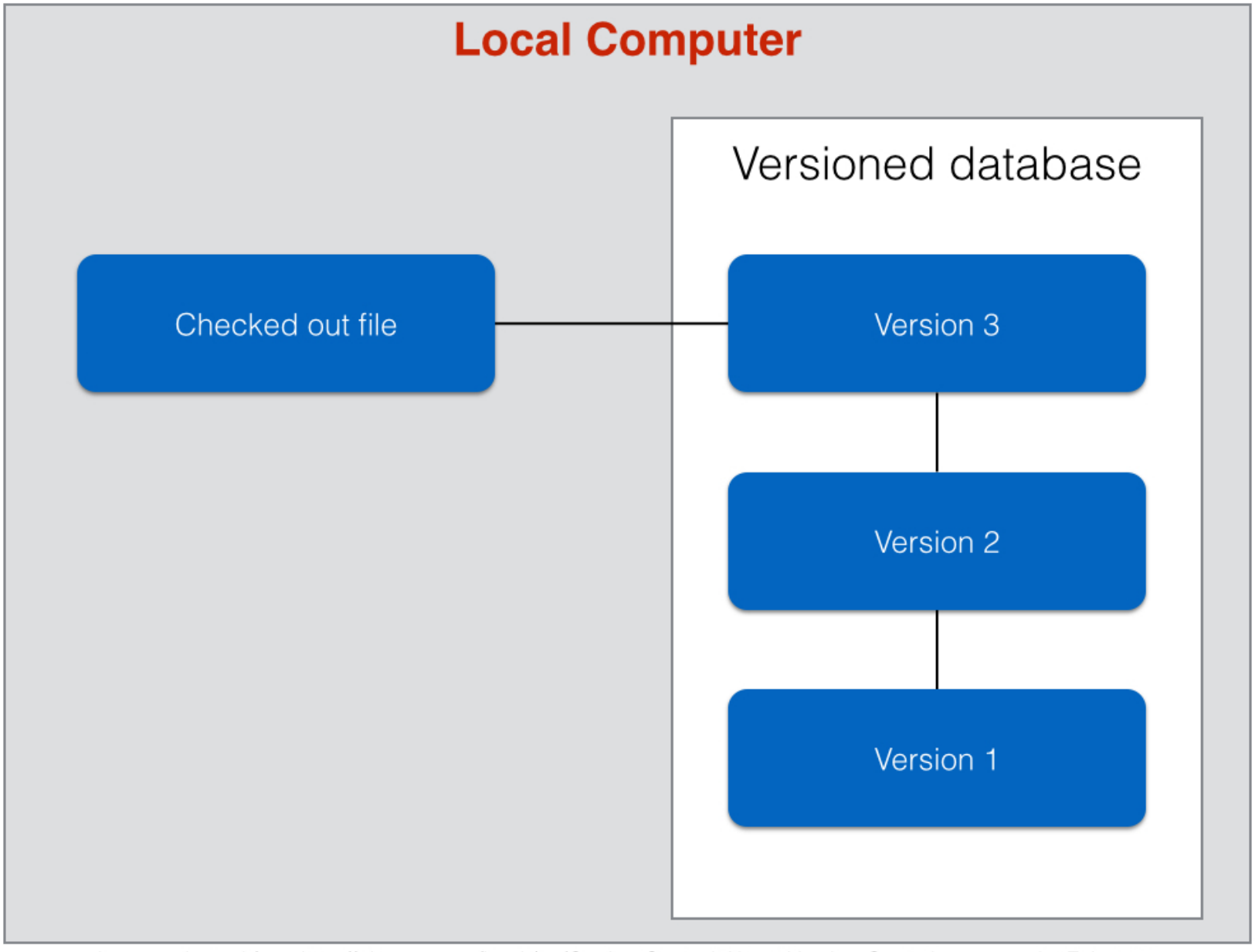


Image adapted from http://git-scm.com/book/en/Getting-Started-About-Version-Control, accessed 6 Feb 2014

Version control systems (git flavored)

