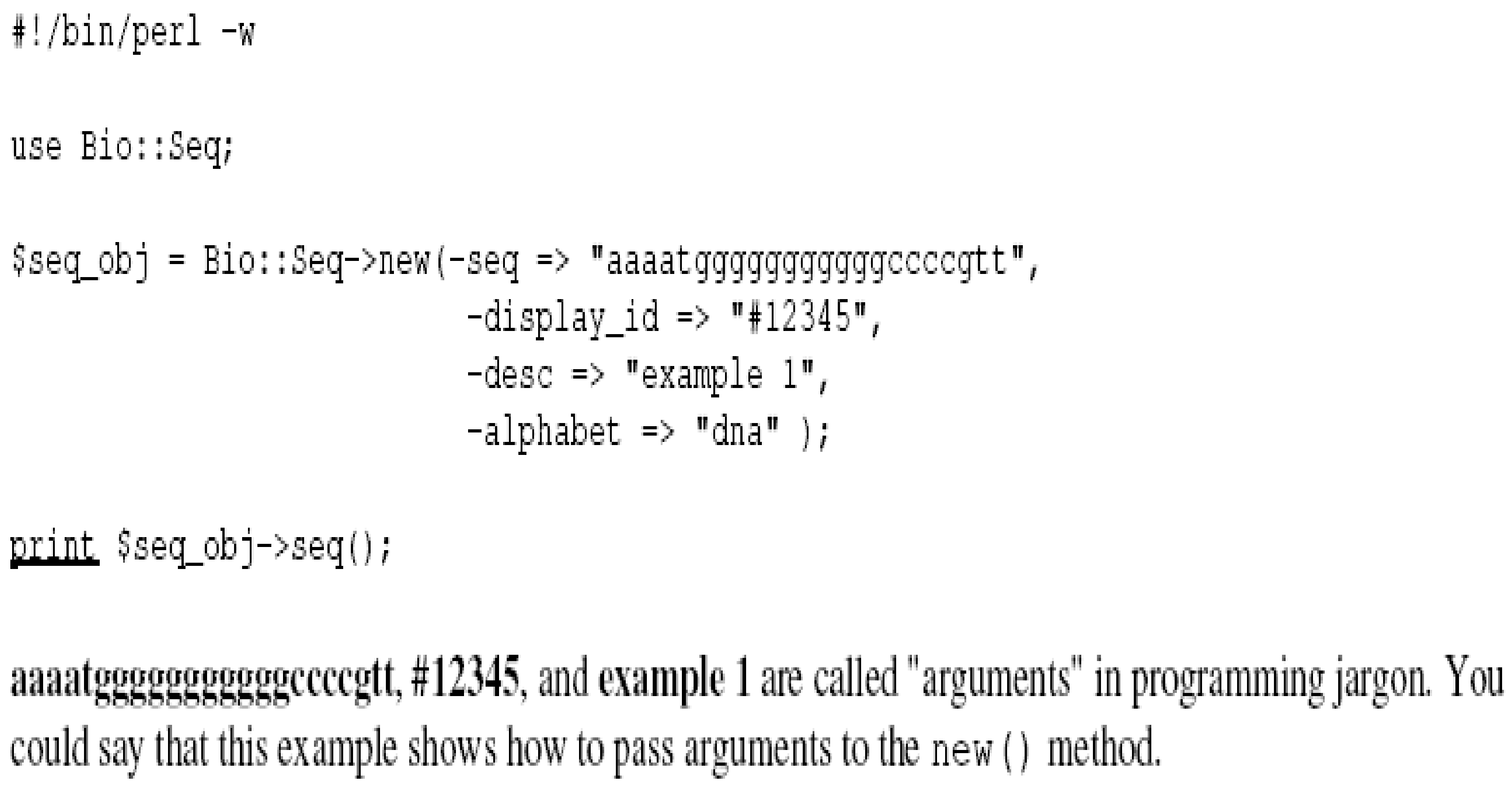
第十一章：BioPerl

生命科学学院

一个简单的程序

- #!/bin/perl -w
- use Bio::Seq;
- $seq_obj = Bio::Seq->new(-seq => ″aaaatgggggggggggccccgtt″, -alphabet => ’dna’);
- print $seq_obj->seq;

