

pamNormalize, v. 0.7: Normalize PAM Matrix

Bernhard Haubold

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

January 7, 2015

1 Introduction

2 Getting Started

pamNormalize 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 pamNormalize_XXX.tgz
```

where XXX indicates the version.

- Change into the newly created directory

```
cd PamNormalize_XXX
```

and list its contents

```
ls
```

- Generate pamNormalize

```
make
```

- List its options

```
./pamNormalize -h
```

- Test program

```
./pamNormalize -a aa.txt pam70.txt
```

3 Listing

The following listing documents the driver program for pamNormalize.

```
1 /***** pamNormalize.c *****/
   * Description: Normalize PAM matrix.
   * Author: Bernhard Haubold, haubold@evolbio.mpg.de
   * Date: Tue Jan 6 11:52:51 2015
   *****/
6 #include <stdio.h>
```

```

#include <stdlib.h>
#include "interface.h"
#include "eprintf.h"
#include "pam.h"
11 #include "tab.h"

void scanFile(FILE *fp, Args *args, Matrix *aa);

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

21     version = "0.7";
    setprogname2("pamNormalize");
    args = getArgs(argc, argv);
    if(args->v)
26     printSplash(version);
    if(args->h || args->e)
        printUsage(version);
    fp = fopen(args->a, "r");
    aa = getMatrix(fp, args->f);
31     fclose(fp);
    if(args->numInputFiles == 0){
        fp = stdin;
        scanFile(fp, args, aa);
    }else{
36     for(i=0; i<args->numInputFiles; i++){
        fp = fopen(args->inputFiles[i], "r");
        scanFile(fp, args, aa);
        fclose(fp);
    }
41 }
    free(args);
    free(progname());
    return 0;
}

46 void scanFile(FILE *fp, Args *args, Matrix *aa){
    SubstitutionMatrix *matrix;
    int i, j;
    double x;

51     matrix = readSubstitutionMatrix(fp);
    if(aa->numRows != matrix->size)
        eprintf("main_\u2014amino_acid_frequencies:_%d;_matrix_rows:_%d\n", aa->
            numRows, matrix->size);
    /* matrix normalization */
56     for(i=0; i<matrix->size; i++){
        for(j=0; j<aa->numRows; j++){
            matrix->mat[i][j] /= aa->mat[i][0];
        }
        /* make matrix symmetrical */
    }
}

```

```

        for (i=1; i<matrix->size; i++)
61     for (j=0; j<i; j++) {
            x = (matrix->mat[i][j] + matrix->mat[j][i]) / 2.0;
            matrix->mat[i][j] = x;
            matrix->mat[j][i] = x;
        }
66     outputMatrix(stdout, matrix, args->f);
    }

```

4 Change Log

- Version 0.7 (January 6, 2015)
 - First version with reorganized code, i.e. with standard interface and nicer output.