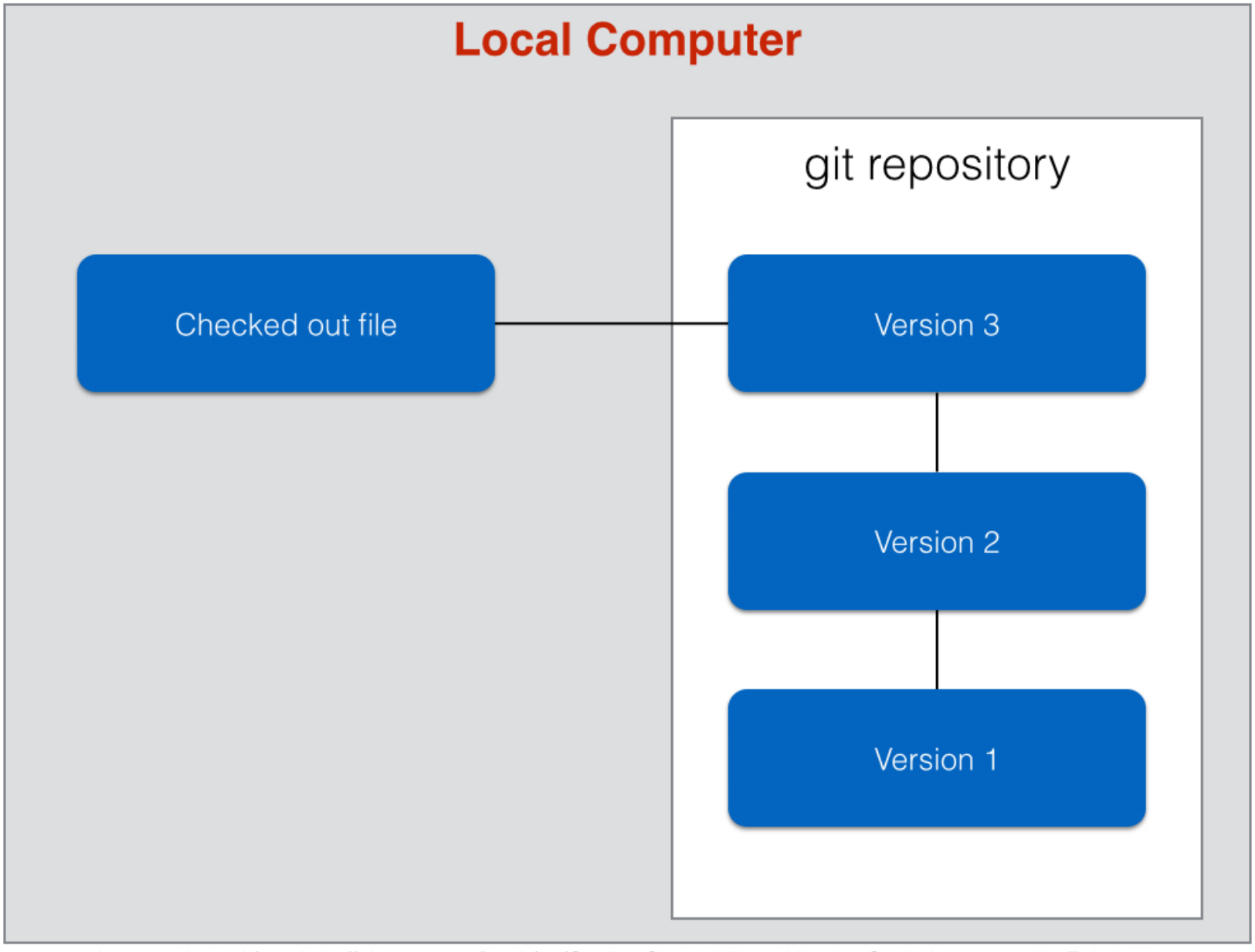


Image adapted from http://git-scm.com/book/en/Getting-Started-About-Version-Control, accessed 6 Feb 2014

