JSAI: BioRuby によるデータベースアクセスと配列解析

Database access and sequence analysis in BioRuby

- 著者名

  片山 俊明
  Toshiaki Katayama

- 所属名

  東京大学医科学研究所ヒトゲノム解析センター
  Human Genome Center, Institute of Medical Science, University of Tokyo.

- 連絡先

  E-mail k@bioruby.org
  URL http://bioruby.org/
  TEL 03-5449-5614
  FAX 03-5449-5434

  〒108-0071
  東京都港区白金台4-6-1
  東京大学医科学研究所ヒトゲノム解析センター
  ゲノムデータベース分野

本文

バイオインフォマティクスという分野は 1990 年代のゲノムプロジェクトの進 展とともに発展を続けてきた。ご存知のように、ゲノムプロジェクトとは、ヒ トから微生物まで様々な生物において、その生命の設計図ともいわれる DNA 配列を全て解読するという壮大な計画である。現在までに 150 種あまりの原 核生物と、一部まだドラフト（完了ではない）のものもあるが酵母、線虫、ハ エ、シロイヌナズナ、そしてヒトといった 20 種ちかくの真核生物まで、多数 の生物の配列解読が終了している。DNA 配列は A, T, G, C の４種類の塩基と 呼ばれる物質が二重螺旋の紐状に並んだものであるが、解読された配列はこの ４文字の並びとしてデータベース化されている。DNA の長さは原核生物のゲノ ムでは数Mb (数百万塩基) 程度であるが、ヒトでは 3Gb (30億塩基) にもなり、 通常は文字列として扱われるため塩基数がそのままバイト数になる。実際に配 列を決定する際には一度に数百塩基ずつしか読むことができないため、この短 い断片配列が互いに重なり合ってゲノム全体をカバーするようにゲノムの長さ の10倍近くを冗長に配列決定する必要があり、データの量はさらに大きくなる。 このような背景から、これらの断片配列を文字列の照合により効率的に繋ぐ方 法や、大量に生成されるデータを管理する技術、解読した配列から遺伝子を探 すといった配列解析のアルゴリズム開発などを中心に、大型計算機やクラスタ を活用したバイオインフォマティクスという分野が広がってきた。

最近では配列解析にとどまらず、数千もの遺伝子の発現状態を一度に計測でき るマイクロアレイのデータ解析や、ツーハイブリッド法などによるタンパク質 同士の相互作用のデータ解析、相互作用のネットワーク的な構造を調べるパス ウェイ解析、マススペクトロメトリーによる大規模な質量分析、タンパク質の 立体構造の予測や機能解析などなど、バイオインフォマティクスの対象も多岐 に及んできているが、本稿では基本的なデータベース検索と配列解析を中心に 解説したい。

なお、本文中のファイル名はサポートページのサンプルのファイル名と揃えて あるので、適宜サポートページの方も参照頂きたい。

配列データベース

配列に基づく研究成果を学術論文に発表する場合はその配列の公共データベー スへの登録が求められるため、ゲノムに限らずこれまでに明らかになった遺伝 子などの塩基配列は日米欧のデータセンターでそれぞれ DDBJ, GenBank, EMBL の３つのデータベースに蓄積されており、相互に補完されている。これらのデー タベースはフォーマットが異なるが内容的には基本的に同じであり、2003年12 月現在の GenBank データベースには3千万エントリ、360億塩基のデータが登 録されている。この他に、遺伝子の塩基配列をアミノ酸に翻訳してできるタン パク質の配列データベースや、特にゲノムの決まった生物を中心に生物種ごと に整理しアノテーション（付加情報）付けされたデータベース（KEGG/GENES）など も作られている。

主な配列データベース

GenBank	http://www.ncbi.nlm.nih.gov/Genbank/
RefSeq	http://www.ncbi.nlm.nih.gov/RefSeq/
DDBJ	http://www.ddbj.nig.ac.jp/
EMBL	http://www.ebi.ac.uk/embl/
UniProt	http://www.uniprot.org/
KEGG/GENES	http://www.genome.ad.jp/kegg/

