

# BioRuby

toshiaki katayama  
k@bioruby.org

好きなメソッド： []

嫌いなメソッド： =~

Rubyっぽいところ？：名前に時間かけてるかも、、、  
(あとリリースが遅め)



2006/6/10-11 日本Rubyカンファレンス2006 Lightning talk (<http://jp.rubyist.net/RubyKaigi2006/>)

# BioRubyって？

- ゲノムとか、生物の解析に使うライブラリ
- 2000年、そそのかされ Perl を捨てて開発開始
- 2005年、YARV のささださんと未踏の同級生に
- 2010年、極限生物クマムシのゲノム解析で活躍



→ <http://kumamushi.net/>

驚異の生物クマムシ！

- 耐冷, 耐熱
- 耐X線
- 耐真空, 耐圧

# who we are?



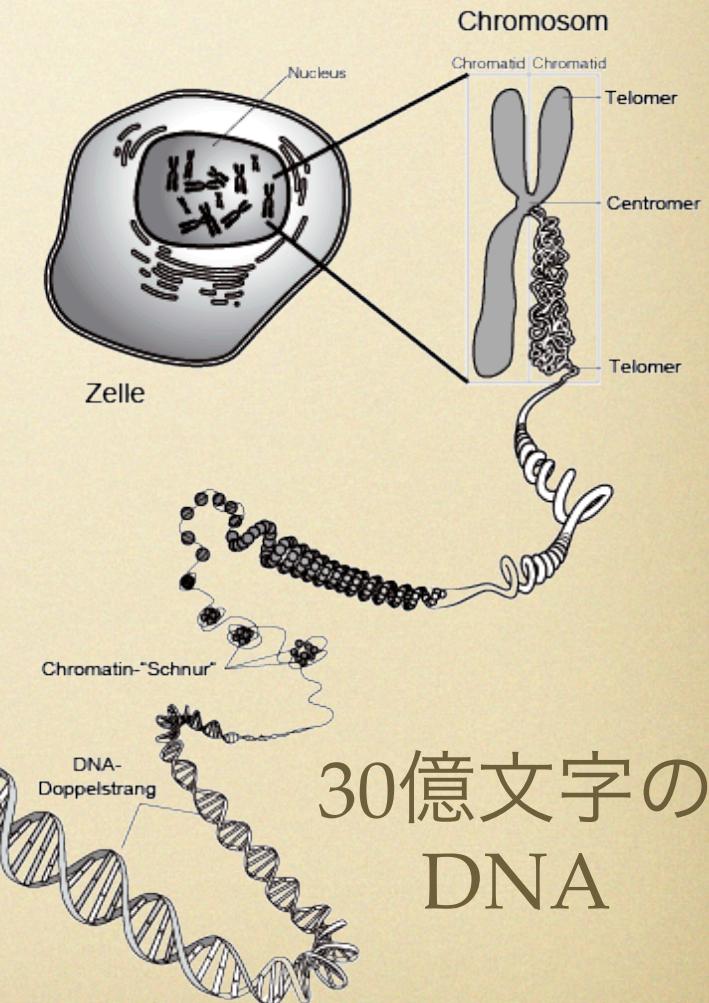
# who we are?

- 中尾光輝：かずさDNA研（RubyKaigi会場確保）
- 後藤直久：阪大
- 田中伸也：京大（ChemRuby）
- 片山俊明：東大医科研
- + 内外のコントリビュータ、コミッタ

