"Essential skills for reproducible research computing"

Universidad Técnica Federico Santa María

3–6 January 2017

A Barba-group workshop for graduate students

`https://barbagroup.github.io/essential_skills_RRC/`

Lorena A. Barba, The George Washington University

with Barba-group members:

Gilbert Forsyth

Natalia Clementi

@gforsyth

@gilforsyth

@ncclementi

@ncclementi

A syllabus for research computing

1. command line utilities in Unix/Linux
2. an open-source scientific software ecosystem (our favorite is Python's)
3. software version control (we advocate the distributed kind: our favorite is git)
4. good practices for scientific software development: code hygiene and testing
5. knowledge of licensing options for sharing software

`https://barbagroup.github.io/essential_skills_RRC/`

Universidad Técnica Federico Santa María, Valparaíso

3 January 2017

Open Licensing

Prof. Lorena A. Barba

Mechanical and Aerospace Engineering Department
The George Washington University

@LorenaABarba

Casting Out Nines

math ∩ tech ∩ edu

4+1 Interview: Lorena Barba

Why do you advocate so strongly for open-source technology in research and education?

`http://rtalbert.org/blog/2015/interview-lorena-barba`

Free & Open-source Software (FOSS)

An invention of great impact:

- an alternative to intellectual-property instruments
- OS licenses allow people to coordinate their work freely

Open-source licenses:

People can coordinate their work freely, within the confines of copyright law, while making access and wide distribution a priority.

Open source

It's not sufficient to make the source public to read. We must attach a **license** that allows others to modify and distribute the code.

The Open Definition

"Open data and content can be freely used, modified, and shared by anyone for any purpose."

http://opendefinition.org

Open source & Reproducibility

Def.— Reproducible research

Authors provide all the necessary data and the computer codes to run the analysis again, re-creating the results.

Schwab, M., Karrenbach, N., Claerbout, J. (2000) “Making scientific computations reproducible,” *Comp. Sci. Eng.* Vol. 2(6):61–67

REPRODUCIBLE RESEARCH

Reproducible Research in Computational Harmonic Analysis

1521-9615/09/$25.00 © 2009 IEEE
Copublished by the IEEE CS and the AIP

David L. Donoho, Arian Maleki, and Morteza Shahram
Stanford University
Inam Ur Rahman
Apple Computer
Victoria Stodden
Harvard Law School

http://dx.doi.org/10.1109/MCSE.2009.15

Guide to Licenses

Everyone developing software in an academic setting should have working knowledge of software licenses.

OPEN ACCESS Freely available online

PLoS COMPUTATIONAL BIOLOGY

Education

A Quick Guide to Software Licensing for the Scientist-Programmer

Andrew Morin[1], Jennifer Urban[2], Piotr Sliz[1]*

1 Department of Biological Chemistry and Molecular Pharmacology, Harvard Medical School, Boston, Massachusetts, United States of America, **2** Samuelson Law, Technology & Public Policy Clinic, School of Law, University of California Berkeley, Berkeley, California, United States of America

Software is a creative work, and copyright is *automatically* attached to it.

Always add a license to software you plan to make public.

Permissive vs. copy-left

Permissive licenses

- Fewest restrictions
- Allow use, distribution, modification
- Only require that code authors be given credit
- Best choice for academic use
- E.g., Berkeley Software Distribution (BSD), MIT License, Apache License

Permissions

- Commercial Use
- Distribution
- Modification
- Private Use

Conditions

- License and Copyright Notice

Limitations

- Liability
- Warranty

Copy-left licenses

- Guarantees perpetual access to the source code
- Requires any derivative work be under the same license
- A.k.a. “share-alike” licenses
- Are considered restrictive
- E.g., GPL license