读



```
#!/bin/perl -w

use Bio::Seq;

$seq_obj = Bio::Seq->new(-seq => "aaaatgggggggggggccccgtt",
                         -display_id => "#12345",
                         -desc => "example 1",
                         -alphabet => "dna" );

print $seq_obj->seq();
```

aaaatgggggggggggccccgtt, **#12345**, and **example 1** are called "arguments" in programming jargon. You could say that this example shows how to pass arguments to the `new()` method.

写

```
#!/bin/perl -w

use Bio::Seq;
use Bio::SeqIO;

$seq_obj = Bio::Seq->new(-seq => "aaaatgggggggggggccccgtt",
                                             -display_id => "#12345",
                                             -desc => "example 1",
                                             -alphabet => "dna" );

$seqio_obj = Bio::SeqIO->new(-file => '>sequence.fasta', -format => 'fasta' );

$seqio_obj->write_seq($seq_obj);
```

读取一条

```
#!/bin/perl -w

use Bio::SeqIO;

$seqio_obj = Bio::SeqIO->new(-file => "sequence.fasta", -format => "fasta" );

$seq_obj = $seqio_obj->next_seq;
```

逐条读取

```
while ($seq_obj = $seqio_obj->next_seq){
    # print the sequence
    print $seq_obj->seq,"\n";
}
```

从网上读取

```
use Bio::DB::GenBank;

$db_obj = Bio::DB::GenBank->new;

$seq_obj = $db_obj->get_Seq_by_id(2);
```

从网上搜索

```
use Bio::DB::Query::GenBank;

$query = "Arabidopsis[ORGN] AND topoisomerase[TITL] and 0:3000[SLEN]";
$query_obj = Bio::DB::Query::GenBank->new(-db    => 'nucleotide', -query => $query );



$query_obj = Bio::DB::Query::GenBank->new(
                      -query   =>'gbdiv est[prop] AND Trypanosoma brucei [organism]',
                      -db      => 'nucleotide' );


use Bio::DB::GenBank;
use Bio::DB::Query::GenBank;

$query = "Arabidopsis[ORGN] AND topoisomerase[TITL] and 0:3000[SLEN]";
$query_obj = Bio::DB::Query::GenBank->new(-db    => 'nucleotide',  -query => $query );

$gb_obj = Bio::DB::GenBank->new;

$stream_obj = $gb_obj->get_Stream_by_query($query_obj);

while ($seq_obj = $stream_obj->next_seq) {
    # do something with the sequence object
    print $seq_obj->display_id, "\t", $seq_obj->length, "\n";
}
```

获取和设定

```
$sequence_as_string = $seq_obj->seq;
```

To set or assign a value:

```
$seq_obj->seq("MMTYDFFFFVVNNNNPPPPAAAW");
```

Table 1: Sequence Object Methods

Name	Returns	Example	Note
accession_number	identifier	$acc = $so->accession_number	get or set an identifier
alphabet	alphabet	$so->alphabet('dna')	get or set the alphabet ('dna','rna','protein')
authority	authority, if available	$so->authority("FlyBase")	get or set the organization
desc	description	$so->desc("Example 1")	get or set a description
display_id	identifier	$so->display_id("NP_123456")	get or set an identifier
division	division, if available (e.g. PRI)	$div = $so->division	get division (e.g. "PRI")
get_dates	array of dates, if available	@dates = $so->get_dates	get dates
get_secondary_accessions	array of secondary accessions, if	@accs = $so->get_secondary_accessions	get other identifiers

