上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

课程报告

生物医学工程领域开源软件综述

518021910971 裴奕博

目录

摘要

如今，开源软件已经成为了软件开发和共享的一大趋势。本文介绍了开源的概念，常用的几种开源协议，开源代码托管平台。列举了几个常用的生物医学工程领域的开源软件，并对开源的这一现象及其价值进行了讨论。

关键字：开源、开源协议、生物医学工程

1 开源的概念

开源（Open Source），是一种软件设计开发和分发的模式。[1] 在传统的软件发行模式中，开发者会以开放应用接口，编译为动态库等方式，对用户隐藏源代码。而开源软件则不同，开源软件的源代码对所有开发者、用户和其他感兴趣的人员开放，在近年来的软件设计和分发中越来越受到开发者的欢迎。开源并不仅仅意味着可以开放访问的源代码，根据 Open Source Init 拟定的标准，此处将开源软件的理念和特点总结为如下几点 [2]：

1. 免费分发（Free Redistribution）。开源许可不应限制任何一方将软件作为包含来自多个不同来源的程序的聚合软件分发的组件进行销售或赠送。许可证不要求为此类销售支付特许权使用费或其他费用。

2. 开放源代码（Source Code）。该程序必须包含源代码，并且必须允许以源代码和编译形式分发。如果某种形式的产品不附带源代码，则必须有一种广为人知的方式以不超过合理的复制成本获得源代码，最好是通过互联网免费下载。源代码必须是程序员修改程序的首选形式。不允许故意混淆源代码。不允许使用中间形式，例如预处理器或翻译器的输出。

3. 允许衍生产品（Derived Work）。许可必须允许修改和衍生作品，并且必须允许它们按照与原始软件许可相同的条款进行分发。

4. 作者源代码的完整性（Integrity of The Author's Source Code）。仅当许可证允许分发带有源代码的“补丁文件”以在构建时修改程序时，许可证才可以限制源代码以修改的形式分发。许可证必须明确允许分发从修改过的源代码构建的软件。许可证可能要求派生作品带有与原始软件不同的名称或版本号。

5. 不得歧视任何开发者和组织（No Discrimination Against Persons or Groups）。许可证不得歧视任何个人或群体。

6. 不得歧视任何应用领域（No Discrimination Against Fields of Endeavor）。许可证不得限制任何人在特定领域使用该程序。例如，它可能不会限制该程序用于企业或用于基因研究。

7. 许可证的分发（Distribution of License）。程序附带的权利必须适用于得到分发程序的所有人，而无需这些方执行额外的许可。

8. 独立于产品的许可（License Must Not Be Specific to a Product）。程序附带的权利不得取决于程序是否属于特定软件分发的一部分。如果该程序是从该分发中提取并在该程序的许可条款内使用或分发的，则该程序被重新分发的所有各方都应拥有与原始软件分发一起授予的权利相同的权利。

9. 许可不得限制其他软件（License Must Not Restrict Other Software）。许可证不得对与许可软件一起分发的其他软件设置限制。例如，许可证不得坚持基于该软件开发的软件中的所有其他程序必须是开源软件。

10. 许可必须是技术中立的（License Must Be Technology-Neutral）。许可的提供不得以任何单独的技术或界面风格为前提。

2 开源协议介绍

开源协议指的是以上满足开源定义的协议，它们的共同特点是允许软件的免费使用、修改和共享。而根据所需开放程度的不同，提供了不同的开源协议，目前开源协议共有上百种之多。而其中常用的开源协议包括：GPL、LGPL、BSD、MIT、Mozilla 和 Apache。此处列举如下：[2][3][4]

2.1 GPL 协议

GPL 协议（GNU General Public License），全称 GNU 通用公共许可协议。GPL 的开源程度在各种协议中最大。GPL 的出发点是代码的开源/免费使用和引用/修改/衍生代码的开源/免费使用。GPL 协议要求只要软件中包含了遵循 GPL 协议的产品或代码，该软件就必须也遵循 GPL 许可协议，也就是必须开源免费，开放所有源代码，因此这个协议并不适合商用软件。

