

BioRuby のインストール

Ruby と BioRuby は Linux, Solaris, IRIX, Mac OS X, Windows XP など一般的な OS で簡単にインストールでき、自由に利用することが出来ます。

1. 前準備

Ruby 本体のインストール

BioRuby は Ruby 1.6.8 でも動作しますが、KEGG API などを利用する場合に、多数のライブラリの追加インストールが必要になるので、Ruby 1.8.1 以降の利用をお勧めします。Ruby に関する情報は <http://www.ruby-lang.org/> から取得できます。

```
% wget ftp://ftp.ruby-lang.org/pub/ruby/ruby-1.8.1.tar.gz
% tar zxvf ruby-1.8.1.tar.gz
% cd ruby-1.8.1
% ./configure
% make
% su
# make install
```

Windowsの場合はまずCygwinをインストールし、その上でRubyをインストールすると環境が整ってよいでしょう。簡単に済ませる場合には、何種類かバイナリインストール用のパッケージが作成されているので、これらを利用する方法もあります。Mac OS Xの場合は、Ruby 1.6.8が最初からインストールされていますが、拡張ライブラリなどに必要となるSDKをOS付属のディスクから追加インストールしておくべきでしょう。Linuxなどでは、多くのディストリビューションでRubyのパッケージが利用可能でしょう。いずれの場合も、上記の方法でRuby 1.8.1以降を追加インストールできるはずですが。

2. BioRuby の入手

リリース版は<http://bioruby.org/archive/>からダウンロードしてください。最新版のソースはCVSから入手できます。ウェブブラウザでViewCVSを使って<http://cvs.bioruby.org/>で眺めることもできます。

リリース版のダウンロード

ウェブブラウザで一番数字の新しいファイル(bioruby-x.x.x.tar.gz)をクリックするか、ファイル名を確認してコマンドラインでwgetなどのツールを使ってダウンロードします。

```
% wget http://bioruby.org/archive/bioruby-x.x.x.tar.gz
```

CVSによる最新版のダウンロード

匿名CVSの公開レポジトリはcvs.open-bio.orgにあります。

```
% cvs -d :pserver:cvs@cvs.open-bio.org:/home/repository/bioruby login
(Logging in to cvs@cvs.open-bio.org)
CVS password: cvs (パスワードは cvs です)
% cvs -d :pserver:cvs@cvs.open-bio.org:/home/repository/bioruby checkout bioruby
```

checkoutすると、カレントディレクトリに**bioruby**フォルダが作成されます。次回からは、この**bioruby**フォルダ内に移動して**cvs update**を実行するだけで最新の内容に更新されます。

```
% cd bioruby
% cvs update
```

なお、匿名CVSのため、登録された開発者以外の方は修正をコミットする事ができません。バグ修正や機能拡張を行われた場合、ぜひメーリングリストなどで報告してください。

3. BioRuby のインストール

基本的には

```
% tar xvfz bioruby-x.x.x.tar.gz
% cd bioruby-x.x.x
% ruby install.rb config
% ruby install.rb setup
% su
# ruby install.rb install
```

これだけです(CVSを使う場合**bioruby-x.x.x**の部分**bioruby**と読み替えてください)。

ホームディレクトリなど、標準でないディレクトリにインストールしたい場合、**install.rb**の機能を使って、**config**の段階で

```
% ruby install.rb config --prefix=/your/homedir/
```

のように**--prefix**でインストール先のディレクトリを指定します。この場合、**BioRuby**を使う際に、環境変数**RUBYLIB**にもインストール先のディレクトリを設定しておく必要があります。

```
% export RUBYLIB=/your/homedir/lib/ruby/site_ruby/1.8/ (bash, zshの場合)
% setenv RUBYLIB /your/homedir/lib/ruby/site_ruby/1.8/ (csh, tcshの場合)
```

また、OBDA <http://obda.open-bio.org>のBioRegistry機能を使えるようにするには、シス

テム管理者の場合、

```
# mkdir /etc/bioinformatics/  
# cp etc/bioinformatics/seqdatabase.ini /etc/bioinformatics
```

または一般ユーザでホームディレクトリに置く場合、

```
% mkdir ~/.bioinformatics  
% cp etc/bioinformatics/seqdatabase.ini ~/.bioinformatics
```