	available		
is_circular	Boolean	if $so->is_circular { # }	get or set
keywords	keywords, if available	@array = $so->keywords	get or set keywords
length	length, a number	$len = $so->length	get the length
molecule	molecule type, if available	$type = $so->molecule	get molecule (e.g. "RNA", "DNA")
namespace	namespace, if available	$so->namespace("Private")	get or set the name space
new	Sequence object	$so = Bio::Seq->new(-seq => "MPQRAS")	create a new one, see Bio::Seq for more
pid	pid, if available	$pid = $so->pid	get pid
primary_id	identifier	$so->primary_id(12345)	get or set an identifier
revcom	Sequence object	$so2 = $so1->revcom	Reverse complement
seq	sequence string	$seq = $so->seq	get or set the sequence
seq_version	version, if available	$so->seq_version("1")	get or set a version
species	Species object	$species_obj = $so->species	See Bio::Species for more
subseq	sequence string	$string = $seq_obj->subseq(10,40)	Arguments are start and end
translate	protein Sequence object	$prot_obj = $dna_obj->translate	
trunc	Sequence object	$so2 = $so1->trunc(10,40)	Arguments are start and end

Table 2: Feature and Annotation Methods

Name	Returns	Note
get_SeqFeatures	array of SeqFeature objects	
get_all_SeqFeatures	array of SeqFeature objects array	includes sub-features
remove_SeqFeatures	array of SeqFeatures removed	
feature_count	number of SeqFeature objects	
add_SeqFeature	annotation array of Annotation objects	get or set

Example Sequence Objects

Let's use some of the methods above and see what they return when the sequence object is obtained from different sources. In the Genbank example we're assuming we've used Genbank to retrieve or create a Sequence object. So this object could have have been retrieved like this:

```
use Bio::DB::GenBank;

$db_obj = Bio::DB::GenBank->new;
$seq_obj = $db_obj->get_Seq_by_acc("J01673");
```

Or it could have been created from a file like this:

```
use Bio::SeqIO;

$seqio_obj = Bio::SeqIO->new(-file => "J01673.gb", -format => "genbank" );
$seq_obj = $seqio_obj->next_seq;
```

What the Genbank file looks like:

```
LOCUS       ECORHO                  1880 bp    DNA     linear   BCT 26-APR-1993
DEFINITION  E.coli rho gene coding for transcription termination factor.
ACCESSION   J01673 J01674
VERSION     J01673.1  GI:147605
KEYWORDS    attenuator; leader peptide; rho gene; transcription terminator.
SOURCE      Escherichia coli
ORGANISM  Escherichia coli
                Bacteria; Proteobacteria; Gammaproteobacteria; Enterobacteriales;
                Enterobacteriaceae; Escherichia.
REFERENCE   1  (bases 1 to 1880)
AUTHORS   Brown,S., Albrechtsen,B., Pedersen,S. and Klemm,P.
TITLE     Localization and regulation of the structural gene for
             transcription-termination factor rho of Escherichia coli
JOURNAL   J. Mol. Biol. 162 (2), 283-298 (1982)
MEDLINE   83138788
PUBMED   6219230
REFERENCE   2  (bases 1 to 1880) AUTHORS   Pinkham,J.L. and Platt,T.
TITLE     The nucleotide sequence of the rho gene of E. coli K-12
JOURNAL   Nucleic Acids Res. 11 (11), 3531-3545 (1983)
MEDLINE   83220759
PUBMED   6304634
COMMENT      Original source text: Escherichia coli (strain K-12) DNA.
                     A clean copy of the sequence for [2] was kindly provided by
                     J.L.Pinkham and T.Platt.
FEATURES        Location/Qualifiers
     source      1..1880
                     /organism="Escherichia coli"
                     /mol_type="genomic DNA"
                     /strain="K-12"
                     /db_xref="taxon:562"
     mRNA       212..>1880
                     /product="rho mRNA"
     CDS             282..383
                        /note="rho operon leader peptide"
                        /codon_start=1
                        /transl_table=11
                        /protein_id="AAA24531.1"
                        /db_xref="GI:147606"
                        /translation="MRSEQISGSSLNPSCRFSSAYSPVTRQRKDMSR"
     gene            468..1727
                        /gene="rho"
     CDS             468..1727
                        /gene="rho"
                        /note="transcription termination factor"
                        /codon_start=1
                        /transl_table=11
                        /protein_id="AAA24532.1"
                        /db_xref="GI:147607"
                        /translation="MNLTELKNTPVSELITLGENMGLENLARMRKQDIIFAILKQHAK
                        SGEDIFGDGVLEILQDGFGFLRSADSSYLAGPDDIYVSPSQIRRFNLRTGDTISGKIR
                        PPKEGERYFALLKVNEVNFDKPENARNKILFENLTPLHANSRLRMERGNGSTEDLTAR
                        VLDLASPIGRGQRGLIVAPPKAGKTMLLQNIAQSIAYNHPDCVLMVLLIDERPEEVTE
                        MQRLVKGEVVASTFDEPASRHVQVAEMVIEKAKRLVEHKKDVIILLDSITRLARAYNT
                        VVPASGKVLTGGVDANALHRPKRFFGAARNVEEGGSLTIIATALIDTGSKMDEVIYEE
                        FKGTGNMELHLSRKIAEKRVFPAIDYNRSGTRKEELLTTQEELQKMWILRKIIHPMGE
                        IDAMEFLINKLAMTKTNDDFFEMMKRS"
ORIGIN      15 bp upstream from HhaI site.
        1 aaccctagca ctgcgccgaa atatggcatc cgtggtatcc cgactctgct gctgttcaaa
       61 aacggtgaag tggcggcaac caaagtgggt gcactgtcta aaggtcagtt gaaagagttc

                                  ...deleted...

     1801 tgggcatgtt aggaaaattc ctggaatttg ctggcatgtt atgcaatttg catatcaaat
     1861 ggttaatttt tgcacaggac
//
```