注：UniProt は Swiss-Prot, PIR, TrEMBL などを統合して作られた。

これらのデータベースはウェブ上で検索したり、フラットファイルのテキスト データとしてダウンロードして使うことができる。特定の遺伝子に興味がある 場合など、個々のエントリをじっくりと目で見て調べるにはウェブのインター フェイスを利用して対話的に検索する方が便利だが、多数のエントリから何ら かの基準でマイニングしたり、順番に同じ処理を繰り返し加工するなどの場合 にはプログラムから利用する方が効率が良く拡張性も高くなる。

今回紹介する BioRuby をはじめ、BioPerl, BioPython, BioJava などのオー プンソースなライブラリ（これらの開発プロジェクトを総称して Open Bio* とよぶこともある）や EMBOSS などのツール群を使うと、インターネット上の 公共サーバにあるデータベースや、ローカルにコピーしてきたデータベースを 用いた検索が容易にできる。

関連ソフトウェアなどのサイト

```
-------------------------------------------------------------
BioRuby     http://www.bioruby.org/
BioPerl     http://www.bioperl.org/
Biopython   http://www.biopython.org/
BioJava     http://www.biojava.org/
Open Bio*   http://www.open-bio.org/
OBDA        http://obda.open-bio.org/
EMBOSS      http://www.emboss.org/
BLAST       http://www.ncbi.nlm.nih.gov/BLAST/
Ruby        http://www.ruby-lang.org/
RAA         http://raa.ruby-lang.org/
-------------------------------------------------------------
```

データベースエントリの取得

まずは何でも良いので自分の興味のある遺伝子（実験生物学者でない情報系の ひとにとっては興味の対象を探すのが一番難しいのかもしれないが）の配列を 取得する。最初は配列データベースのウェブサイトにある検索フォームを用い て探してみるのが良いだろう。例としてゲノムネットのページ http://www.genome.ad.jp/ を開き、フォームの Search となっている部分で 対象データベースに GENES や GenBank を選んで、kinase や caffeine、SARS など思いつく適当なワードで検索してみる。

ここでは GenBank のアクセッション番号 AJ617376 のエントリを用いること にした。このエントリは、ある生物の 18S リボソーマル RNA 遺伝子の部分配 列らしい。rRNA は蛋白質を製造するリボソーム複合体に含まれる RNA 分子で、 大多数の遺伝子とは異なり蛋白質には翻訳されず塩基のまま機能する。この分 子は進化系統の解析に使われることが多く、rRNA の配列だけ決まっている生 物種も多い。

1. http://www.genome.ad.jp/dbget-bin/www_bget?genbank+AJ617376
2. http://www.genome.ad.jp/dbget-bin/www_bget?-f+genbank+AJ617376

エントリ全体は (1) の URL で表示することができるが、必要なのはこれの配 列部分だけなので、(2) を testseq.txt と名付けたファイルに保存する。こ の形式は FASTA フォーマットとよばれ、'>' で始まるのがコメント行、続く 行が塩基やアミノ酸の配列を表す文字列となる。

ファイル `testseq.txt` の内容

```
-------------------------------------------------------------
>gb:AJ617376 [AJ617376] Tardigrada sp. A001 MOTU012 partial 18S rRNA gene ...
taagccatgcatgtctcagtacttgctttaacaaggcgaaaccgcgaatggctcattaaa
tcagttatggttcactagatcgtatatcttacatggataactgtggtaattctagagcta
atacatgcattcattccgtgccctcgtggtgcggaagcagttattttctcaaaaccaatc
gggccttcgggttcgttaattggtgactctgaataaccgaagcgaagcgaatggtctcgt
accgtcgctagatctttcaagtgtctgacttatcagcttgttgttaggttatgttcctaa
caaggcagttacgggtaacggggtgtcagggcccgacaccggagagggagcctgagaaac
ggctaccacatccaaggaaggcagcaggcgcgcaaattacccactcccggcacggggagg
tagtgacgaaaaataacgatgcgagagctttatgcttctcgtaatcggaatgggtacact
ttaaatcctttaacgaggatctattggag
-------------------------------------------------------------
```

BLAST の実行

BLAST は塩基配列やアミノ酸配列の類似性（相同性）を検出するプログラムと して最も広く使われている。アルゴリズムの詳細は文献を参照して頂くとして、 まずは使用例を見ることにしたい。

