Towards a Census of the Use of Licenses in Free and Open Source

Daniel M German
Professor
Department of Computer Science
University of Victoria
Canada

FOSS licenses

- OSI has approved more than 70 different licenses
- Many more in the “wild”
  - Original General Public License
  - Beerware v42
  - Do What the F**k You Want To Public License v2

How systems state their license

- The license of a system is stated in:
  - source code files
  - README files
  - COPYING files
  - Metadata of the project
    - Sourceforge/Launchpad/GoogleCode (often outdated and/or incorrect)
- Others abstract this information:
  - RedHat
  - Debian

Understanding and Auditing the Licensing of Open Source Software Distributions

Daniel M. German†, Massimiliano Di Penta‡, Julius Davies†
† Dept. of Computer Science, University of Victoria, Canada
‡ Dept of Engineering, University of Sannio, Italy
dmg@uvic.ca, dipenta@unisannio.it, juliusd@uvic.ca

***Abstract*—Free and open source software (FOSS) is often distributed in binary packages, sometimes part of GNU/Linux operating system distributions, or part of products distributed/sold to users.**

FOSS creates great opportunities for users, developers and integrators, however it is important for them to understand the licensing requirements of any package they use. Determining the license of a package and assessing whether it depends on other software with incompatible licenses is not trivial. Although this task has been done in a labor intensive manner by software distributions, automatic tools to perform this analysis are highly desired.

This paper proposes a method to understand licensing compatibility issues in software packages, and reports an empirical study aimed at auditing licensing issues in binary packages of the Fedora-12 GNU/Linux distribution. The objective of this study is (i) to understand how the license declared in packages is consistent with those of source code files, and (ii) to audit the licensing information of Fedora-12, highlighting cases of incompatibilities between dependent packages.

The obtained results—supported by feedback received from Fedora contributors—show that there exist many nuances in determining the license of a binary package from its source code, as well as cases of license incompatibility issues due to package dependencies.

When one installs a new application/library in a Unix system, this is often done from what is known as a binary package (such as *RPM* packages in Fedora/Redhat-like distributions or *.deb* packages in Debian-like distributions). Other than the various artifacts composing the application/library, the package also contains metadata describing, among other things, (i) under what open source license the package is distributed (which we call the *declared license of the binary package)*, and (ii) the list of other packages required in order to successfully install and use the current package (its *required packages*) [2].

From a legal point of view, modifying and redistributing a FOSS package poses two important issues:

1) *Can we trust the declared license of the package? i.e.*, is that license consistent with those of the files the package contains?
2) *Do the dependency requirements of a binary package create potential legal concerns?* Software with different licenses can be combined to create larger systems, but such combinations can increase the chance for license incompatibilities.

Measuring FOSS Licenses use

- Empirical vs anecdotal
  - Can I replicate the results?
- Ex:
  - Blackduck yearly census

Rank	License	%
1.	GNU General Public License (GPL) 2.0	32.65%
2.	Apache License 2.0	12.84%
3.	GNU General Public License (GPL) 3.0	11.62%
4.	MIT License	11.28%
5.	BSD License 2.0	6.83%
6.	Artistic License (Perl)	6.27%
7.	GNU Lesser General Public License (LGPL) 2.1	6.19%
8.	GNU Lesser General Public License (LGPL) 3.0	2.62%
9.	Eclipse Public License (EPL)	1.61%
10.	Code Project Open 1.02 License	1.33%
11.	Microsoft Public License	1.32%
12.	Mozilla Public License (MPL) 1.1	1.08%
13.	Common Development and Distribution License (CDDL)	0.31%
14.	BSD 2-clause "Simplified" or "FreeBSD" License	0.30%
15.	Common Public License (CPL)	0.26%
16.	zlib/libpng License	0.23%
17.	Academic Free License	0.20%
18.	GNU Affero GPL v3	0.16%
19.	Microsoft Reciprocal License (Ms-RL)	0.14%
20.	Open Software License (OSL)	0.14%

Source: Black Duck Software

Danger: be suspicious of a census

- Methodology?
- Tool used?
- Accuracy?
- Licenses identified?
- Names used?

Rank	License	%
1.	GNU General Public License (GPL) 2.0	32.65%
2.	Apache License 2.0	12.84%
3.	GNU General Public License (GPL) 3.0	11.62%
4.	MIT License	11.28%
5.	BSD License 2.0	6.83%
6.	Artistic License (Perl)	6.27%
7.	GNU Lesser General Public License (LGPL) 2.1	6.19%
8.	GNU Lesser General Public License (LGPL) 3.0	2.62%
9.	Eclipse Public License (EPL)	1.61%
10.	Code Project Open 1.02 License	1.33%
11.	Microsoft Public License	1.32%
12.	Mozilla Public License (MPL) 1.1	1.08%
13.	Common Development and Distribution License (CDDL)	0.31%
14.	BSD 2-clause "Simplified" or "FreeBSD" License	0.30%
15.	Common Public License (CPL)	0.26%
16.	zlib/libpng License	0.23%
17.	Academic Free License	0.20%
18.	GNU Affero GPL v3	0.16%
19.	Microsoft Reciprocal License (Ms-RL)	0.14%
20.	Open Software License (OSL)	0.14%

Source: Black Duck Software

The challenges

- What is the universe of FOSS?
- How do I find it?
- What is an individual?
  - New versions v. old?
  - Forks v original?
    - Github?
  - Embedded copies?
    - Common to copy dependencies to simplify dependency management

Towards a Census

- Choose a corpus
  - It will be biased
  - Repositories vs distributions
    - Debian, RedHat, …

      or
    - Maven2, github, sourceforge, CPAN, CTAN, etc.