Table 3: Values from the Sequence object (Genbank)

Method	Returns
display_id	ECORHO
desc	E.coli rho gene coding for transcription termination factor.
display_name	ECORHO
accession	J01673
primary_id	147605
seq_version	1
keywords	attenuator; leader peptide; rho gene; transcription terminator
is_circular	
namespace	
authority	
length	1880
seq	AACCCT...ACAGGAC
division	BCT
molecule	DNA
get_dates	26-APR-1993
get_secondary_accessions	J01674

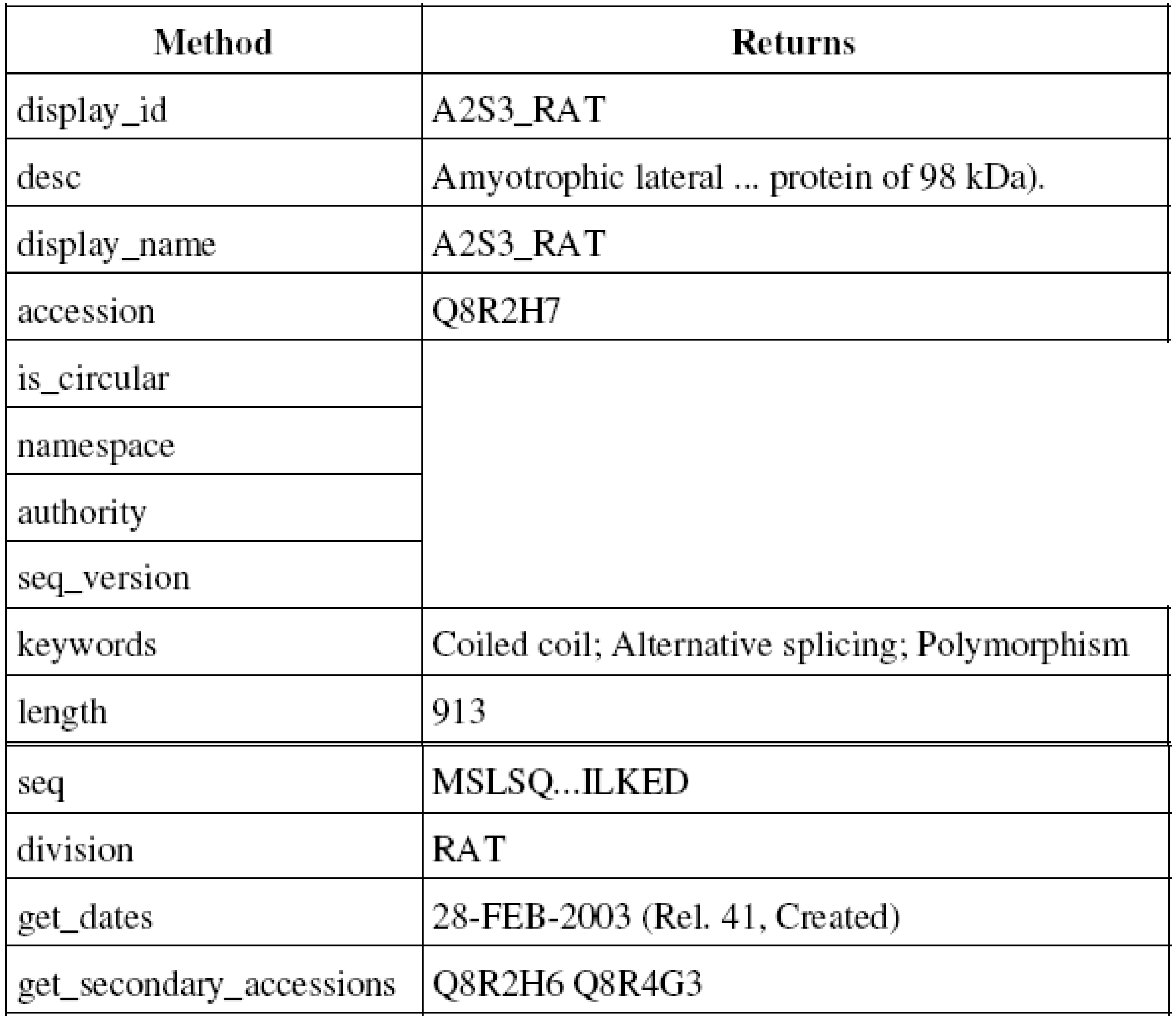
FASTA格式列子

```
>gi|147605|gb|J01673.1|ECORHO E.coli rho gene coding for transcription termination factor
AACCCTAGCACTGCGCCGAAATATGGCATCCGTGGTATCCCGACTCTGCTGCTGTTCAAAAACGGTGAAG
TGGCGGCAACCAAAGTGGGTGCACTGTCTAAAGGTCAGTTGAAAGAGTTCCTCGACGCTAACCTGGCGTA

                        ...deleted...

ACGTGTTTACGTGGCGTTTTGCTTTTATATCTGTAATCTTAATGCCGCGCTGGGCATGTTAGGAAAATTC
CTGGAATTTGCTGGCATGTTATGCAATTTGCATATCAAATGGTTAATTTTTGCACAGGAC
```

Table 4: Values from the Sequence object (Fasta)

Method	Returns
display_id	147605|gb|J01673.1|ECORHO
desc	E.coli rho gene coding for transcription termination factor
display_name	147605|gb|J01673.1|ECORHO
accession	unknown
primary_id	147605|gb|J01673.1|ECORHO
is_circular	
namespace	
authority	
length	1880
seq	AACCCT...ACAGGAC

Swissprot文件格式的调用参数



Method	Returns
display_id	A2S3_RAT
desc	Amyotrophic lateral ... protein of 98 kDa).
display_name	A2S3_RAT
accession	Q8R2H7
is_circular	
namespace	
authority	
seq_version	
keywords	Coiled coil; Alternative splicing; Polymorphism
length	913
seq	MSLSQ...ILKED
division	RAT
get_dates	28-FEB-2003 (Rel. 41, Created)
get_secondary_accessions	Q8R2H6 Q8R4G3