上記の testseq.txt に保存した配列と類似な配列を BLAST プログラムを使っ て検索してみることにする。BLAST 検索が可能なウェブサイトはいくつもある が、ここでは例としてゲノムネットの http://blast.genome.ad.jp/ を用いる。 ページを開き、保存した問い合わせ配列が塩基配列なので BLASTN プログラム を選択、検索対象データベースに塩基配列データベースの GenBank を選択す る。あとは testseq.txt ファイルをアップロードして Exec ボタンを押せば 検索できるはずである（図１）。検索結果はスコアの高い順に リストアップされ、配列の類似部分についてのアラインメントが続く。ここで 表示される bit スコアは、検索に使用する置換行列間の違いをなくすように 補正された値で、類似性が高いほど大きくなる。一方、E-value はランダムな 同じ大きさのデータベース中に問い合わせ配列が偶然一致する期待値で、小さ いほど類似性が高いことになる。アラインメントを見ると、スコアが低くなっ ていくにしたがって不一致部分が増えたり、類似領域の長さが減少していくの が分かると思う。

図１(`blast-web.png`)：

```
-------------------------------------------------------------
ウェブによる BLAST 検索の実行例
-------------------------------------------------------------
```

BLAST のプログラム名と配列種別の対応

プログラム名	問い合わせ配列	検索対象データベース
blastp	アミノ酸配列	アミノ酸配列
blastn	塩基配列	塩基配列
blastx	塩基配列	アミノ酸配列
tblastn	アミノ酸配列	塩基配列

このように、調べたい配列が少数の場合はウェブだけで簡単に検索できるが、 ある生物の遺伝子を全て処理して類似配列を検索し、その結果をサマライズす るとなると手作業では厳しくなってくる。そこで、BLAST 検索をローカルのコ ンピュータで実行できる環境を作り、プログラムを使って結果をフィルタリン グするようなパイプラインを準備することになる。

BLAST インストール手順

1. BLAST はソースも配布されているがバイナリも提供されているので使用す るコンピュータ用のものを取得するのが簡単である。

BLAST のダウンロードサイト

```
ftp://ftp.ncbi.nlm.nih.gov/blast/executables/release/2.x.x/blast-2.x.x-arch-os.tar.gz
```

2. 展開して適当なディレクトリにインストールする。必要に応じて同梱のマ ニュアルも参照する。

```
% mkdir blast-2.x.x
% cd blast-2.x.x
% tar zxvf ../blast-2.x.x-arch-os.tar.gz
% sudo mkdir -p /usr/local/bin /usr/local/share/ncbi
% sudo cp bl2seq blastall blastclust blastpgp copymat fastacmd ¥
    formatdb impala makemat megablast rpsblast seedtop /usr/local/bin/
% sudo cp -r data /usr/local/share/ncbi/
```

3. data ディレクトリの内容をコピーしたディレクトリの場所を ~/.ncbirc に書く。

ファイル ~/.ncbirc の内容

```
[NCBI]
Data=/usr/local/share/ncbi/data
```

4. 検索対象となる FASTA フォーマットの配列データベースを用意する。ここ では、例として KEGG の GENES データベースから、大腸菌の全遺伝子のアミ ノ酸配列を格納したファイルを取得して用いることにする。

KEGG/GENES データベースのダウンロードサイト

```
ftp://ftp.genome.ad.jp/pub/kegg/genomes/sequences/e.coli.pep
```

5. BLAST に付属の formatdb プログラムで用意したデータベースファイルを フォーマットする。アミノ酸配列のデータベースなので -p オプションに T を指定する。fastacmd プログラムで遺伝子名から配列を取り出せるように -o オプションにも T を指定した。

```
% formatdb -i e.coli.pep -p T -o T
```

6. 例として、ある遺伝子（この例では大腸菌 E. coli の b0002 遺伝子）の アミノ酸配列を BLAST に付属の fastacmd プログラムで取り出してみる。

```
% fastacmd -d e.coli.pep -s eco:b0002 > b0002.txt
```

7. 取り出した配列と類似の配列を検索し、BLAST の動作確認をする。さらに オプション -m 7 で XML、-m 8 でタブ区切りなど、出力フォーマットが変更 できることを確認しておく。

```
% blastall -p blastp -d e.coli.pep -i b0002.txt
% blastall -p blastp -d e.coli.pep -i b0002.txt -m 7
% blastall -p blastp -d e.coli.pep -i b0002.txt -m 8
```

