`ranseq`, v. 0.9: Generate random DNA sequence

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

June 2, 2017

1 Introduction

When testing a program for analyzing DNA sequences, I often need a random sequence of defined G/C content. I wrote the program `ranseq` to quickly generate such sequences.

2 Getting Started

`ranseq` 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 ranseq_XXX.tgz
  ```

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

  ```
  cd Ranseq_XXX
  ```

  and list its contents

  ```
  ls
  ```
- Generate `ranseq`

  ```
  make
  ```
- List its options

  ```
  ./ranseq -h
  ```

3 Listing

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

```
/***** ranseq.c ***********************************
 * Description:
 * Author: Bernhard Haubold, haubold@evolbio.mpg.de
 * Date: Mon Jun  4 09:48:10 2012
 **************************************************/
#include <stdio.h>
```

```
#include <stdlib.h>
#include <time.h>
#include "interface.h"
#include "eprintf.h"
#include "ran.h"

int main(int argc, char *argv[]){
  int i, j, counter;
  long idum;
  char *c, *version;
  Args *args;
  FILE *fp;

  version = "0.9";
  setprogname2("ranseq");
  args = getArgs(argc, argv);
  if(args->h || args->e)
    printUsage(version);
  if(args->v)
    printSplash(version);

  /* seed and initialize random number generator */
  if(args->s != 0){
    idum = args->s;
  }else if((fp = fopen("randomSeed.dat","r")) != NULL){
    if(!fscanf(fp,"%ld",&idum))
      printf("WARNING[sample.initializeSample]:␣Something␣is␣wrong␣reading␣
          the␣the␣seed␣of␣the␣random␣number␣generator␣from␣randomSeed.dat.\n
          ");
    fclose(fp);
  }else
    idum = -time(NULL);
  init_genrand(idum);

  for(i=0; i<args->n; i++){
    printf(">Rand_%d;␣G/C=%.2f\n",i+1,args->g);
    counter = 0;
    for(j=0; j<args->l; j++){
      if(genrand_real1() < args->g){
        if(genrand_real1() < 0.5)
          c = "G";
        else
          c = "C";
      }else{
        if(genrand_real1() < 0.5)
          c = "A";
        else
          c = "T";
      }
      if(++counter < args->L)
        printf("%s",c);
      else{
        printf("%s\n",c);
        counter = 0;
      }
    }
    if(counter != 0)
      printf("\n");
  }
  /* save seed of random number generator */
  if(args->s == 0){
    fp = fopen("randomSeed.dat","w");
    fprintf(fp,"%d\n",(int)genrand_int32());
    fclose(fp);
  }
  free(args);
  free(progname());

  return 0;
}
```

4 Change Log

- Version 0.7 (June 4, 2012)
  - Complete rewrite; most important change: proper handling of seed for random number generator.
- Version 0.8 (August 16, 2012)
  - Fixed bug in handling of the `-g` option.
- Version 0.9 (June 2, 2017)
  - Removed `[inputfiles]` from interface.