できる！ BioPerl

infobiologist: 第二回研究集会(2003)＠遺伝研

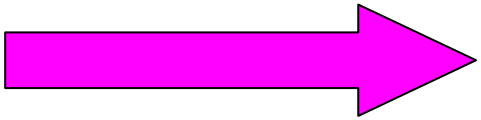
2003/1/28

大浦智紀

タカラバイオ株式会社

自己紹介

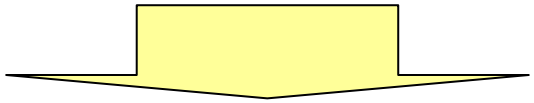
- タカラバイオ株式会社
- DNAチップ（1999年ころから）
  - 受託製造。
  - 自社製品企画。（コンテンツの準備）
  - プローブ設計。
  - 製造管理。
  - アノテーション。



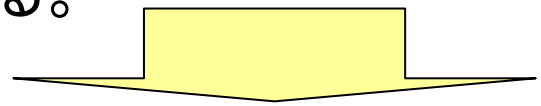
Perlとの必然的な出会い。

Perlの導入

- 3000個の塩基配列のそれぞれに、PCR用プライマーを設計する。
- 3000個の遺伝子名のそれぞれについて、データベースを検索する。
- 市販アプリケーションでは難しい。



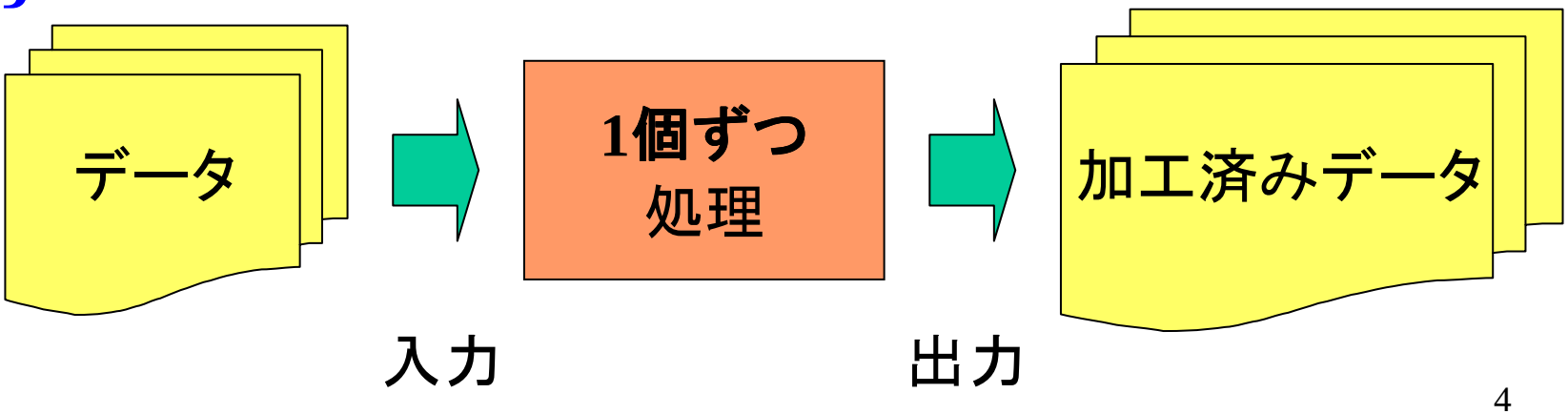
- 頼りになったPerl言語。しかし、自作プログラムを作るのは時間がかかる。



- bioperlを導入してみよう。

プログラムの基本形の構造

```
while ([読み込み]) {
  [処理]
}
```



Perlプログラムの基本形

```
#!perl
while($line = <>) {
    print $line;
}
```

sample01.pl

Perlプログラムの基本形

```
#!perl
#sample01.pl
while($line = <>) {#１行を$lineに読み込む。
   print $line;#$lineを出力する
}
#whileは読み込み動作が真である場合ループ処理
   を続ける。
```