BioRuby

BLAST の動作環境が用意できたら、次は BLAST の膨大な出力結果から必要な 情報を抽出できるようにしたい。自分で出力をパースするプログラムを書いて もよいが、BioPerl, BioPython, BioJava など各言語用のライブラリがすでに 作られているのでこれらを利用するのが簡単だろう。ここでは Ruby 言語用 に筆者らが開発したバイオインフォマティクス関連のライブラリである BioRuby を使った方法を紹介する。まずは Ruby を用意しよう。

Ruby は、まつもとゆきひろ氏によって作られたオブジェクト指向のスクリプ ト言語である。もともとバイオインフォマティクスでは文字列データを取り扱 うことが多いため、正規表現が強力で手軽に使える Perl が広く使われてきた。 しかし、Perl では少しプログラムが大規模になってくると Perl5 以降に追加 されたオブジェクト指向機能を導入することが多いが、あと付けの機能である ため文法が複雑になってしまった。実際に BioPerl は Perl のオブジェクト 指向機能を全面的に取り入れているが、Perl 初心者には敷居が高くなってい る。これに対し Ruby、Python、Java ではオブジェクト指向プログラムが書き やすく、特に Ruby では必要な処理を短く簡単にプログラミングでき Java の ようなコンパイル作業も不要であるため、実験生物学者が必要な仕事を手軽に 済ませるための言語としても最適ではないかと考えている。各言語の特性だけ でなく、生物学のデータがオブジェクト指向を使うと表現しやすいという点も、 Open Bio* の各ライブラリがオブジェクト指向言語で書かれている要因かもし れない。各言語でプログラムの書き方がどのように違うかを、ほぼ同じ実行内 容の BLAST 出力の処理方法で見比べてみるのも興味深い。

```
Open Bio* 各言語での BLAST 出力の処理方法
-------------------------------------------------------------
BioPerl    http://bioperl.org/HOWTOs/html/SearchIO.html#use
BioPython  http://www.biopython.org/docs/tutorial/Tutorial004.html#htoc24
BioJava    http://www.biojava.org/docs/bj_in_anger/BlastParser.htm
BioRuby    http://wiki.bioruby.org/English/?How+do+I+set+up+a+BLAST+parser%3F
-------------------------------------------------------------
```

BioRuby のインストール

BioRuby に先立って、Ruby がない場合は Ruby のインストールを済ませてお く必要がある。最近は標準で Ruby が入っている OS も増えてきたが、バイナ リパッケージも見つからない場合には Ruby のソースを取ってきてインストー ルすることになる。その場合は、SOAP など標準添付ライブラリが増えている Ruby 1.8.1 以降の新しいものを選択するのがお勧めである。同様に、BioRuby をダウンロードする際も、リリース番号の新しいものを選択する。

```
Ruby のダウンロードサイト
-------------------------------------------------------------
ftp://ftp.ruby-lang.org/pub/ruby/ruby-1.8.x.tar.gz
-------------------------------------------------------------
```

1. Ruby のインストール

```
% tar zxvf ruby-1.8.x.tar.gz
% cd ruby-1.8.x
% ./configure
% make
% sudo make install
```

2. BioRuby の取得

```
BioRuby のダウンロードサイト
-------------------------------------------------------------
http://bioruby.org/archive/bioruby-0.x.x.tar.gz
-------------------------------------------------------------
```

3. BioRuby のインストール

```
% tar zxvf bioruby-0.x.x.tar.gz
% cd bioruby-0.x.x
% ruby install.rb config
% ruby install.rb setup
% sudo ruby install.rb install
```

4. データベースのアクセスに必要な定義ファイルのインストール（システム ワイドに設定を行うには /etc/bioinformatics にコピーする）

```
% cp -r etc/bioinformatics ~/.bioinformatics
```

