SOFTWARE PATENTS AS A BARRIER TO SCIENTIFIC TRANSPARENCY: AN UNEXPECTED CONSEQUENCE OF BAYH-DOLE

Victoria Stodden and Isabel Reich
Columbia University

ABSTRACT

Computing has taken a central role in the practice of science. From the ability to capture data, methods, create simulations, and provide dissemination mechanisms, science has gone digital. We examine the interaction between the digitization of science and Intellectual Property Law, specifically the incentives created by the Bayh-Dole Act to patent inventions associated with university-based research. We first show that the number of software patents granted to faculty researchers has more than doubled over the last ten years among top patenting universities and colleges. In a traditional scientific setting methods are usually openly shared in the methods section of a scientific publication, but due to increased levels of complexity and detail deep intellectual contributions to science are now being captured only in the software and codes that generate published computational results. In computational science reproducibility of results can typically only be effected with knowledge of the underlying code and data – the traditional methods section is insufficient for computational science. Because of the ability to patent software, incentives to patent academic code are potentially at odds with scientific norms of transparency and reproducibility.

We collect a novel dataset to understand the rise in academic software patenting and its potential for occluding reproducible research – where the code and data underlying published results is openly shared. We implement a predictive model and find preliminary evidence that the increasing number of academic software patents is associated with increasing levels of industry collaboration (defined as at least one industry-based patent author) in patentable research, but there a weakly negative association with published articles (defined as those cited in the patent application). This extends a result in the literature to software patents: that patents arise more frequently from collaborations with industry, but does not confirm a second result, that positive reinforcement effect on publication by faculty who seek patents. From the perspective of reproducible computational research, this indicates that prior publications are either not be made if there is an intent to patent, or not be listed on the patent application. These findings indicate that the question of whether code needed to reproducible computational results is being patented is open and important.

JASA June Issue	Computational Articles
1996	9 of 20 (45%)
2006	33 of 35 (94%)
2009	32 of 32 (100%)
2011	29 or 29 (100%)

Proportion of Articles Published in the Journal of the American Statistical Association with a Computational Component, by Year

UNIVERSITY SOFTWARE PATENTING

Background: In 1980 Congress enacted two laws, the Stevenson-Wydler Act and the Bayh-Dole Act, intended to promote the development of technologies arising from federally funded research by permitting licensing agreements between research entities and for-profit companies. The Bayh-Dole Act explicitly gave federal agency grantees and contractors, most notably universities and research institutions, title to government-funded inventions and charged them with using the patent system to encourage disclosure and commercialization of the inventions.

From 2000 to 2012 our sample of universities obtains more than twice as many patents, from 44 to 104. Note also that the increase in software patents is not steady and is subject to variability over these 13 years. Note that Rai et al. used fewer school and a different definition of software patent. Although our trends are congruent, the absolute numbers in the two figures differ.

There is an unexpected decrease in the percentage of software patents starting around 2010, even though their absolute numbers continue to rise.

Percent Software Patents of Total Academic Patents

Total Number of Software Patents filed by the top 23 University Patent Filers, 2000-2012

Software Patents as a Percent of the Total University Patent Portfolio, 2000-2012

We compare the overall patent rank of universities in our sample, to their rank for software patents in 2008.

The adjacent table shows the dominance of both the University of California and MIT in software and non-software patents in both 2002 and 2008. The University of Wisconsin's rank has risen in the six years since 2002, and Caltech's and Stanford University's have fallen. The software patenting rank and overall patenting rank tend to be quite similar for each school, with notable exceptions of the University of Pennsylvania (much lower software rank than overall patent rank in 2008, compared to about even in 2002) and both Georgia Tech and the University of Illinois rank lower for software patenting in 2008 compared to overall patenting than they did in 2002.

Assignees	Software Rank 2002	Overall Rank 2002	Software Rank 2008	Overall Rank 2008
University of California	2	1	1	1
Massachusetts Institute of Technology	1	2	2	2
Wisconsin Alumni Research Foundation	7	7	4	6
University of Texas	5	5	5	4
Stanford University	3	4	6	5
California Institute of Technology	4	3	8	3
Johns Hopkins University	10	6	11	7
Columbia University	15	14	12	13
Georgia Tech Research Corp.	8	20	15	16
University of Illinois	14	28	19	17
University of Pennsylvania	13	13	20	15

Comparison of University Software Patenting to Overall Patenting, by Rank for 2002 and 2008. The 2002 Results are from Rai et al. (2009)

SOFTWARE PATENTS

The adjacent figure shows the distribution of the patent classes chosen for inclusion in this study, over all patents granted in those classes. The largest proportion is class 370, Multiplex Communication, including wireless communication algorithms, and the second largest is 707, Data Processing, including algorithms for filtering and analyzing datasets.

All Patents (Academic and Non-Academic) by Class and by Year

The table below lists the patent classifications used in this study, obtained by an examination of subclasses for a mix predominantly containing software.

Patent Codes for Software Patenting

PTO Classification Code	Definition
341	Coded Data Generation or Conversion
345	Computer Graphics Processing
370	Multiplex Communications
706	Data Processing: Artificial Intelligence
707	Data Processing: Database and File Management or Data Structures
708	Electrical Computers: Arithmetic Processing and Calculating
716	Computer-aided Design and Analysis of Circuits and Semiconductor Masks
717	Data Processing: Software Development, Installation, and Management

UNIVERSITY SOFTWARE PATENT TRENDS

The adjacent set of graphs shows software patent counts from 2000 to 2012 for the top 6 patenting universities. The overall number is increasing, but this figure shows that increase is driven primarily by two schools, the University of Wisconsin and MIT. The other school are largely flat in their software patenting behavior over 2000 to 2012, and the University of California shows a decline since 2007.