sample01.pl

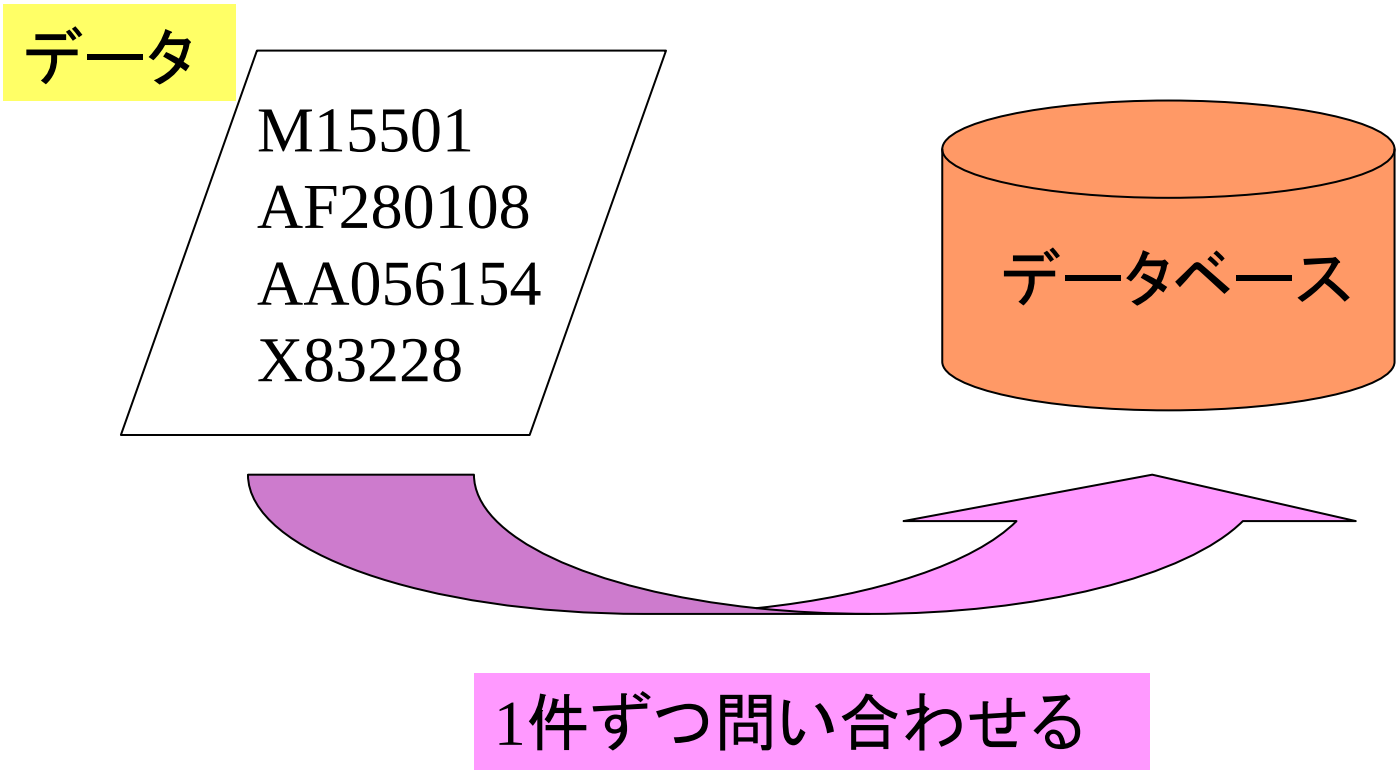
perl SAMPLE01.PL < DATA.TXT

Bioperlとは

- すでに誰かが問題を解決している。
- データの形式やデータの取り扱い方法がパッケージとして提供されている。

http://bioperl.org

課題：データベースからデータ取得



データベースからデータ取得

```
#!perl
use Bio::DB::GenBank;
$gb = new Bio::DB::GenBank();
while($line = <>) {
  chomp $line;
  $seq = $gb->get_Seq_by_acc($line);
}
```

sample02.pl

データ取得準備

```
#!perl
use Bio::DB::GenBank;
$gb = new Bio::DB::GenBank();
while($line = <>) {
  chomp $line;
  $seq = $gb->get_Seq_by_acc($line);
}
```

データ取得と内容表示

```
#!perl
use Bio::DB::GenBank;
$gb = new Bio::DB::GenBank();
while($line = <>) {
  chomp $line;
  $seq = $gb->get_Seq_by_acc($line);
  print $seq->desc,"¥n";
}#while
```

sample03.pl

データ取得と内容表示2 (1 of 2)

```
#!perl
use Bio::DB::GenBank;
$gb = new Bio::DB::GenBank();
while($line = <>) {
  chomp $line;
  $seq = $gb->get_Seq_by_acc($line);
```

sample04.pl

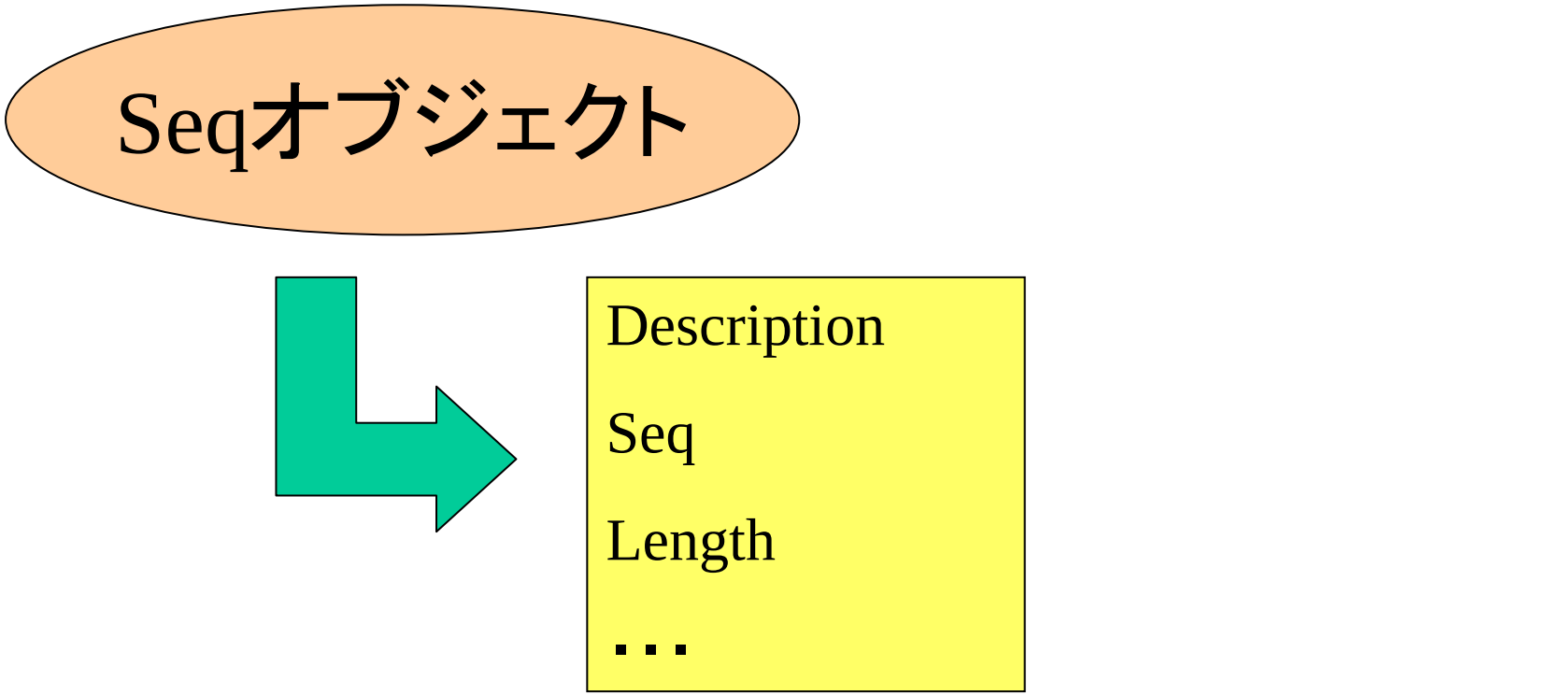
データ取得と内容表示2 (2 of 2)

```
    print $seq->desc,"¥n";
    print $seq->seq,"¥n";
    print $seq->length,"¥n";
    print $seq->primary_id,"¥n";
    print $seq->id,"¥n";
    print '--¥n';
}#while
```

