Parsing GenBank Files

BIFS 617

Dr. Alkharouf

1

Topics

- Parsing GenBank Files
- More regular expression modifiers
  - /m
  - /s

2

Parsing GenBank Libraries

- Parsing = systematically taking apart some unstructured data so that it can be used in further analysis
- GenBank (Genetic Sequence Data Bank) is a rapidly growing international repository of known genetic sequences from a variety of organisms
- Maintained at the National Center for Biotechnology Information (NCBI) at the National Institutes of Health (NIH) (http://www.ncbi.nlm.nih.gov)
- Bioinformatics work often involves downloading records from GenBank, and then extracting certain parts of the record for further analysis

A GenBank Record

```
LOCUS       1NHV_B                578 aa            linear   VRL 19-DEC-2002
DEFINITION  Chain B, Hepatitis C Virus Rna Polymerase In Complex With Non-
            Nucleoside Analogue Inhibitor.
ACCESSION   1NHV_B
VERSION     1NHV_B  GI:29726693
DBSOURCE    pdb: molecule 1NHV, chain 66, release Dec 19, 2002;
            deposition: Dec 19, 2002;
            class: Transferase;
            source: Mol_id: 1; Organism_scientific: Hepatitis C Virus;
            Organism_common: Virus; Variant: Type 1b; Expression_system:
            Escherichia Coli; Expression_system_common: Bacteria;
            Expression_system_strain: Bl21 (De3);
            Exp. method: X-Ray Diffraction.
KEYWORDS    .
SOURCE      Hepatitis C virus
  ORGANISM  Hepatitis C virus
            Viruses; ssRNA positive-strand viruses, no DNA stage; Flaviviridae;
            Hepacivirus.
REFERENCE   1  (residues 1 to 578)
  AUTHORS   Wang,M., Ng,K.K.S., Cherney,M.M., Chan,L., Yannopoulos,C.G.,
            Bedard,J., Morin,N., Nguyen-Ba,N., Alaoui-Ismaili,M.H.,
            Bethell,R.C. and James,M.N.G.
  TITLE     Non-nucleoside Analogue Inhibitors Bind to an Allosteric Site on
            HCV NS5B Polymerase. CRYSTAL STRUCTURES AND MECHANISM OF INHIBITION
  JOURNAL   J. Biol. Chem. 278 (11), 9489-9495 (2003)
  MEDLINE   22513896
   PUBMED   12509436
REFERENCE   2  (residues 1 to 578)
  AUTHORS   Wang,M., Ng,K.K.S., Cherney,M.M., Chan,L., Yannopoulos,C.G.
  TITLE     Direct Submission
  JOURNAL   Submitted (19-DEC-2002)
COMMENT     Revision History:
            MAR 18 3 Initial Entry.
FEATURES             Location/Qualifiers
     source          1..578
                     /organism="Hepatitis C virus"
                     /db_xref="taxon:11103"
     Region          1..365
                     /region_name="Domain 3"
                     /note="NCBI Domains"
ORIGIN
        1 ashhhhhhsm sytwtgalit pcaaeesklp inalsnsllr hhnmvyatts rsaglrqkkv
       61 tfdrlqvldd hyrdvlkemk akastvkakl lsveeacklt pphsakskfg ygakdvrnls
      121 skavnhihsv wkdlledtvt pidttimakn evfcvqpekg grkparlivf pdlgvrvcek
      181 malydvvstl pqvvmgssyg fqyspgqrve flvntwkskk npmgfsydtr cfdstvtend
      241 irveesiyqc cdlapearqa ikslterlyi ggpltnskgq ncgyrrcras gvlttscgnt
      301 ltcylkasaa craaklqdct mlvngddlvv icesagtqed aaslrvftea mtrysappgd
      361 ppqpeydlel itscssnvsv ahdasgkrvy yltrdpttpl araawetarh tpvnswlgni
      421 imyaptlwar milmthffsi llaqeqleka ldcqiygacy siepldlpqi ierlhglsaf
      481 slhsyspgei nrvasclrkl gvpplrvwrh rarsvrarll sqggraatcg kylfnwavkt
      541 klkltpipaa sqldlsgwfv agysggdiyh slsrarpr
//
```

