

testMeans, v. 0.2:

Bernhard Haubold

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

May 12, 2017

1 Introduction

TestMeans implements the t-test.

2 Getting Started

TestMeans 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 testMeans_XXX.tgz
```

where XXX indicates the version.

- Change into the newly created directory

```
cd TestMeans_XXX
```

and list its contents

```
ls
```

- Generate testMeans

```
make
```

- List its options

```
./testMeans -h
```

- Inspect the example data, which consists of paired measurements, which you can think of as control

```
cat exampleControl.txt
```

```
Cwc22 11.961 12.401 11.661 11.96 10.454 11.584 11.175 11.343
```

```
Yars2 9.347 9.341 9.29 9.441 9.602 9.892 10.058 9.99
```

and treatment

```
cat exampleTreatment.txt
```

```
Cwc22 8.479 8.523 8.793 8.726 9.677 8.728 8.383 11.086
```

```
Yars2 9.924 9.953 9.568 9.79 10.381 10.156 10.045 10.079
```

where for each gene (first column) eight measures were made (columns 2–9). `TestMeans` can deal with an arbitrary number of measurements as long as the files being compared have the same number of columns.

- Test program

```
./testMeans exampleControl.txt exampleTreatment.txt
Cwc22 1.157e+01 9.049e+00 2.807e-05
Yars2 9.620e+00 9.987e+00 2.166e-02
```

where the four columns in each row indicate

1. Sample name (gene in this case)
2. Average of measurements for sample 1
3. Average of measurements for sample 2
4. *P*-value, that is, the error probability when rejecting the null hypothesis that both samples were drawn from the same population.

3 Listing

The following listing documents the driver program for `testMeans`.

```
1  /***** testMeans.c *****/
   * Description: Compare two means.
   * Author: Bernhard Haubold, haubold@evolbio.mpg.de
   * Date: Thu Dec 15 10:44:31 2016
   *****/
6  #include <stdio.h>
   #include <stdlib.h>
   #include <string.h>
   #include "interface.h"
   #include <gsl/gsl_statistics_double.h>
11 #include "eprintf.h"
   #include "tab.h"
   #include "tTest.h"
   #include "mcTest.h"
   #include "gsl_rng.h"
16
   #define MAX_ID_LEN 256

   void test(FILE *fp1, FILE *fp2, Args *args){
       char *line, *id1, *id2;
21   double *s1, *s2;
       int n1, n2, f, l, i;
       Result *res;
       gsl_rng *ran;

26   s1 = s2 = NULL;
       n1 = n2 = 0;
       l = 0;
       id1 = (char *)emalloc(MAX_ID_LEN);
       id2 = (char *)emalloc(MAX_ID_LEN);
31   res = (Result *)emalloc(sizeof(Result));
       while((line = tabGetLine(fp1)) != NULL){
           l++;
```

```

f = tabNfield() - 1;
if(f > n1)
36     s1 = (double *)erealloc(s1, f * sizeof(double));
n1 = f;
id1 = strncpy(id1,tabField(0),MAX_ID_LEN);
for(i=1; i<=n1; i++)
    s1[i-1] = atof(tabField(i));
41 line = tabGetLine(fp2);
if(!line){
    printf("ERROR: unpaired_line_#d\n",l);
    exit(-1);
}
46 id2 = strncpy(id2,tabField(0),MAX_ID_LEN);
if(strcmp(id1,id2)){
    printf("ERROR: s_on_line_d_paired_with_s.\n", id1, l, id2);
    exit(-1);
}
51 f = tabNfield() - 1;
if(f > n2)
    s2 = (double *)erealloc(s2, f * sizeof(double));
n2 = f;
for(i=1; i<=n2; i++)
56     s2[i-1] = atof(tabField(i));
if(args->t == 'm'){
    ran = ini_gsl_rng(args);
    res = mcTest(res, ran, args->i, s1, n1, s2, n2);
    free_gsl_rng(ran, args);
61 }else if(args->t == 's')
    res = student(res, s1, n1, s2, n2);
else if(args->t == 'w')
    res = welch(res, s1, n1, s2, n2);
    printf("%s\t%.3e\t%.3e\t%.3e\n", id1, res->m1, res->m2, res->p);
66 }
free(id1);
free(id2);
}

71 int main(int argc, char *argv[]){
    int i, j;
    char *version;
    Args *args;
76 FILE *fp1, *fp2;

    version = "0.2";
    setprogname2("testMeans");
    args = getArgs(argc, argv);
81 if(args->v)
    printSplash(version);
if(args->h || args->e)
    printUsage(version);
for(i=0; i<args->numInputFiles; i++){
86     for(j=i+1; j<args->numInputFiles; j++){
        fp1 = fopen(args->inputFiles[i], "r");

```

```

        fp2 = fopen(args->inputFiles[j], "r");
        test(fp1, fp2, args);
        fclose(fp1);
91      fclose(fp2);
    }
  }
  free(args);
  free(progname());
96  return 0;
}

```

4 Change Log

- Version 0.1 (March 3, 2017)
 - First running version
- Version 0.2 (May 12, 2017)
 - Explain data format in help output (-h).