Seq
オブジェクトの内容表示

sample04.pl

$seqって何だ？



複数のSeqオブジェクト

SAMPLE04_local.plの出力。

desc:Mouse alpha-cardiac actin mRNA, 3' end.
length:1226
prim
id:M

desc:Homo sapiens clone 15e cytochrome P450 subfamily IIIA
polype
length
primar
id:AF

desc:Homo sapiens clone 15e cytochrome P450 subfamily IIIA
polypeptide 43 (CYP3A43) mRNA, complete cds.
length:1512
primary_id:11225237
id:AF280108

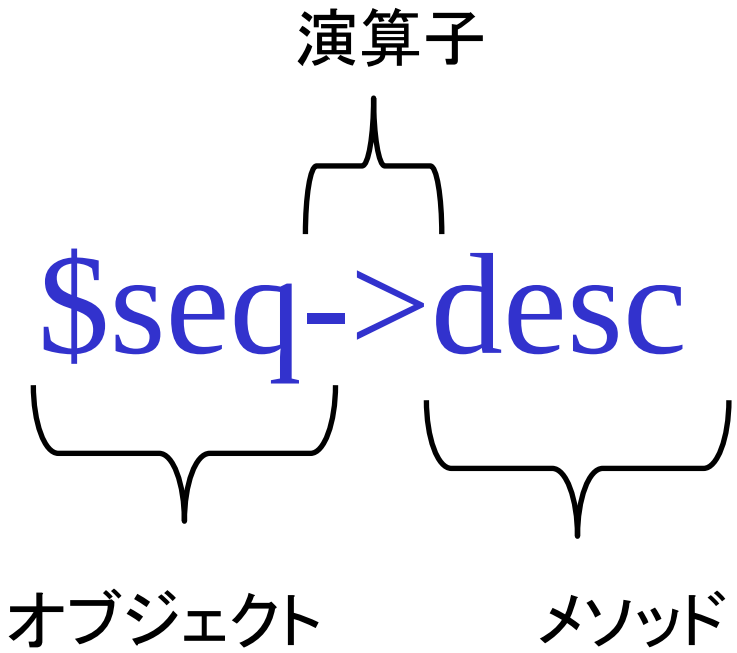
データ取得と内容表示2 (2 of 2)

```
    print $seq->desc,"¥n";
    print $seq->seq,"¥n";
    print $seq->length,"¥n";
    print $seq->primary_id,"¥n";
    print $seq->id,"¥n";
    print '--¥n';
}#while
```

Seq
オブジェクトの内容表示

sample04.pl

オブジェクトを操作する



descriptionを出せ。

操作する手段がメソッド

オブジェクトとメソッド

- メソッドの主な使い方は3つ
  - Seqオブジェクトを作る。
  - Seqオブジェクトを見る。
  - Seqオブジェクトの中身を変える。

メソッドの使い方 (1)

オブジェクトを自前で作る

```
$seq = Bio::Seq->new( -display_id => 'my_id',
-seq => ‘ATGCCGGTA’);
```

オブジェクトを外から読みこんで作る

```
$seqio = Bio::SeqIO->new( ‘-format’ => ‘embl’ , -file =>
‘myfile.dat’); #読み込み準備
$seq = $seqio->next_seq();
```

メソッドの使い方 (2)

オブジェクトの中身を見る

```
print $seq->desc,"¥n";
```

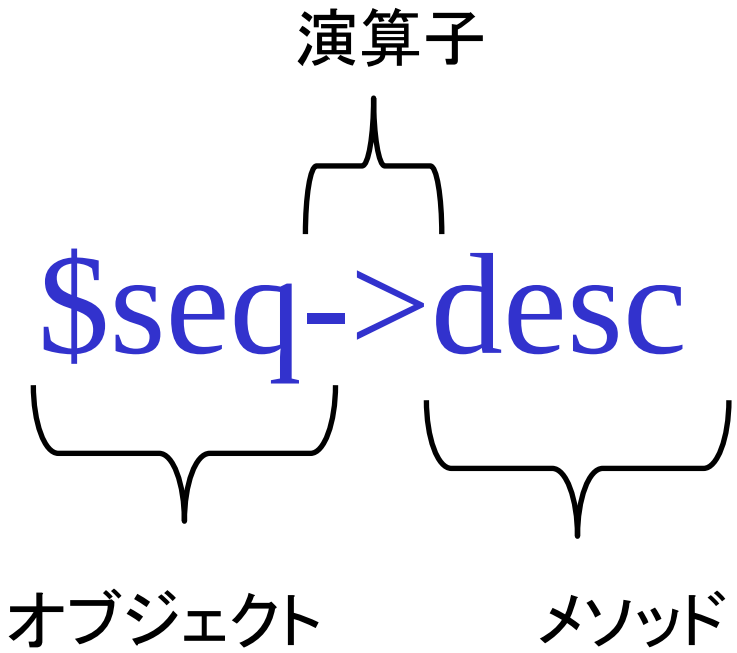
メソッドの使い方 (3)