Linux

- Licensed under the GPLv2
  - Contains:

Licenses found in Linux
General Public License v2 (GPLv2)
General Public License v2 or any later version (GPLv2+)
Library General Public License v 2 (LGPLv2)
Lesser General Public License v 2.1 (LGPLv2.1)
New BSD –3 clauses (BSD-3)
BSD 2 Clauses (BSD-2)
MIT/X11
… and many others

FreeBSD

- FreeBSD:
  - Licensed under the BSD-2 license
- It contains GPL code!!!
  - Disabled by default
  - If enabled
    - the license of the kernel can't be BSD-2 any more

```
/* $FreeBSD: src/sys/gnu/dev/sound/pci/maestro3_dsp.h,v 1.5 2005/01/06 18:27:30 imp Exp $ *
/*-
 *      ESS Technology allegro audio driver.
 *
 *      Copyright (C) 1992-2000  Don Kim (don.kim@esstech.com)
 *
 *      This program is free software; you can redistribute it and/or modify
 *      it under the terms of the GNU General Public License as published by
 *      the Free Software Foundation; either version 2 of the License, or
 *      (at your option) any later version
```

Source code license identification

- Originally we started using Fossology (Fossology.org)
  - But:
    - it was way too slow (computational expensive):
      - Few seconds to minutes per file
    - It wasn't able to distinguish between *NO License*, and *I don't know the license*
- So we created our own:
  - Ninka
    - http://github.com/dmgerman/ninka

This is no longer true in
It is fast, and detects "No license"

A sentence-matching method for automatic license identification of source code files

Daniel M. German
University of Victoria, Canada
dmg@uvic.ca

Yuki Manabe
Osaka University, Japan
y-manabe@ist.osaka-u.ac.jp

Katsuro Inoue
Osaka University, Japan
inoue@ist.osaka-u.ac.jp

ABSTRACT

The reuse of free and open source software (FOSS) components is becoming more prevalent. One of the major challenges in finding the right component is finding one that has a license that is adequate for its intended use. The license of a FOSS component is determined by the licenses of its source code files. In this paper, we describe the challenges of identifying the license under which source code is made available, and propose a sentence-based matching algorithm to automatically do it. We demonstrate the feasibility of our approach by implementing a tool named `Ninka`. We performed an evaluation that shows that `Ninka` outperforms other methods of license identification in precision and speed. We also performed an empirical study on 0.8 million source code files of Debian that highlight interesting facts about the manner in which licenses are used by FOSS.

version 2.1 license, allowing both Free/Open source software and proprietary software development."

In an empirical study, Li et al. reported that 37% of companies that used OSS components modify their source code [14]. They also emphasize that any organization wanting to reuse FOSS components (either with or without modification) should consider the legal implications of such changes.

The legal issues of reusing FOSS components affect not only companies, but other FOSS components and applications. As reported in [9], FOSS applications are also concerned about licensing issues, primarily if the license of the FOSS component is compatible with the license of the application that uses it. If they are not, then the component cannot be used.

One of the major challenges of intellectual property clearance is to identify the license under which a FOSS component, and each of its files, is made available. This is due

