




NCBI PowerScripting

Lecture 3: Framework for Programming



April 20-22, 2005






Overview



- Cygwin LINUX emulation environment
- nedit: text editor
- Module of eUtil subroutines in Perl
 - NCBI_PowerScripting.pm





Perl Module

- Programs will use `NCBI_PowerScripting;`
- Module contains 16 subroutines:
 - input: `read_params`
 - processing: 12 eutil calling routines
 - output: 3 printing routines




Module Header

```
package NCBI_PowerScripting;
use LWP::Simple;
use LWP::UserAgent;
use vars qw ($VERSION @ISA @EXPORT @EXPORT_OK);
require "ctime.pl";
use Exporter;

@ISA = qw (Exporter);
@EXPORT = qw (read_params egquery esearch esummary efetch
    efetch_batch elink epost_uids epost_file
    elink_history elink_batch elink_by_id
    epost_set print_summary
    print_links print_link_summaries );

$VERSION = '1.4';

my $delay = 0;
my $maxdelay = 3;
```

©-utilities 

What you will write...

command

perl myscript.pl my_input

my_input


db|nucleotide
term|mouse[orgn]
usehistory|y

myscript.pl

#!/usr/bin/perl
use NCBI_PowerScripting;
my (%params, %results);

%params = read_params();

%results = esearch(%params);
print "Found \$results{'count'} records";

©-utilities 

sub read_params

Reads an input file of parameter values and stores them in %params

input file


db|nucleotide
term|mouse[orgn]
usehistory|y

you write this!

↓

%params

\$params{'db'} = 'nucleotide'
\$params{'term'} = 'mouse[orgn]'
\$params{'usehistory'} = 'y'

©-utilities 

sub esearch

```
local %params = @_; #get input parameters
[declare variables]

$url = $base . "esearch.fcgi?db=$params{'db'}";
$url .= "&term=$params{'term'}...";

$raw = get($url); #post URL, results into $raw
$raw =~ /<Count>(/d+)<\/Count>/s; #parse count
$results{'count'} = $1; #(/d+) into $1

[continue parsing]
return(%results);
```

© utilities

PowerScripting

Some Perlsh Tricks

Working with multidimensional arrays and hashes

```
$results{'count'} = 5287
$results{'WebEnv'} = A13dkJ_34Kdkfhj305
$results{'query_key'} = 2
@{$results{'uids'}} = (345,1201,930,20,595)
```

```
$results{'uids'}[1] = 1201
```

```
foreach ( @{$results{'uids'}} ) {
    print "$_\n";
}
```

© utilities

PowerScripting

ESummary Routines

sub esummary

Runs ESummary and parses XML output into %results

```
%results = esummary(%params);
```

sub print_summary

Prints %results in a human-readable format

```
print_summary(%results);
```

© utilities

PowerScripting

sub esummary

Download the resolutions of structures corresponding to a list of MMDB-IDs

ESummary is particularly useful for databases NOT supported by EFetch!

db|structure
id|6492,10919

→

\$params{'db'} = 'structure'
\$params{'id'} = '6492,10919'

\$params{'db'} = 'structure'
\$params{'id'} = '6492,10919'

↓

\$results{'uid'}{'item name'}

PowerScripting

esummary Output

Download the resolutions of structures corresponding to a list of MMDB-IDs

```
<?xml version="1.0"?>
<DOCTYPE eSummaryResult PUBLIC "-//NLM//DTD eSummaryResult, 11 May 2003//EN" "http://www.ncbi.nlm.nih.gov/efetch/DTD/eSummaryResult.dtd">
<eSummaryResult>
  <DocSum>
    <Id>6492</Id>
    <Item Name="PdbAcc" Type="String">1DAN</Item>
    <Item Name="PdbDescr" Type="String">Complex Of Active Site Inhibited Human Blood Coagulation Factor VIIa With Human Recombinant Soluble Tissue Factor</Item>
    <Item Name="PdbClass" Type="String">Complex(Serine ProteaseCOFACTORLIGAND)</Item>
    <Item Name="PdbReleaseDate" Type="String">1997/9/4</Item>
    <Item Name="LigCode" Type="String">CA|CAC|CL|FUC|GLC</Item>
    <Item Name="LigCount" Type="String">5</Item>
    <Item Name="ModProteinResCount" Type="String">13</Item>
    <Item Name="ModDNAResCount" Type="String">0</Item>
    <Item Name="ModRNAResCount" Type="String">0</Item>
    <Item Name="ProteinChainCount" Type="String">5</Item>
    <Item Name="RNAChainCount" Type="String">0</Item>
    <Item Name="DNAChainCount" Type="String">0</Item>
  </DocSum>
</eSummaryResult>
```

