# `genCode`, v. 0.4: Measuring Error Minimization in the Genetic Code

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

November 6, 2018

## 1 Introduction

Haig and Hurst (1991) investigated the average effect of changing codon assignments in the genetic code. They found that the natural genetic code minimizes such changes when compared to random genetic codes of the same redundancy. The program `genCode` implements the simulations described by Haig and Hurst (1991)

## 2 Getting Started

`genCode` was written in C on a computer running Linux. It depends on the Gnu Scientific Library (GSL) 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 genCode_XXX.tgz
  ```

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

  ```
  cd GenCode_XXX
  ```

  and list its contents

  ```
  ls
  ```
- Generate `genCode`

  ```
  make
  ```
- List its options

  ```
  ./genCode -h
  ```
- Test the program by running it in default mode, which is, $10^4$ iterations and printing of codes that are better with respect to ms0 than the natural code; the polarity data is taken from Table 1 in Haig and Hurst (1991):

```
./genCode polarity.dat
   T   C   A   G
T Lys Gly Arg Asp T
  Lys Gly Arg Asp C
  Ala Gly Ter Ter A
  Ala Gly Ter Glu G
C Ala Met Trp Gln T
  Ala Met Trp Gln C
  Ala Met Ile Gln A
  Ala Met Ile Gln G
A Ser Leu Phe Gly T
  Ser Leu Phe Gly C
  Ser Leu Tyr Gln A
  Cys Leu Tyr Gln G
G Asn Val Thr His T
  Asn Val Thr His C
  Asn Val Pro His A
  Asn Val Pro His G
ms0: 4.51829
        ms1     ms2     ms3     ms0
nc      4.88    10.56   0.14    5.19
m(rc)   12.09   12.65   3.59    9.43
```

The simulated codes are bound to differ in your simulation, since they are drawn from a total of $20! \approx 2.4 \times 10^{18}$ possibilities. The table at the end shows the statistics for the natural code (`nc`), which should be identical to the first row of Table 3 in Haig and Hurst (1991). The second row in the table shows the mean for random codes (`m(rc)`), which should be similar to the first row of Table 2 in Haig and Hurst (1991).

## 3 Listing

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

```
/***** genCode.c **********************************
 * Description:
 * Author: Bernhard Haubold, haubold@evolbio.mpg.de
 * Date: Tue Nov  1 17:40:48 2016
 **************************************************/
#include <stdio.h>
#include <stdlib.h>
#include "interface.h"
#include "eprintf.h"
#include "code.h"
#include "prop.h"

void runAnalysis(FILE *fp, Args *args, gsl_rng *ran){
  int i;
  Code *code;
  double ms0, ms1, ms2, ms3;
  double sms0, sms1, sms2, sms3;

  code = getCode(fp);
  if(args->d){
    printCode(code);
    printProp(code);
    printf("fn1:␣%d;␣fn2:␣%d;␣fn3:␣%d\n",code->fn1,code->fn2,code->fn3);
    printf("n1:␣%d;␣n2:␣%d;␣n3:␣%d\n",code->n1, code->n2, code->n3);
    return;
  }
  analyzeCode(code);
  ms0 = code->ms0;
  ms1 = code->ms1;
  ms2 = code->ms2;
  ms3 = code->ms3;
  sms0 = sms1 = sms2 = sms3 = 0.;
  for(i=0;i<args->n;i++){
    shuffleMap(code, ran);
    analyzeCode(code);
    sms0 += code->ms0;
    sms1 += code->ms1;
    sms2 += code->ms2;
    sms3 += code->ms3;
    if(args->p)
      printf("ms0:\t%g\n",code->ms0);
    else if(!args->q && code->ms0 < ms0){
      printCode(code);
      printf("ms0:␣%g\n",code->ms0);
    }
  }
  sms0 /= args->n;
  sms1 /= args->n;
  sms2 /= args->n;
  sms3 /= args->n;
  if(args->m){
    printf("\tms1\tms2\tms3\tms0\n");
    printf("nc\t%.2f\t%.2f\t%.2f\t%.2f\n",ms1,ms2,ms3,ms0);
    printf("m(rc)\t%.2f\t%.2f\t%.2f\t%.2f\n",sms1,sms2,sms3,sms0);
  }else{
    printf("\tms0\n");
    printf("nc\t%.2f\n",ms0);
    printf("m(rc)\t%.2f\n",sms0);
  }
}

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

  version = "0.4";
  setprogname2("genCode");
  args = getArgs(argc, argv);
  ran = ini_gsl_rng(args);
  if(args->v)
    printSplash(version);
  if(args->h || args->e)
    printUsage(version);
  if(args->numInputFiles == 0){
    fp = stdin;
    runAnalysis(fp, args, ran);
  }else{
    for(i=0;i<args->numInputFiles;i++){
      fp = efopen(args->inputFiles[i],"r");
      runAnalysis(fp, args, ran);
      fclose(fp);
    }
  }
  free_gsl_rng(ran, args);
  free(args);
  free(progname());
  return 0;
}
```

## 4 Change Log

- Version 0.1 (Nov. 2, 2016)
  - First running version.
- Version 0.2 (March 30, 2017)
  - Included `-p` option for plotting all `ms0` values.
- Version 0.3 (April 12, 2017)
  - Reduced default output to ms0.
- Version 0.4 (November 6, 2018)
  - Fixed bug in `interface.c`.

## References

D. Haig and L. D. Hurst. A quantitative measure of error minimization in the genetic code. *Journal of Molecular Evolution*, 33:412–417, 1991.