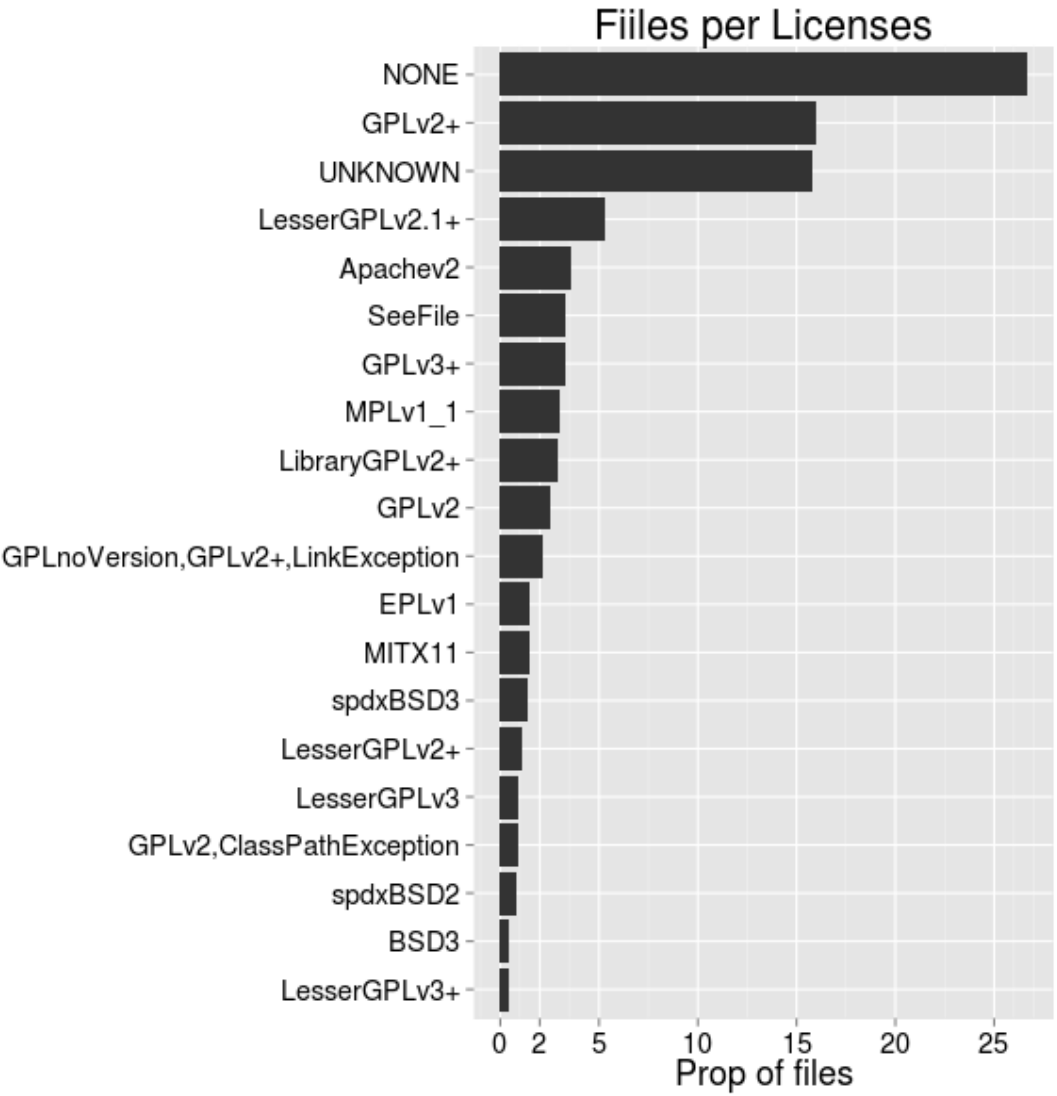
Ninka showed us

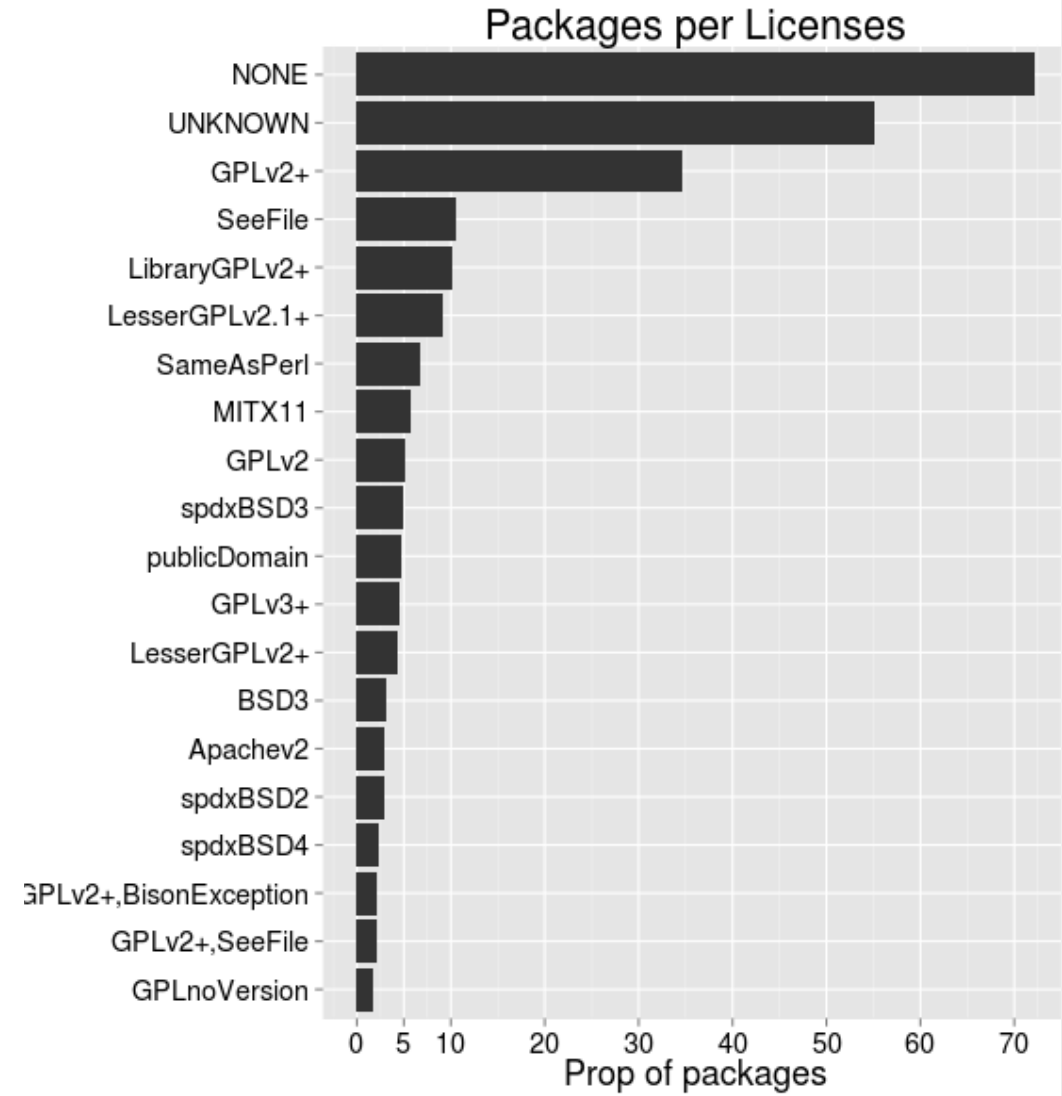
- That license identification of source code is far from trivial
  - Finding the license statement
  - How many licenses does it contain?
    - How do they interact?
  - Language related issues
  - License customization