Version control systems

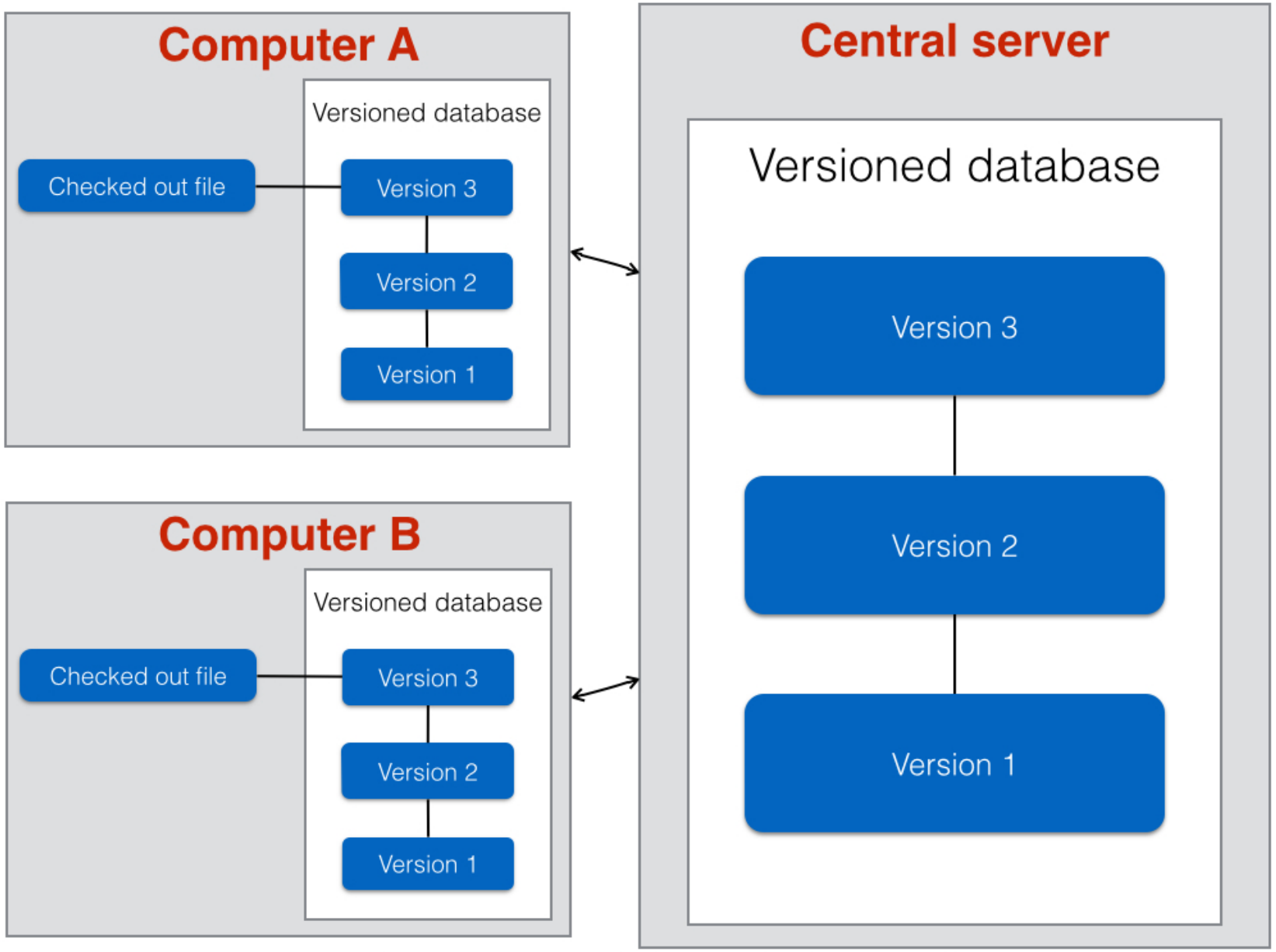


Image adapted from http://git-scm.com/book/en/Getting-Started-About-Version-Control, accessed 6 Feb 2014