# ヒトゲノム計画とバイオインフォ

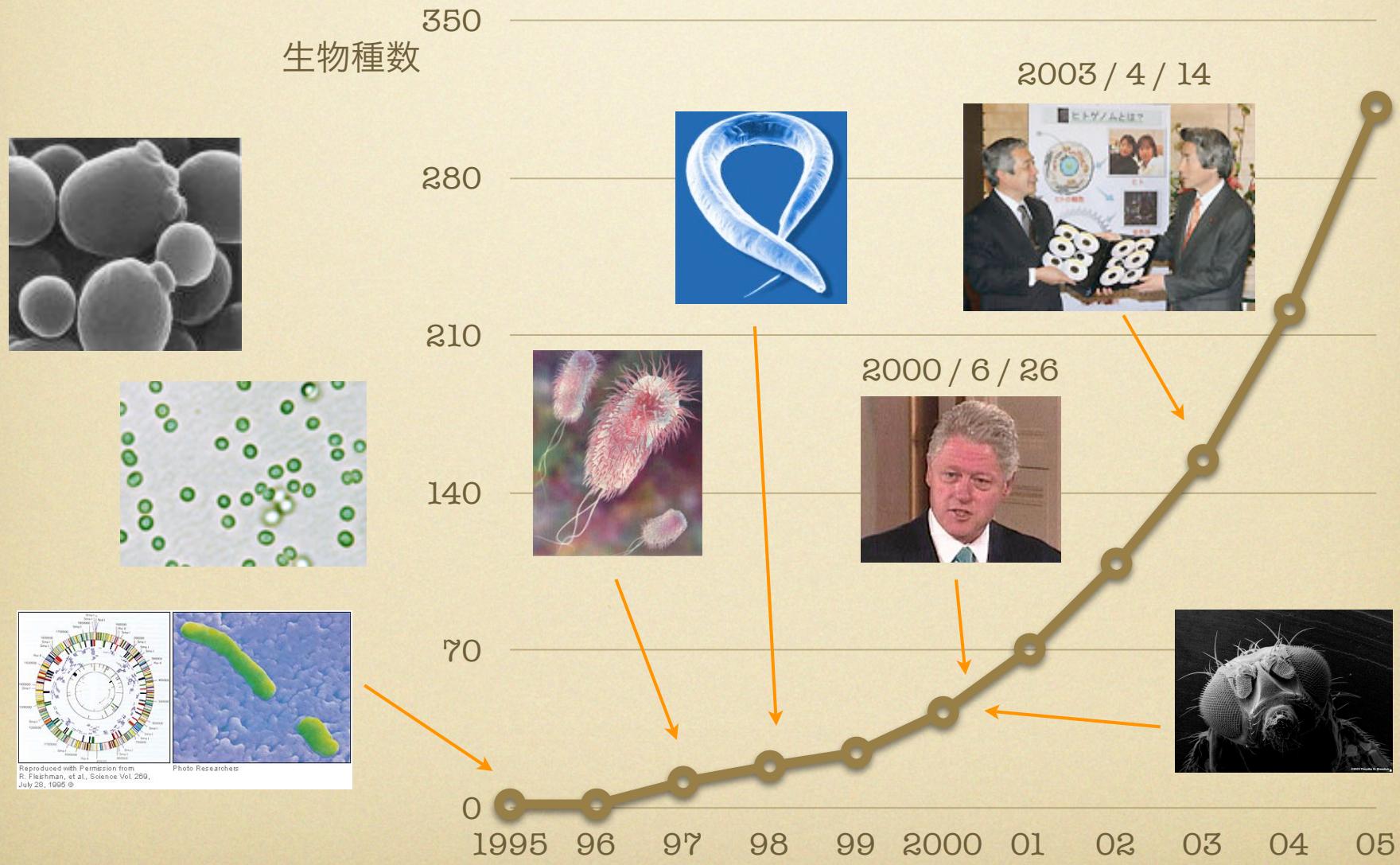
The screenshot shows the Ensembl Human genome browser. On the left, there's a sidebar with links for various Ensembl services like BLAST search, Data mining, and Ensemble data. The main area displays a chromosome map of Homo sapiens (NCBI 36 assembly) with numbered chromosomes. A specific genomic region on chromosome 21 is highlighted with a green bar at the top labeled '606,496'. Below this, a detailed track shows gene annotations, SNPs, and other genomic features. A large green arrow points from the text 'テラバイト' (TeraByte) towards this genomic data.

- TBのテキストデータ
- 文字列処理で Perl の成功  
→ BioPerl



<http://ja.wikipedia.org/wiki/染色体> より

# 300生物種以上のゲノム解読が完了



# ドラフトゲノムも続々と進展中！

*Canis familiaris*

犬

*Gallus gallus*

鶏

*Pan troglodytes*

猿  
(チンパンジー)

*Ovis aries*

羊

*Sus scrofa*

豚

*Mus musculus*

子(マウス)

*Bos taurus*

牛

TIGR

*Oryctolagus cuniculus*

兎

タツノオトシゴ /  
リュウグウノツカイ /  
トンボ ?



