BLAST

Jon-Michael Deldin

Dept. of Computer Science
University of Montana
jon-michael.deldin@mso.umt.edu

2011-09-19 Mon

Outline

Topic

Goals

- get familiar with BLAST

Goals

- get familiar with BLAST
- show how you can integrate scripting and external tools to save time

Topic

The easy way

If you're on a Mac or Linux machine, run setup.pl from your project root to

- create `data` directory

Be sure to read through the script before executing it.

The easy way

If you're on a Mac or Linux machine, run setup.pl from your project root to

- create `data` directory
- download all sequences

Be sure to read through the script before executing it.

The easy way

If you're on a Mac or Linux machine, run setup.pl from your project root to

- create `data` directory
- download all sequences
- download the query file

Be sure to read through the script before executing it.

The easy way

If you're on a Mac or Linux machine, run setup.pl from your project root to

- create `data` directory
- download all sequences
- download the query file
- create the database

Be sure to read through the script before executing it.

The easy way

If you're on a Mac or Linux machine, run setup.pl from your project root to

- create `data` directory
- download all sequences
- download the query file
- create the database
- index the database

Be sure to read through the script before executing it.

The easy way

If you're on a Mac or Linux machine, run setup.pl from your project root to

- create `data` directory
- download all sequences
- download the query file
- create the database
- index the database
- create stub files

Be sure to read through the script before executing it.

Installing BLAST

Download one of the installers from the NCBI (links on the HTML version)

Project layout

Example (Tree)

```
README
data/
 NC_003997.fna
 NC_005945.fna
 NC_012581.fna
 NC_012659.fna
 anthraxDB.fna
 queryNuc.txt
setup.pl
src/
    query_nuc.pl
    query_prot.pl
```

Project layout

Example (Tree)

```
README
data/
 NC_003997.fna
 NC_005945.fna
 NC_012581.fna
 NC_012659.fna
 anthraxDB.fna
 queryNuc.txt
setup.pl
src/
    query_nuc.pl
    query_prot.pl
```

Explanation

File	Purpose
`README`	How to run your program, etc.
`data/`	Contains data for your project
`setup.pl`	Automate setting up data
`src/`	Source code
`src/query_nuc.pl`	Performs nucleotide searches
`src/query_prot.pl`	Performs protein searches

Downloading sequences

Downloading sequences

Downloading sequences

- Downloading from the command-line

Downloading sequences

- Downloading from the command-line

  OS X Use `curl -O URL`

Downloading sequences

- Downloading from the command-line

  OS X Use `curl -O URL`

  Linux Use `wget URL`

Downloading sequences

- Downloading from the command-line

OS X	Use `curl -O URL`
Linux	Use `wget URL`
Windows	Use your browser

Downloading sequences

- Downloading from the command-line

OS X	Use `curl -O URL`
Linux	Use `wget URL`
Windows	Use your browser

Downloading sequences

- Downloading from the command-line

OS X	Use `curl -O URL`
Linux	Use `wget URL`
Windows	Use your browser

- Files

Downloading sequences

- Downloading from the command-line

OS X	Use `curl -O URL`
Linux	Use `wget URL`
Windows	Use your browser

- Files
  - NC_012659.fna

Downloading sequences

- Downloading from the command-line

OS X	Use `curl -O URL`
Linux	Use `wget URL`
Windows	Use your browser

- Files
  - NC_012659.fna
  - NC_003997.fna

Downloading sequences

- Downloading from the command-line

OS X	Use `curl -O URL`
Linux	Use `wget URL`
Windows	Use your browser

- Files
  - NC_012659.fna
  - NC_003997.fna
  - NC_012581.fna

Downloading sequences

- Downloading from the command-line

OS X	Use `curl -O URL`
Linux	Use `wget URL`
Windows	Use your browser

- Files
  - NC_012659.fna
  - NC_003997.fna
  - NC_012581.fna
  - NC_005945.fna

Preparing the DB

Need to merge all of our `FNA` files into `anthraxDB.fna` so BLAST can search it

- Mac & Linux

```
cd data
cat *.fna > anthraxDB.fna
```

- Windows

```
cd data
copy /a *.fna anthraxDB.fna
```

Index the database

From your project's working directory (i.e., above `data`), type

```
makeblastdb -in data/anthraxDB.fna -dbtype nucl
```

Download the query file

Save queryNuc.txt to `data/` (we'll use it in the next tutorial)

Topic

From the shell

Do a search against the database:

```
blastn -db data/anthraxDB.fna -query data/query.txt
```

- look at the E-values (smaller is better)

Topic

Remember our goal

We are trying to automate queries against BLAST to determine whether ~100 fragments are from the A0248 strain.