オブジェクトの中身を変える

```
print $seq->desc,"¥n";

$seq->desc( 'hogehoge' );

print $seq->desc,"¥n";
```

SAMPLE05_local.pl

オブジェクトを操作する



descriptionを出せ。

操作する手段がメソッド

オブジェクトの操作

```
#!perl
use Bio::DB::GenBank;
$gb = new Bio::DB::GenBank();
while($line = <>) {
  chomp $line;
  $seq = $gb->get_Seq_by_acc($line);
  print $seq->desc,"¥n";
}#while
```

オブジェクトとメソッドのまとめ

- メソッドの主な使い方は3つ
  - Seqオブジェクトを作る。
  - Seqオブジェクトを見る。
  - Seqオブジェクトの中身を変える。

Seqオブジェクトのメソッド

Methods		
new	Description	Code
seq	Description	Code
validate_seq	Description	Code
length	Description	Code
subseq	Description	Code
display_id	Description	Code
accession_number	Description	Code
desc	Description	Code
primary_id	Description	Code
can_call_new	Description	Code
alphabet	Description	Code
object_id	Description	Code
version	Description	Code
authority	Description	Code
namespace	Description	Code

$seq->desc

$seq->seq

$seq->length

$seq->accession_number

$seq->primary_id

...

http://doc.bioperl.org/releases/bioperl-1.2/

クラスって何？（用語の整理）

- Seqクラスのオブジェクトが，Seqオブジェクトである。
- Bioperlは，バイオインフォマティクス用クラスのライブラリ。
- クラスの一覧表，使えるメソッド。
  - http://doc.bioperl.org/releases/bioperl-1.2/

Perlのお約束

```
#!perl
use Bio::DB::GenBank;
$gb = new Bio::DB::GenBank();
while($line = <>) {
  chomp $line;
  $seq = $gb->get_Seq_by_acc($line);
  print $seq->desc,"¥n";
}
```

Perlのお約束

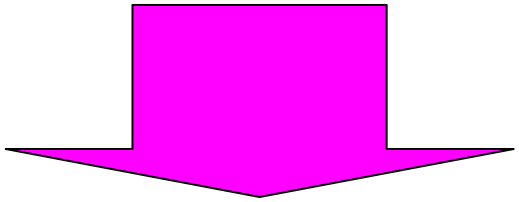
```
#!perl -w
use strict;
use vars qw($gb);
use Bio::DB::GenBank;
$gb = new Bio::DB::GenBank();
while( defined (my $line = <>) ) {
   chomp $line;
   my $seq = $gb->get_Seq_by_acc($line);
   print $seq->desc,"¥n";
}
exit;
```

お約束の意味

```
#!perl –w #1回しか出現しない変数などいろいろとチェック。
use strict; #構文を制限
use vars qw($gb); #グローバル変数
use Bio::DB::GenBank;
$gb = new Bio::DB::GenBank();
while( defined(my $line = <>) ) { #ローカル変数, 入力確認
   chomp $line;
   my $seq = $gb->get_Seq_by_acc($line); #ローカル変数
   print $seq->desc,"¥n";
}
exit; #プログラムの終了
```

