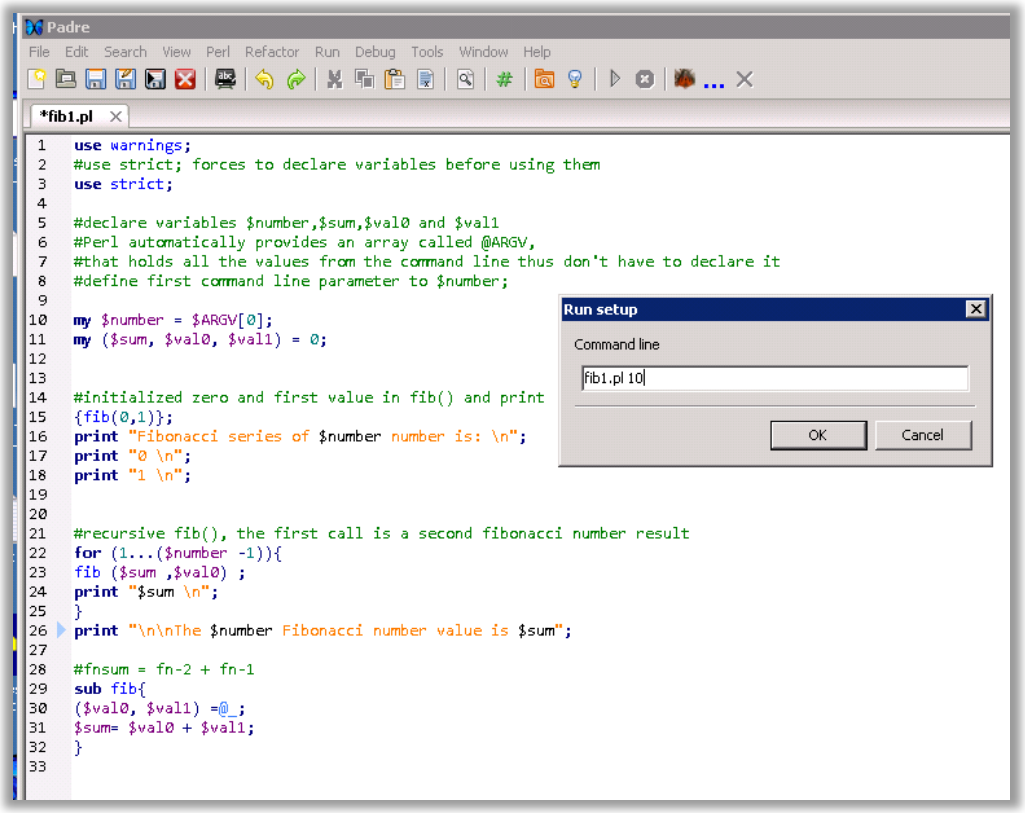
Name: Ho Ping Chong
Student no: 1155057016
Course: CUHK MSc Bioinformatics - GNBF5010

