J. Plasma Fusion Res. Vol.84, No.1 (2008) 31-32

講座

オープンソースソフトウェアを使った実践データ解析

Practical Data Analysis Using Open Source Software

1．はじめに

鈴木康浩，稲垣　滋[1)]
核融合科学研究所，[1)]九州大学応用力学研究所

（原稿受付：2007年11月16日）

「オープンソース」という言葉を聞いたことがありますか？オープンソースとは英語表記で「Open Source」となり固有名詞です．オープンという言葉のニュアンスから，公開されたソフトウェアを意味するであろうことは容易に想像できると思います．Wikipedia によると[1]，オープンソースとは「ソースコードを公開して，プログラムを自由に使用・修正・配布できるようにする」という「考え方」を意味します[2]．オープンソースを名乗るには単にソースコードを公開するだけではなく，厳密に定められた定義[1]を満たす必要があります．全文はリンク先を参照してください．

オープンソースである以上，ソースコードを公開するのは当然ですが，特に重要なことはソフトウェアの変更と再頒布を認めなければならないことです．これはソフトウェア開発者にとって重要な問題です．動機が趣味であれ業務であれ，開発者はコストを費やしてソフトウェアを開発します．オープンソフトを名乗るにはソースコードを公開するだけでなく，第3者によるソフトの改変と改変者による頒布を認めなければならないのです！こう書くと，オープンソースはソフトウェア開発者にメリットはないように思えます．では，なぜソフトウェア開発者はオープンソフトを選択するのでしょうか？開発者にとって一番重要なことはソフトウェアの著作権です．せっかく開発したプログラムを公開したばかりに，第3者に横取りされてしまえば，ソフトウェア開発者の苦労が報われません．このようなことを防ぐ仕組みがソフトウェアのライセンスです．オープンソースはソースコードを公開することを義務づけていますが，同時に改変，再頒布する場合はオリジナルのソースコードとソースコードに対するパッチファイルとして提供しなければならないことを定めています．つまり，改変点をオリジナルに対する差分として提供することで，オリジナル開発者の完全性（integrity）を保証するのです．再頒布されたプログラムは元々のソースコードど差分ファイルを組み合わせることで，初めて改変されたプログラムを動かすことができるのです．再頒布されたソースコードには改変者の著作権も記されますが，元々の著作権を変更することは認められません．このようなライセンスを採用することにより，ソフトウェア開発者の著作権を保護しつつ，ソースコードを広く公開することができるわけです．

一方，オープンソースと対になるライセンスの概念としてプロプライエタリ・ソフトウェアがあります．プロプライエタリ・ソフトウェアとはパッケージ形式で提供される商用ソフトウェアと同義で使われることが多いですが，厳密にはソフトウェアの改変・再頒布を認めないライセンス形態を指します．第3者による改変・再頒布を防ぐために多くの場合，ソースコードは頒布されず，実行形式のバイナリファイルで提供されます．当然，実行バイナリに対するリーバスエンジニアリング[3]も禁止されています．マイクロソフトをはじめとする，ソフトウェアメーカーの多くは，プロプライエタリとしてソフトウェアを提供しています．ソフトウェアメーカーは多くの予算と人材を費やしてソフトウェアを開発します．開発に費やしたコストは価格に反映させ販売し，数年かけて回収します．また，プロプライエタリ・ソフトウェアの多くは商用のソフトウェアで

1　http://en.wikipedia.org/wiki/Opensource
2　本講座で紹介するソフトウェアのすべては，厳密にはフリーソフトウェアです．ですが，オープンソースはフリーソフトウェアを含んだ概念ですので，本講座ではオープンソースで統一します．
3　ソフトウェアの動作を解析して，ソースコードを書き起こすこと．

Practical Data Analysis Using Open Source Software 1. Introduction
SUZUKI Yasuhiro and INAGAKI Shigeru
author's e-mail: ysuzuki@LHD.nifs.ac.jp

©2008 The Japan Society of Plasma Science and Nuclear Fusion Research

あるために数年にわたる保証期間とサポートのコストも同時に含んでいます[4].