遵循 GPL 协议的开源软件数量很多，包括我们如今看到的 Linux 的各种发行版、gcc 编译器等均采用了 GPL 协议。

2.2 LGPL 协议

LGPL 协议（GNU Lesser General Public License），全称 GNU 宽通用公共许可证，同样由 GNU 发行并推出，是 GPL 协议的一个衍生版本。由于 GPL 协议对商业软件不友好，GNU 推出了允许商业软件通过类库引用（link）的方式使用 LGPL 类库，而不需要开源商业软件的代码的 LGPL 协议。这使得采用 LGPL 协议的开源代码可以被商业软件作为类库引用并发布和销售。

但是如果修改 LGPL 协议的代码或者衍生品，则所有修改的代码，涉及修改部分的额外代码和衍生的代码都必须采用 LGPL 协议。因此，LGPL 协议适合于作为类库引用，而不适合在此基础上进行修改和衍生的商业软件。

使用 LGPL 协议的软件有：Qt 等

2.3 BSD 协议

与 GPL 相反，BSD 的协议规定比较宽松。在满足以下几个条件时，遵循 BSD 协议的软件可以作为商业软件发布和销售：

1. 如果再发布的软件中包含源代码，则源代码必须继续遵循 BSD 许可协议。

2. 如果再发布的软件中只有二进制程序，则需要在相关文档或版权文件中声明原始代码遵循了 BSD 协议。

3. 不允许用原始软件的名字、作者名字或机构名称进行市场推广。

BSD 协议由于允许使用者修改和重新发布代码，也允许使用或在 BSD 代码上开发商业软件发布和销售，因此是对商业集成很友好的协议。而很多的公司企业在选用开源产品的时候都首选 BSD 协议，因为可以完全控制这些第三方的代码，在必要的时候可以修改或者二次开发。

采用 BSD 协议的开源软甲有：Flask、Flutter、Chromium 等。

2.4 MIT 协议

MIT 协议是目前开源限制最少的协议，只要程序的开发者在修改后的源代码中保留原作者的许可信息即可，因此普遍被商业软件所使用。

使用 MIT 协议的软件有 Python、React、vue、jquery、Node.js 等。

2.5 Mozilla 协议

Mozilla 协议，简称 MPL (Mozilla Public License)。MPL 协议允许免费重发布、免费修改，但要求修改后的代码版权归软件的发起者。这种授权维护了商业软件的利益，它要求基于这种软件的修改无偿贡献版权给该软件。这样，围绕该软件的所有代码的版权都集中在发起开发人的手中。但 MPL 是允许修改，无偿使用的。

2.6 Apache 协议

Apache 和 BSD 类似，都适用于商业软件，由著名的非盈利组织 Apache 基金会提出。Apache 协议在为开发人员提供版权及专利许可的同时，允许用户拥有修改代码及再发布的自由。Apache 协议要求程序开发人员在开发遵循该协议的软件时，要严格遵守下面的四个条件：

1. 该软件及其衍生品必须继续使用 Apache 许可协议。

2. 如果修改了程序源代码，需要在文档中进行声明。

3. 若软件是基于他人的源代码编写而成的，则需要保留原始代码的协议、商标、专利声明及其他原作者声明的内容信息。

4. 如果再发布的软件中有声明文件，则需在此文件中标注 Apache 许可协议及其他许可协议。

使用 Apache 协议的软件有 TypeScript、MongoDB、tensorflow、OpenCV 等。

3 常见的代码托管平台

3.1 GitHub

GitHub 是一个面向开源及私有软件项目的托管平台，因为只支持 Git 作为唯一的版本库格式进行托管，故名 GitHub。是世界上最大的开源代码托管平台，目前属于 MicroSoft 公司。

3.2 GitLab

GitLab 是由 GitLabInc. 开发，使用 MIT 许可证的基于网络的 Git 仓库管理工具，且具有 wiki 和 issue 跟踪功能。使用 Git 作为代码管理工具，并在此基础上搭建起来的 web 服务。其本身基于 MIT 协议。