Debian 6.0

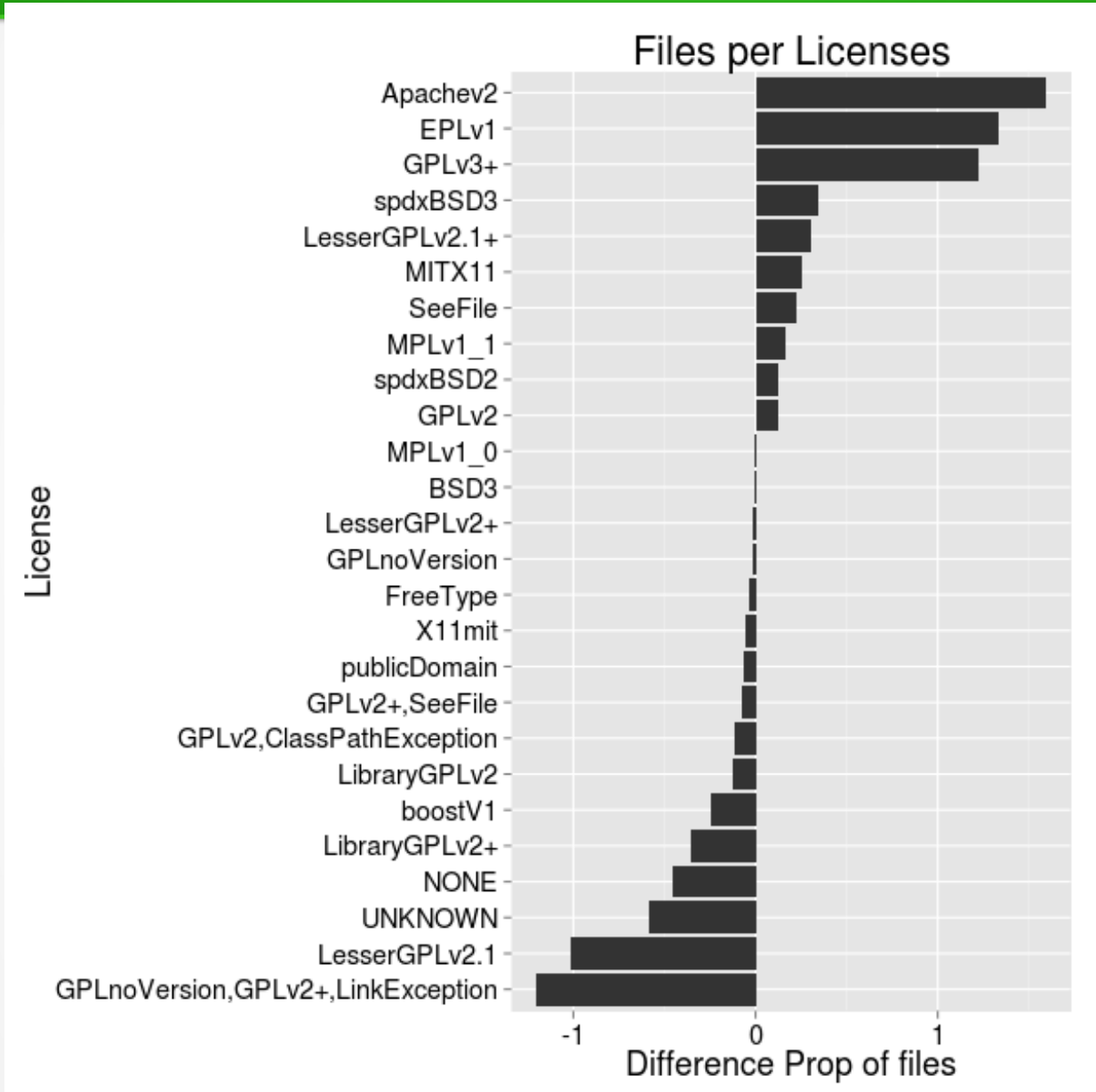
- Used Ninka
- Scanned source code of few main programming languages
  - Perl, python, C, java