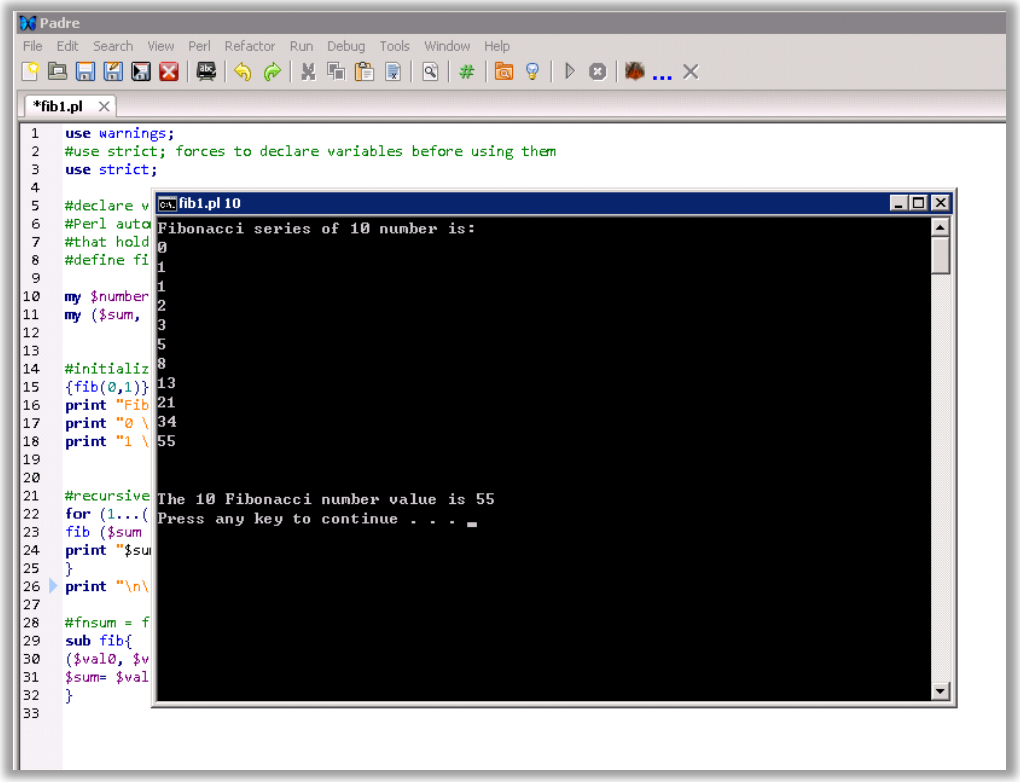
Q1: Write a program to print Fibonacci sequence. The length of output sequence is specified by the first command line parameter. (fibonacci.pl)

To fulfill requirement on “The length of output sequence is specified by the first command line parameter”. We need to use command argument. For example “fib1.pl 10”.

fib1.pl coding:-

```
use warnings;
#use strict; forces to declare variables before using them
use strict;

#declare variables $number,$sum,$val0 and $val1
#Perl automatically provides an array called @ARGV without declaration,
#that holds all the values from the command line
#define first command line parameter to $number;
my $number = $ARGV[0];
my ($sum, $val0, $val1) = 0;

#initialized zero and first value in fib() and print
{fib(0,1)};
print "Fibonacci series of $number number is: \n";
print "0 \n";
print "1 \n";

#recursive fib(), the first call is a second fibonacci number result
for (1...($number -1)){
fib ($sum ,$val0) ;
print "$sum \n";
}
print "\n\nThe $number Fibonacci number value is $sum";

#fnsum = fn-2 + fn-1
sub fib{
($val0, $val1) =@_;
$sum= $val0 + $val1;
}
```





In extreme example, one line codes is possible.

Q2: Try to optimize the pos_annotate.pl as much as you can. (pos_annotateV3.pl)

The current algorithm is failed to match correct chromosome number even though chromosome number is available pos.txt and hg19_refGene.txt file. It was because the lines 16 only return Boolean value instead of chromosome.

my $refChr = $fields[2] =~ m/chr(\d+)/;

Amend as

my ($refChr) = ($fields[2] =~ m/chr(\d+)/);

It will return the number following “chr” in $field[2] follow the regular expression statement. “~m/chr(\d+)/” (http://www.tutorialspoint.com/perl/perl_regular_expression.htm) which will match (m/) with character “chr” (chr) and extract following one or more numbers ((\d+)) which is extract from $fields[2] then store in $refChr.

This only fix the logic on annotation but nil significant improve in performance. Hash is unorder and the search mechansim is control by Perl (http://perl101.org/hashes.html). To improve the search performance in hash. We can reduce the hash size (http://mailman.anu.edu.au/pipermail/perl.sig/2007-June/000033.html). For example reduce the record in %anno from hg19_refGene.txt. For this approach the search time will be same as V4 if input pos.txt cover 1-22 chromosome because of result in same hash size between V4 and V6.

To improve this condition. We try to stratified the whole hash table into multiple hash on each chromonsome number. Thus for each record in pos.txt only one small chromosome hash table will be run through instead of a large merged hash table across chromosome.

To adopt all chromosone number such as X and Y. We can’t use number as chromosome number thus we can’t use array on hash but hash on hash only. On the whole a pair of hash table on chromosome will be use as initial lookup then following hash on hash (http://docstore.mik.ua/orelly/perl/prog3/ch09_04.htm) of h19_refGene.txt. An annotation will result after compare pos.txt position with h19_refGene.txt start and end position.