Parsing GenBank Files

- GenBank uses a flat-file format
  - All data is in plain ASCII text
  - Variable size
  - Variable set of fields
- How do we recognize a record?
  - Records are separated by a line containing //
- Given a record, how do parse the fields?
  - Fields are introduced by keywords
  - No explicit separators

8

```
% cat kinase.gb
LOCUS       AAH35715               396 aa            linear   PRI 29-OCT-2004
DEFINITION  Protein kinase Njmu-R1 [Homo sapiens].
ACCESSION   AAH35715
VERSION     AAH35715.1  GI:54887327
ORIGIN
        1 mlpslqesmd gdekelesse eggsaeerrl eppssshycl ysyrgsrlaq qrgdsedgsp
       61 sgtnaetpsg ddfslsladt nlpsevepel rsfiakrlsr gavfeglgnv asvelkipgy
      121 rvgcyyclfq neklpetvt idsernpsey vvcflggsek glelfrleld kyiqglknnm
      181 neaerglesh iksylsswfe dvvcpiqrvv llfqekltfl lhaalsytpv evkesdektk
      241 rdinrflsva slqglihegt mtslcmamte eqhksvvidc sssqpqfcna gsnrfcedwm
      301 qaflngakgg npflfrqvle nfklkaiqdt nnlkrfirqa emnhyalfkc ymflkncgsg
      361 dillkivkve heempeaknv iavleefmke aldqsf
//

% gb_print3 kinase.gb
ANNOTATION:
LOCUS       AAH35715               396 aa            linear   PRI 29-OCT-2004
DEFINITION  Protein kinase Njmu-R1 [Homo sapiens].
ACCESSION   AAH35715
VERSION     AAH35715.1  GI:54887327
ORIGIN
SEQUENCE:
        1 mlpslqesmd gdekelesse eggsaeerrl eppssshycl ysyrgsrlaq qrgdsedgsp
       61 sgtnaetpsg ddfslsladt nlpsevepel rsfiakrlsr gavfeglgnv asvelkipgy
      121 rvgcyyclfq neklpetvt idsernpsey vvcflggsek glelfrleld kyiqglknnm
      181 neaerglesh iksylsswfe dvvcpiqrvv llfqekltfl lhaalsytpv evkesdektk
      241 rdinrflsva slqglihegt mtslcmamte eqhksvvidc sssqpqfcna gsnrfcedwm
      301 qaflngakgg npflfrqvle nfklkaiqdt nnlkrfirqa emnhyalfkc ymflkncgsg
      361 dillkivkve heempeaknv iavleefmke aldqsf
```

GenBank Record recognition

- When we use a statement like

  ```
  $string = <FH>
  ```

  the read operation < FH > reads ***one record***, defined by the *record separator* variable $/
  - Normally, $/ is defined to be newline "\n"
- We can store an entire GenBank record in one scalar variable ($record) by changing the record separator to "//\n"
- Remember that Perl strings can contain newline characters, for example:

  ```
  $string = "This string will\nprint on\nthree lines.\n";
  print $string;
  ```

  Output:

  ```
  This string will
  print on
  three lines.
  ```

@ARGV

- @ARGV is a special built in array in Perl that stores command line arguments, which are files name or other user input passed to your Perl program through the command line (i.e. when you run your program).

```
#!/usr/bin/perl
# File: gb_print
# read in and print out GenBank records
use strict;
use warnings;

if (not $ARGV[0]) {
   die "usage: gb_print genbank_library\n";
}
open my $fh, $ARGV[0] or die "Can't open file $ARGV[0]";

while ( my $record = get_gb_record($fh) ) {
   print $record;    # each records contains multiple lines
}
exit;

sub get_gb_record {
   my ( $fh ) = @_;
   my $record = '';
   my $saved_separator = $/;
   $/ = "//\n";
   $record = <$fh>;
   $/ = $saved_separator;
   return $record;
}

% perl gb_print.pl library.gb   # prints out the records in library.gb
```

