BIOINFORMATICS EDITORIAL Vol. 22 no. 12 2006, pages 1416–1417
doi:10.1093/bioinformatics/btl167

It is time to end the patenting of software

John Quackenbush[1] and Steven L. Salzberg[2]

[1]Department of Biostatistics, Dana-Farber Cancer Institute, Harvard School of Public Health, Boston, MA 02115, USA. E-mail: johnq@jimmy.harvard.edu. and [2]Center for Bioinformatics and Computational Biology, University of Maryland, College Park, MD 20742, USA. E-mail: salzberg@umd.edu.

One of the most significant outcomes of genomics has been a rapid increase in the rate that we as a community can generate data on interesting biological systems. Rapid improvements in technologies such as DNA microarrays and proteomics applications have produced a climate where the challenge is no longer collecting high quality data but rather managing and analyzing it. As we in the bioinformatics community have addressed this challenge, we have had to carefully consider the way in which the results of our intellectual efforts—the software tools that we develop—are made available to the wider research community. Increasingly, bioinformatics scientists are coming to call for development in an open source environment in which software is distributed with its underlying source code under a license that generally allows broad reuse and redistribution of the code under certain, usually minimal, restrictions.

This model for software distribution has certain distinct advantages that add significant value to tools and techniques that we create. Developing software in a free and open community allows us to build rapidly on the advances of others, rather than having to re-invent or re-engineer software that came before and in doing so greatly accelerates the progress of research both in bioinformatics and in the fields that it touches. Further, making source code available provides an opportunity for methods to be checked and verified in ways that would not be possible if the software or the method it was based upon were proprietary. Admittedly, this model may not work best for all software, but in the research community, which is funded primarily by the public sector, we believe it makes sense for scientists to share their software just as they share their discoveries through publications. Of course, open source software can be copyrighted, which gives the developers of the software proper credit for their creative efforts, much like putting one's name on a research paper gives credit for the results described within it.

However, there is another way to protect software: patents. Starting in 1981 (http://www.bitlaw.com/software-patent/history.html), the United States patent office (USPTO) decided that they would issue patents to software, even though copyright protection was already available. Patents offer much stronger protection than copyright, since they protect not just a software implementation, but also the algorithms and ideas underlying the software. If a program is patented, then others are not permitted to re-implement the algorithms on their own unless they are willing to pay a fee to the patent holder.

In practice, software patents have been a boon for attorneys and the USPTO, which is currently granting around 300 000 new patents per year, but a disaster for scientists and engineers who want to build useful artifacts. They have spurred the development of a mini-industry of shell companies whose primary goal is to file patents on software and then to sue others in the hope of collecting monetary damages. These companies often do not even attempt to commercialize their software, but exist solely to extract money from the productive efforts of others. The availability of patents has also spurred large software companies to file literally hundreds of patent applications, many of which have been granted, on their software. Sometimes these patents are filed to defend their turf, warding off competitors, and other times they are filed defensively, to protect against the vulturous shell companies that might attempt to sue. Regardless of the reason, it seems clear that the efforts to file and defend patents primarily benefit lawyers, predatory special interests and others who are not developing software themselves. Although patents are intended to encourage innovation by disclosing the method used to achieve a particular goal while protecting the inventor's intellectual property, in the software world they act primarily to prevent it. A patent is supposed to instruct 'someone skilled in the art' how to 'practice' a particular invention while claiming rights to the method and use of that invention. In practice, software patents generally disclose very little of practical use and then issue broad, vague claims about the applicability of their method that often goes far beyond what one might reasonably infer they have described. As a result, software patents often create roadblocks to the development of new methods and drain away valuable resources from companies attempting to build new tools.

Consider one example. There has been a lengthy patent dispute between Research in Motion (RIM), the makers of the popular Blackberry hand-held PDAs, and a small company called NTP, whose only noteworthy assets are a series of software patents (Austen and Guernsey, 2005). This dispute recently threatened to shut down Blackberry's enormously popular email service, despite the fact that NTP has no competing service to offer. The basic facts are not in dispute: several years ago, NTP was granted a patent on technology for wireless email services. Around the same time, RIM independently invented similar algorithms, and unlike NTP, RIM created a device and started selling it, eventually becoming a highly successful company. NTP sued, and is now seeking some $1 billion USD in damages. Judges initially ruled in NTP's favor, and RIM appealed to a higher court, while continuing to ask the US patent office to rule that the original patents were invalid. Finally, in March 2006, RIM settled the case (under tremendous pressure from the judge) and paid $612.5 million to NTP. Thus the patent 'blackmail' strategy worked: RIM invested enormous amounts of time and money, finally paying a huge sum to NTP, and Blackberry users are no better off. All of this to allow RIM to provide a

service based on a technology that they clearly developed and implemented.

Now consider a hypothetical example relevant to bioinformatics: imagine that the BLAST algorithm (Altschul *et al*, 1990) had been patented, and further that the patent holders insisted on collecting fees from everyone who wished to use it. The incredibly valuable BLAST servers at NCBI would never have even been built, nor would the hundreds of other sites providing BLAST search services for numerous genome databases. Countless discoveries built on the use of BLAST would have been missed or at best slowed down. BLAST servers all over the world would be shut down or forced to pay fees that would produce no benefit to the scientific community. This scenario is not far-fetched: patent applications have already been filed for numerous sequence alignment algorithms, though fortunately none of them pre-dates the free availability of BLAST. If they did, the patent claims themselves would likely be broad enough that they would prevent BLAST or any other sequence alignment method from being used without a license.

We believe that the practice of issuing patents for software should end. While we realize that we may not be able to make patent offices stop issuing software patents, we can at least discourage members of our own research community from patenting software. One way we can take action is to reject any manuscripts submitted to scientific journals if they describe software that is protected by patents or is subject to a pending patent application. We urge all our scientific colleagues to denounce software patents, as we have, and to embrace the practice of open-source development. Otherwise you may find yourself one day asking a lawyer's permission to run software that you wrote yourself.

REFERENCES

Austen,I. and Guernsey,L. (2005) A payday for patents 'R' Us; huge blackberry settlement is grist for holding company. *New York Times*, May 2, C1.

Altschul,S.F. *et al*. (1990) Basic local alignment search tool. *J. Mol. Biol*., **215**, 403–410.