# BioRuby

大阪大学遺伝情報実験センター
ゲノム情報解析分野

後藤　直久

2005年3月11日

# Rubyとは？

- オブジェクト指向スクリプト言語
- http://www.ruby-lang.org/
- 日本で開発され、海外にも普及したプログラム言語
  - 作者: まつもとゆきひろ氏
- Perlとの類似点
  - テキスト（文字列）処理が得意
  - スクリプト言語（コンパイル不要）
- Javaとの類似点
  - オブジェクト指向

# Rubyを選択した理由

- オブジェクト指向
  - データ構造を容易に記述できる
  - データとデータに対する処理を一括管理可能
    →構造化されたデータが数多く存在する生物学分野では特に有用
- 簡潔な文法
  - 書きやすく読みやすい
  - 開発効率が高い
- 日本で誕生し海外にも普及した言語

# BioRuby

- 2000/11/21 BioRubyプロジェクト開始
- 2001/06/21 バージョン0.1をリリース
- …（この間, リリース18回，学会発表8回など）
- 2004/12/13 バージョン0.62をリリース
- 現在
  - ファイル数:　130以上
  - 行数:　37,000行以上
  - 開発者:　累計 9人以上(うち海外2人)

# BioRubyの機能(1)

## 基本的なデータ構造・アルゴリズム

- 塩基・アミノ酸配列 (Bio::Sequence)
  - 部分配列の切り出し・翻訳など
- 配列上の位置情報 (Bio::Locations)
- アノテーション (Bio::Features)
- マルチプルアライメント (Bio::Alignment)
- 二項関係 (Bio::Relation)
- パスウェイ (Bio::Pathway)
- 文献情報 (Bio::References)
- …

# BioRubyの機能(2)

## データベース等のデータフォーマット対応

- FASTA形式 (Bio::FastaFormat)
- GenBank/DDBJ (Bio::GenBank)
- EMBL (Bio::EMBL)
- SwissProt/TrEMBL (Bio::SPTR)
- PIR(NBRF形式) (Bio::NBRF)
- PDB (Bio::PDB)
- PROSITE (Bio::PROSITE)
- KEGG (Bio::KEGG::*)
- TRANSFAC (Bio::TRANSFAC)
- FANTOM (Bio::FANTOM)
- MEDLINE (Bio::MEDLINE)
- Gene Ontology (Bio::GO)
- 他、合計約26種類のデータ形式に対応

# BioRubyの機能(3)

## 解析ソフトウェアの結果処理

- BLAST (Bio::Blast)
- FASTA (Bio::Fasta)
- HMMER (Bio::HMMER)
- CLUSTAL W (Bio::ClustalW)
- MAFFT (Bio::MAFFT)
- sim4 (Bio::Sim4)
- BLAT (Bio::BLAT)
- Spidey (Bio;;Spidey)
- GenScan (Bio::GenScan)
- PSORT (Bio::PSORT)
- TarrgetP (Bio::TargetP)
- SOSUI (Bio::SOSUI)
- TMHMM (Bio::TMHMM)
- 他、合計約15種類の解析ソフトウェアに対応

# BioRubyの機能(4)

## ファイルやネットワーク経由のデータ入出力

- Bio::FlatFile
- Bio::FlatFileIndex
- Bio::Fetch
- Bio::SQL
- Bio::Registry
- Bio::DAS
- Bio::KEGG::API
- Bio::DDBJ::XML
- Bio::PubMed
- …

# Bio::PDBクラス
# （PDBパーサ）について

# Bio::PDB

- PDBフォーマットのパーサ
- 2003年9月30日新規追加
  - 仕様書を見ながら作った
  - 最低限のパース機能のみ
- 2004年2月頃
  - Alex Gutteridge氏から意欲的なパッチ&機能拡張が届く
  - さっそく採用(2004/3/2)
  - 規模が大きくなったのでファイル分割