5. Ruby と、インストールされた BioRuby のバージョンを確認する

```
% ruby -v -r bio -e 'p Bio::BIORUBY_VERSION'
```

これで BioRuby を使う準備は整ったので、配列の操作やデータベースファイ ルの取り扱いなど多くの機能を使うことができる。できれば、BLAST の出力や SOAP など XML の処理を高速にするために、xmlparser と依存する expat ラ イブラリをインストールしておくとよい。

```
Ruby 1.6 と 1.8 用の追加ライブラリ
-------------------------------------------------------------
xmlparser     http://raa.ruby-lang.org/list.rhtml?name=xmlparser
expat         http://expat.sf.net/
-------------------------------------------------------------
```

また、Ruby 1.6 系のユーザは、BioRuby の様々な機能を有効にするために、 ruby-sumo や soap4r などいくつかのライブラリを追加インストールするのが よいだろう。Ruby 1.8.0 のユーザは 1.8.1 以降にバージョンアップすることを お勧め

する。

```
Ruby 1.6 用の追加ライブラリ
-------------------------------------------------------------
ruby-sumo     http://raa.ruby-lang.org/list.rhtml?name=ruby-sumo
soap4r        http://raa.ruby-lang.org/list.rhtml?name=soap4r
http-access2  http://raa.ruby-lang.org/list.rhtml?name=http-access2
date2         http://raa.ruby-lang.org/list.rhtml?name=date2
-------------------------------------------------------------
```

依存ライブラリなどについて最新の状況は BioRuby のウェブページで確認し て頂きたい。

```
BioRuby のインストールに関するページ
-------------------------------------------------------------
http://wiki.bioruby.org/Japanese/?BioRuby_Install
-------------------------------------------------------------
```

BioRuby を使ったデータベースエントリの取得

2002 年と 2003 年に開かれた BioHackathon というミーティングにおいて、 BioRuby をはじめ各 Open Bio* の開発者間で、特に配列データベースへのアク セスに関して、共通な仕様を作ることを取り決めた。この仕様は OBDA (Open Bio Database Access) とよばれ、以下の４つが考えられている。

- BioFetch
- BioFlat
- BioSQL
- BioRegistry

BioFetch は、ウェブを使ってサーバの持つ配列データベースの任意のエント リを決まった引数で取得できる CGI で、インターネット（HTTP）にアクセス できる環境があれば利用可能なので手軽である。現在 BioFetch のサーバはイ ギリスにあるバイオインフォマティクスのメッカ EBI と BioRuby のサーバ （実際には内部でゲノムネットにアクセスしている）にある。利用可能なデー タベースの種類とバージョンはサーバに依存する。

```
BioFetch のサーバ
-------------------------------------------------------------
EBI          http://www.ebi.ac.uk/cgi-bin/dbfetch
BioRuby      http://bioruby.org/cgi-bin/biofetch.rb
-------------------------------------------------------------
```

BioFlat はダウンロードしてきたフラットファイルの配列データベースに対し てインデックスを作成し、エントリの ID による高速な検索を可能にするもの である。特定の生物種や興味のあるファミリーの遺伝子だけを集めるなどして、 独自の規準で作成したデータベースのサブセットを活用する、といった場合に も手軽に使うことができる。

BioSQL は複雑なデータ構造をもつ配列データベースのエントリを正規化し、 MySQL, PostgreSQL, Oracle などの関係データベースに格納するためのスキー マで、フィールドごとに高度な検索を行える。また、BioPerl などの各ライブ ラリで抽象化された配列オブジェクトのシリアライズを行うことができるとい う意味もある。

BioRegistry は、GenBank や SwissProt など様々な配列データベースに対し、 それぞれのデータベースごとに上記のどの方法でどこにアクセスして取得する かを、サイトやユーザが設定できるようにする仕組みである。BioRuby のイン ストールの時にコピーしたのがその設定ファイルで、ユーザ用の設定ファイル が優先される。

```
BioRegistry の設定ファイル
-------------------------------------------------------------
ユーザ用      ${HOME}/.bioinformatics/seqdatabase.ini
サイト用      /etc/bioinformatics/seqdatabase.ini
サンプル      http://www.open-bio.org/registry/seqdatabase.ini
-------------------------------------------------------------
```

Open Bio* では BioRegistry を使う biogetseq コマンドを作ることになって おり、BioRuby でもインストールされているはずである。試しに、本稿の最初 に取得した rRNA のエントリなどを biogetseq で取得してみよう（※脚注）。

```
% biogetseq --dbname genbank AJ617376 > AJ617376.txt
% biogetseq --dbname embl HSPRP > HSPRP.txt
% biogetseq --dbname swissprot CG11_YEAST > CG11_YEAST.txt
```