↓

\$results{'6492'}{'Resolution'} = '2'

PowerScripting

print_summary

print_summary(%results);

Writes to standard output, not a file

```
ID 6492:
DNAChainCount: 0
EC: 3.4.21.21
ExpMethod: X-Ray Diffraction
LigCode: CA|CAC|CL|FUC|GLC
LigCount: 5
ModDNAResCount: 0
ModProteinResCount: 13
ModRNAResCount: 0
PdbAcc: 1DAN
PdbClass: Complex(Serine ProteaseCOFACTORLIGAND)
PdbDescr: Complex Of Active Site Inhibited Human Blood Coagulation Factor VIIa With Human Recombinant Soluble Tissue Factor
PdbReleaseDate: 1997/9/4
ProteinChainCount: 5
RNAChainCount: 0
Resolution: 2
```

PowerScripting

EPost Routines

sub epost_uids

- Runs EPost with GET method
- &id is a comma-delimited list
- use for **small** sets of UIDs (< 250)

```
%results = epost_uids(%params);
```

sub epost_set

- Runs EPost with POST method
- &id is a comma-delimited list
- use for **large** sets of UIDs (> 250)

```
%results = epost_set(%params);
```

sub epost_file

- Runs EPost with POST method
- &id is a **file name** containing one UID per line
- use for **large** sets of UIDs (> 250)

```
%results = epost_file(%params);
```

© utilities PowerScripting

sub epost_uids

Upload a list of 35 PMIDs

db|pubmed
id|15075265,15068336,15007097,15003443...

↓


```
$params{'db'} = 'pubmed'
$params{'id'} = '15075265,15068336,15007097,15003443...'
```

↓

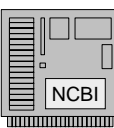
```
$results{'query_key'}
$results{'WebEnv'}
```