3.3 Gitee

Gitee 是开源中国（OSChina）推出的基于 Git 的代码托管服务。

4 生物医学工程领域的开源软件

开源软件用途多样，开源协议也规定了开源软件不得限制其应用领域。在生物医学工程领域，开源软件也得到了广泛的应用。生物医学工程领域的开源软件可按照应用领域大致分为如下几类：

4.1 开源 PACS/Dicom Server 软件

4.1.1 Dicoogle

Dicoogle 是一个企业级的开源 PACS 系统，它具有模块化结构，配备适合开发人员的 SDK，允许开发人员构建医疗成像服务器就绪应用程序 [5]。Dicoogle 拥有强大的存档、索引和查询选项，具有高度扩展性。Dicoogle 配备了一组 API 来构建基于云的 DICOM 应用程序。Dicoogle 的文件包括有关 Dicoogle 安装、配置和设置的详细手册，并提供了为 Dicoogle 和 Web/Cloud 应用程序构建插件的全面指南。

Dicoogle 使用 Java 和 JS 开发，可作为 Web 应用运行，Dicoogle 同样基于 GPL 开源协议。

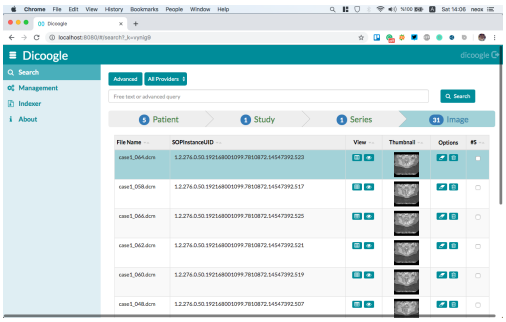

图 1: Dicoogle 界面

4.1.2 Orthanc

Orthanc 是一个开源、模块化、轻量级的 DICOM 服务器项目，由比利时的列日大学医院与 2012 年 9 月发起 [5]。它配备了丰富的 API 和几个插件，支持不同的数据库和 DICOM 查看者。Orthanc 可在 Linux、Windows 等多平台下运行，执行 GPL 开源协议。

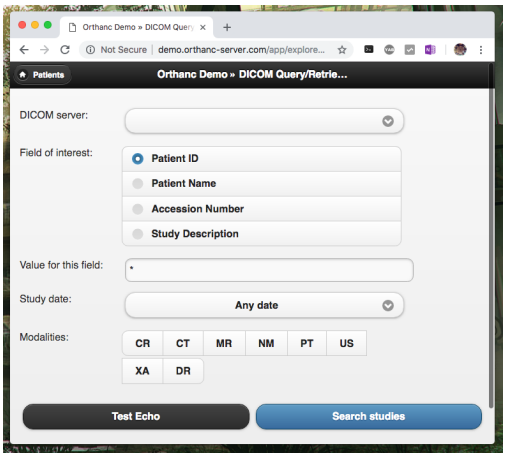

图 2: Orthanc 界面

4.2 开源医学图像处理软件

4.2.1 ITK

ITK（Insight Segmentation and Registration Toolkit）是一个跨平台、开源的应用程序开发框架，广泛用于图像分割和图像配准程序的开发 [6]。ITK 是在 NLM 的资助下开发的。该工具包提供了两个、三个和更多维度的前沿分割和配准算法。ITK 使用 CMake 构建环境来管理配置过程。该软件是用 C++ 实现的，并为 Python 包装。SimpleITK 是 ITK 项目的一个分支，以八种编程语言为 ITK 提供简化的接口，也在积极开发中。

4.2.2 pydicom

pydicom 是一个使用纯 Python 语言的第三方库，它提供了在 Python 环境下读取、修改和写入 Dicom 文件的 API。pydicom 支持单张 Dicom 和序列的读取，并可以很容易的将 Dicom 像素数据转化为 Numpy.ndarray 格式，极大方便了医学图像处理工作者的进一步处理和可视化。同时 pydicom 也支持 tag 解析，和 JPEG 压缩方式的文件写入。

