

Lizenz für Softwareentwicklung am IPHT

Claudia Beleites

2012-07-16 Mo

Contents

1	Überlegungen zu einer Lizenz für cbmodels	1
1.1	Vorschlag	1
2	Überlegungen, die in Triest zur GPL geführt haben	2
2.1	Infrastruktur	2
2.2	Langzeit-Pflege	2
2.3	Veröffentlichung als wissenschaftlicher Artikel	3
2.4	Nutzung, Weiterentwicklung und Derivate	4

1 Überlegungen zu einer Lizenz für cbmodels

- sollen Lizenzen nach außen (IPC, Rest der Welt) gegeben werden oder nicht?
- anderweitige Veröffentlichung (eigenes Paper)?

1.1 Vorschlag

- ich schlage GPL \geq 3.0 vor
- da das Paket die Pakete MASS und pls nutzt, die selbst unter GPL stehen, ist sowieso nur eine GPL-kompatible Lizenz möglich.
- Veröffentlichung als eigener Artikel im Journal of Statistical Software (peer-review des Quellcodes) ist angestrebt.

2 Überlegungen, die in Triest zur GPL geführt haben

Vielleicht ist es ganz hilfreich, wenn ich mal zusammenstelle, wie wir an der Uni Triest für hyperSpec zu dem Schluss gekommen sind, es unter einer freien, quelloffenen Lizenz (FOSS = free and open source software) zu veröffentlichen. Letztlich ist die Entscheidung zwischen FOSS und proprietärer Verwertung eine ziemliche Weichenstellung.

2.1 Infrastruktur

- Auf das CRAN (weltweit verfügbare Struktur zum Vertrieb von R Packages) darf nur FOSS Software
- (“Drittmittelgeber” Google Summer of Code verlangt FOSS Veröffentlichung)
- FOSS Software kann r-forge nutzen (Entwicklungsserver; weniger wichtig, wenn wir einen eigenen solchen Server betreiben)

2.2 Langzeit-Pflege

- geringeres Risiko, dass das gesamte Wissen mit Weggang des Entwicklers verloren geht
- höhere Chance, von Entwicklung vor und nach dem Arbeitsvertrag zu profitieren
- bei FOSS größerer Nutzerkreis, das bedeutet auch: mehr und verteiltes Wissen (Nutzer können Fragen im Internet stellen und bekommen sinnvolle Antworten)
- Chance auf Beiträge von Nutzern zur Software
- Bei größerem Nutzerkreis “amortisiert” sich der hohe Aufwand beim Schreiben von langfristig verständlichem Code
- Thema Aufwand:
 - Bei hyperSpec schätze ich, dass zwischen 1/2 und 2/3 des Entwicklungsaufwandes in die Dokumentation und Handbücher gehen.

- softclassval implementiert komplett neue chemometrische Methoden; hier gibt es kein “Nutzerhandbuch” wie bei hyperSpec (satt dessen gleich einen Artikel), aber viele Unit-Tests zum Überprüfen, ob die Rechnungen auch stimmen. Von diesem Paket sind etwas über 1/3 der Codezeilen Dokumentation und Kommentare, und jeweils etwas unter 1/3 Code, der die eigentliche Funktionalität bereitstellt und ebensoviel Code für die Unit-Tests.
- bei FOSS kann ich als Entwickler/Maintainer sehr viel einfacher als bei proprietären Lizenzen Nutzeranfragen nach Erweiterungen beantworten mit
 - dem Hinweis, dass die Software FOSS ist, der Nutzer also alle erforderlichen Anpassungen selbst vornehmen darf und kann.
 - Je nachdem, wie ich das allgemeine Interesse an der angesprochenen Funktionalität einschätze, füge ich hinzu, dass ich aber gern bereit bin bzw. bitte darum, den so entstandenen Code unter Nennung des jeweiligen Autors in das Paket aufzunehmen (und damit die Wartung übernehme), wenn er allgemeingültig geschrieben und ordentlich dokumentiert ist. (Das macht auch dem Nutzer das Leben dann viel bequemer).

2.3 Veröffentlichung als wissenschaftlicher Artikel

- Veröffentlichung mit peer-review des Quellcodes beim Journal of Statistical Software an FOSS-Lizenz gebunden, ähnlich bei anderen Zeitschriften die Software als wissenschaftliche Veröffentlichung behandeln.
- Für chemometrische Methoden mit entsprechender “Erfindungshöhe” ist ggf. eine Veröffentlichung ohne Offenlegung des Quellcodes möglich. Ich gehe aber davon aus, dass das in der Regel nicht erstrebenswert ist:
 - Entweder die Methode ist im Artikel genau genug beschrieben, dass sie durch Neuprogrammieren reproduziert werden kann. Dann halte ich es für besser, die Implementierung FOSS zu veröffentlichen. Dadurch wird unsere Software verwendet, anderenfalls verwenden fremde Gruppen ihre eigene.
 - Ist die Methode nicht so genau beschrieben, dass sie einfach implementiert werden kann, so ist die Reproduzierbarkeit im Sinne guter wissenschaftlicher Praxis nicht gegeben.

- Soll die Methode nicht offengelegt werden, dann wäre eher Markenrecht gefragt als wissenschaftliche Veröffentlichungen über die Methode (im Unterschied zu wiss. Veröffentlichungen von Anwendungen, die mit dieser Methode bearbeitet wurden).
- Zitat klar einforderbar

2.4 Nutzung, Weiterentwicklung und Derivate

- In Triest haben wir beschlossen, die Software unter GPL zu veröffentlichen. So ist einerseits FOSS ist sichergestellt, und andererseits, dass Derivate ebenfalls unter einer solchen Lizenz stehen müssen.
- sozusagen als akademische Form der Lizenzgebühr fordern wir Zitate ein
- Die Behandlung einer proprietären Lizenz haben wir als wesentlich zu aufwändig eingestuft (zumal da ja eine ganze Menge Infrastruktur und ggf. langfristige Verpflichtungen dranhängen).
- mir als Entwickler gibt die FOSS Lizenz Rechtssicherheit über die Nutzung & Entwicklung über den jeweiligen Arbeitsvertrag hinaus
 - es ist unerheblich, wo gewonnenes Erfahrungswissen aufhört und die Verwertungsrechte des IPHT beginnen
 - es ist vollkommen unproblematisch, wenn ich nochmal in den Code schauen soll, um Fragen zu beantworten.
- Gemeinsame Nutzung an IPC und IPHT (oder ggf. mit anderen Kooperationspartnern) braucht keine weiteren Vereinbarungen.