*Equus caballus*

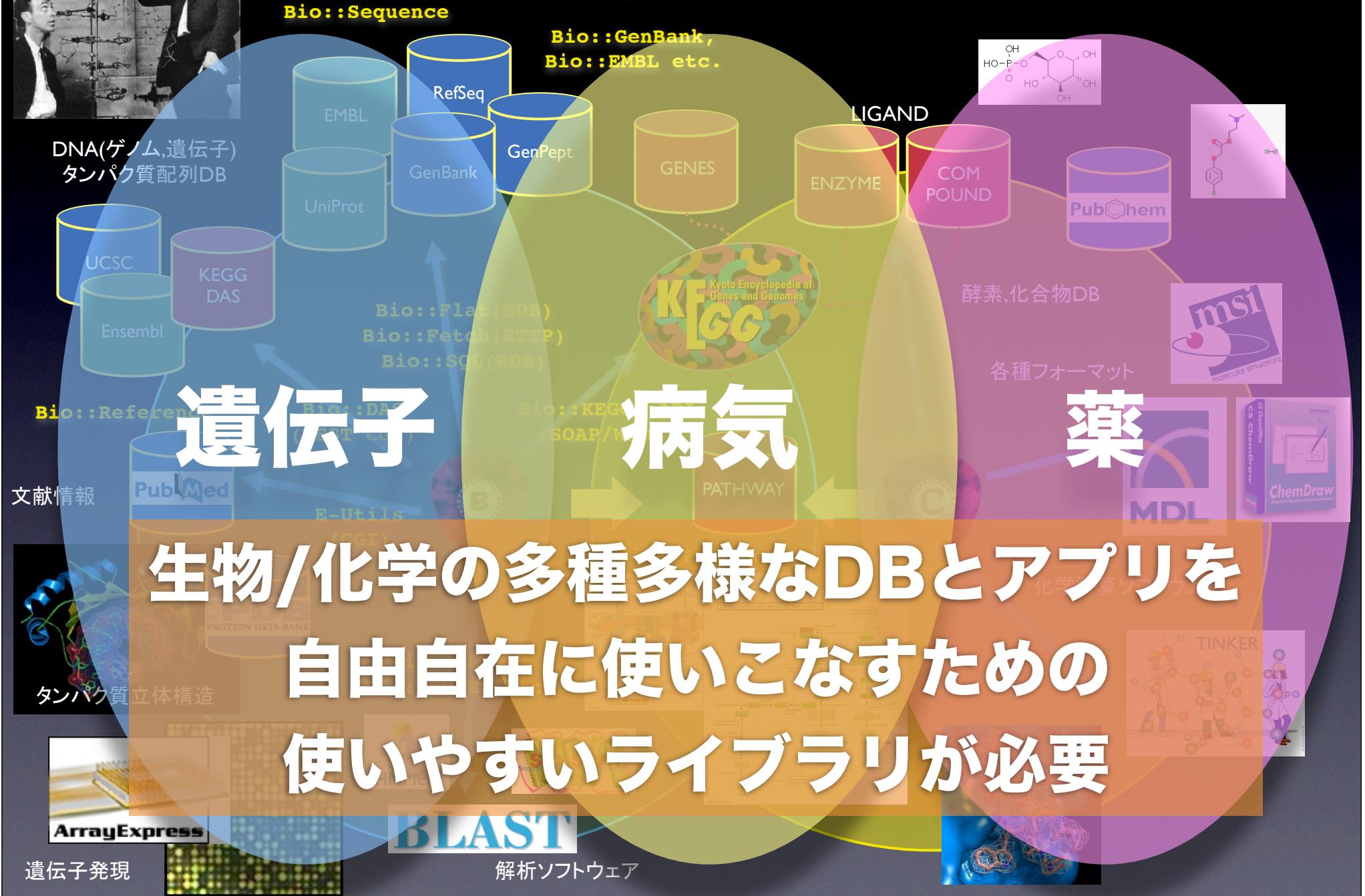
馬

*Caenorhabditis elegans*

線虫



# BioRuby + ChemRuby



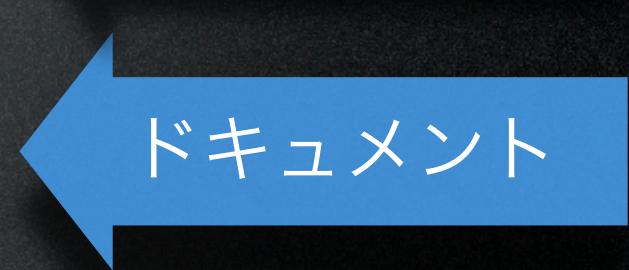
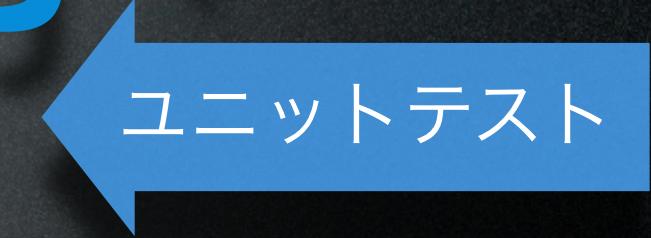
# 未踏による開発の進展

GUI: ウェブ (Rails)

CUI: BioRubyシェル

BioRuby/ChemRuby

Ruby : 文字列処理, 正規表現, 入出力, Test::Unit, SOAP



新規開発

ユニットテスト

ドキュメント

# ライブラリとシェル (irbベースのCUI)

- 生物屋 (≠プログラマ) でも利用できる?
- 怠惰指向 (Net::HTTP vs open-uri)

TAMWTDI vs TIOOWTDI

too

```
#!/usr/bin/env ruby

require 'bio'

# 文字列から生成する場合
dna = Bio::Sequence::NA.new("atgc")

# KEGG API を使う場合
kegg = Bio::KEGG::API.new
entry = kegg.bget("hsa:217")
gene = Bio::KEGG::GENES.new(entry)
dna = gene.naseq

# OBDA BioFetch を使う場合
obda = Bio::Fetch.new
entry = obda.fetch("gb:AF237819")
gb = Bio::GenBank.new(entry)
dna = gb.naseq
```

# EMBOSs を使う場合

```
entry = Bio::EMBOSS.entret("gb:AF237819")
gb = Bio::GenBank.new(entry)
dna = gb.naseq
```

# FlatFile を使う場合

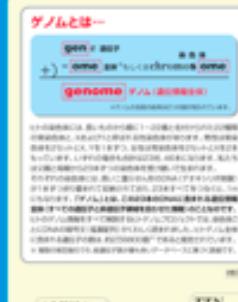
```
Bio::FlatFile.auto("file.txt") do |ff|
  ff.each do |entry|
    dna = entry.naseq
  end
end
```

% bioruby

```
bioruby> dna = seq("atgc")
bioruby> dna = seq("hsa:217")
bioruby> dna = seq("gb:AF237819")
bioruby> flatfile("file.txt") do |entry|
bioruby+   dna = seq(entry)
bioruby+ end
```