として設定ファイルをコピーし、必要に応じてseqdatabase.iniの中身を編集します。OBDAはデータベースエントリを指定した方法で簡単に取得できるようにする仕組みで、データベースをBioPerl, BioJava, Biopythonなど他のプロジェクトと共通に利用することができます。

4. 追加情報

追加インストールすることで機能が有効になるもの

BioRubyで利用可能なコマンドやライブラリを追加インストールすると、それらを使った機能が有効になります。

- アプリケーション類
 - **BLAST** - ローカルでBLAST検索を実行
 - ◇ <ftp://ftp.ncbi.nih.gov/blast/executables/>
 - **FASTA** - ローカルでFASTA, SSEARCHなどの検索を実行
 - ◇ <ftp://ftp.virginia.edu/pub/fasta/>
 - **HMMER** - ローカルでhmmsearchやhmmpfamを実行
 - ◇ <http://hmmer.wustl.edu/>
 - **MySQL**や**PostgreSQL** - OBDAのBioSQLサーバを構築
 - ◇ <http://www.mysql.com/>, <http://www.postgresql.org/>
- Rubyのライブラリ (<http://raa.ruby-lang.org/>で取得可能)
 - **RAA:xmlparser** - BLASTのXML出力解析を**RAA:rexml**より高速に
 - **RAA:ruby-dbi**(DBI)と**RAA:mysql-ruby**, **RAA:postgres**, **RAA:oracle**いずれか(DBD) - OBDAのBioSQLを利用する場合に使用
 - **RAA:bdb** - OBDAのBioFlatでBerkeley DBも選択可能に

Ruby 1.6系をお使いの方は、多くのライブラリが含まれている**RAA:ruby-sumo**と、追加で**RAA:soap4r**をインストールされる事をお勧めします(この場合、SOAP4Rに**RAA:date2**, **RAA:http-access2**も必要)。

Ruby のヘルプコマンド

Rubyにはriというヘルプコマンドがついてきますが、残念ながら準備をしないと使うことが出来ない上に、Ruby 1.8.1付属のものはきちんと動作しません。しかし、あると非常に便利なので現状の最新版であるRuby 1.8.2-preview2を利用してインストールしてみます。

```
% wget ftp://ftp.ruby-lang.org/pub/ruby/1.8/ruby-1.8.2-preview2.tar.gz
% tar zxvf ruby-1.8.2-preview2.tar.gz
% cd ruby-1.8.2/
% RUBYLIB=./lib rdoc --ri
```

少し時間がかかりますが、これによりヘルプがホームディレクトリの~/rdoc/に自動生成されます。--riオプションの代わりにrootになって--ri-siteオプションを使用すればシステムワイドにインストールされます。

使い方は、例えば配列クラスArrayのヘルプを参照したい場合、

```
% ri Array (Arrayクラスの概要とメソッドの一覧などが表示される)
----- Class: Array
Arrays are ordered, integer-indexed collections of any object.
Array indexing starts at 0, as in C or Java. A negative index is
assumed to be relative to the end of the array--that is, an index
of -1 indicates the last element of the array, -2 is the next to
last element in the array, and so on.
-----

Includes:
-----
Enumerable(all?, any?, collect, detect, each with index, entries,
find, find_all, grep, include?, inject, map, max, member?, min,
partition, reject, select, sort, sort by, to a, to set, zip)

Class methods:
-----
[], new

Instance methods:
-----
&, *, +, -, <<, <=>, ==, [], []=, abbrev, assoc, at, clear,
collect, collect!, compact, compact!, concat, delete, delete at,
delete if, each, each index, empty?, eql?, fetch, fill, first,
flatten, flatten!, frozen?, hash, include?, index, indexes,
indices, initialize copy, insert, inspect, join, last, length, map,
map!, nitems, pack, pop, push, rassoc, reject, reject!, replace,
reverse, reverse!, reverse_each, rindex, select, shift, size,
slice, slice!, sort, sort!, to a, to ary, to s, transpose, uniq,
uniq!, unshift, values_at, zip, |

% ri Array.sort (Arrayクラスのsortメソッドの解説が表示される)
----- Array#uniq
array.uniq -> an array
-----
Returns a new array by removing duplicate values in self .

a = [ "a", "a", "b", "b", "c" ]
a.uniq #=> ["a", "b", "c"]
```

のようになります。このヘルプは標準添付の拡張ライブラリについても生成されますが、ドキュメントがまだ付属していないものが多く、全てのライブラリで利用できるわけではないようです。