Version	Executation time in sec (n=15)	Executation time in sec (n=15 X 10)	Remarks
V2	2.187	--	Orginal V2
V4	2.414	--	Fix regular expression
V6	1.361	2.894	Reduce hash size
V7	1.319	2.241	Hash on hash

The performance on hash on hash has similar as reduce hash size. But when increase in chromosome type variation. It can expect V7 is superior then V6.

V2 result:

```
HO-PCs-MacBook:Desktop chpcz01$ time perl pos_annotateV2.pl
1	948921	ADAP1
1	948921	TMEM175
1	948921	ARID3A
1	948921	ADAP1
1	948921	ADAP1
1	948921	SNTG2
1	948921	ERICH1-AS1
1	948921	AP2A2
1	948921	RSPO4
1	948921	LMF1
1	948921	ISG15
……
…..
…..
1	67705958	SUCLG2-AS1
1	67705958	C8orf44-SGK3
1	67705958	SUCLG2-AS1
1	67705958	RTTN
1	67705958	SGK3
1	67705958	PCDH9
1	67705958	SGK3
1	67705958	CTNNA3
1	67705958	LOC101928122
1	67705958	CAND1
1	67705958	IL23R
1	67705958	IQCH
1	67705958	CTNNA3

real 0m2.187s
user0m2.090s
sys  0m0.029s
```

V4 reuslt:

```
HO-PCs-MacBook:Desktop chpcz01$ time perl pos_annotateV4
1	948921	ISG15
1	1404001	ATAD3C
1	5935162	NPHP4
1	162736463	DDR2
1	84875173	DNASE2B
1	84875173	DNASE2B
```

1 67705958 IL23R
2 234183368 ATG16L1
16 50745926 NOD2
16 50745926 NOD2
16 50756540 NOD2
16 50756540 NOD2
16 50763778 NOD2
16 50763778 NOD2
13 20763686 GJB2
13 20797176 GJB6
13 20797176 GJB6

real 0m2.414s
user 0m2.174s
sys 0m0.036s

V6 result:

HO-PCs-MacBook:Desktop chpcz01$ time perl pos_annotateV6.pl
1 948921 ISG15
1 1404001 ATAD3C
1 5935162 NPHP4
1 162736463 DDR2
1 84875173 DNASE2B
1 84875173 DNASE2B
1 67705958 IL23R
2 234183368 ATG16L1
16 50745926 NOD2
16 50745926 NOD2
16 50756540 NOD2
16 50756540 NOD2
16 50763778 NOD2
16 50763778 NOD2
13 20763686 GJB2
13 20797176 GJB6
13 20797176 GJB6

real 0m1.361s
user 0m1.336s
sys 0m0.022s

V7 result:

HO-PCs-MacBook:Desktop chpcz01$ time perl pos_annotateV7.pl

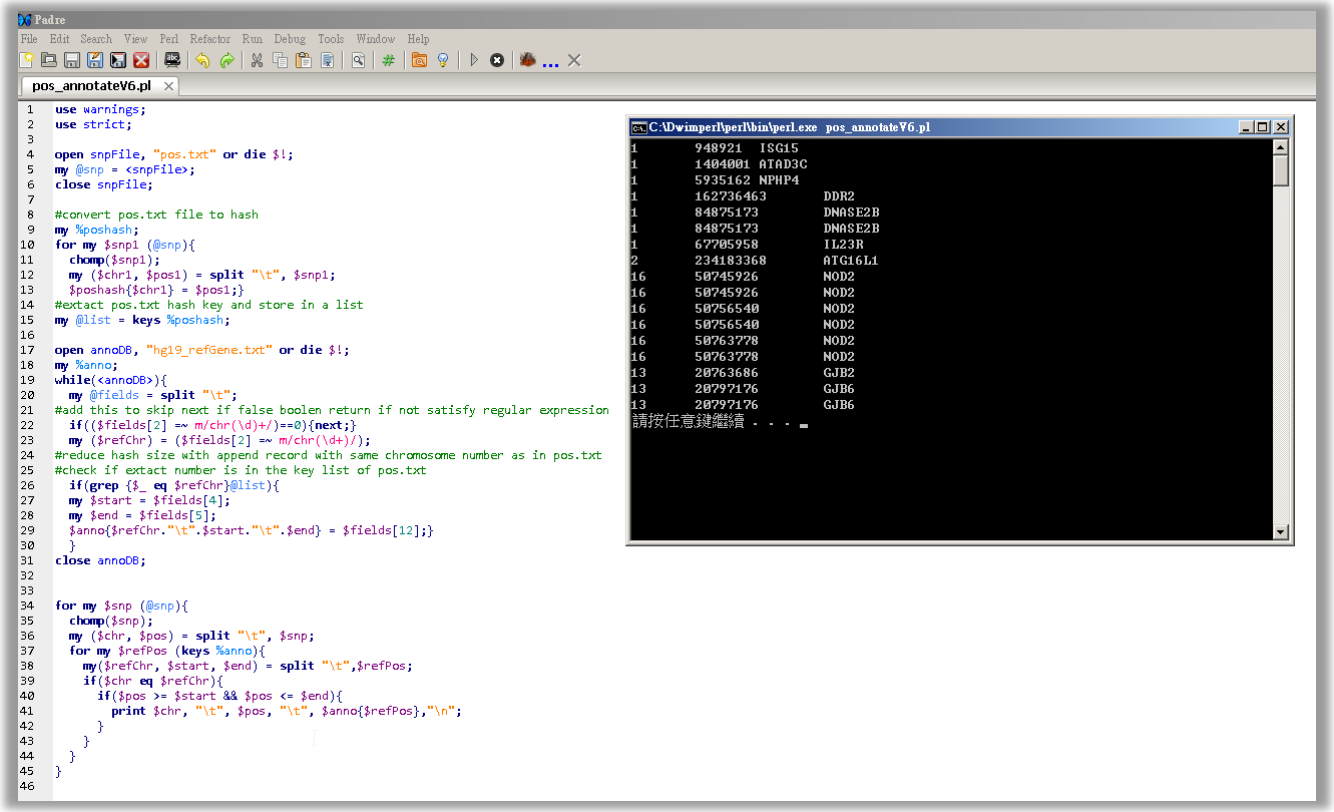
1	948921	ISG15
1	1404001	ATAD3C
1	5935162	NPHP4
1	162736463	DDR2
1	84875173	DNASE2B
1	84875173	DNASE2B
1	67705958	IL23R
2	234183368	ATG16L1
16	50745926	NOD2
16	50745926	NOD2
16	50756540	NOD2
16	50756540	NOD2
16	50763778	NOD2
16	50763778	NOD2
13	20763686	GJB2
13	20797176	GJB6
13	20797176	GJB6

real 0m1.319s
user 0m1.289s
sys 0m0.026s

V6 coding:

```
use warnings;
use strict;

open snpFile, "pos.txt" or die $!;
my @snp = <snpFile>;
close snpFile;

#convert pos.txt file to hash
my %poshash;
for my $snp1 (@snp){
   chomp($snp1);
   my ($chr1, $pos1) = split "\t", $snp1;
   $poshash{$chr1} = $pos1;}
#extact pos.txt hash key and store in a list
my @list = keys %poshash;

open annoDB, "hg19_refGene.txt" or die $!;
my %anno;
while(<annoDB>){
   my @fields = split "\t";
#add this to skip next if false boolen return if not satisfy regular expression
   if(($fields[2] =~ m/chr(\d)+/)==0){next;}
   my ($refChr) = ($fields[2] =~ m/chr(\d+)/);
#reduce hash size with append record with same chromosome number as in pos.txt
#check if extact number is in the key list of pos.txt
   if(grep {$_ eq $refChr}@list){
   my $start = $fields[4];
   my $end = $fields[5];
   $anno{$refChr."\t".$start."\t".$end} = $fields[12];}
   }
close annoDB;



for my $snp (@snp){
   chomp($snp);
   my ($chr, $pos) = split "\t", $snp;
   for my $refPos (keys %anno){
      my($refChr, $start, $end) = split "\t",$refPos;
      if($chr eq $refChr){
         if($pos >= $start && $pos <= $end){
```

```
            print $chr, "\t", $pos, "\t", $anno{$refPos},"\n";
        }
      }
    }
}
```