では，なぜデータ解析にオープンソースを利用するのでしょうか．オープンソースのソフトウェアを使ってデータ解析をしなくても，製品として発売されているグラフソフトや解析ツールを使ってもデータ解析はできます．市販されているグラフソフトは高度なデータ解析機能を持つ場合がほとんどです．「プロプライエタリ・ソフトウェアを廃してオープンソースのみを使え！」というような思想のもとに，オープンソースの使用を強制しているわけでもありません．

オープンソースを採用する理由は，誰でも自由に使えるからです．また，用途に応じて自由に改良できるからです．このことは，例えば共同研究等で重要な意味を持ちます．次のような状況を考えます．A さんは，B 社が開発したソフトウェアを購入して，すべてのデータ解析を行っています．A さんの共同研究者が，自分の所属先で A さんと同じ作業をするには，A さんと同じ環境を整える必要があります．つまり，B 社のソフトウェアを購入する必要があります[5]．ですが，はじめからオープンソースソフトウェアを用いてデータ解析環境を構築しておけば，常に最新の環境をすべての共同研究者同士が共有できるのです．本講座で紹介する，オープンソースのスクリプト言語や解析ツールのほとんどは，様々なOS環境に移植されています．そのようなスクリプト言語で解析環境を構築すれば，日本国内に限らず世界中で環境を構築できます．オープンソースソフトウェアは，無料で公開され自由に使える代わりに，メーカー製品のような充実したサポートはありません．また，簡単なマニュアルが整備されているわけではないので，プログラム言語を習得するような努力が必要です．しかし，そのような負担があっても，例えば，大学の研究室のような規模と予算が限られている環境では，オープンソースを採用するメリットは大きいと思います．大学・大学院は学生を教育して人材を生み出す必要があります．企業は大学等とは異なり，収益を生み出すために研究活動をするわけですから，専門分野をそのまま生かすことは難しい状況です．ですが，たとえ専門分野が異なっても，データ解析のノウハウを自分自身で構築できる力をもった学生を育成できれば，社会に役立つ人材を生み出すことにつながるのではないでしょうか．

次に，オープンソースの考え方が，サイエンス的な物の考え方と相通じるものがあるからです．サイエンスの世界では，研究者の成果は論文として公表されます．別の研究者が，論文を引用しつつ，さらなるアイデアを加えることによって研究を発展させます．このとき，引用元の成果のオリジナリティは保証されています．つまり，サイエンス的な物の考え方は，結果を公表することで先駆性とオリジナリティを保証し，他の研究者は成果を自由に研究し引用することで，サイエンス全体が発展します．サイエンスの世界では，分野が異なるにもかかわらず同じ解析手法が通じる場合が多数あります．例えば，フーリエ変換によってスペクトル解析をすると，物事の見通しが良くなることは分野を問わずよく知られていることです．ということは，オープンソースソフトウェアを使ってデータ解析を行おうとした場合，他分野で同じソフトウェアを使って解析している成果を利用することができます．一方，逆もまた真で，我々のコミュニティの成果が，他の分野で使われることがあるでしょう．従って，我々のこのような活動は，プラズマ・核融合コミュニティの成果になるだけでなく，計算科学，サイエンス全体への貢献になります．このような活動は，研究者だけでなく，大学・大学院で学ぶ学生への教育効果ももたらすでしょう．

この講座では，筆者らが開発している環境を前提に，オープンソースソフトウェアの使用例と使い方を紹介します．第 2 章では，本格的にスクリプト言語や解析ツールを紹介する前に，便利なツールを使った小技を紹介します．第 3 章では，スクリプト言語 Python を使った，データ処理やグラフ作成を紹介します．第 4 章では，スクリプト言語 Ruby を使った，データ処理を紹介します．Ruby は，日本発のスクリプト言語として注目されているプログラミング環境です．第 5 章と第 6 章は，外部から著者を招きました．第 5 章では，地球流体分野で活用されている Ruby を使ったツールの紹介と活動の経緯を紹介していただきます．この分野での活動を知ったことが，今回の講座を立ち上げた経緯でもあります．第 6 章では，Octave というソフトウェアを使ったデータ解析を紹介していただきます．第 7 章は，古式ゆかしい Fortran を使ったデータ解析例を紹介します．オープンソースで開発されている Fortran コンパイラの使い方を説明し，さらに Python といったスクリプト言語と組み合わせる使い方を紹介します．この講座で紹介したツールやスクリプトは原則，のちに立ち上げるホームページ上でダウンロードできるようにしたいと考えています．ご期待ください．

参 考 文 献

［1］The Open Source Definition (http://www.opensource.org/docs/osd)

4　ソフトウェアメーカにとって品質の保証とその他の効率を考えると，ソフトウェアの改変と再頒布を禁止することは選択肢の一つとして認められるべきだと思われます．

5　もちろん，作業は A さんのもとを訪れて行うか，ネットワーク経由で A さんの環境を利用するとか，何とかソフトウェアを購入しなくても同じ環境を構築することはできますが．