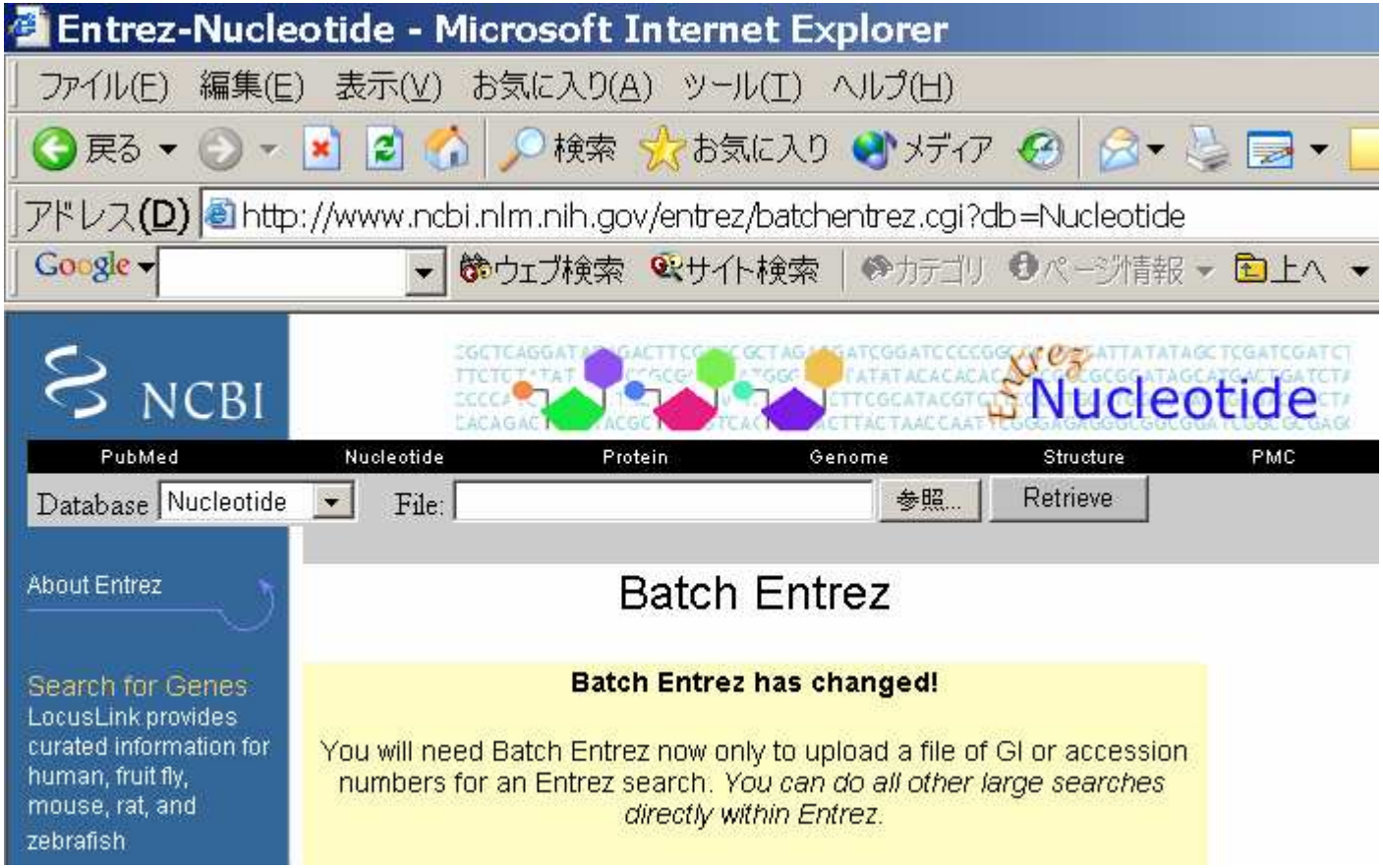
なぜ約束があるの？

- スペルミスのチェックを自動的にしてもらえる。
- 文法チェック。
- 変数のグローバル／ローカル。



- プログラムの規模が大きくなるのに備える。

Batch Entrez



何でもPerlでやらなくてよい。

パーサを作る

- パーサって何？
  - Parser，Parseする。
  - 構文解析。
- SeqIOクラス
  - フォーマット変換ツール
  - GenBankパーサ
  - BLASTパーサ
  - ○○パーサ

見ていきます。

GenBank -> Fasta変換プログラム

```
#!perl -w
use strict;
use Bio::SeqIO;
use vars qw($in $out);
$in = Bio::SeqIO->new(
   -fh => ¥*STDIN , -format => 'GenBank ' );
$out = Bio::SeqIO->new(
   -fh => ¥*STDOUT , -format =>  'Fasta'  );
while(my $seq = $in->next_seq) {
   $out->write_seq($seq);
}
exit;
```

SeqIO
オブジェクト作成

sample06_gb2fasta.pl

SeqIOオブジェクト

```
$in = Bio::SeqIO->new(
  -fh => ¥*STDIN , -format =>
  'GenBank ‘ );
$out = Bio::SeqIO->new(
  -fh => ¥*STDOUT , -format =>
   ‘Fasta’  );
```

ファイルハンドル:

標準入力, 標準出力

GenBank -> Fasta 出力部分

```
#!perl -w
use strict;
use Bio::SeqIO;
use vars qw($in $out);
$in = Bio::SeqIO->new(
   -fh => ¥*STDIN , -format => 'GenBank ‘ );
$out = Bio::SeqIO->new(
   -fh => ¥*STDOUT , -format =>  ‘Fasta’  );
while(my $seq = $in->next_seq) {
   $out->write_seq($seq);
}
exit;
```

sample06_gb2fasta.pl

GenBank flatfileのパース (1 of 2)

```
LOCUS       AA056154                582 bp    mRNA    linear   EST 02-FEB-1997
DEFINITION  zf55e12.r1 Soares retina N2b4HR Homo sapiens cDNA clone
            IMAGE:380878 5' similar to gb:K02281_cds1 RHODOPSIN (HUMAN);, mRNA
            sequence.
ACCESSION   AA056154
VERSION     AA056154.1  GI:1548492
KEYWORDS    EST.
SOURCE      Homo sapiens (human).
  ORGANISM  Homo sapiens
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
            Mammalia; Eutheria; Primates; Catarrhini; Hominidae; Homo.
REFERENCE   1  (bases 1 to 582)
  AUTHORS   Hillier,L., Lennon,G., Becker,M., Bonaldo,M.F., Chiapelli,B.
  TITLE     Generation and analysis of 280,000 human expressed sequence tags
  JOURNAL   Genome Res. 6 (9), 807-828 (1996)
  MEDLINE   97044478
   PUBMED   8889549
COMMENT     Contact: Wilson RK Washington University School of Medicine 4444
            Forest Park Parkway, Box 8501, St. Louis, MO 63108 Tel: 314 286
            ...
FEATURES             Location/Qualifiers
     source          1..582
                     /dev_stage="55 year old"
                     /tissue_type="retina"
```

GenBankのパース (2 of 2)

```
COMMENT     Original source text: Mouse cardiac muscle, cDNA to mRNA, clone
            pmC1. Draft entry and computer-readable sequence for [1] kindly
            provided by D.P.Leader, 27-MAY-1987.
FEATURES             Location/Qualifiers
     source          1..1226
                     /organism="Mus musculus"
                     /db_xref="taxon:10090"
     mRNA            <1..>1226
                     /note="actin mRNA [1]"
     CDS             <1..1128
                     /protein_id="AAA37167.1"
                     /codon_start=1
                     /translation="DDEETTALVCDNGSGLVKAGFAGDDAPRAVFPSIVGRPRHQGVM
                     VGMGQKDSYVGDEAQSKRGILTLKYPIEHGIITNWDDMEKIWHHTFYNELRVAPEEHP
                     TLLTEAPLNPKANREKMTQIMFETFNVPAMYVAIQAVLSLYASGRTTGIVLDSGDGVT
                     HNVPIYEGYALPHAIMRLDLAGRDLTDYLMKILTERGYSFVTTAEREIVRDIKEKLCY
                     VALDFENEMATAASSSSLEKSYELPDGQVITIGNERFRCPETLFQPSFIGMESAGIHE
                     TTYNSIMKCDIDIRKDLYANNVLSGGTTMYPGIADRMQKEITALAPSTMKIKIIAPPE
                     RKYSVWIGGSILASLSTFQQMWISKQEYDEAGPSIVHRKCF"
                     /db_xref="GI:387090"
                     /note="alpha-cardiac actin"
BASE COUNT      286 a    346 c    294 g    300 t
ORIGIN
```