4.3 生物信息学开源软件

4.3.1 EMBOSS

EMBOSS（The European Molecular Biology Open Software Suite），是专为分子生物学研究用户的需要而开发的免费开源软件分析包 [7]。该软件自动处理各种格式的数据，甚至允许从 Web 上透明地检索序列数据。此外，由于提供了广泛的库，这是一个平台，让其他科学家开发和发布软件在真正的开源精神。EMBOSS 还将一系列当前可用的软件包和工具集成到整体中，以便进行序列分析。EMBOSS 运行在 Linux 上，支持序列的信息提取、比对等各种常用功能，同时也提供了 Web 版。EMBOSS 使用的是 GPL 和 LGPL 开源协议。

4.3.2 Bioconductor

Bioconductor 是一款基于 R 语言开发的开源软件，可用于高通量的基因组数据 [8]。Bioconductor 追求的是透明开源、可重复性和高效开发。R 语言在统计中本来就占有压倒性的地位，R 语言支持面向对象、互联网接口、并行计算、可视化、统计模拟和建模等各类适合处理高通量基因组处理的语言特性，这也为 Bioconductor 打下了基础，Bioconductor 建议同时发表数据和每一步执行的命令，而不是只提供算法，这大大提高了可重复性实验的操作难度。此外，Bioconductor 还支持动态的生物注释，可以随着研究的进行更新生物信息的注释。

5 结果与讨论

如今，开源已经成为越来越多软件开发的语言。尤其是对于构建开发“基础设施”的公司来说，其开源软件及其背后的社区的影响力越发明显。近年来越来越多的框架产品由于其封装了底层的特性，极大地方便了开发者，已经越来越多的成为了行业通用的解决方案和趋势。如谷歌的 Chromium、AOSP、TenserFlow、Golang，Facebook 的 PyTorch、React，微软的.NET、Jetbrain 的 Kotlin 等开源语言、框架和底层 SDK 都已经在各自的领域极大地影响了行业的生态。我认为开源软件具有如此强大生命力的原因和优势有如下几点：

1. 免费。作为开源软件最大的特点，其本身即是最大的优势。专业软件动辄几百上千美元/年的价格不仅对于企业不友好，其高昂的价格对个人开发者和学习者来说更是劝退的存在。在能提供同等条件的软件中，免费软件显然能吸引更多人使用，这是毋庸置疑的。

2. 普通开发者的参与度。在传统的商业软件开发模式中，官方开发团队负责软件版本的迭代更新，用户在使用软件的同时可以向软件的提供者提出建议，但却无法真正的影响产品的代码本身，无法参与到整个软件设计和开发的环节中来。而开源软件则不同，我们可以看到 GitHub 上万的 Star 数和上千条的 Issue。任何开发者都可以提出 Issue 对现有的产品进行反馈，也可以通过 pull request 的方式直接提供源代码，直接改进产品的设计。如果被 repo 的拥有者采用，便可以真正的为开源项目贡献一份属于自己的代码。“任何人都可以修改”这一开源的基本原则，大大提升了普通开发者的参与度，使得所有开发者甚至是个人开发者，都可以为开源软件贡献自己的力量，这是传统商业软件开发无法想象的。

3. 活跃的社区。开源的社区和论坛是开源生态得以维持的一大重要原因。软件的使用者可以在此提出自己遇到的问题，由其他社区成员甚至软件的开发团队来解答。传统的闭源软件虽然也有社区，但由于闭源社区无法真正影响软件设计和开发，因此与开源社区的影响力根本无法相提并论。