# GENOME MAP

ヒトゲノムマップ



AI	アレル	遺伝子	機能
アレル	アレル	アレル	アレル
アレル	アレル	アレル	アレル
アレル	アレル	アレル	アレル
アレル	アレル	アレル	アレル

Number.  
**12**  
1億4200万bp  
1268個

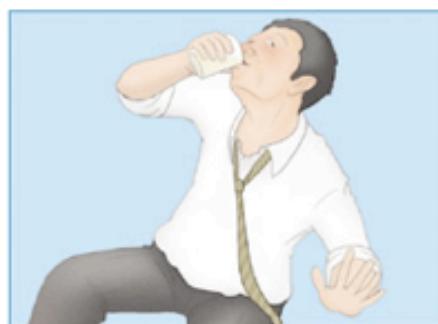


遺伝子名

通称名

*ALDH2*

アルデヒド分解酵素2



◎アルコールから生成される有毒なアセトアルデヒドを無毒な酢酸に変える酵素。

◎お酒に弱い人は、この酵素のはたらきが弱い。

同等の遺伝子（オーソログ）を持つ生物



# ALDH2 の変異

対立遺伝子 - Wikipedia

<http://ja.wikipedia.org/wiki/対立遺伝子>

Google

ログインまたはアカウント作成

本文 ノート 編集 履歴

## 対立遺伝子

出典: フリー百科事典『ウィキペディア (Wikipedia)』

相同的な遺伝子座を占める遺伝子の種類が複数ある場合、これらを対立遺伝子（たいりついでんし、allele）と呼ぶ。平易に言い換えると、対立遺伝子とはある遺伝子の種類のことである。

対立遺伝子の内、正常な機能をするものを野生型 (wild type) という。対立遺伝子は突然変異によって生じる。突然変異を二種類にわけると、機能欠失型変異型 (loss of function) と機能獲得型 (gain of function) がある。前者には完全に機能を失った amorph と、部分的に機能を失った hypomorph がある。後者には野生型の機能を妨げるように働く antimorph と、全く新たな機能を獲得した neomorph がある。

通常の生物は、父母由来の2つの対立遺伝子を持ち、両親から同じ遺伝子を引き継いでいる場合、ホモ接合とよばれ、異なる遺伝子を引き継いでいる場合、ヘテロ接合と呼ばれる。

例 [編集]

ヒトのアルコール代謝経路ではALDH2というアルデヒドデヒドロゲナーゼが重要な働きをしている。ALDH2にはALDH2\*1とALDH2\*2が存在することが知られ、その違いは第12染色体にあるALDH2遺伝子のエクソン12の変異に由来している。487番目のアミノ酸コドンがGAA（グルタミン酸）ならALDH2\*1ができ、AAA（リシン）ではALDH2\*2が作られる。このときALDH2遺伝子にはALDH2\*1とALDH2\*2の2つの対立遺伝子があるといえる。ALDH2\*1対立遺伝子から作られる酵素は活性が高く、俗に酒に強い遺伝子と呼ばれている。ALDH2\*2は活性が弱いため、この対立遺伝子を両親から受け継いだ人（ALDH2\*2のホモ接合型）は非常にアルコールに弱くなる。

一方、ヒトのアルデヒド脱水素酵素はさらにALDH1が知られるが、これは異なった遺伝子座（第9染色体）にあり、ALDH2とALDH1は別の対立遺伝子とは呼ばれない。

ヘルプ

ヘルプ  
井戸端  
連絡先  
バグの報告  
寄付

検索

表示 検索

ツールボックス

リンク元  
リンク先の更新状況

- 遺伝子の配列を取得
- 翻訳とか配列操作
- パスウェイにマップ
- 遺伝子音楽に

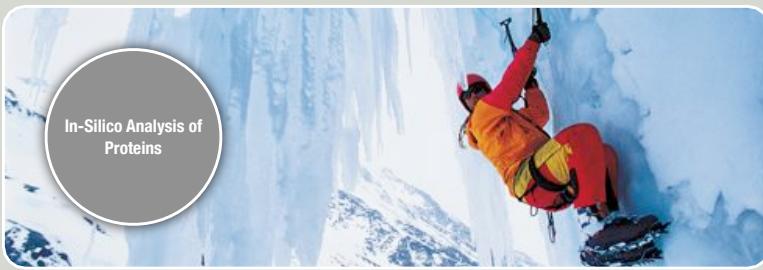
# デモのストーリー

# <http://bioruby.org/>

**Swiss-Prot 20 Years**

# BIORUBY

**Fortaleza:**  
Celebrating the  
20th Anniversary  
of Swiss-Prot  
Jul 30 - Aug 04  
2006 / Brazil



In-Silico Analysis of Proteins

## SAMURAI SWORD V.S. SWISSKNIFE

Mitsuteru Nakao, Naohisa Goto, Nobuya Tanaka, Toshiaki Katayama

We have been developing an open source library to utilize various life science databases and bioinformatics applications. The BioRuby library is written in an object oriented scripting language Ruby, and by its nature the code becomes very clean and efficient.