GenBankパーサ

```
#!perl -w
use strict;
use Bio::SeqIO;
use vars qw($in $out);
$in = Bio::SeqIO->new(-fh => ¥*STDIN , -format =>
   'GenBank');
while(my $seq = $in->next_seq) {
   print 'display_id:',$seq->display_id,"¥n";
   print 'desc:',$seq->desc,"¥n";
   print 'accession:',$seq->accession_number,"¥n";
   print 'length:',$seq->length,"¥n";
   print "----¥n";
}
exit;
```

属性を表示する。

sample07_gb.pl

GenBankパーサ feature付き

```
#!perl -w
use strict;
use Bio::SeqIO;
use vars qw($in $out);
$in = Bio::SeqIO->new(-fh => ¥*STDIN , '-format' => 'GenBank');
while(my $seq = $in->next_seq) {
    print 'desc:',$seq->desc,"¥n";
    my @feature_array = $seq->get_SeqFeatures;
    foreach my $feat (@feature_array) {
          my $primary_tag = $feat -> primary_tag();
          my $start = $feat -> start;
          my $end = $feat -> end;
          print ("¥$primary_tag:$primary_tag,$start,$end¥n");
    }#for each feature
    print "----¥n";
}
exit;
```

属性を表示する。

sample08_gb_feat.pl

get_seqFeaturesメソッド

```
@feature_array = $seq->get_SeqFeatures;
```

オブジェクト

メソッド

全てのfeatureを出せ。

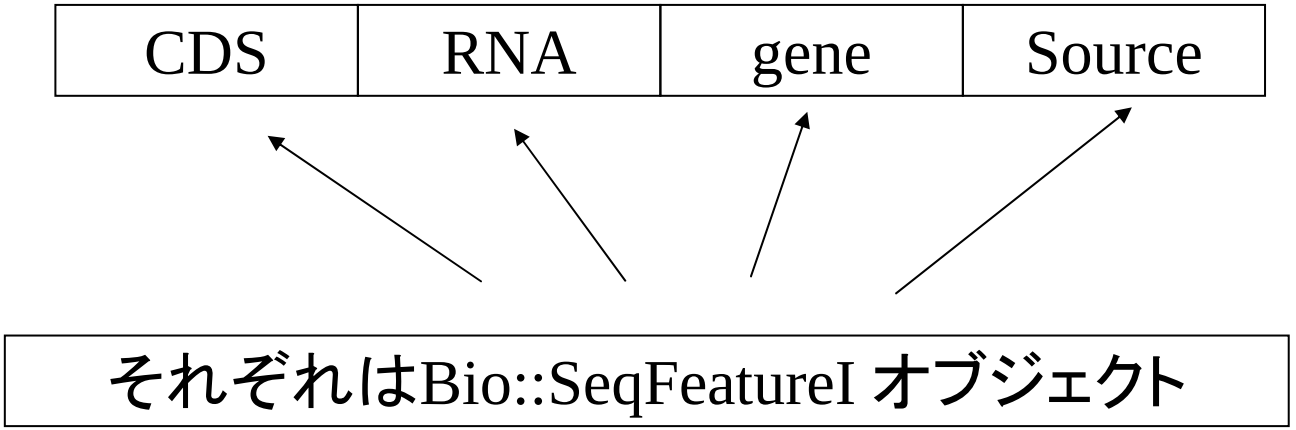
```
get_SeqFeatures            code                top                prev

 Title   : get_SeqFeatures
 Usage   :
 Function: Get the feature objects held by this feature holder.
           Features which are not top-level are subfeatures of one or
           more of the returned feature objects, which means that you
           must traverse the subfeature arrays of each top-level
           feature object in order to traverse all features associated
           with this sequence.
           Use get_all_SeqFeatures() if you want the feature tree
           flattened into one single array.
 Example :
 Returns : an array of Bio::SeqFeatureI implementing objects
 Args    : none
At some day we may want to expand this method to allow for a feature
filter to be passed in.
```

メソッドが何を返すのか。

配列とは

- 複数の要素を持つ変数。

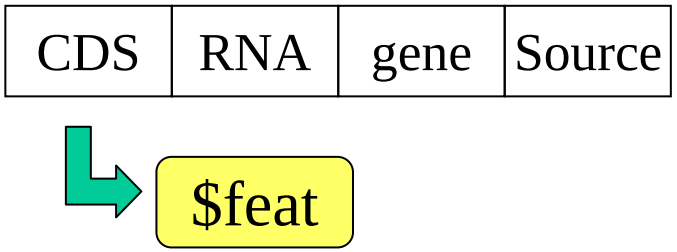
@feature_arrayのイメージ



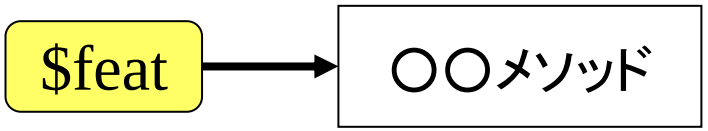
feature配列の中身にアクセス

```
foreach my $feat (@feature_array) {
    my $primary_tag = $feat->primary_tag();
    my $start = $feat->start;
    my $end = $feat->end;
    print ("¥$primary_tag:$primary_tag,$start,$end¥n");
}#for each feature
```