翻译序列

Any sequence object with alphabet 'dna' or 'rna' can be translated by simply using `translate` which will return a protein sequence object:

```
$prot_obj = $my_seq_object->translate;
```

However, the `translate()` method can also be passed several optional parameters to modify its behavior. For example, you can tell `translate()` to modify the characters used to represent terminator (default is *) and unknown amino acids (default is **X**).

```
$prot_obj = $my_seq_object->translate(-terminator => '-');
$prot_obj = $my_seq_object->translate(-unknown => '_');
```

You can also determine the frame of the translation. The default frame starts at the first nucleotide (frame 0). To get translation in the next frame we would write:

```
$prot_obj = $my_seq_object->translate(-frame => 1);
```

If we want to translate full coding regions (CDS) the way major nucleotide databanks EMBL, GenBank and DDBJ do it, the `translate()` method has to perform more checks. Specifically, `translate()` needs to confirm that the open reading frame has appropriate start and terminator codons at the very beginning and the very end of the sequence and that there are no terminator codons present within the sequence in frame 0. In addition, if the genetic code being used has an atypical (non-ATG) start codon, the `translate()` method needs to convert the initial amino acid to methionine. These checks and conversions are triggered by setting "complete" to 1:

```
$prot_obj = $my_seq_object->translate(-complete => 1);
```

If "complete" is set to true and the criteria for a proper CDS are not met, the method, by default, issues a warning. By setting "throw" to 1, one can instead instruct the program to die if an improper CDS is found, e.g.

```
$prot_obj = $my_seq_object->translate(-complete => 1,
                                      -throw => 1);
```

密码子表

The codontable_id argument to `translate()` makes it possible to use alternative genetic codes. There are currently 16 codon tables defined, including 'Standard', 'Vertebrate Mitochondrial', 'Bacterial', 'Alternative Yeast Nuclear' and 'Ciliate, Dasycladacean and Hexamita Nuclear'. All these tables can be seen in Bio::Tools::CodonTable. For example, for mitochondrial translation:

```
$prot_obj = $seq_obj->translate(-codontable_id => 2);
```

You can also create a custom codon table and pass this to `translate`, the code will look something like this:

```
use Bio::Tools::CodonTable;

@custom_table =
    ( 'test1',
      'FFLLSSSSYY**CC*WLLLL**PPHHQQR*RRIIIFT*TT*NKKSSRRV*VVAA*ADDEE*GGG'
    );

$codon_table = Bio::Tools::CodonTable->new;

$id = $codon_table->add_table(@custom_table);

$prot_obj = $my_seq_object->translate(-codontable_id => $id);
```

See Bio::Tools::CodonTable for information on the format of a codon table.

`translate()` can also find the open reading frame (ORF) starting at the 1st initiation codon in the nucleotide sequence, regardless of its frame, and translate that:

```
$prot_obj = $my_seq_object->translate(-orf => 1);
```

Most of the codon tables, including the default codon table NCBI "Standard", have initiation codons in addition to ATG. To tell `translate()` to use only ATG or atg as the initiation codon set -start to "atg":

```
$prot_obj = $my_seq_object->translate(-orf => 1,
                                      -start => "atg" );
```

The -start argument only applies when -orf is set to 1.

Last trick. By default `translate()` will translate the termination codon to some special character (the default is *, but this can be reset using the -terminator argument).

When -complete is set to 1 this character is removed. So, with this:

```
$prot_obj = $my_seq_object->translate(-orf => 1,
                                      -complete => 1);
```

the sequence **ttttatgccctaggggg** will be translated to **MP**, not **MP***.

