

Ruby Association Grant 2014 Final report

Author: Naoki Nishida

Contents

1. Overview
2. Results
 - Mikon
 - IRuby
 - Nyaplot
3. Future work
4. Conclusion

Summary

- New DataFrame gem Mikon has become available.
- Bidirectional communication between IRuby kernel and JavaScript front-end has been implemented.
- Plotting library Nyaplot has been totally redesigned.

Overview

Rubyはその黎明期から多くの科学者・研究者に利用されてきたが、近年 R や Python が築いたような統合環境を中心とした科学用途のライブラリのエコシステムを構築できていない。私が昨年制作したプロットライブラリの Nyaplot は、その一部として簡単なデータフレームを実装し、統合環境として Web 上での Ruby 実行環境 IRuby を利用することでその問題の一部を解決した。

Many scientists and researchers have used Ruby for their research and education from the beginning of the language but these days it seems that other languages like Python and R are defeating Ruby in scientific field. As a result an eco-system of scientific libraries like one of R with its core environment RStudio cannot be seen in Ruby world. IRuby, a Web-based Ruby environment and a plotting library named Nyaplot resolved a part of this issue last year.

しかしそのデータフレームは解析用のメソッドをほとんど備えておらず、実際の計算部分は JavaScript に大きく依存するものであった。これによって Nyaplot の自由度は大きく落ちた。例えば Statsample で解析した結果をデータフレームに変換してから Nyaplot のメソッドに渡す必要があること、また新しいグラフを実装する際に JavaScript を書く必要がある他、genome browser のようにフロントエンドの GUI から解析を実行するようなインターフェイスを実装できないことなどが主要な問題点として挙げられた。

However Nyaplot's dependency on its back-end library Nyaplotjs dramatically reduced its flexibility and extendibility. Its DataFrame do not have methods for data analyzing enough and most code for generating plots was written in JavaScript. Because of that users had to write some JavaScript code to create plots which were not bundled with pure Nyaplot, and interactive GUI like genome browser cannot be implemented on Nyaplot.

そこで本プロジェクトでは新しく Ruby 製のデータフレームライブラリ Mikon を作り,解析用のメソッドを実装すること,そしてその成果を用い IRuby 上に対話的な GUI を実装することを目標として設定した.

Therefore I created a new DataFrame library named Mikon and set the goal of this project to implement methods for data analyzing and to construct a new interactive GUI like web-based genome browsers on it.

Results

Mikon

Mikon は既存の Nyaplot::DataFrame と API に互換性を持たせた Ruby 製データフレームをプロジェクトである.データストレージとして Array の代わりに多次元配列 gem の NMatrix を用いることで高速かつ安定したデータ処理を見込んでいる. Nyplot::DataFrame からは一部のコードを流用したものの,メモリ上でのデータの持ち方を変更したためほぼフルクラッチでの開始となった.

At the beginning of this project, I separated DataFrame from Nyaplot and released it as a new gem named Mikon. After that I replaced its storage from pure Array to NMatrix, a fast linear algebra library developed by SciRuby community.

コンストラクタや基本的なアクセサを実装した後に select や map など,DSL を使いデータを操作するメソッドを複数追加した.例えば以下のコードではデータフレームの中で year 列に 2012 という値が入っている行のみを選択し,新しいデータフレームを作る.

Moreover I implemented methods for data manipulating like DataFrame#select and #map. These methods take block as one of their arguments and new DSL is available in it as:

```
require 'mikon'
df1 = Mikon::DataFrame.new(year: [1994, 1994, 2000, 2000, 2012, 2012], value:
[10, 3, 4, 19, 8, 12])
df1.select{year==2012} #-> [{year: 2012, value: 8}, {year: 2012, value: 12}]
```

また,統計処理ライブラリ Statsample が持つデータ構造との互換性を得るため,基本的な統計処理機能を持つメソッドを実装した.

これによって Statsample の持つ豊富な解析用のメソッドが Mikon を介して利用できるようになった.例えば以下のように CSV ファイルからデータを読み込んだ後,Statsample の持つメソッドに Mikon のデータ構造を渡すことができる.