@feature_array



それぞれの$featに対して
メソッドで中身を見る。



GenBankパーサ feature付き

```
#!perl -w
use strict;
use Bio::SeqIO;
use vars qw($in $out);
$in = Bio::SeqIO->new(-fh => ¥*STDIN , '-format' => 'GenBank');
while(my $seq = $in->next_seq) {
    print 'desc:',$seq->desc,"¥n";
    my @feature_array = $seq->get_SeqFeatures;
    foreach my $feat (@feature_array) {
          my $primary_tag = $feat -> primary_tag();
          my $start = $feat -> start;
          my $end = $feat -> end;
          print ("¥$primary_tag:$primary_tag,$start,$end¥n");
    }#for each feature
    print "----¥n";
}
exit;
```

属性を表示する。

sample08_gb_feat.pl

GenBankのパースのまとめ (1)

```
LOCUS       AA056154                 582 bp    mRNA    linear   EST 02-FEB-1997
DEFINITION  zf55e12.r1 Soares retina N2b4HR Homo sapiens cDNA clone
            IMAGE:380878 5' similar to gb:K02281_cds1 RHODOPSIN (HUMAN);, mRNA
            sequence.
ACCESSION   AA056154
VERSION     AA056154.1  GI:1548492
KEYWORDS    EST.
SOURCE      Homo sapiens (human).
  ORGANISM  Homo sapiens
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
            Mammalia; Eutheria; Primates; Catarrhini; Hominidae; Homo.
REFERENCE   1  (bases 1 to 582)
  AUTHORS   Hillier,L., Lennon,G., Becker,M., Bonaldo,M.F., Chiapelli,B.
  TITLE     Generation and analysis of 280,000 human expressed sequence tags
  JOURNAL   Genome Res. 6 (9), 807-828 (1996)
  MEDLINE   97044478
   PUBMED   8889549
COMMENT     Contact: Wilson RK Washington University School of Medicine 4444
            Forest Park Parkway, Box 8501, St. Louis, MO 63108 Tel: 314 286
            ...
FEATURES             Location/Qualifiers
     source          1..582
                     /dev_stage="55 year old"
                     /tissue_type="retina"
```

値を取り出すメソッドが用意されている。

Bio::SeqFeatureI オブジェクトのメソッド(1)

- $feat -> primary_tag;
  - primary tagを返す。(CDS, mRNA, geneなどの文字列。)
- $feat -> start
  - featureのスタート位置を返す。
- $feat -> end
  - featureの終了位置を返す。

GenBankパーサ feature付きその2 (1 of 2)

```
#!perl -w
use strict;
use Bio::SeqIO;
use vars qw($in $out);
$in = Bio::SeqIO->new(-fh => ¥*STDIN , '-format' =>
   'GenBank');
while(my $seq = $in->next_seq) {
   print 'desc:',$seq->desc,"¥n";
   my @feature_array = $seq->all_SeqFeatures;
   foreach my $feat (@feature_array) {
        my $primary_tag = $feat -> primary_tag();
        my $start = $feat -> start;
        my $end = $feat -> end;
        print ("¥$primary_tag:$primary_tag,$start,$end¥n");
```

sample09_gb_feat.pl

GenBankパーサ feature付きその2 (2 of 2)

```
        foreach my $each_tag ($feat->get_all_tags()) {
                my @tag_values = $feat-
    >each_tag_value($each_tag);
                print ("¥$each_tag:$each_tag,");
                print (¥"@tag_values¥"¥n");
        }#for each tag
    }#for each feature
    print "----¥n";
}#while
exit;
```