# Bio::PDBの特徴

- Bio::FlatFile によるファイル形式自動判別に対応
  - ent = Bio::FlatFile.auto('pdb1a00.ent') { |f| f.next_entry }
- 遅延評価のようなことをしている
  - 必要になったときに必要な部分だけをパース
- 原子の座標データはもちろん簡単に得られる
- REMARK レコードの情報も漏らさず得られる

# Bio::PDBサンプル

```
# ファイルから読み込む
ent = Bio::FlatFile.auto("pdb1a00.ent") { |f| f.next_entry }
# ID と Definition
puts ent.entry_id;  puts ent.definition
# REMARK
p ent.remark
# 原子の座標
ent.each_atom { |a| p a.xyz }
# プロリンについてだけ原子の座標を表示
ent.each_residue { |r|
        r.each_atom { |a| p a.xyz } if r.resName == "PRO"
}
# 他に each_model, each_chain もある
```

# ホモロジー検索ソフトウェアおよびmRNAマッピングソフトウェアのパーサについて

# 「検索＆ペアワイズアライメント」という共通点を持つソフト群

- BLAST (Bio::Blast::Report 他)
- FASTA (Bio::Fasta::Report)
- HMMER (Bio::HMMER::Report)
- Spidey (Bio::Spidey::Report)
- Sim4 (Bio::Sim4::Report)
- BLAT (Bio::Blat::Report)

APIもできるだけ共通にすると便利

# データ構造とクラス

- XXXXX::Report
  - 単一のクエリに対する結果全体
  - 複数の XXXXX::Hit をインスタンスとして持つ
- XXXXX::Hit
  - 単一のヒット
  - 複数の XXXXX::Hsp をインスタンスとして持つ
- XXXXX::Hsp (またはHSP, SegmentPair)
  - 配列断片の位置やアライメントなどの情報
  - 結果の最小単位

# BLASTの例

```
BLASTN 2.2.6 [Apr-09-2003]


Reference: Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer,
Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997),
"Gapped BLAST and PSI-BLAST: a new generation of protein database search
programs",  Nucleic Acids Res. 25:3389-3402.

Query= ri|0610005A07|R000001A15|1277 contigs=2 ver=1 seqid=2
         (1277 letters)

Database: fantom2.00.seq
           60,770 sequences; 119,956,725 total letters

Searching.................................................done
```

バージョン

Reference

Queryの情報

データベースの情報

```
                                                                  Score    E

Sequences producing significant

ri|0610005A07|R000001A15|1277 c
ri|0610039M06|R000004L05|1061 c
ri|4930431E11|PX00030N13|1181 c
ri|1110004G14|R000015H01|1462 c
ri|1700124M20|ZX00096C11|926 co
ri|2900019E12|ZX00083B15|841 co
ri|0610033N11|R000004G20|840 co
ri|9430011C20|PX00107J21|1874 c
ri|B830049N13|PX00073P19|1106 c

>ri|0610005A07|R000001A15|1277
          Length = 1277

 Score = 2531 bits (1277), Expect = 0.0
 Identities = 1277/1277 (100%)
 Strand = Plus / Plus


Query: 1    gggcagctctctgaacagccaaggctagattgacactgagcctgtccgttcagacctcgg 60
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct: 1    gggcagctctctgaacagccaaggctagattgacactgagcctgtccgttcagacctcgg 60

~~~~~~~~~~~~~~~~~~~~~~~~(中略)~~~~~~~~~~~~~~~~~~~~~~~~

>ri|1110004G14|R000015H01|1462 contigs=2 ver=1 seqid=1271
          Length = 1462

 Score =  297 bits (150), Expect = 3e-79
 Identities = 207/226 (91%)
```

HSP

High-Scoring Segment Pair の略。

BLASTによる相同性検索結果の最小単位

HSP

Hit