At the second quarter of grand period, methods for statistics compatible with data structure of Statsample have become available in DataFrame and Series. These methods enabled Statsample to accept Mikon as arguments of its methods as follows:

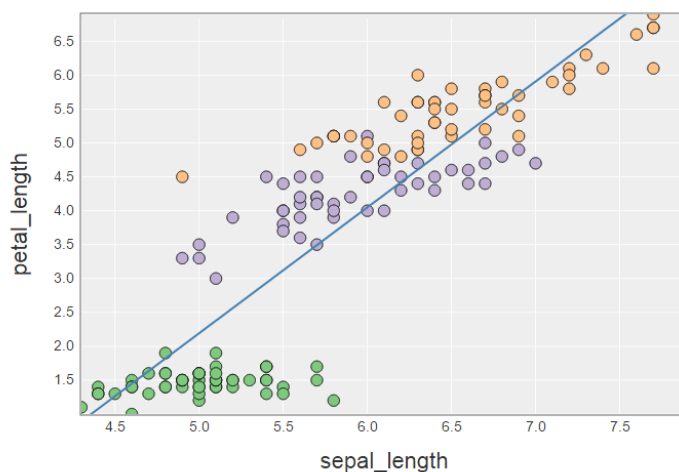
```
df = Mikon::DataFrame.from_csv(path_to_iris_dataset)
lr = Statsample::Regression.simple(df[:sepal_length], df[:petal_length])
```

他には Nyaplot を用い,データフレームから直接プロットができるメソッドとして DataFrame#plot を実装した.以上に関してデモ用の IRuby notebook を公開した.¹

Some methods for plotting are also implemented. An IRuby notebook using features shown above was published to web.

```
In [14]: x = (df[:sepal_length].min.round..df[:sepal_length].max.round).to_a
         y = x.map{|v| b*v+a}
         plot.add(:line, x, y)
         plot
```

Out[14]:



また,NMatrix のインストールが煩雑であることから,NMatrix への依存をなくした簡易版を助成金期間中にリリースすることを予定していた.

¹ <http://nbviewer.ipython.org/github/domitry/Iris/blob/master/Iris.ipynb>

これに関しては本年2月に NMatrix の β リリースが始まり,今後インストール環境の整備が見込まれることから中止とした.

The installation method of NMatrix was too complicated and it was planned to release simple version of Mikon that do not depend on NMatrix in the proposal. However the plan was canceled because the beta version of NMatrix was released this February and the installation method will be simplified before next releasing.

IRuby

IRuby²は SciRuby プロジェクトによりメンテナンスが行われている IPython の他言語カーネルの Ruby 版である. IRuby を用いることで対話的なデータの加工・可視化ができる他, nbviewer³などの web サービスを利用することでその結果をすぐに共有することができる. Nyaplot・Mikon は共にこの環境で実行されることを前提に開発されている.

IRuby is one of other language kernels of IPython, a web-based interactive programming language environment. Both Nyaplot and Mikon are designed to work on IRuby notebook.

プロジェクトの目標である対話的な GUI の実装には,フロントエンドで動く JavaScript と Ruby カーネルでの双方向通信が必要である.そのため IPython では双方向通信用のインターフェイスの Comm が提供されているが,それを Ruby カーネルから利用することがこれまでできなかった.

Bidirectional communication between IRuby kernel and IPython front-end is necessary to achieve the goal of this projects. Therefore in the midterm of the grant term, I implemented the wrapper for Comm, the bidirectional communication module supported by IPython. The Wrapper for html-based interface named Widget was also implemented. Real-time simulator of double-pendulum was published as a demo application of Widget and Comm. These patches are merged to the master branch of IRuby.

そこで助成金の期間中に Comm のラップ,そしてそれを用いて実装される Widgets を IRuby でサポートするパッチを作成した.その後このパッチのマージをメンテナに要請し,この変更は本家リポジトリにマージされた.その際にデモとして二重振り子のシミュレーション

² <https://github.com/minad/iruby>

³ <http://nbviewer.ipython.org/>

ヨンの可視化部分を Widgets で実装し,公開した⁴.

```
In [4]: sim.save_movie(3000)
```

Out[4]:



課題として IPython の新しいバージョンのサポートが挙げられる.新しく実装した機能の内,Commはすでに安定しているが Widgets は新しいバージョンで大きな変更が加えられており,IPython3.0.0dev を用いた場合今回の実装では動作しないことが確認されている.2015年2月にβリリースが始まり,そのサポートは喫緊の課題である.