Version control systems (git/GitHub flavored)

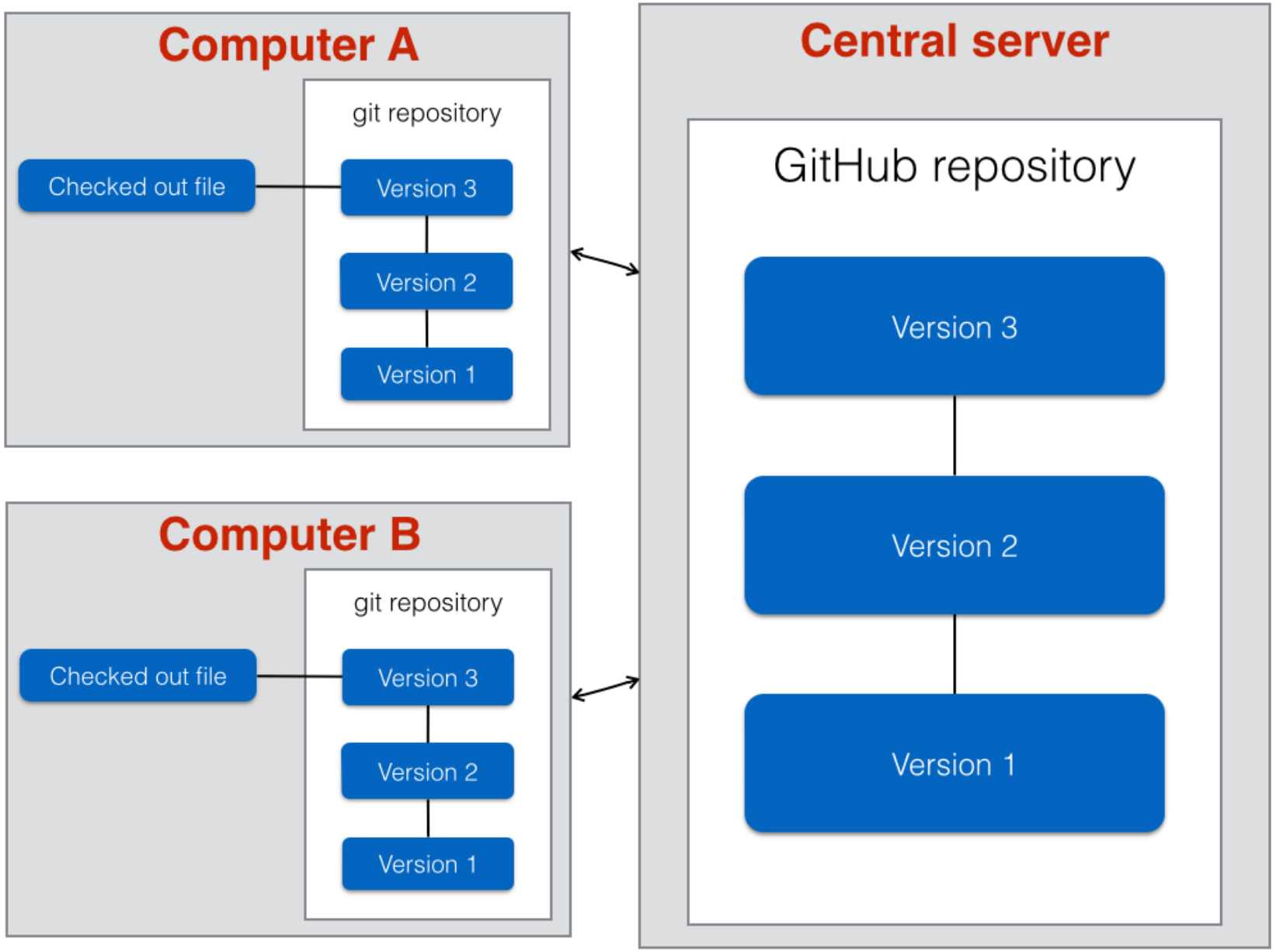


Image adapted from http://git-scm.com/book/en/Getting-Started-About-Version-Control, accessed 6 Feb 2014