多少時間がかかるがローカルにデータベースを構築し、定期的に更新する労力 を考えると簡単である。

```
※脚注：
-------------------------------------------------------------
コマンド名が他のプロジェクトとかぶるので将来のリリースでは
biogetseq.rb になっているかもしれない
-------------------------------------------------------------
```

BioRuby を使った BLAST 出力のパース

ようやく BLAST 出力結果のパースに話が戻ってきた。すでに見たように BLAST にはいくつかの出力フォーマットがある。目で結果を見るにはデフォル トのフォーマットが分かりやすいが、コンピュータで処理する場合 XML など のフォーマット

の方が処理しやすい。BioRuby では最初に XML フォーマット のパーザが開発され、後にデフォルトのフォーマットのパーザが追加された経 緯があり、まだこれらを同じインターフェイスで扱うようにはなっていない。 いずれの出力フォーマットも Bio::FlatFile クラスで自動判定できるように してメソッドも揃えたいのだが（本稿執筆中には間に合わなかったがそのうち 対応しよう）、今回はとりあえずできるだけ違いの出ないサンプルプログラム を使って、フォーマットに応じて 1 行だけ修正して実行する。

```
blast_parse.rb (blast_parse_xml.rb)
-------------------------------------------------------------
#!/usr/bin/env ruby

require 'bio'

### デフォルトフォーマットの場合
reports = Bio::FlatFile.open(Bio::Blast::Default::Report, ARGF)
### XML フォーマットの場合
#reports = Bio::Blast.reports(ARGF)

reports.each do |report|
  print "--- BLAST hits : ", report.db, "¥n"
  print "Query = ", report.query_def, "¥n"
  report.each do |hit|
    print "Hit = ", hit.definition, "¥n"
    hit.each do |hsp|
      if hsp.align_len > 100
        print " Length = ", hsp.align_len, "¥n"
        print " E-value = ", hsp.evalue, "¥n"
      else
        print " ..skipped..¥n"
      end
    end
  end
end
-------------------------------------------------------------
```

BLAST の出力は、クエリ配列である b0002 遺伝子と相同性のある遺伝子がス コアの高い順に並んでいる。blast_parse.rb プログラムはそれぞれの遺伝子 について、アラインメント領域の長さが 100 残基より長いものをピックアッ プして E-value とともに結果を表示する（出力結果はサポートサイトにも同 じファイル名で置かれているので参照して頂きたい）。実際には他の様々な値 や配列のアラインメントを活用するプログラムを書く必要があり、そのために BLAST の出力結果の全てのパーツについて値を取り出すためのメソッドが BioRuby に準備されているので、ドキュメントを参照して活用して頂きたい。

1. デフォルトフォーマットの場合

```
% blastall -p blastp -d e.coli.pep -i b0002.txt > b0002.out
% ruby blast_parse.rb b0002.out > b0002_out.txt
```

2. XML フォーマットの場合

```
% blastall -p blastp -d e.coli.pep -i b0002.txt -m 7 > b0002.xml
% ruby blast_parse_xml.rb b0002.xml > b0002_xml.txt
```

先に紹介した bit スコアと E-value は、-m 7 の XML 出力ではそれぞれ <Hsp_bit-score> と <Hsp_evalue> タグに書かれている。逆に、アラインメン ト自体は必要がなく、スコアだけ欲しいという場合には -m 8 のタブ区切り フォーマットが有効である。また、BLAST の実行自体も BioRuby のプログラ ムから可能で、さらにゲノムネットなどリモートのサーバを利用した BLAST 検索を行う機能もあるのだが、いずれも本稿では割愛する。実際の運用では、 大型計算機やクラスタを利用して並列化する場合も多い。