Currently, BioRuby contains parsers for over 30 biological databases including Swiss-Prot and also extensible for other database formats.

We will also represent our recent development including a new user interface, which is intended to be useful for biological researchers to utilize the power of BioRuby as their daily tool.

In this presentation, we will describe our framework to develop parsers and a benchmark focused on parsing of the Swiss-Prot database.

Functionality of the BioRuby is not limited to database parsing but covers biological sequence manipulation, execution of bioinformatics applications, 3D structures, biological pathways, and interfaces for various web services.

We will also represent our recent development including a new user interface, which is intended to be useful for biological researchers to utilize the power of BioRuby as their daily tool.

As one of the Open Bio\* projects hosted by Open Bio Foundation, we have been developing the BioRuby, a Ruby library for bioinformatics. In this February, we have released the BioRuby version 1.0 coming with various new features, bug fixes, unit tests and documentations. In this talk, I will describe what we have achieved with the 1.0 release and our future plans.

The Open Bio\* libraries have been successful as toolkits to develop customized bioinformatics pipelines, however, it was still difficult for the biologist to utilize the libraries as their daily tool. There can be two main reasons, (1) learning a programming language is a burden for them, (2) there are "too" many ways to do it. Thus, we included the BioRuby shell, a newly developed CLI (command line user interface) for the BioRuby library. BioRuby shell integrates and abstracts various ways of entry retrieval, flatfile processing, and accessing of web services, even with enabling users to utilize all functionality of Ruby and BioRuby without writing any script file. In the shell, objects and the history are conserved across the sessions, and a script to reproduce the procedure can be automatically generated.

Other enhancements in BioRuby 1.0 include unit tests and documentations. Holman started to add unit tests for some essential classes in BioRuby and Nakao has completed >1000 tests to make our library stable. We also added a English guideline to contribute our projects, and thanks to it, several developers joined to our project. Aerts and Reaum have been worked on the documentation format specification (RDoc format) and adding detailed API documentations. Goto translated Japanese tutorial and Prins improved the English version. Wernblom launched bioruby-doc.org site to develop further BioRuby documentations.

**BioRuby**  
<http://bioruby.org/>  
**Swiss-Prot 20th**  
<http://swissprot20.org/>



**BIORUBY 1.0 & THE BIORUBY SHELL**  
Toshiaki Katayama

August 4-5, 2006  
BOSC2006  
Fortaleza, Brazil

[http://open-bio.org/wiki/BOSC\\_2006](http://open-bio.org/wiki/BOSC_2006)  
<http://bioruby.org/>

As one of the Open Bio\* projects hosted by Open Bio Foundation, we have been developing the BioRuby, a Ruby library for bioinformatics. In this February, we have released the BioRuby version 1.0 coming with various new features, bug fixes, unit tests and documentations. In this talk, I will describe what we have achieved with the 1.0 release and our future plans.

The Open Bio\* libraries have been successful as toolkits to develop customized bioinformatics pipelines, however, it was still difficult for the biologist to utilize the libraries as their daily tool. There can be two main reasons, (1) learning a pro-

gramming language is a burden for them, (2) there are "too" many ways to do it. Thus, we included the BioRuby shell, a newly developed CLI (command line user interface) for the BioRuby library. BioRuby shell integrates and abstracts various ways of entry retrieval, flatfile processing, and accessing of web services, even with enabling users to utilize all functionality of Ruby and BioRuby without writing any script file. In the shell, objects and the history are conserved across the sessions, and a script to reproduce the procedure can be automatically generated. Other enhancements in BioRuby 1.0 include unit tests and documentations. Holman started to add unit tests for some essential classes in BioRuby and Nakao has completed >1000 tests to make our library stable. We also added a English guideline to contribute our projects, and thanks to it, several developers joined to our project. Aerts and Reaum have been worked on the documentation format specification (RDoc format) and adding detailed API documentations. Goto translated Japanese tutorial and Prins improved the English version. Wernblom launched bioruby-doc.org site to develop further BioRuby documentations.