© utilities PowerScripting

POSTing a URL



```
$response = $ua->request($req)
```

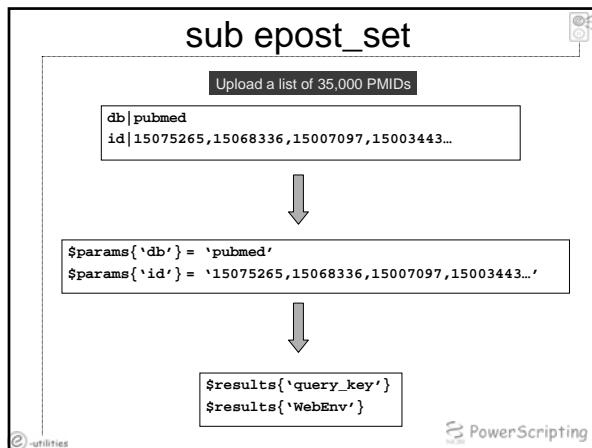


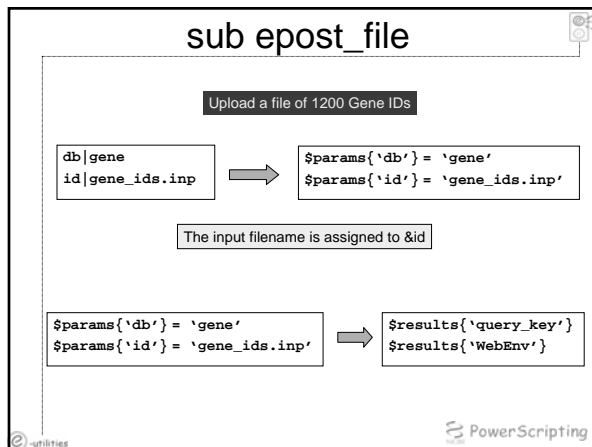
```
$ua = LWP::UserAgent->new
```

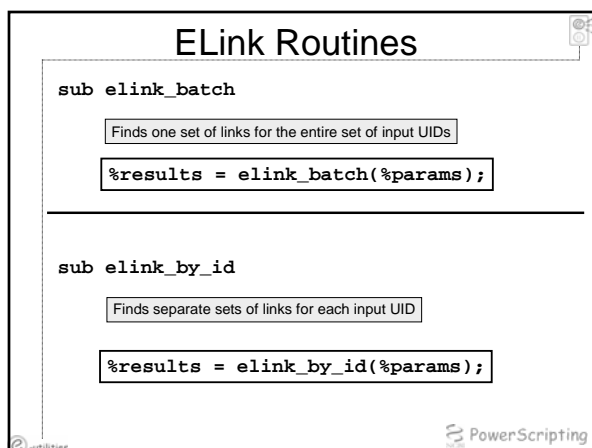
```
$req = new HTTP::Request POST
```

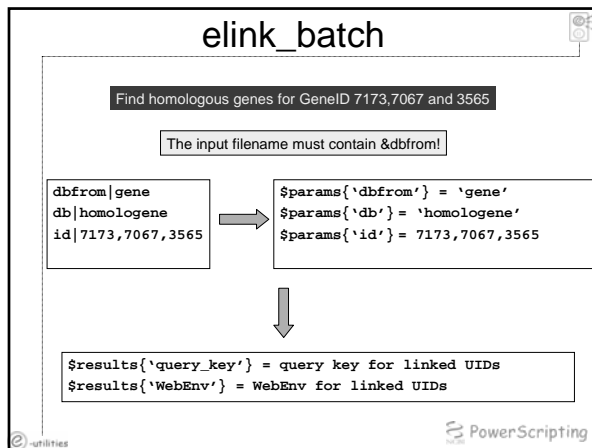
For Perl, you need libwww-perl-5.800

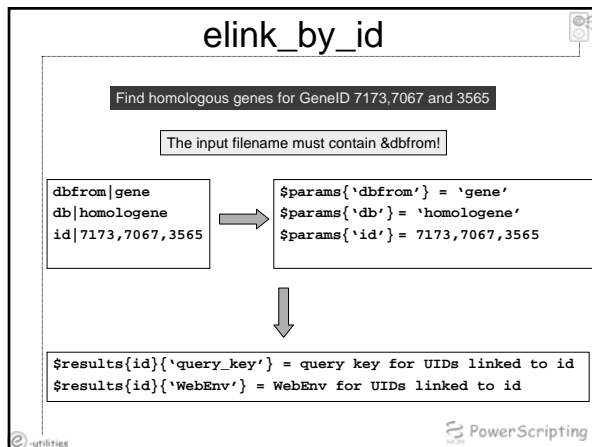
© utilities PowerScripting

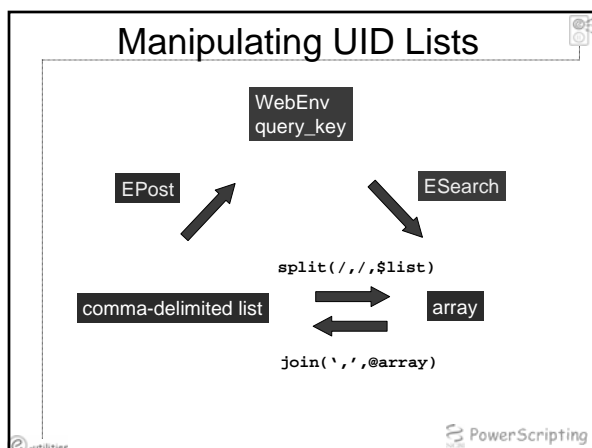












Retrieving a UID List from History

Run ESearch twice: first to get the size of the list;
second to retrieve the UIDs

```
%params1 = read_params();  
%results1 = esearch(%params1);  
  
$params1{'retmax'} = $results1{'count'};  
%results1 = esearch(%params1);  
@uid_list = @{$results1{'uids'}};
```