KEGG API

最後に、筆者らの開発している KEGG API を BioRuby から使ってパスウェイ など配列以外の情報を扱う例を紹介する。KEGG は京都大学化学研究所バイオ インフォマティクスセンターで開発され公開されているシステムで、すでに利 用した遺伝子のデータベース GENES 以外に、タンパク質や化合物間の機能的 な関係をグラフで表した PATHWAY など、様々なデータベースを含んでいる。KEGG ではこれらのデータベースを組み合わせたデータマイニングや解析などに利用 できるプログラムを CGI としてウェブ上で公開している。具体的なイメージ はページを開いて参照して頂きたい。一方で、これらの機能は手作業でデータ 解析するために作られたものであり、プログラムから利用するには向いていな かった。そこで SOAP と WSDL を用いたウェブサービスとしてこれらの機能を 実装したのが KEGG API である。KEGG API を用いると、入力データだけ替え て同じような処理を繰り返し実行するようなプログラムを簡単に実行できる他、 他のウェブサービスと連携するなど複雑な解析のパイプラインを作ることがで きる。

```
KEGG API 関連ページ
-------------------------------------------------------------
KEGG API      http://www.genome.ad.jp/kegg/soap/
KEGG/PATHWAY  http://www.genome.ad.jp/kegg/pathway.html
LinkDB        http://www.genome.ad.jp/dbget/dbget_manual.html
-------------------------------------------------------------
```

ここでは、指定した KEGG のパスウェイ上に載っている遺伝子のリストを取得 して、LinkDB を使って立体構造データベース PDB へのリンクをたどり、関連 する構造データが見つかった遺伝子についてマップに色を付けた画像を取得す る、という例をあげてみる。

```
map_pdb_on_pathway.rb
------------------------------------------------------------
#!/usr/bin/env ruby

require 'bio'
require 'uri'
require 'net/http'

# KEGG API のサーバに接続し WSDL から利用可能なメソッドを取得
serv = Bio::KEGG::API.new

# 指定したパスウェイに載っている遺伝子名のリストを取得
path = ARGV.shift || 'path:eco00010'
genes = serv.get_genes_by_pathway(path)

# LinkDB を用いて、各遺伝子から PDB データベースへのリンクを探す
results = Hash.new
genes.each do |gene|
  print gene
  if links = serv.get_linkdb_by_entry(gene, 'pdb')
    # リンクがあれば、順番にリンク先のエントリ名を表示する
    links.each do |link|
      results[gene] ||= Array.new
      results[gene] << link.entry_id2
      print "¥t" + link.entry_id2
    end
  end
  puts
end

# リンクがあった遺伝子のリストを渡して枠に色付けした画像を生成する
url, = serv.mark_pathway_by_genes(path, results.keys)
STDERR.puts url

# 画像の URL が返されるので、実際の画像ファイルを取得し、ファイルに保存する
host = URI.parse(url).host
path = URI.parse(url).path

filename = 'pdb_on_' + File.basename(path)

File.open(filename, File::CREAT|File::TRUNC|File::RDWR) do |file|
  file.print Net::HTTP.get_response(host, path).body
end
------------------------------------------------------------
```

map_pdb_on_pathway.rb プログラムを実行すると「pdb_on_パスウェイ番号.gif」 のような画像ファイルが生成される。標準出力には、指定したパスウェイに載っ ていた遺伝子と、そこから辿ることのできた PDB のエントリ名がタブ区切りで出力される。デフォルトでは大腸菌の解糖系パスウェイ path:eco00010 を 使用するが、KEGG/PATHWAY のページに載っている任意のパスウェイを引数に 指定できる。図２はヒトのビオチン代謝経路を使った例である。四角の箱で表 されているものが酵素（遺伝子から作られるタンパク質）で、箱の中には酵素 反応を分類した酵素番号が示されている。丸いノードは酵素によって代謝され る化合物を表しており、矢印の方向に反応が進む。塗られている箱はヒトが対 応する酵素遺伝子を持つことを示し、PDB へのリンクを辿ることができた箱は 枠の色が変わっている。

```
% ruby map_pdb_on_pathway.rb > pdb_on_eco00010.txt
% ruby map_pdb_on_pathway.rb path:hsa00780 > pdb_on_hsa00780.txt
```

```
図２(pdb_on_hsa00780.gif)：
------------------------------------------------------------
ヒトのビオチン代謝経路に関わる遺伝子から PDB へのリンクを辿ってみる
------------------------------------------------------------
```