Software Patent Counts from 2000-2012 for the Top Six Patenting Universities

DOES SOFTWARE PATENTING HINDER REPRODUCIBILITY?

We want to understand whether the increase in academic software patenting has a relationship to the open availability of the software that underlies published results. The idea is that in order for computational results to be reproducible, the code and data must nearly always be made available. If incentives to patent prevent a researcher from releasing the code associated with his or her published results, this is a public policy concern since the code is less widely available than it would have been if released openly (With the exception of code revealed less than a year before the patent was filed. See 35 U.S.C. 102(b)).

Patents can provide disincentives to reveal code in two primary ways. Revealing code you intend to patent may create prior art, reducing the chance that the patent will be granted, and revealing the code openly undercuts the ability to charge licensing fees for use of the software after the patent is granted. In this study we do not measure this relationship directly but contribute preliminary results based on existing literature combined with data analytic methods to show that software patent growth rates in academia appear to be correlated with levels of collaboration with industry but not to related academic publications. We chose a 20% random sample of the patents for each year to be the basis of further data collection and modeling, for a total of 172 patents. This presumes, of course, that the researcher would have made the code available if it was not submitted for patenting. This is not the case. In 2011 we found only about 20% of the code in the articles published in JASA in 2011 was available, but the number of articles with available code is increasing dramatically, from 0 in 1996 (see panel 1). However, we are interested in any occlusion of code through patenting, since this is a subversion of the scientific norms of transparency and reproducibility since after patenting the code becomes available only through a license fee.

We develop the following model:

$$Y_t = \beta_t + Ind.Collab_t + Ac.Pub_t + t + \epsilon_t$$

where $t = 2000 \dots 2012$. Y_t is the number of software patents filed by a university or college on the list of twenty-three used in this study indexed by year, $Ind.Collab_t$ counts the number of patents with an industry research as co-author, and $Ac.Pub_t$ counts the number of patents that reference an academic article published within the 5 years of the filing date, and t is a time effect.

We refine the model by taking into account that 35 U.S.C. 102(b) exempts references from being classified as prior art if they are within one year of the patent application date. We include a third variable indicating if there was a referenced academic publication within a year of the patent filing date, and redefine the academic publication variable to refer to publications referred to in the patent filing within 2 to 5 years of the date of application.

$$Y_t = \beta_t + Ind.Collab_t + Ac.Pub.2to5_t + Ac.Pub.1_t + t + \epsilon_t$$

This model could be augmented with other variables with the collection of more data, such as R&D expenditures or a control variable for originating university. The coefficients for fitting this model with ordinary least squares appear in the table below.

Independent Variable	Coefficient Estimate	Standard Error	p-value
β_t	36.78	9.71	0.0053
$Ind.Collab_t$	7.34	2.64	0.0239
$Ac.Pub.2to5_t$	-3.36	2.61	0.2342
$Ac.Pub.1_t$	0.12	1.79	0.9478
t	4.47	1.10	0.0036

Model Coefficients for Academic Software Patenting Growth

Industry collaboration appears to be significantly correlated with the growth in academic software patents, implying that industry collaboration has a positive effect on likelihood to patent confirming previous literature. There was no effect, or a weakly negative effect, stemming from concurrent academic publications, either those that were 2 to 5 years before filing and those that posed no prior art issues being a year before filing. This is perhaps not surprising as prior publications would either not be made if there was an intent to patent, or they would not be listed on the patent application. This does not provide evidence against there being overlap between published articles and patentable scientific code that supports them.

Further research could extend this model by collecting detailed information from the remaining 659 patents in our dataset. This would provide greater degrees of freedom and permit the inclusion of a school variable, to account for different software patenting practices at different institutions. Other data could be collected and included in the model, such as R&D expenditures on computational research. Information could be collected on publications by university faculty submitting patent applications, and whether or not data and code are available to reproduce those findings. In a larger study this could be compared to similar non-patent-filing faculty members to ascertain whether or not incentives to patent provide a disincentive to reveal academic code openly. Note that there are two possible control groups in this study, similar academics who have published but not sought patents, and the patent applicant themselves. If the applicant releases software regularly for research for which no patent is sought this can provide evidence that patent seeking incentives restrict the sharing of academic software.

SOLUTION? DUAL-LICENSING

We propose a solution of designing patent licensing terms that distinguish between commercial and research applications, and so permitting transparency in research methodologies while maintaining incentives for commercialization and technology transfer provided by the Bayh-Dole Act. The Stanford Natural Language Processing group uses such a dual licensing strategy. Openly licensed code is available for download by researchers and licensing fees exist for commercial applications, since the code is patented. The empirical results presented in this paper support the notion that a dual licensing agreement for patented scientific software can accelerate scientific progress. See e.g. <http://nlp.stanford.edu/software/>

CONCLUSION

We have made efforts to delineate the potential conflict between the scientific method, including reproducible research, and the Bayh-Dole Act that has been brought about by the digitization of science. We have hand curated a new dataset of faculty software patents from 2000 to 2012, and we use it to examine the effects of industry collaboration on academic software patent rates, and the effect of concurrent publications, both those that pose to threat of creating prior art (less than one year from the patent filing date) and those that at two to five years old. In our analysis, industry collaboration drives software patenting among academic, but concurrent publications has no effect or a weakly negative effect.