`pim`, Version 0.3: Alignment-free Estimation of Nucleotide Diversity

Bernhard Haubold, Floyd A. Reed, and Peter Pfaffelhuber

January 20, 2011

1 Introduction

`pim` is a program for estimating the number of mismatches per site between two unaligned sequences [1].

2 Getting Started

`pim` was written in C on a computer running Mac OS X; it is intended to work on any UNIX system with a C compiler. However, please contact BH at `haubold@evolbio.mpg.de` if you have any problems with the program.

- Unpack the program

  ```
  tar -xvzf pim_XXX.tgz
  ```

  where `XXX` indicates the version.
- Change into the newly created directory

  ```
  cd Pim_XXX
  ```

  and list its contents

  ```
  ls
  ```
- Generate `pim`

  ```
  make
  ```
- List its options

  ```
  ./pim -h
  ```
- Test the program on two pairs of simulated 100 kb sequences containing 1000 SNPs:

  ```
  ./pim -s Data/sbjct.fasta Data/query.fasta
  ```
- For the following test, the programs `ms2dna` and `getSeq` need to be installed on your computer. These are available from

  `http://guanine.evolbio.mpg.de/bioBox/`

  Given that these programs are in your path, you can repeatedly simulate pairs of 100 kb sequences with 1000 SNPs and compute $\hat{\pi}_{\mathrm{m}}$:

  ```
  sh ./test.sh
  ```
- In order to calculate the average $\hat{\pi}_{\mathrm{m}}$ value, you can try:

  ```
  sh ./test.sh |
  awk '{s += $2; c++}END{print "avg:", s/c}'
  ```

3 Application to *Drosophila* Chromosomes

- Apply `pim` to the the unaligned left arm of chromosome 3 sequenced from two members of the Raleigh strain collection of *Drosophila melanogaster*[1]

  ```
  ./pim  -s Data/chr3L_RAL-303_1.fasta Data/chr3L_RAL-301_1.fasta
  ```

- Carry out a sliding window analysis of the *Drosophila* data

  ```
  ./pim -w 150000 -s Data/chr3L_RAL-303_1.fasta Data/chr3L_RAL-301_1.fasta
  ```

  To visualize the resulting curve you can, for example, pipe it though the program `graph`, which is part of the GNU Plotting Utilities:

  ```
  ./pim -w 150000 -s Data/chr3L_RAL-303_1.fasta Data/chr3L_RAL-301_1.fasta |
  graph -T X
  ```

4 Change Log

1. Version 0.1 (May 24, 2010)
   - First release.
2. Version 0.2 (November 23, 2010)
   - Fixed serious bug in `pim.windowAnalysis`.
3. Version 0.3 (January 20, 2011)
4. Removed spurious reference to the Gnu Scientific Library from `Makefile`.

5 Acknowledgement

This software is based on the `dss_sort` library by G. Manzini [2].

6 Listing

The following listing documents the central part of the code for `pim`.

```
/***** pim.c ************************************
 * Description: Compute the estimator of the number
 *   of pairwise mismatches from shustring lengths
 *   as described in Haubold, Reed, Pfaffelhuber
 *   (2010). Alignment-free estimation of
 *   nucleotide diversity identifies a
 *   novel genomic outlier of unusually low genetic
 *   diversity in Drosophila melanogaster. In
 *   preparation.
 * Author: Bernhard Haubold, haubold@evolbio.mpg.de
 * Date: Thu Mar 12 15:40:15 2009
 ***********************************************/
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
```