~~~~~~~~~~~~~~~~~~~~~~~~(中略)~~~~~~~~~~~~~~~~~~~~~~~~

```
>ri|1110004G14|R000015H01|1462 contigs=2 ver=1 seqid=1271
          Length = 1462
```

```
 Score =  297 bits (150), Expect = 3e-79
 Identities = 207/226 (91%)
 Strand = Plus / Plus


Query: 113 attcgcctgttcctggaatacacagactcaagctatgaggagaagagatacaccatgggt 172
           ||||| ||| |||| |||||||||| |||||||||||| ||||||||||||||||||||
Sbjct: 29  attcggctgctcctagaatacacaggctcaagctatgaagagaagagatacaccatggga 88


Query: 173 gatgctcctgactatgaccaaagccagtggctgaatgagaaattcaagctgggcctggac 232
           || |||||||||||||| |||||||||||||| |||||| |||||  ||||||||||||
Sbjct: 89  gacgctcctgactatgaccgaagccagtggctgagtgagaagttcaaattgggcctggac 148


Query: 233 tttcctaacctgccctacttgatcgatgggtcacacaagatcacgcagagcaatgccatc 292
           ||||| ||  |||| |||||||| ||||||||||||||||||||||||||||||||||
Sbjct: 149 tttcccaatttgccttacttgattgatgggtcacacaagatcacgcagagcaatgccatc 208


Query: 293 ctgcgctaccttggccgcaagcacaacctgtgtggggagacagagg 338
           ||||||||| ||| ||||||||||||||||||||||||||||||
Sbjct: 209 ctgcgctacattgcccgcaagcacaacctgtgtggggagacagagg 254
```

HSP

```
 Score = 93.7 bits (47), Expect = 1e-17
 Identities = 110/131 (83%)
 Strand = Plus / Plus


Query: 583 gtgcctggatgcgttcccaaacctgaaggacttcatagcgcgctttgagggcctgaagaa 642
           ||||||||| || |||||||||||||||||||  | || ||||||||||  |||||||
Sbjct: 499 gtgcctggacgccttcccaaacctgaaggactttgtggcccgctttgaggtactgaagag 558


Query: 643 gatctccgactacatgaagaccagtcgcttcctcccaagacccatgttcacaaagatggc 702
           |||||| |  |||||||||||||| |||||||||| || |||| | |  |||||| ||||
Sbjct: 559 gatctctgcttacatgaagaccagccgcttcctccgaacacccctatatacaaaggtggc 618


Query: 703 aacttggggca 713
            ||||||||||
Sbjct: 619 cacttggggca 629
```

HSP

```
 Score = 56.0 bits (28), Expect = 2e-06
 Identities = 106/132 (80%)
 Strand = Plus / Plus
```

Hit
~~~~~~~~~~~~~~~~~~~~~~~~

```
Query: 419 gactttgagaagctgaagccagggtacctggagcaactccctggaatgatgaggctttac 478
           |||||||||||| ||||||   |  ||| ||||||| |||||||||||| ||| ||| | |
Sbjct: 335 gactttgagaaactgaaggtggaatacttggagcagctccctggaatggtgaagctcttc 394


Query: 479 tctgagttcctgggcaagcggccatggttcgcaggggacaagatcacctttgtggatttc 538
           ||  |||||||||||| ||||| ||||||| |  || || ||||| || ||||| ||||||
Sbjct: 395 tcacagttcctgggccagcggacatggtttgttggtgaaaagattactttgtagatttc 454


Query: 539 attgcttacgat 550
           | |||||||||
Sbjct: 455 ctggcttacgat 466
```

HSP