However, there is one known issue about IRuby and its new module Widget. The design of Widget is still not stable and the API of Widgets are dramatically changed in new version of IPython, IPython3.0.0dev. It was already confirmed that `IRuby::Widget` and IRuby kernel itself do not work on new IPython, so IRuby should be renewed as soon as possible.

Nyaplot&Nyaplotjs

Nyaplot は Google Summer of Code 2014 を利用し作成された新しいグラフ作成ライブラリである.Mikon はこのソフトウェアの一部である `Nyaplot::DataFrame` から派生したものである.

Nyaplot is a new plotting library developed during Google Summer of Code 2014 term. Mikon is a folk gem of its sub-module `Nyaplot::DataFrame`.

⁴ https://github.com/domitry/double_pendulum/

プロジェクトのゴールである対話的な GUI の実装のためには Ruby 側で計算をした結果を JavaScript 側に渡すことでフロントエンドの機能をほぼ描画のみに限る必要があるが、以前の設計ではそれが不可能であった。本プロジェクトの期間中にはフロントエンド・バックエンド共に大幅な設計変更を施し、Ruby 側でほとんどの計算を行わせることにした。Nyaplotjs は設計方針の変更により Mikon と同様ほぼフルスクラッチでの開始となった。

To implement interactive GUI and achieve the goal, it is necessary to move the code from Nyaplotjs to Nyaplot. During the latter half of the grant term, I redesigned Nyaplot and Nyaplotjs and transport JavaScript code to Ruby.

まずはバックエンドである Nyaplotjs のグラフオブジェクト(Nyaplot::Diagram)を廃止し、代わりに矩形や線分などの基本図形を実装した。これにより、新しい図を定義する際 Rect や LineSegment など既存の図形の組み合わせで書ける図であれば Ruby で完結することが可能になった。

また、合わせて箱ひげ図(Glyph::Box)や棒グラフ(Bar::Glyph)など既存のグラフを基本図形の集合として実装しなおした。

At first Nyaplot::Diagram, a root-class of Glyph was duplicated and new basic glyphs as Rectangle and Line Segment are implemented. New glyphs enable to implement more complex charts like Box plot (Glyph::Box) and Bar chart (Glyph::Bar) as a set of basic glyphs.

次に、IRuby 上で対話的な操作を行う機能を Interactive module として実装した。この機能は IRuby のトップレベルで include Interactive とすることで利用できる。

以下の例で add_filter に渡したブロックはブラウザ上の図形がクリックされた際に呼び出され、その結果は EmptyPlot#update を介してフロントエンド側に反映される。

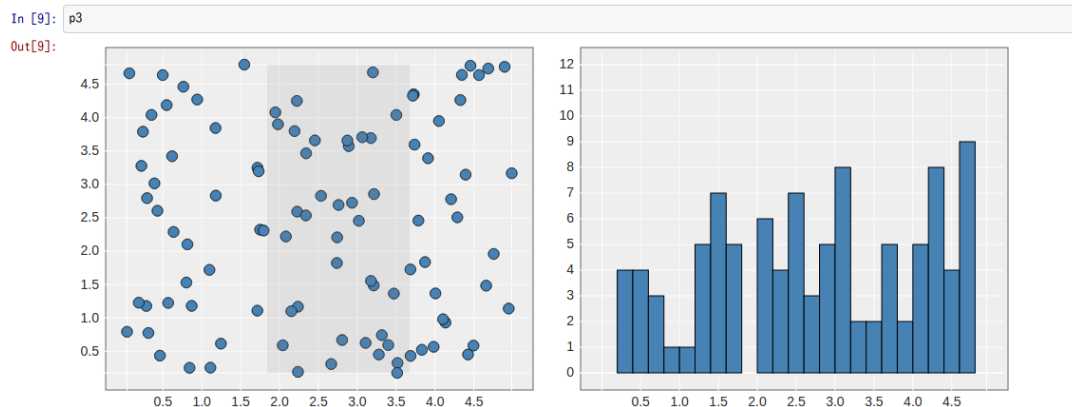
Next module named Interactive was implemented to introduce interactivity between front-end and IRuby kernel.