---

[1] Sequence data obtained from the Drosophila Population Genomics Project, `http://www.dpgp.org/`

```c
#include <fcntl.h>
#include "eprintf.h"
#include "sequenceData.h"
#include "interface.h"
#include "lcpTree.h"
#include "pim.h"

void windowAnalysis(int *sl, int n, int w, int stepLen);

int main(int argc, char *argv[]){
  int queryDscr;
  Args *args;
  char *version;
  int i;

  version = "0.3";
  setprogname2("pim");
  args = getArgs(argc, argv);
  if(args->p)
    printSplash(version);
  if(args->h || args->e)
    printUsage(version);
  if(args->numInputFiles == 0){
    queryDscr = 0;
    runAnalysis(queryDscr, args);
  }else{
    for(i=0;i<args->numInputFiles;i++){
      queryDscr = open(args->inputFiles[i],0);
      if(queryDscr < 0)
        eprintf("ERROR:␣could␣not␣open␣query␣file␣%s\n",args->inputFiles[i]);
      runAnalysis(queryDscr, args);
      close(queryDscr);
    }
  }
  free(args);
  free(progname());
  return 0;
}


void runAnalysis(int queryDscr, Args *args){
  Sequence *query, *sbjct;
  Sequence *seq;
  int *sl;
  double p;
  int sbjctDscr;

  sbjctDscr = open(args->s,0);
  if(sbjctDscr < 0)
    eprintf("ERROR:␣could␣not␣open␣sbjct␣file␣%s\n",args->s);
  query = readFasta(queryDscr);
  sbjct = readFasta(sbjctDscr);
  close(sbjctDscr);
  prepareSeq(query);
  prepareSeq(sbjct);
  seq = catSeq(query,sbjct);
  seq->sbjctGc = gcContent(sbjct);
  seq->queryGc = gcContent(query);
  query = freeSequence(query);
  sbjct = freeSequence(sbjct);
  sl = getLcpTreeShulens(args, seq);
  if(args->w){
    windowAnalysis(sl,seq->numQueryNuc/2,args->w,args->S);
    return;
  }
  if(args->u)
    printShulens(sl, seq->numQueryNuc/2);
  else{
    p = pim(args, seq, sl);
    printf("pi_m:␣%8.4e\n",p);
  }
  freeSequence(seq);
}

void printShulens(int *sl, int n){
  int i;
  for(i=0;i<n;i++){
    printf("%d\t%d\n",i+1,sl[i]);
  }
}

/* windowAnalysis: sliding window analysis of shulens */
void windowAnalysis(int *sl, int n, int winLen, int stepLen){
  int i, j, s;
  double pos;
  double t;

  s = 0;
  for(i=0;i<winLen;i++)
    s += sl[i];
  pos = (double)i - winLen/2.;
  t = (double)winLen/(double)s;
  printf("%.1f\t%g\n",pos,t);

  for(i=winLen;i<n;i+=stepLen){
    for(j=0;j<stepLen;j++){
      s += sl[i+j];
      s -= sl[i+j-winLen];
    }
    pos = (double)i - winLen/2.;
    t = (double)winLen/(double)s;
    printf("%.1f\t%g\n",pos,t);
  }
}


double pim(Args *args, Sequence *seq, int *sl){
  int i;
  Result *res;
  double avgShulen;

  res = countShulens(sl,seq->numQueryNuc);

  avgShulen = 0;
  for(i=0;i<res->n;i++)
    avgShulen += res->c[i] * (i);

  return seq->numSbjctNuc/avgShulen;
}

Result *countShulens(int *sl, int n){
  int max = -1;
  Result *res;
  int i, *c;

  /* find maximum */
  for(i=0;i<n;i++)
    if(max < sl[i])
      max = sl[i];
  c = (int *)emalloc((max+1)*sizeof(int));
  for(i=0;i<=max;i++)
    c[i] = 0;
  for(i=0;i<n;i++)
    c[sl[i]]++;
  res = (Result *)emalloc(sizeof(Result));
  res->n = max+1;
  res->c = c;
  return res;
}
```

References

[1] B. Haubold, F. A. Reed, and P. Pfaffelhuber. Alignment-free estimation of nucleotide diversity identifies a novel genomic outlier of unusually low genetic diversity in *Drosophila melanogaster*. *In preparation*, 2010.

[2] G. Manzini and P. Ferragina. Engineering a lightweight suffix array construction algorithm. In *ESA '02: Proceedings of the 10th Annual European Symposium on Algorithms*, pages 698–710, London, UK, 2002. Springer-Verlag.