~~~~~~~~~~~~~~~~~~~~~~~~(中略)~~~~~~~~~~~~~~~~~~~~~~~~

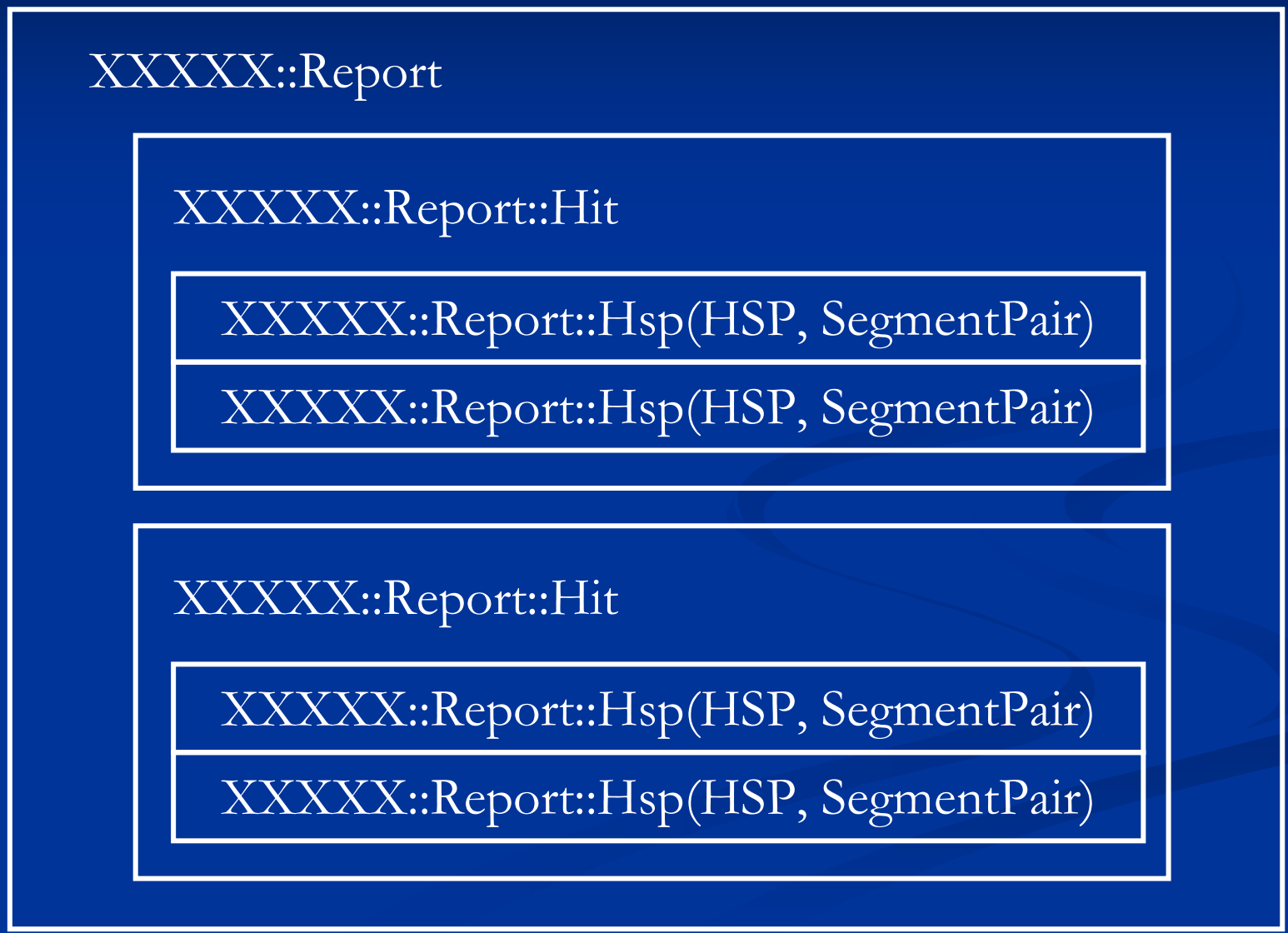
```
  Database: fantom2.00.seq
    Posted date:  Dec 7, 2003  4:50 PM
  Number of letters in database: 119,956,725
  Number of sequences in database:  60,770

Lambda     K      H
    1.37    0.711     1.31

Gapped
Lambda     K      H
    1.37    0.711     1.31


Matrix: blastn matrix:1 -3
Gap Penalties: Existence: 5, Extension: 2
Number of Hits to DB: 107,501
Number of Sequences: 60770
Number of extensions: 107501
Number of successful extensions: 2506
Number of sequences better than 1.0e-01: 9
Number of HSP's better than  0.1 without gapping: 9
Number of HSP's successfully gapped in prelim test: 0
Number of HSP's that attempted gapping in prelim test: 2471
Number of HSP's gapped (non-prelim): 31
length of query: 1277
length of database: 119,956,725
effective HSP length: 19
effective length of query: 1258
effective length of database: 118,802,095
effective search space: 149453035510
effective search space used: 149453035510
T: 0
A: 0
X1: 6 (11.9 bits)
X2: 15 (29.7 bits)
S1: 12 (24.3 bits)
S2: 21 (42.1 bits)
```