获得序列基本信息

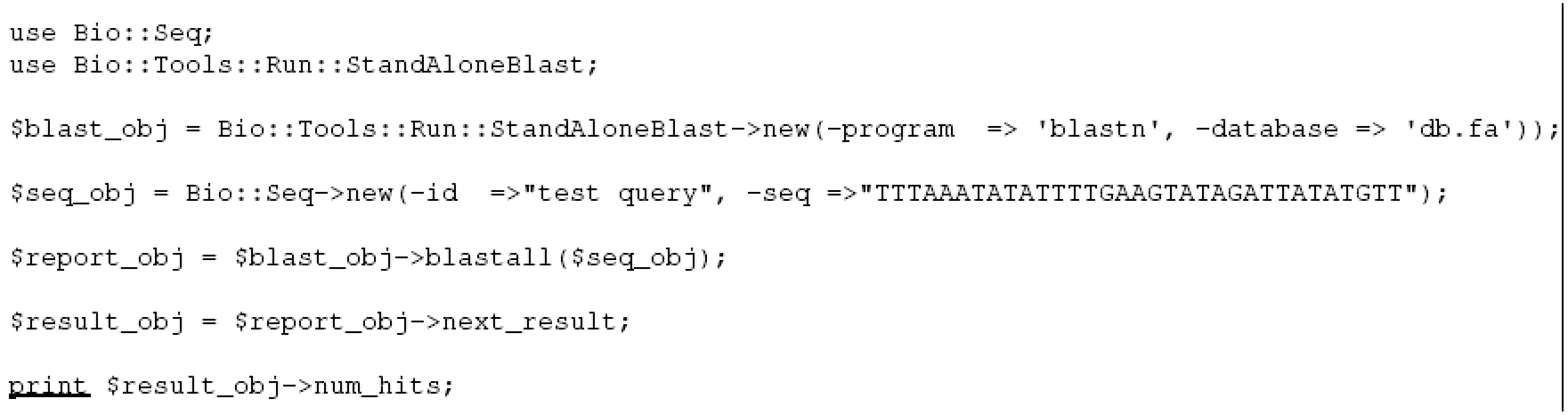
In addition to the methods directly available in the Seq object, Bioperl provides various helper objects to determine additional information about a sequence. For example, SeqStats object provides methods for obtaining the molecular weight of the sequence as well the number of occurrences of each of the component residues (bases for a nucleic acid or amino acids for a protein.) For nucleic acids, SeqStats also returns counts of the number of codons used. For example:

```
use Bio::Tools::SeqStats;
$seq_stats  = Bio::Tools::SeqStats->new($seqobj);
$weight = $seq_stats->get_mol_wt();
$monomer_ref = $seq_stats->count_monomers();
$codon_ref = $seq_stats->count_codons();  # for nucleic acid sequence
```

Note: sometimes sequences will contain ambiguous codes. For this reason, `get_mol_wt()` returns a reference to a two element array containing a greatest lower bound and a least upper bound of the molecular weight.

The SeqWords object is similar to SeqStats and provides methods for calculating frequencies of "words" (e.g. tetramers or hexamers) within the sequence. See Bio::Tools::SeqStats and Bio::Tools::SeqWords for more information.

BLAST



```
use Bio::Seq;
use Bio::Tools::Run::StandAloneBlast;

$blast_obj = Bio::Tools::Run::StandAloneBlast->new(-program  => 'blastn', -database => 'db.fa'));

$seq_obj = Bio::Seq->new(-id  =>"test query", -seq =>"TTTAAATATATTTTGAAGTATAGATTATATGTT");

$report_obj = $blast_obj->blastall($seq_obj);

$result_obj = $report_obj->next_result;

print $result_obj->num_hits;
```

Here's an example of how one would use SearchIO to extract data from a BLAST report:

```
use Bio::SearchIO;
$report_obj = new Bio::SearchIO(-format => 'blast',
                                  -file   => 'report.bls');
while( $result = $report_obj->next_result ) {
    while( $hit = $result->next_hit ) {
      while( $hsp = $hit->next_hsp ) {
         if ( $hsp->percent_identity > 75 ) {
           print "Hit\t", $hit->name, "\n", "Length\t", $hsp->length('total'),
                   "\n", "Percent_id\t", $hsp->percent_identity, "\n";
         }
       }
     }
}
```