A GenBank Record

```
LOCUS       1NHV_B                578 aa            linear   VRL 19-DEC-2002
DEFINITION  Chain B, Hepatitis C Virus Rna Polymerase In Complex With Non-
            Nucleoside Analogue Inhibitor.
ACCESSION   1NHV_B
VERSION     1NHV_B  GI:29726693
DBSOURCE    pdb: molecule 1NHV, chain 66, release Dec 19, 2002;
            deposition: Dec 19, 2002;
            class: Transferase;
            source: Mol_id: 1; Organism_scientific: Hepatitis C Virus;
            Organism_common: Virus; Variant: Type 1b; Expression_system:
            Escherichia Coli; Expression_system_common: Bacteria;
            Expression_system_strain: Bl21 (De3);
            Exp. method: X-Ray Diffraction.
KEYWORDS    .
SOURCE      Hepatitis C virus
  ORGANISM  Hepatitis C virus
            Viruses; ssRNA positive-strand viruses, no DNA stage; Flaviviridae;
            Hepacivirus.
REFERENCE   1  (residues 1 to 578)
  AUTHORS   Wang,M., Ng,K.K.S., Cherney,M.M., Chan,L., Yannopoulos,C.G.,
            Bedard,J., Morin,N., Nguyen-Ba,N., Alaoui-Ismaili,M.H.,
            Bethell,R.C. and James,M.N.G.
  TITLE     Non-nucleoside Analogue Inhibitors Bind to an Allosteric Site on
            HCV NS5B Polymerase. CRYSTAL STRUCTURES AND MECHANISM OF INHIBITION
  JOURNAL   J. Biol. Chem. 278 (11), 9489-9495 (2003)
  MEDLINE   22513896
   PUBMED   12509436
REFERENCE   2  (residues 1 to 578)
  AUTHORS   Wang,M., Ng,K.K.S., Cherney,M.M., Chan,L., Yannopoulos,C.G.
  TITLE     Direct Submission
  JOURNAL   Submitted (19-DEC-2002)
COMMENT     Revision History:
            MAR 18 3 Initial Entry.
FEATURES             Location/Qualifiers
     source          1..578
                     /organism="Hepatitis C virus"
                     /db_xref="taxon:11103"
     Region          1..365
                     /region_name="Domain 3"
                     /note="NCBI Domains"
ORIGIN
        1 asmnnnnnsm sytwtgafl pcaaeeskip inalshsill nnnmvyatts rsagirqkkv
       61 tfdrlqvldd hyrdvlkemk akastvkakl lsveeacklt pphsakskfg ygakdvrnls
      121 skavnhihsv wkdlledtvt pidttimakn evfcvqpekg grkparlivf pdlgvrvcek
      181 malydvvstl pqvvmgssyg fqyspgqrve flvntwkskk npmgfsydtr cfdstvtend
      241 irveesiyqc cdlapearqa ikslterlyi ggpltnskgq ncgyrrcras gvlttscgnt
      301 ltcylkasaa craaklqdct mlvngddlvv icesagtqed aaslrvftea mtrysappgd
      361 ppqpeydlel itscssnvsv ahdasgkrvy yltrdpttpl araawetarh tpvnswlgni
      421 imyaptlwar milmthffsi llaqeqleka ldcqiygacy siepldlpqi ierlhglsaf
      481 slhsyspgei nrvasclrkl gvpplrvwrh rarsvrarll sqgggraatcg kylfnwavkt
      541 klkltpipaa sqldlsgwfv agysgggdiyh slsrarpr
//
```

annotation

sequence

Parsing Records: Separating Annotation from Sequence

- GenBank records contain **annotation** (or meta-data) and **sequence** data
- In GenBank records, the **annotation** begins with the keyword LOCUS and runs until the end of the line containing the keyword ORIGIN
- The **sequence** contains the remaining lines until the line containing the record separator //