Debian 6.0

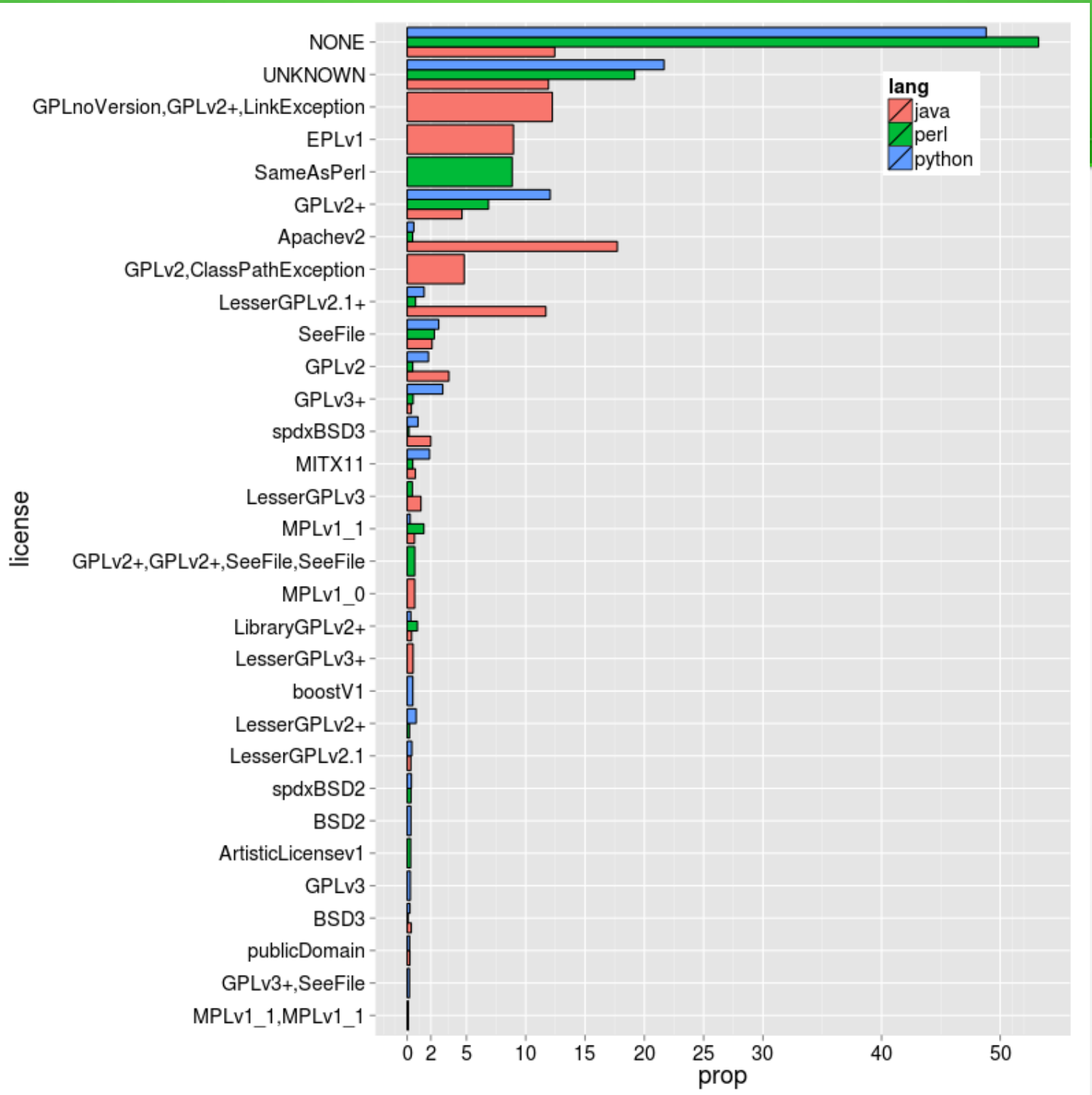

Fiiles per Licenses
NONE
GPLv2+
UNKNOWN
LesserGPLv2.1+
Apachev2
SeeFile
GPLv3+
MPLv1_1
LibraryGPLv2+
GPLv2
GPLnoVersion,GPLv2+,LinkException
EPLv1
MITX11
spdxBSD3
LesserGPLv2+
LesserGPLv3
GPLv2,ClassPathException
spdxBSD2
BSD3
LesserGPLv3+
0 2 5 10 15 20 25
Prop of files