Permissions

- Commercial Use
- Distribution
- Modification
- Patent Use
- Private Use

Conditions

- Disclose Source
- License and Copyright Notice
- Same License
- State Changes

Limitations

- Liability
- Warranty

Preamble

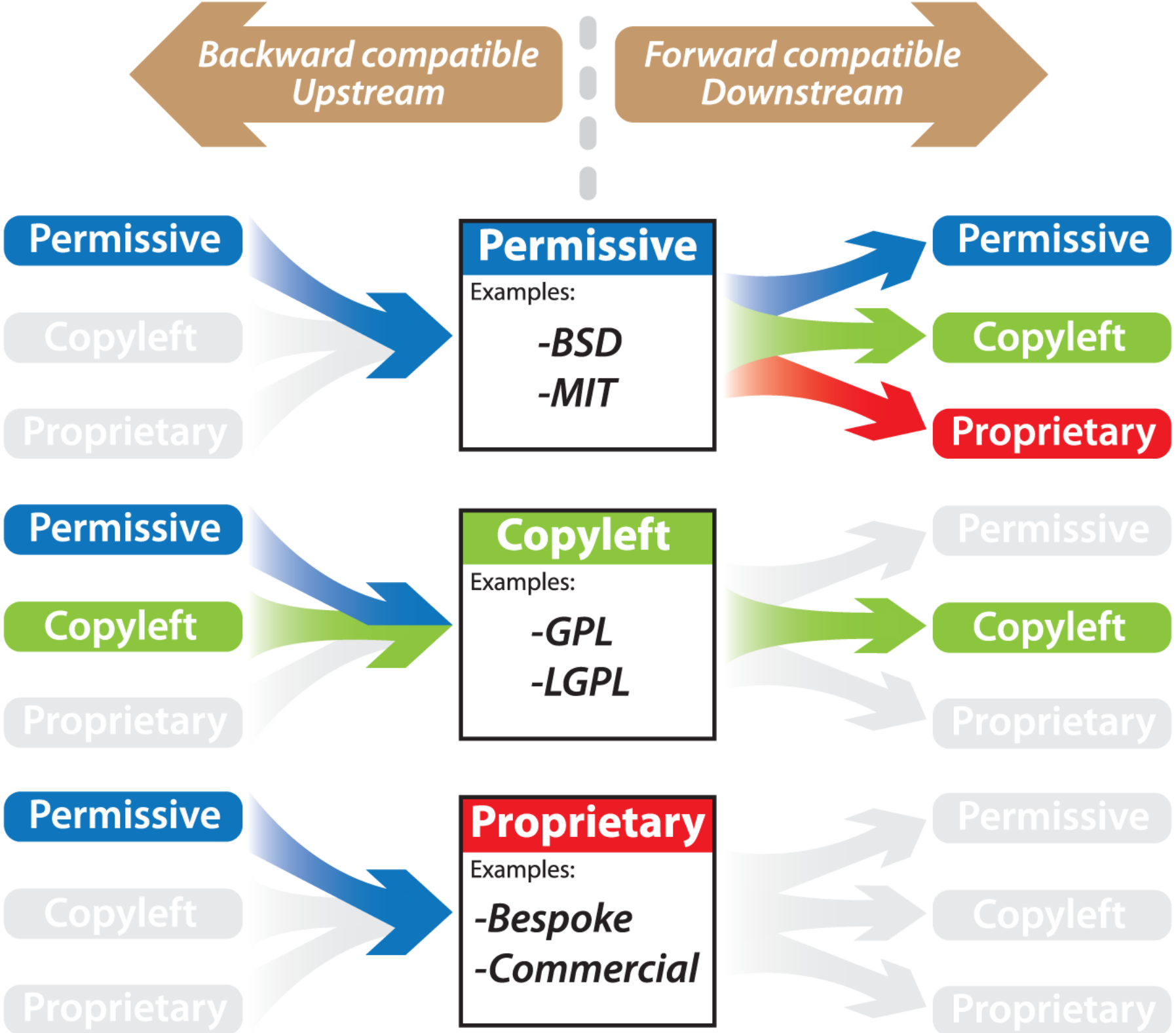
The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you

License compatibility

Compatible licenses allow source code from different works to be combined to make new software. Not all licenses are compatible!


Backward compatible
Upstream
Forward compatible
Downstream
Permissive
Copyleft
Proprietary
Permissive
Examples:
-BSD
-MIT
Permissive
Copyleft
Proprietary
Permissive
Copyleft
Proprietary
Copyleft
Examples:
-GPL
-LGPL
Permissive
Copyleft
Proprietary
Permissive
Copyleft
Proprietary
Proprietary
Examples:
-Bespoke
-Commercial
Permissive
Copyleft
Proprietary
Credit: Morin et al. (2012)

How to choose?

For academic work: simple & permissive is best.

`http://choosealicense.com/`

Subtlety: MIT vs. BSD 3-clause

- MIT License:

  "The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software."

- BSD 3-clause:

  "Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer."

Bonus advice:

Write into your grant proposals that your research software will be released under an OSI-approved license.