然而，开源软件作为一种新兴的开发模式，并不是百利而无一害的。我认为，开源软件的劣势有如下几点：

1. 产品的混乱。就拿如今大火的 Python 来说，由于所有人都可以贡献 Python 扩展库，同样的功能可以通过多个库来实现，而这些库许多都彼此不兼容。除了在某些领域成为通用选择的 Numpy、Scipy、Pandas、Sklearn 等之外，在许多应用领域，由于开源社区的开放性，都已经出现了混乱的局面。初学者需要在比对多个库之后才能选择最符合自己需求的扩展库来使用。而这一点在闭源软件上是不存在的，使用者只需要使用官方软件的操作逻辑就可以实现一切，避免了选择的困难，而这样通用的实现在使用者很多的情况下，初学者很容易找到类似的示例，再通过官方的 API 组合来实现，不必担心软件版本和兼容性的问题。

2. 专业性的缺失。由于开源社区中开发者水平的参差不齐，开源软件的专业性也难以得到保证，因此通常只有最核心的“官方”的代码可以得到大范围的使用，这也是开源软件中的框架类项目如此受到人们欢迎的原因。开发者只需要使用官方的代码即可，只要不使用被非官方修改的框架，就可以不用担心专业性的问题。此外，许多核心的功能和算法都是作为商业机密存在，因此我们可能感受到，闭源的软件在许多体验上仍然胜过开源软件。如 JetBrains 公司各种 IDE 的智能搜索和代码补全，仍然是我在开源的 VSCode、Eclipse 等编辑器和 IDE 上无法感受到的。

3. 安全性的缺失。任何人都能修改的代码也会带来一些问题，如代码的安全性问题。如果开源软件被有人恶意地利用，向其中插入恶意病毒、钓鱼的代码，会产生十分严重的后果。而在闭源软件中，由于所有软件版本的更新都由官方的开发小组控制，因此安全性可以得到很好的保证。

综上，我们可以看到开源软件的巨大优势，但也不可忽视其劣势。因此我们在使用相关软件时，仍需要根据自己的需求和偏好来选择。我们就拿科学计算领域的软件作对比，在这一领域的传统软件代表自然是大名鼎鼎的 Matlab，而近年大火的 Python 和 R 语言等开源软件，俨然有与之分庭抗礼的趋势。如果你更喜欢用更简洁的方式和 API 来解决问题而不想“折腾”的话，那么 Matlab 显然是更适合你的选择，Matlab 有更多封装好的底层算法来供你选择。然而如果你需要更好的扩展性，需要更多

自定义的功能，甚至嵌入其他 Web 或桌面端软件来执行时，那么 Python 这样根正苗红的编程语言是更好的选择。只要你花时间，你几乎可以在 Python 的开源社区中找到任何你需要的功能。你可以使用 Python 搭建网页（Django、Flask），进行科学计算（Numpy、Scipy），进行数据挖掘和分析（Pandas，Matplotlib），执行机器学习和深度学习（Sklearn、PyTorch、TensorFlow）……作为一个通用编程语言的开源社区，Python 提供了你想要的几乎所有功能的扩展，而这也是近年来 Python 这一新兴语言大火的重要原因吧。

如今，开源显然已经成为软件开发的大趋势。我相信在将来，会有越来越多优秀的开源项目和软件推出。但对于“开源”这一问题本身，我们仍然要辩证看待，根据自己的需要和喜好选择最适合自己的软件产品。

参考文献

[1] Wikipedia contributors. Open source — Wikipedia, the free encyclopedia, 2021. [Online; accessed 27-September-2021].

[2] Digital Ocean. Open source initiative. https://opensource.org, 2021.

[3] Runoob. 各种开源协议介绍. https://www.runoob.com/w3cnote/open-source-license.html, 2021.

[4] 机智玛莉. 关于开源协议，你知多少. https://zhuanlan.zhihu.com/p/78998314, 2019.

[5] Hamza Musa. Top 10 free open source pacs/ dicom server projects. https://medevel.com/10-open-source-pacs-dicom/, 2019.

[6] Wikipedia contributors. Insight segmentation and registration toolkit — Wikipedia, the free encyclopedia, 2021. [Online; accessed 27-September-2021].

[7] I. Rice, P. Longden and A. Bleasby. Emboss: The european molecular biology open software suite. http://emboss.sourceforge.net/, 2000.

[8] Bioconductor. Bioconductor. http://www.bioconductor.org/, 2021.