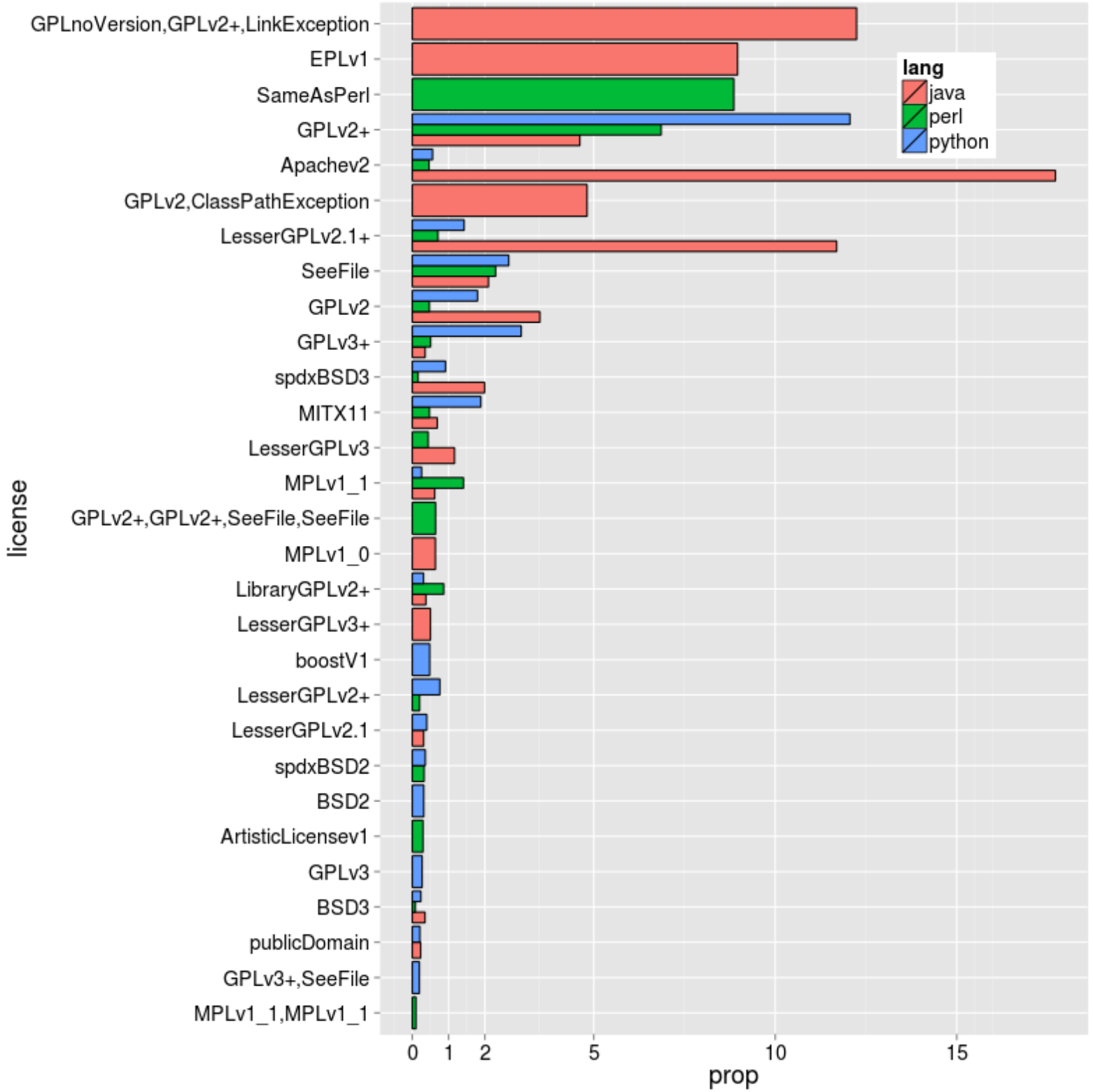

Packages per Licenses
NONE
UNKNOWN
GPLv2+
SeeFile
LibraryGPLv2+
LesserGPLv2.1+
SameAsPerl
MITX11
GPLv2
spdxBSD3
publicDomain
GPLv3+
LesserGPLv2+
BSD3
Apachev2
spdxBSD2
spdxBSD4
GPLv2+,BisonException
GPLv2+,SeeFile
GPLnoVersion
0 5 10 20 30 40 50 60 70
Prop of packages