- Two ways to do it:
  - Using Arrays
  - Using Regular Expressions

Using arrays to separate annotation from sequence

```
#!/usr/bin/perl
# Example 10-1   Extract annotation and sequence from
        GenBank file

use strict;
use warnings;
use BeginPerlBioinfo;     # see Chapter 6 about this module

# declare and initialize variables
my @annotation = (  );
my $sequence = '';
my $filename = 'record.gb';

parse1(\@annotation, \$sequence, $filename);

# Print the annotation, and then
#   print the DNA in new format just to check if we got it okay.
print @annotation;

print_sequence($sequence, 50);

exit;
##############################
# Subroutine
##############################
# parse1
#
# -parse annotation and sequence from GenBank record
```

```
sub parse1 {
   my($annotation, $dna, $filename) = @_;
   # $annotation-reference to array
   # $dna       -reference to scalar
   # $filename  -scalar
   # declare and initialize variables
   my $in_sequence = 0;
   my @GenBankFile = (  );

   # Get the GenBank data into an array from a file
   @GenBankFile = get_file_data($filename);

   # Extract all the sequence lines
   foreach my $line (@GenBankFile) {

      if( $line =~ /^\/\/\n/ ) { # If $line is end-of-record line //\n,
         last; #break out of the foreach loop.
      } elsif( $in_sequence) { # If we know we're in a sequence,
         $dna .= $line; # add the current line to $$dna.
      } elsif ( $line =~ /^ORIGIN/ ) { # If $line begins a sequence,
         $in_sequence = 1; # set the $in_sequence flag.
      } else{ # Otherwise
         push( @$annotation, $line); # add the current line to
        @annotation.
      }
   }
   # remove whitespace and line numbers from DNA sequence
   $$dna =~ s/[\s0-9]//g;
}
```

Using regular expressions to separate annotation from sequence

- We want a regular expression that returns the two parts:

```
$record  =~ /^(LOCUS.*ORIGIN\s*\n)(.*)\/\/\n/s;
$annotation = $1;
$sequence = $2;
```

Normally, dot matches any character EXCEPT "\n".
With the modifier /s, dot WILL match any character including "\n".

16

```
#!/usr/bin/perl
# File: gb_print2
# read in and print out GenBank records

use strict;
use warnings;

if (not $ARGV[0]) {
   die "usage: gb_print genbank_library\n";
}
open my $fh, $ARGV[0] or die "Can't open file $ARGV[0]";

while ( my $record = get_gb_record($fh) ) {
   $record  =~ /^(LOCUS.*ORIGIN\s*\n)(.*)\/\/\n/s;
   my $annotation = $1;
   my $sequence = $2;
   print "ANNOTATION:\n";
   print $annotation;
   print "SEQUENCE:\n";
   print $sequence;
}
exit;
```

```
#!/usr/bin/perl
# File: gb_print_3
# read in and print out GenBank records

use strict;
use warnings;

if (not $ARGV[0]) {
   die "usage: gb_print genbank_library\n";
}
open my $fh, $ARGV[0] or die "Can't open file $ARGV[0]";

while ( my $record = get_gb_record($fh) ) {
   my ($annotation, $sequence) =
            ($record  =~ /^(LOCUS.*ORIGIN\s*\n)(.*)\/\/\n/s);
   print "ANNOTATION:\n";
   print $annotation;
   print "SEQUENCE:\n";
   print $sequence;
}
exit;
```

Now we will clean up the Sequence

- Remove numbers at beginning of lines

  $sequence =~ s/^\s*\d*//gm;

  The modifier /m makes ^ and $ match next to embedded newlines. Without the /m the above would only remove the first set of line numbers.