統計情報など
~~~~~~~~~~~~~~~~~~~~~~~~

# BLAST結果の構造

- Bio::Blast::Default::Report
  - 単一クエリに対する結果を格納
- Bio::Blast::Default::Report::Iteration
  - PSI-BLASTの繰り返し
  - PSI-BLAST以外では意識する必要なし
- Bio::Blast::Default::Report::Hit
- Bio::Blast::Default::Report::HSP

# 基本構造


XXXXX::Report
XXXXX::Report::Hit
XXXXX::Report::Hsp(HSP, SegmentPair)
XXXXX::Report::Hsp(HSP, SegmentPair)
XXXXX::Report::Hit
XXXXX::Report::Hsp(HSP, SegmentPair)
XXXXX::Report::Hsp(HSP, SegmentPair)

# BLAST

- 3種類の出力形式をサポート
- パーサに使用するクラスが異なる
  - XML(-m 7), タブ区切り(-m 8)
    - Bio::BLAST::Report クラス
    - Bio::FlatFile 非対応
  - デフォルト出力(-m 0)
    - Bio::Blast::Default::Report クラス
    - Bio::FlatFile 対応

# WU-BLAST

- ワシントン大学でNCBIとは別に開発

- Bio::Blast::WU::Report
- Bio::Blast::WU::Iteration
- Bio::Blast::WU::Hit
- Bio::Blast::WU::HSP

- Bio::FlatFile 対応

# FASTA (SSEARCH)

- -m 10 オプションの出力のみサポート
- Bio::Fasta::Report
- Bio::Fasta::Report::Hit
- HSPクラスは存在しない

- Bio::FlatFile非対応

# HMMER

- Bio::HMMER::Report
- Bio::HMMER::Report::Hit
- Bio::HMMER::Report::HSP

- Bio::FlatFile非対応

# Spidey

- NCBIのmRNA-ゲノムアライメントソフト
  - スプライシングを意識(GT-AGなど)
- 内部でBLASTアルゴリズムを使用？

- Bio::FlatFile対応（自動判別には未対応）

- Bio::Spidey::Report
- Bio::Spidey::Report::Hit
- Bio::Spidey::Report::SegmentPair

# sim4

- mRNA-ゲノム間マッピングソフト
  - スプライシングを意識(GT-AGなど)