Difference

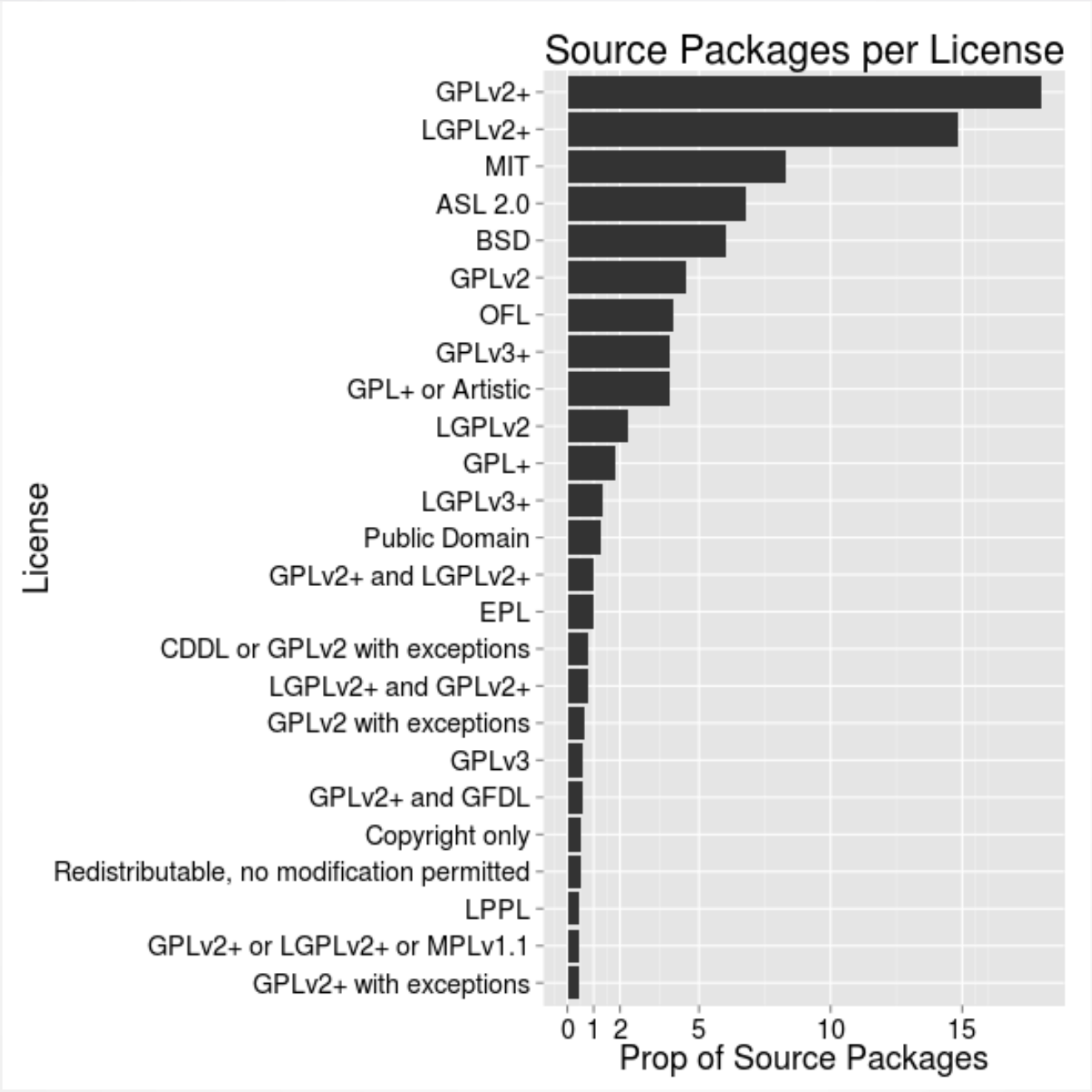


Not included: lesserGPLv3, sameAsPerl

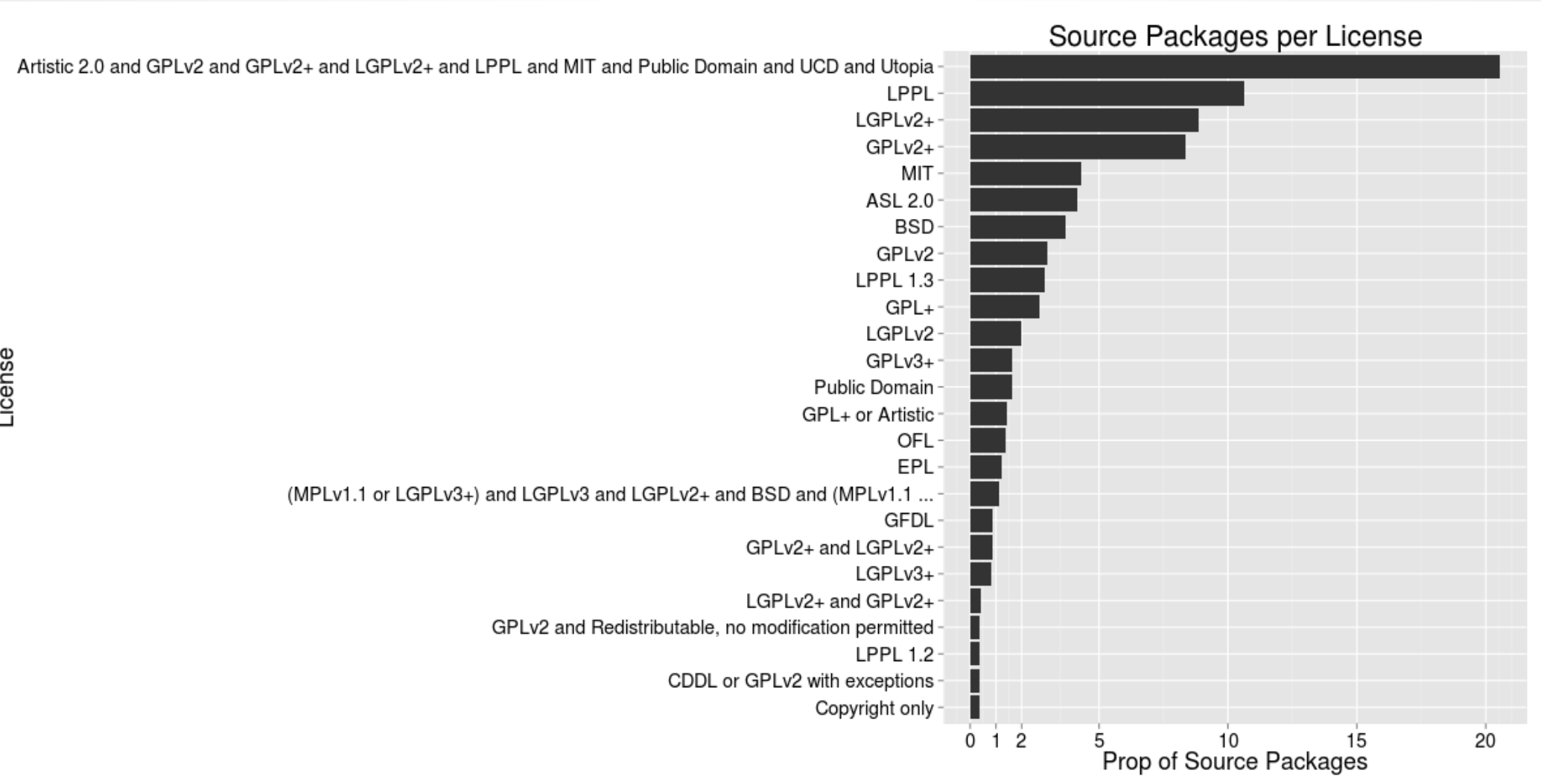

NONE
UNKNOWN
GPLnoVersion,GPLv2+,LinkException
EPLv1
SameAsPerl
GPLv2+
Apachev2
GPLv2,ClassPathException
LesserGPLv2.1+
SeeFile
GPLv2
GPLv3+
spdxBSD3
MITX11
LesserGPLv3
MPLv1_1
GPLv2+,GPLv2+,SeeFile,SeeFile
MPLv1_0
LibraryGPLv2+
LesserGPLv3+
boostV1
LesserGPLv2+
LesserGPLv2.1
spdxBSD2
BSD2
ArtisticLicensev1
GPLv3
BSD3
publicDomain
GPLv3+,SeeFile
MPLv1_1,MPLv1_1
license
lang
java
perl
python
0
2
5
10
15
20
25
30
40
50
prop