実際には GENES の遺伝子名から PDB へのリンクのほとんどが酵素番号を辿っ てしまうため、ここで色がついたものがどれくらい意味を持っているのかは怪 しいところだが、意外と多くの遺伝子にリンクが見つかった。重要なパスウェ イの中で色がつかなかったタンパク質の立体構造を優先的に決定していくと良 いのかもしれない。

KEGG API の応用として、マイクロアレイの実験データをもとにパスウェイ上 の各遺伝子に色づけを行うことで活性化しているパスウェイを解析したり、新 しく配列決定した遺伝子群にオーソログクラスターなどを利用して機能推定を 行うといった例が考えられるが、KEGG API の機能とユーザのプログラムを組 み合わせて何ができるかは今後の工夫次第である。

おわりに

当初の予定では、塩基やアミノ酸の配列操作や、EMBOSS や BioFlat などを用 いたローカルデータベースの構築、それぞれのエントリを BioRuby の機能を 使ってパースする例なども取り上げるつもりだったが、残念ながら触れること ができなかった。また、BioRegistry の設定方法や BioSQL についても詳しく 取り上げることはできなかった。これらについては BioRuby 付属のドキュメ ント等を参照して頂きたい（doc ディレクトリの中にチュートリアルなどがあ る）。この他、BioRuby プロジェクトでは、バイオインフォマティクス関連の リソースへのリンクやソフトウェアの更新情報などを掲載したニュースサイト も運用しているので見て頂けたらと思う。また、本稿で興味を持たれた方はぜ ひメーリングリストに

も参加して応援して頂けると幸いである。

```
BioRuby 関連の情報サイト
------------------------------------------------------------
トップ       http://bioruby.org/
ニュース     http://news.bioruby.org/
Wiki         http://wiki.bioruby.org/
------------------------------------------------------------
```

もう一点、今回紹介した BioRuby も KEGG API もまだ開発途上にあるため、 今後の改良で仕様が変更になる可能性もある。そのような場合、ドキュメント 等を参照して適宜対応して頂きたい。

参考文献

- 書籍

バイオインフォマティクス David W. Mount 著 メディカル・サイエンス・インターナショナル ISBN4-89592-307-X

ゲノムネットの利用法 第３版 金久實編 共立出版 ISBN4-320-05595-0

The Ruby Way Hal Fulton 著 翔泳社 ISBN4-7981-0228-8

BLAST Ian Korf, Mark Yandell, Joseph Bedell 著 O'Reilly ISBN0-596-00299-8

SEQUENCE ANALYSIS in a nutshell Scott Markel, Darryl Leon 著 O'Reilly ISBN0-596-00494-X

- 論文

¥bibitem{PMID:2231712} Altschul, S. F., Gish, W., Miller, W., Myers, E. W., Lipman, D. J. Basic local alignment search tool., {¥em J Mol Biol}, 215(3):403--410, 1990.

¥bibitem{PMID:9254694} Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., Lipman, D. J. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs., {¥em Nucleic Acids Res}, 25(17):3389--3402, 1997.

¥bibitem{PMID:14681412} Kanehisa, M., Goto, S., Kawashima, S., Okuno, Y., Hattori, M. The KEGG resource for deciphering the genome., {¥em Nucleic Acids Res}, 32 Database issue():D277--D280, 2004.

¥bibitem{PMID:12368254} Stajich, J. E., Block, D., Boulez, K., Brenner, S. E., Chervitz, S. A., Dagdigian, C., Fuellen, G., Gilbert, J. G., Korf, I., Lapp, H., Lehvaslaiho, H., Matsalla, C., Mungall, C. J., Osborne, B. I., Pocock, M. R., Schattner, P., Senger, M., Stein, L. D., Stupka, E., Wilkinson, M. D., Birney, E. The Bioperl toolkit: Perl modules for the life sciences., {¥em Genome Res}, 12(10):1611--1618, 2002.

¥bibitem{PMID:12000935} Stein, L. Creating a bioinformatics nation., {¥em Nature}, 417(6885):119--120, 2002.

著者紹介

片山 俊明

1972年生 1996年京都大学理学部卒業 1998年同大学院修士課程修了 2001年同大学院博士課程単位取得退学 同年より京都大学にて教務補佐員 2003年東京大学医科学研究所助手

更新日時:2004-03-04 (木) 20:05:35
キーワード:
参照:[BioRuby によるデータベースアクセスと配列解析]