- Bio::FlatFile対応
  - 自動判別には未対応(CVS最新版にて対応)
- Bio::Sim4::Report
- Bio::Sim4::Report::Hit
- Bio::Sim4::Report::SegmentPair

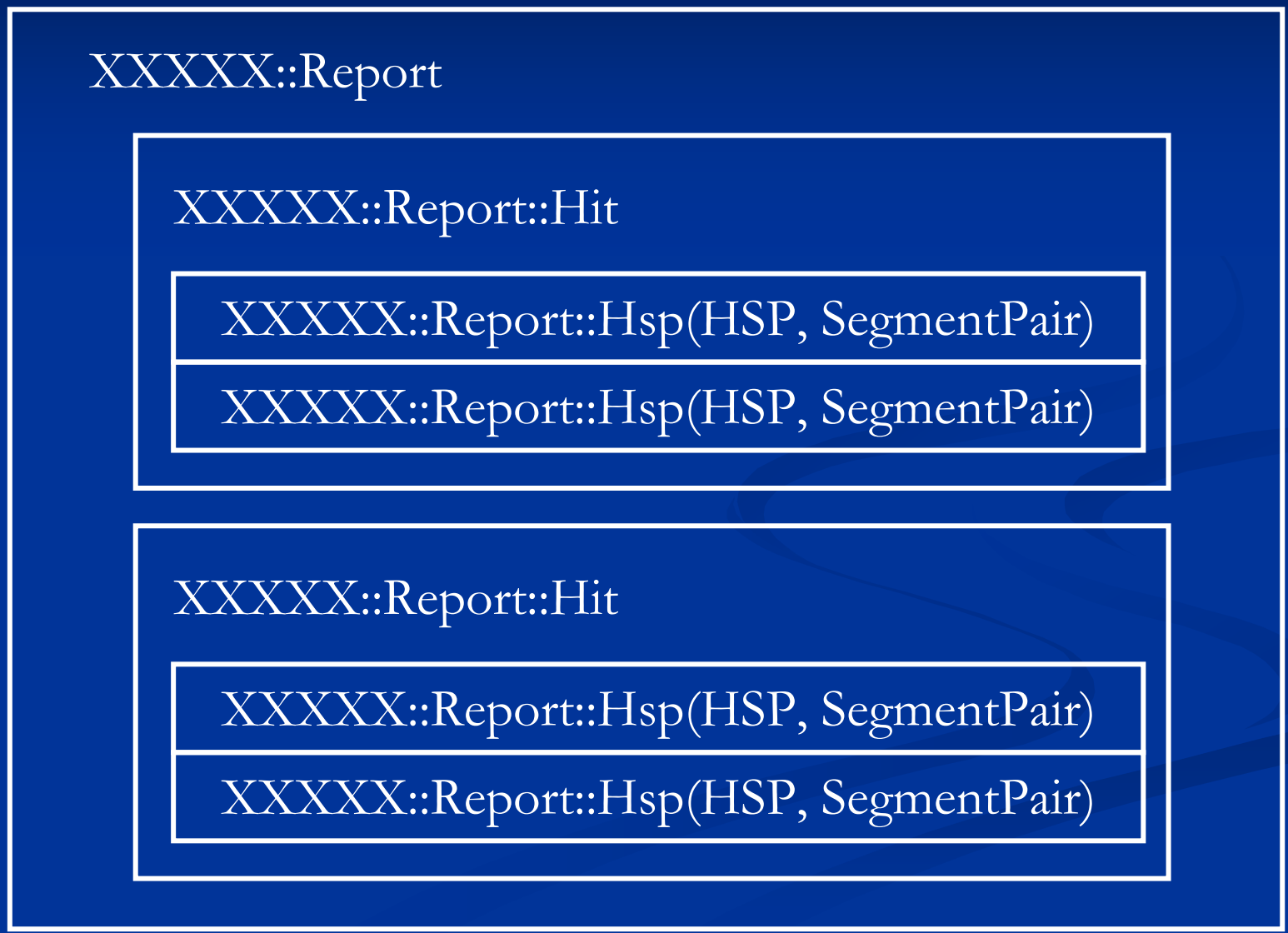
# BLAT

- 高速配列検索ツール
- スプライシングを考慮(GT-AGなど)

- Bio::FlatFile対応
  - 自動判別には未対応(CVS最新版で対応)
- 他のソフトと異なり、配列の1塩基目が0
  - BioRubyでは他との一貫性を保つため+1している

- Bio::Blat::Report
- Bio::Blat::Report::Hit
- Bio::Blat::Report::SegmentPair

# 基本構造


XXXXX::Report
XXXXX::Report::Hit
XXXXX::Report::Hsp(HSP, SegmentPair)
XXXXX::Report::Hsp(HSP, SegmentPair)
XXXXX::Report::Hit
XXXXX::Report::Hsp(HSP, SegmentPair)
XXXXX::Report::Hsp(HSP, SegmentPair)

# 共通API

- rep = XXXXX::Report.new(str)
  - 文字列strをパースする
- ff = Bio::FlatFile.auto( “filename”)
  - ファイルを読み込む
    - （ファイル形式自動判別対応の場合）
    - 未対応なら Bio::FlatFile.open(XXXXX::Report, “filename”)
  - その後は
    - rep = ff.next_entry
    - ff.each { |rep| … }
  - 等としてエントリ単位で読み込む

# 共通API --- Reportクラス

- rep は XXXXX::Report のインスタンス
  - rep.query_def, rep.query_len
    - クエリの定義（名前やIDなど）, クエリ配列の長さ
  - rep.hits
    - ヒットを格納したArray
  - rep.each { |hit| ... }
  - または rep.each_hit { |hit| ... }
    - ヒット1個ずつに対して処理を行う

# 共通API --- Hitクラス

- hit は XXXXX::Hit のインスタンス
  - hit.definition または hit.target_def
    - ヒットした配列の定義（名前やIDなど）
  - hit.target_len
    - ヒットした配列の長さ
  - hit.hsps
    - Hsp, HSP または SegmentPair を格納したArray