Lots of (mostly free) options for cloud-based version controlling

Most services host multiple types of VCS

- sourceforge.net
- github.com
- bitbucket.org
- springloops.io
- ... [what have you used?]

git is a dialect

Key command-line operations

- `git init`: initializes a repository locally
- `git clone`: clones a repository from a remote source (i.e. GitHub.com)
- `git branch`: creates a new "branch" of code
- `git add`, `git rm`: manipulating files
- `git commit`: commits changes you have made

Using git with RStudio

Demo...

- clone the BiPSandbox repository from GitHub
- simple commit/push/pull examples

Tools for reproducible data analysis with R

Topics

- R/RStudio
- Version control: git & GitHub.com
- ggplot2
- Dynamic documents: knitr, RMarkdown, Sweave

Arguments for using ggplot2

From ggplot2 users

- "literate programming: you describe the plot almost in a natural language. It makes it much easier to reuse code and come back to a graph in a few years time without feeling lost."
- "flexibility, intuitiveness, and logic of the mapping between the data and its representation. Once one gets his mind wrapped around the grammar of graphics concepts (and particularly the aesthetic mapping), figuring out how to best represent a dataset is much easier than with other graphical representation methods."
- "I think the big strength of ggplot2 is graphic-artist quality default output. And an information-focussed point of view."
- Lots more reasons here..

Today, we will be using very simple ggplot2 commands.

Tools for reproducible data analysis with R

Topics

- R/RStudio
- Version control: git & GitHub.com
- ggplot2
- Dynamic documents: knitr, RMarkdown, Sweave

Dyanamic Documents in R

- Dynamic R documents allow a user to combine text, R code, and R output, including tables and figures, into one document.
- Why is this useful?
  - Writing code and producing reports are now one document rather than many.
  - When the analysis changes, the results in the report change automatically.
  - What else?
- There are several options for how to do this.
  - R Markdown
  - Sweave
  - knitr

Dyanamic R Reports: Summary

- R Markdown: creates HTML document
- Sweave: creates pdf document AND incorporates LaTeX
- knitr: $\approx$ Sweave + cacheSweave + pgfSweave + weaver + animation::saveLatex + R2HTML::RweaveHTML + highlight::HighlightWeaveLatex + 0.2 * brew + 0.1 * SweaveListingUtils + more

R Markdown

- R Markdown creates HTML files
- Reference: http://www.rstudio.com/ide/docs/authoring/using_markdown
- Markdown files (.Rmd) act just like text files, except they allow a user to embed R code in chunks
- The syntax for a chunk in R Markdown:

Regular text
```{r}
Code goes here
```

Sweave

- Sweave creates pdf files as output
- Reference: http://leisch.userweb.mwn.de/Sweave/
- Sweave Manual: http://www.stat.uni-muenchen.de/ leisch/Sweave/Sweave-manual.pdf
- Sweave not only integrates R code, but also LaTeX!
- The syntax for a chunk in Sweave:

Regular text with $LaTeX$ if you want it.
<<OPTIONS >>=
Code goes here
@

Sweave: Options

- fig=TRUE (or FALSE): This indicates that the code in the chunk will print the figure to the output pdf document
- echo=TRUE (or FALSE): Should the R input code be displayed in the output pdf document
- eval=TRUE (or FALSE): Should the R input code be evaluated

knitr

- Created by Ph.D. student Yihui Xie (what have you done?)
- knitr creates pdf files as output.
- It also allows the use of LaTeX, like Sweave, whereas R Markown does not.
- Syntax for knitr is largely the same as Sweave.
- Xie describes knitr $\approx$ Sweave + cacheSweave + pgfSweave + weaver + animation::saveLatex + R2HTML::RweaveHTML + highlight::HighlightWeaveLatex + 0.2 * brew + 0.1 * SweaveListingUtils + more