V7 coding:

```
use warnings;
use strict;

open snpFile, "pos.txt" or die $!;
my @snp = <snpFile>;
close snpFile;

open annoDB, "hg19_refGene.txt" or die $!;
my %anno;
while(<annoDB>){
   my @fields = split "\t";
#add this to skip next if false boolen return if not satisfy regular expression
   if(($fields[2] =~ m/chr(\d)+/)==0){next;}
   my ($refChr) = ($fields[2] =~ m/chr(\d+)/);
   my $start = $fields[4];
   my $end = $fields[5];
#build a hash on hash from chromosome to start+end the value as gene
   $anno{$refChr}{$start."\t".$end}=$fields[12];
   }
close annoDB;

for my $snp (@snp){
   chomp($snp);
   my ($chr, $pos) = split "\t", $snp;
#directly use pos.txt chromosome to lookup key in hg19_refGene hashes
   my $chromosome = $chr;
      for my $reflocation (keys %{$anno{$chromosome}}) {
            my ($start, $end) = split "\t", $reflocation;
            my $gene =$anno{$chromosome}{$reflocation};
                  if($pos >= $start && $pos <= $end){
                    print $chr, "\t", $pos, "\t", $gene,"\n";
              }
           }
        }
```

pos_annotateV7.pl

```perl
use warnings;
use strict;

open snpFile, "pos.txt" or die $!;
my @snp = <snpFile>;
close snpFile;

open annoDB, "hg19_refGene.txt" or die $!;
my %anno;
while(<annoDB>){
  my @fields = split "\t";
#add this to skip next if false boolen return if not satisfy regular expression
  if(($fields[2] =~ m/chr(\d)+/)==0){next;}
  my ($refChr) = ($fields[2] =~ m/chr(\d+)/);
  my $start = $fields[4];
  my $end = $fields[5];
#build a hash on hash from chromosome to start+end the value as gene
  $anno{$refChr}{$start."\t".$end}=$fields[12];
  }
close annoDB;

for my $snp (@snp){
  chomp($snp);
  my ($chr, $pos) = split "\t", $snp;
#directly use pos.txt chromosome to lookup key in hg19_refGene hashes
  my $chromosome = $chr;
    for my $reflocation (keys %{$anno{$chromosome}}) {
         my ($start, $end) = split "\t", $reflocation;
         my $gene =$anno{$chromosome}{$reflocation};
             if($pos >= $start && $pos <= $end){
              print $chr, "\t", $pos, "\t", $gene,"\n";
          }
        }
      }
```

```
C:\Dwimperl\perl\bin\perl.exe  pos_annotateV7.pl
1       948921  ISG15
1       1404001 ATAD3C
1       5935162 NPHP4
1       162736463       DDR2
1       84875173        DNASE2B
1       84875173        DNASE2B
1       67705958        IL23R
2       234183368       ATG16L1
16      50745926        NOD2
16      50745926        NOD2
16      50756540        NOD2
16      50756540        NOD2
16      50763778        NOD2
16      50763778        NOD2
13      20763686        GJB2
13      20797176        GJB6
13      20797176        GJB6
請按任意鍵繼續 . . .
```

I am submitting the assignment for:

x□ an individual project or

□ a group project on behalf of all members of the group. It is hereby confirmed that the submission is authorized by all members of the group, and all members of the group are required to sign this declaration.

I/We declare that the assignment here submitted is original except for source material explicitly acknowledged, the piece of work, or a part of the piece of work has not been submitted for more than one purpose (i.e. to satisfy the requirements in two different courses) without declaration, and that the submitted soft copy with details listed in the <Submission Details> is identical to the hard copy(ies), if any, which has(have) been / is(are) going to be submitted. I/We also acknowledge that I am/we are aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the University website http://www.cuhk.edu.hk/policy/academichonesty/. In the case of a group project, we are aware that each student is responsible and liable to disciplinary actions should there be any plagiarized contents in the group project, irrespective of whether he/she has signed the declaration and whether he/she has contributed directly or indirectly to the plagiarized contents.

It is also understood that assignments without a properly signed declaration by the student concerned and in the case of a group project, by all members of the group concerned, will not be graded by the teacher(s).

Ho Ping Chong	24 Oct 2014
Signature(s)	Date

Ho Ping Chong	1155057016
Name(s)	Student ID(s)

GNBF5010	Introduction to programming
Course code	Course title