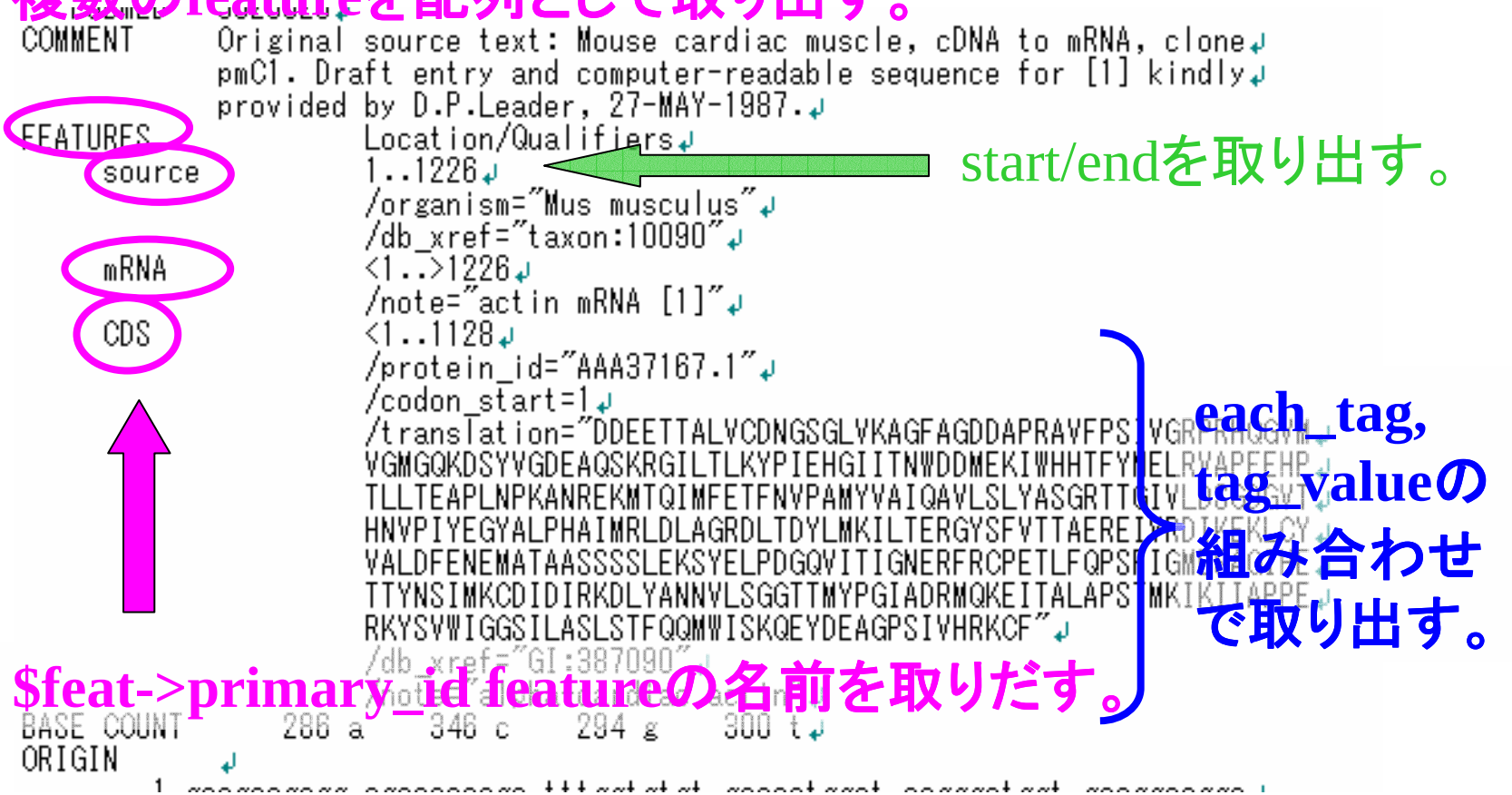
sample09_gb_feat.pl

Bio::SeqFeatureI オブジェクトのメソッド（２）

- $feat->get_all_tags
  - すべてのタグを配列で返す。
- $feat->each_tag_value($tag)
  - 指定したタグの値を配列で返す。

GenBankのパースのまとめ (その2)

複数のfeatureを配列として取り出す。



start/endを取り出す。

each_tag, tag_valueの組み合わせで取り出す。

$feat->primary_id featureの名前を取りだす。

本日のまとめ

- **While**を使って繰り返し処理を行う。
- **Bioperl**を使ってデータをダウンロードしたり、パースしたり、フォーマット変換したりできる。
- Bioperlを扱うということは**オブジェクト**を扱うということである。

本日のまとめ（細かいこと）

- 最も重要なクラスは、**Seq**, **SeqIO,** **SeqFeatureI**クラスである。
- オブジェクトは**メソッド**によって、（属性の値を）参照したり、変更したりできる。
- メソッドによって返されるものはいろいろ。
  - 値
  - オブジェクト
  - オブジェクトのリスト

プログラムを作ってみる

- 簡単なものから作る。
- 必要に応じて作る。
- あれば便利、を考える。
- とりあえずサンプルプログラムを拡張していく。
- Bioperlのオブジェクトは少しずつ、必要に応じて覚えていく。

サンプルプログラム

- bioperl-1.2/scripts
- bioperl-1.2/examples

パッケージを解凍したディレクトリに入っている。

チュートリアル

- bioperl-1.2/bptutorial.pl

  パッケージを解凍したディレクトリに入っている。

- http://www.pasteur.fr/recherche/unites/sis/formation/bioperl/

  パスツール研究所のチュートリアル。

より進んだ学習のために

- Perlを基礎から勉強したい。
- 複雑なデータ型を使いたい。
- オブジェクト指向をより深く理解したい。
- 仕事に必要な新しいクラスを自由に作りたい。
- 大量のデータを高速にアクセスしたい。

作業環境整備に

初心者でもわかる!
バイオインフォマティクス入門—
やさしいUNIX操作から遺伝子・
タンパク質解析まで
坊農 秀雅 (著)
出版社: 羊土社
ISBN: 489706290X

- bioperlのインストール
- 各種プログラムのインストール。

Perlの勉強

Perl言語プログラミングレッスン 入門編
結城 浩 (著)
出版社: ソフトバンクパブリッシング ;
ISBN: 4797312211

- プログラムの初心者から。
- 丁寧な解説。

Perlのリファレンス

プログラミングPerl〈VOLUME1〉
ラリー ウォール (著), ジョン オーワント (著), トム クリスチャンセン (著), 近藤 嘉雪 (翻訳)
出版社: オライリー・ジャパン ;
ISBN: 4873110963

- リファレンスとして。
- 初心者には言い回しが難しい。

Perlのプログラム事例集

Perlクックブック—Perlの鉄人が贈るレシピ集
トム クリスチャンセン (著), ネイザン トーキントン (著), 田和 勝 (翻訳)
出版社: オライリー・ジャパン ;
ISBN: 4873110378

- データベースアクセス。
- ほかのプログラムとの協調。
- CGI

複雑なデータ型を扱いたい

Effective Perl　ASCII Addison Wesley Programming Series
ジョセフ・N. ホール (著), ランドル・L. シュワォーツ (著), 吉川 邦夫 (翻訳)
出版社: アスキー ; ISBN: 4756130577

- 配列、ハッシュ。
- リファレンス、デリファレンス
- 配列の配列、ハッシュの配列。
- 複雑なソート。
- オブジェクト指向。

モジュールの使い方を知りたい

Perlモジュール活用ガイド―
かんたんオブジェクト指向プログラミング
エリック フォスター・ジョンソン (著), Eric Foster-Jonson (原著), アークシンクタンク (翻訳), 三島 俊司
出版社: 翔泳社
ISBN: 4881356682

- モジュールのインストール。
- 標準的なモジュールの使い方。

大量のデータを扱いたい

入門Perl DBI

アリゲータ デカルト (著), ティム バンス (著),
Alligator Descartes (原著), Tim Bunce (原著),
田中 幸 (翻訳)

出版社: オライリー・ジャパン

ISBN: 4873110505

- 大量のデータをディスクに保存して、後からアクセスしたい。
- データベースシステムをPerlから使いたい。
- データベースってどんなもの？