Example Run

```
$ perl query_nuc.pl
best hit:       ambiguous
best hit:       ambiguous
:
best hit:       Bacillus anthracis str. A0248
:
:
best hit:       ambiguous

votes for ambiguous: XX
votes for Bacillus anthracis str. A0248: YY
```

Input & Output

input results from a BLAST command

Input & Output

input results from a BLAST command

output how many hits were ambiguous or conclusive

Process (pseudocode)

1. Read the sequences in from the query file (`queryNuc.txt`)

Process (pseudocode)

1. Read the sequences in from the query file (`queryNuc.txt`)
2. For each query sequence:

Process (pseudocode)

1. Read the sequences in from the query file (`queryNuc.txt`)
2. For each query sequence:
    1. Write the query to a text file to use in a BLAST command

Process (pseudocode)

1. Read the sequences in from the query file (`queryNuc.txt`)
2. For each query sequence:
    1. Write the query to a text file to use in a BLAST command
    2. Run the BLAST command

Process (pseudocode)

1. Read the sequences in from the query file (`queryNuc.txt`)
2. For each query sequence:
    1. Write the query to a text file to use in a BLAST command
    2. Run the BLAST command
    3. Parse the output to determine what strain

Process (pseudocode)

1. Read the sequences in from the query file (`queryNuc.txt`)
2. For each query sequence:
    1. Write the query to a text file to use in a BLAST command
    2. Run the BLAST command
    3. Parse the output to determine what strain
3. Print out how many ambiguous and conclusive strains were found

Topic

At the top...

```
use strict;
use warnings;
```

At the top...

```
use strict;
use warnings;

my $db = 'data/anthraxDB.fna'; # path to BLAST DB
```

At the top...

```
use strict;
use warnings;

my $db = 'data/anthraxDB.fna'; # path to BLAST DB

my $query_fn = 'tmp_query.txt'; # file we generate each seq
```

At the top...

```
use strict;
use warnings;

my $db = 'data/anthraxDB.fna'; # path to BLAST DB

my $query_fn = 'tmp_query.txt'; # file we generate each seq

# this is defined in the subroutines tutorial -- paste
# the definition into your file
my @queries = fasta_to_array('data/queryNuc.txt');
```

At the top...

```
use strict;
use warnings;

my $db = 'data/anthraxDB.fna'; # path to BLAST DB

my $query_fn = 'tmp_query.txt'; # file we generate each seq

# this is defined in the subroutines tutorial -- paste
# the definition into your file
my @queries = fasta_to_array('data/queryNuc.txt');

my $num_ambiguous  = 0; # number of ambiguous hits
my $num_conclusive = 0; # number of conclusive hits
```

At the top...

```
use strict;
use warnings;

my $db = 'data/anthraxDB.fna'; # path to BLAST DB

my $query_fn = 'tmp_query.txt'; # file we generate each seq

# this is defined in the subroutines tutorial -- paste
# the definition into your file
my @queries = fasta_to_array('data/queryNuc.txt');

my $num_ambiguous  = 0; # number of ambiguous hits
my $num_conclusive = 0; # number of conclusive hits

# command to r un for each query (i.e., 100 times)
my $cmd = "blastn -db $db -query $query_fn -evalue 1e-10";
```

In the middle...

```
for my $query (@queries) {
    # create a BLAST-query for the current sequence ($query)
    # ...write it to $query_fn

    # execute BLAST
    # my $result = `$cmd`;

    # see what we got (parse the output)
}
```

At the bottom...

```
print "Total ambiguous: $num_ambiguous\n";
print "Total conclusive: $num_conclusive\n";
```