  - hit.each { |hsp| … }
    - Hsp1個ずつに対して処理を行う

# 共通API – Hspクラス

- 注: Hspクラスの名称はクラスにより微妙に異なる
  - Bio::Blast::Report::Hsp
  - Bio::Blast::Default::Report::HSP
  - Bio::Blast::WU::Report::HSP
  - Bio::Fasta::Report には Hspクラスが存在しない
  - Bio::HMMER::Report::Hsp
  - Bio::Spidey::Report::SegmentPair
  - Bio::Sim4::Report::SegmentPair
  - Bio::Blat::Report::SegmentPair

# 共通API --- Hspクラス

- hsp は XXXXX::Hspクラスのインスタンス
  - hsp.query_from, hsp.query_to, hsp.qseq
    - クエリ上の開始位置, 終了位置, 配列データ
  - hsp.hit_from, hsp.hit_to, hsp.hseq
    - ヒット上の開始位置, 終了位置, 配列データ
  - hsp.query_strand, hsp.hit_strand
    - クエリおよびヒットの鎖(strand)の方向
  - hsp.midline
    - アライメントの状況を示した行
  - hsp.align_len
    - アライメント長

```
ATCCATGC
||  |||||
ATGCATGC
```

# 現状はまだ不完全です

- 存在しないメソッドもある
  - 順次揃えていきたい
- 情報の記述方法がバラバラな場合がある
  - query_strand, hit_strand など
    - 現状では原則として対象ソフトの記述を可能な限りそのまま使用している
- アプリケーション実行支援の方法がばらばら
  - オプションや入出力データの与え方がソフトによって全然違うため困難だが何とかしたい

# 今後の課題

- ドキュメントやサンプルの整備
- UnitTest
- 対応データベース・ソフトウェアの拡大
- より使いやすくするための改良・機能拡張
- リファクタリング
- BioRubyを使用したソフトウェアの開発
- …

# Acknowledgements


- BioRuby Developers
  - Toshiaki Katayama
  - Mitsuteru Nakao
  - Yoshinori Okuji
  - Shuichi Kawashima
  - Masumi Itoh
  - Alex Gutteridge
  - Moses Hohman
  - and some other contributors on the internet.

http://bioruby.org/