New GUI can be used by running the command include Interactive on IRuby notebook. In the example code as follows the block passed to Plot#add_filter will be called when Histogram is hovered on the browser, and the front-end was updated by EmptyPlot#add_filter. This feature is demonstrated on a movie ⁵and notebook shown as

⁵ <http://youtu.be/iNKaYDzMOpE>

follows:

```
p1 = Plot.from(df).add(:scatter, x: :x, y: :y)
p2 = Plot.from(df).add(:histogram, value: :z)
p3 = columns(p1, p2)
p1.add_filter {|range|
  new_df = df.select{x >= range.min && x <= range.max}
  p2.glyph.data(new_df)
  EmptyPlot.new(p3.pane, p2.stage, p2.glyph).update
}
```



以上のデモとして動画とノートブック⁶を用意した.

デモでは非常に簡単な例しか扱っていないが,これで Epivizr のようなゲノムブラウザを実装することは可能になった.ただしこの際に足りない図形やコントローラを JavaScript で加える必要があるため,今後汎用的なものに限って実装を進めていく予定である.

The demo code is very simple, but more complex interface like Epivizr ⁷ can be implemented as a set of basic glyphs and other components. However more glyphs and components written in JavaScript should be bundled with Nyaplot.

Future Work

当プロジェクトの今後の課題としてまず Mikon のデータ操作の機能が貧弱であることが

⁶ <http://youtu.be/iNKaYDzMOpE>

⁷ <http://bioconductor.org/packages/release/bioc/html/epivizr.html>

挙げられる。データを読み込んだ後そのまま可視化する場合、また genome browser のように標準化されたフォーマットを持つファイルを読み込んで可視化する場合には問題ないが、様々なフォーマットのファイルを手動で解析していく場合には問題となる。例えば N/A を含むようなデータの扱いには難があり、さらなる機能拡充が必要である。

One of future issues of this project is to implement more data manipulation methods to Mikon. Even now Mikon has enough power to visualize standardized data as genome browser, but it is not good at handling "wild" data like one have many N/A values in it. This is a serious problem as a DataFrame gem, and should be resolved as soon as possible.

次に, Nyaplot の大幅な設計変更による影響の克服が課題として挙げられる。この設計変更により Nyaplot の柔軟性は大幅に高まったが、代わりに以前のバージョンとの互換性をほぼ失ったため、ツールチップなどこれまでサポートしていた多くの機能が利用不可能になった。また地図用の Mapnya や 3D グラフ用の Nyaplot3D などの既存の拡張モジュールも新しいバージョンとは互換性がない。これらは新しい設計に基づいて再度実装する必要があり、正式版のリリースまでにはまだ時間がかかることが予想される。

To resolve bad effects caused by whole redesigning of Nyaplot is another issue. The redesigning made Nyaplot flexible and extendable, but it also destroyed the compatibility with older version of Nyaplot. Therefore many important features like zooming and small tooltip are disabled in new version of Nyaplot. (v0.2.0.rc1) In addition three extensions Mapnya, Bionya and Nyaplot3D cannot be used in this version. I have to implement these features according to new design of Nyaplot, so official release of Nyaplot v0.2.x seems to be delayed.

Conclusion

上記のようにプロジェクト開始当初に掲げた、解析と可視化をスムーズにつなげるデータフレームの作成とその周辺環境の整備という目標はほぼ達成できたと評価する。ただしそれぞれのライブラリの完成度は未だ十分ではない。これを踏まえ、当プロジェクトでは今後も機能拡充を図っていく予定である。

I conclude that this project had achieved the goal which I set before the grant period started. Now the new dataframe library Mikon help gems for visualization and analysis to work with each other. It also enabled Nyaplot to behave as a simple tool for analysis on IRuby. However, all libraries are still uncompleted and this project will continue till they get enough features and satiability.