pamLog, v. 0.7: Take log of PAM Matrix

Bernhard Haubold
Max-Planck-Institute for Evolutionary Biology, Plön, Germany

July 11, 2017

1 Introduction

2 Getting Started

pamLog was written in C on a computer running Linux and should work on any standard UNIX system. However, please contact me at haubold@evolbio.mpg.de if you have any problems with the program.

- Unpack the program

  ```
  tar -xvzf pamLog_XXX.tgz
  ```

  where XXX indicates the version.

- Change into the newly created directory

  ```
  cd PamLog_XXX
  ```

  and list its contents

  ```
  ls
  ```

- Generate pamLog

  ```
  make
  ```

- List its options

  ```
  ./pamLog -h
  ```

- Test program

  ```
  ./pamLog pam70n.txt
  ```

3 Listing

The following listing documents the driver program for pamLog.

```
1 /***** pamLog.c *********************************
   * Description:
   * Author: Bernhard Haubold, haubold@evolbio.mpg.de
   * Date: Tue Jan  6 11:52:39 2015
   ********************************************/
6 #include <stdio.h>
```

```c
#include <stdlib.h>
#include <math.h>
#include "interface.h"
#include "eprintf.h"
#include "pam.h"

void scanFile(FILE *fp, Args *args);
int round2(float x);

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

  version = "0.7";
  setprogname2("pamLog");
  args = getArgs(argc, argv);
  if(args->v)
    printSplash(version);
  if(args->h || args->e)
    printUsage(version);
  if(args->numInputFiles == 0){
    fp = stdin;
    scanFile(fp, args);
  }else{
    for(i=0;i<args->numInputFiles;i++){
      fp = efopen(args->inputFiles[i],"r");
      scanFile(fp, args);
      fclose(fp);
    }
  }
  free(args);
  free(progname());
  return 0;
}

void scanFile(FILE *fp, Args *args){
  SubstitutionMatrix *matrix;
  int i, j;

  matrix = readSubstitutionMatrix(fp);
  /* take log and round */
  for(i=0;i<matrix->size;i++)
    for(j=0;j<matrix->size;j++)
      matrix->mat[i][j] = round2(log(matrix->mat[i][j])/log(2.0)/args->b);
  outputMatrixInt(stdout,matrix);
}

int round2(float x){
  double ip;

  if(modf(x, &ip) >= 0.5){
    return (int) ++ip;
```

```
61    }else if(modf(x, &ip) <= -0.5){
        return (int) --ip;
      }else{
        return (int) ip;
      }
66  }
```

4 Change Log

- Version 0.7 (January 6, 2015)
  - First version with standardized interface.