- Remove white space (including embedded newlines)

  $sequence =~ s/\s//g;

19

More on /s and /m

- Lets take an example:

"AAC\nGTT" =~ /^.*$/;

Print $&, "\n";

What do you expect this to return? NOTHING!

"AAC\nGTT" =~ /^.*$/m;

Print $&, "\n";

Output: AAC

"AAC\nGTT" =~ /^.*$/s;

Print $&, "\n";

Output:

AAC

GTT


```
#!/usr/bin/perl
# Example 10-2   Extract the annotation and
      sequence sections from the first
#   record of a GenBank library

use strict;
use warnings;
use BeginPerlBioinfo;     # see Chapter 6 about
      this module

# Declare and initialize variables
my $annotation = '';
my $dna = '';
my $record = '';
my $filename = 'record.gb';
my $save_input_separator = $/;

# Open GenBank library file
unless (open(GBFILE, $filename)) {
   print "Cannot open GenBank file
      \"$filename\"\n\n";
   exit;
}

# Set input separator to "//\n" and read in a
      record to a scalar
$/ = "//\n";

$record = <GBFILE>;

# reset input separator
$/ = $save_input_separator;

# Now separate the annotation from the
      sequence data
($annotation, $dna) = ($record =~
      /^(LOCUS.*ORIGIN\s*\n)(.*)\/\/\n/s);

# Print the two pieces, which should give us the
      same as the
#  original GenBank file, minus the // at the end
print $annotation, $dna;

exit;
```

Parsing the Annotation

- Now that we have the annotation separated from the sequence, we will parse the annotation into top level fields
- GenBank records have many possible top-level fields:
  - SOURCE, DEFINITION, KEYWORDS, VERSION, FEATURES ACCESSION, REFERENCE, LOCUS, BASE
- Some fields (such as FEATURES and REFERENCE) have several subfields
- We will put fields into a hash where the **key** is the name of the field, and the **value** is the set of lines associated with the field.

22

Parsing the Annotation

- All top level fields begin in column 1 (with no preceding white space)
- Field names are ALL CAPITAL LETTERS
- We want a regular expression that matches
  - a line that starts with an UPPER CASE word (the field name)
  - **zero or more lines that do start with white space**

```
my %fields = ();
while ( $annotation =~ /^([A-Z]+).*\n(^\s+.*\n)*/gm ) {
  my $field_name = $1;
  $fields{$field_name} = $&;
}
```

23

Parsing the Annotation

```
while ( $annotation =~ /^([A-Z]+).*\n(^\s+.*\n)*/gm ) {
  my $field_name = $1;
  $fields{$field_name} = $&;
}
```

```
SOURCE      Hepatitis C virus
  ORGANISM  Hepatitis C virus
            Viruses; ssRNA positive-strand viruses, no DNA stage;
            Flaviviridae; Hepacivirus.
REFERENCE   1  (residues 1 to 578)
  AUTHORS   Wang,M., Ng,K.K.S., Cherney,M.M., Chan,L.,
            Yannopoulos,C.G.,
            Bedard,J., Morin,N., Nguyen-Ba,N., Alaoui-Ismaili,M.H.,
            Bethell,R.C. and James,M.N.G.
```


Output:

```
********* LOCUS ********
LOCUS       AAH35715               396 aa            linear   PRI 29-OCT-2004
********* VERSION ********
VERSION     AAH35715.1  GI:54887327
********* DEFINITION ********
DEFINITION  Protein kinase Njmu-R1 [Homo sapiens].
********* ORIGIN ********
ORIGIN
********* ACCESSION ********
ACCESSION   AAH35715
```

Summary

- Parsing means systematically taking apart unstructured data, for further analysis
- GenBank records are plain text files with multiple fields of annotation, and sequence data
- $/ is a special variable that stores the current record separator used in reading.
  - Usually set to \n
  - Can be changed to allow records with embedded newlines
- /s modifies matching property of "." so that it matches the newline character \n
- /m modifies ^ and $ to match next to embedded newlines

30