GPLnoVersion,GPLv2+,LinkException
EPLv1
SameAsPerl
GPLv2+
Apachev2
GPLv2,ClassPathException
LesserGPLv2.1+
SeeFile
GPLv2
GPLv3+
spdxBSD3
MITX11
LesserGPLv3
MPLv1_1
GPLv2+,GPLv2+,SeeFile,SeeFile
MPLv1_0
LibraryGPLv2+
LesserGPLv3+
boostV1
LesserGPLv2+
LesserGPLv2.1
spdxBSD2
BSD2
ArtisticLicensev1
GPLv3
BSD3
publicDomain
GPLv3+,SeeFile
MPLv1_1,MPLv1_1
license
lang
java
perl
python
0
1
2
5
10
15
prop

Fedora 18 Source Packages



Fedora18 Binary Packages



Conclusions

- Counting licenses is tricky and dangerous
  - Decide on a corpus
  - Decide on what an “entity” is
  - Keep in mind that identification is hard
    - Tools make mistakes: errors + ignore
- Different communities/languages/products tend to use different licenses
  - Any census will be biased

Towards a Census of the Use of Licenses in Free and Open Source

Daniel M German
Professor
Department of Computer Science
University of Victoria
Canada

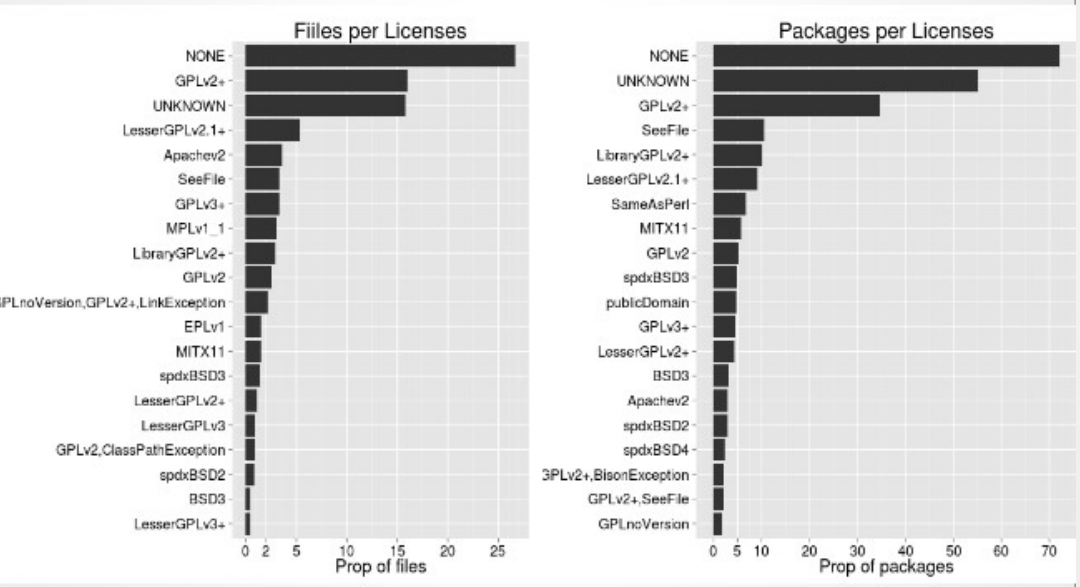
Danger: be suspicious of a census

- Methodology?
- Tool used?
- Accuracy?
- Licenses identified?
- Names used?

Rank	License	%
1.	GNU General Public License (GPL) 2.0	32.65%
2.	Apache License 2.0	12.84%
3.	GNU General Public License (GPL) 3.0	11.62%
4.	MIT License	11.28%
5.	BSD License 2.0	6.83%
6.	Artistic License (Perl)	6.27%
7.	GNU Lesser General Public License (LGPL) 2.1	6.19%
8.	GNU Lesser General Public License (LGPL) 3.0	2.62%
9.	Eclipse Public License (EPL)	1.61%
10.	Code Project Open 1.02 License	1.33%
11.	Microsoft Public License	1.32%
12.	Mozilla Public License (MPL) 1.1	1.08%
13.	Common Development and Distribution License (CDDL)	0.31%
14.	BSD 2-clause "Simplified" or "FreeBSD" License	0.30%
15.	Common Public License (CPL)	0.26%
16.	zlib/libpng License	0.23%
17.	Academic Free License	0.20%
18.	GNU Affero GPL v3	0.16%
19.	Microsoft Reciprocal License (Ms-RL)	0.14%
20.	Open Software License (OSL)	0.14%

Source: Black Duck Software

Debian 6.0



Conclusions

- Counting licenses is tricky and dangerous
  - Decide on a corpus
  - Decide on what an “entity” is
  - Keep in mind that identification is hard
    - Tools make mistakes: errors + ignore
- Different communities/languages/products tend to use different licenses
  - Any census will be biased