

revComp, v. 1.6: Reverse-Complement a DNA Sequence

Bernhard Haubold

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

January 7, 2015

1 Introduction

2 Getting Started

revComp 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 revComp_XXX.tgz
```

where XXX indicates the version.

- Change into the newly created directory

```
cd RevComp_XXX
```

and list its contents

```
ls
```

- Generate revComp

```
make
```

- List its options

```
./revComp -h
```

- Test program

```
./revComp test.fasta
```

3 Listing

The following listing documents the driver program for revComp.

```
1 /***** revComp.c *****/
   * Description:
   * Author: Bernhard Haubold, haubold@evolbio.mpg.de
   * Date: Tue Jan 6 16:32:34 2015
   *****/
6 #include <stdio.h>
```

```

#include <stdlib.h>
#include <string.h>
#include "sequenceData.h"
#include "stringUtil.h"
11 #include "interface.h"
#include "eprintf.h"

#define LINELENGTH 70

16 void scanFile(FILE *fp, Args *args);

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

    version = "1.6";
    setprogname2("revComp");
    args = getArgs(argc, argv);
26 if(args->v)
    printSplash(version);
    if(args->h || args->e)
    printUsage(version);
31 if(args->numInputFiles == 0){
    fp = stdin;
    scanFile(fp, args);
} else{
    for(i=0; i<args->numInputFiles; i++){
36 fp = fopen(args->inputFiles[i], "r");
    scanFile(fp, args);
    fclose(fp);
    }
}
41 free(args);
free(progname());
return 0;
}

46 void scanFile(FILE *fp, Args *args){
    Sequence *seq;
    int i, c, residues;

    seq = NULL;
51 while((seq = getNextSequence(fp)) != NULL){
    if(args->r)
        rev(seq->seq);
    if(args->c)
        comp(seq->seq);
56 if(!(args->r || args->c))
        revcomp(seq->seq);
    c = 0;
    i = 0;
    residues = strlen(seq->seq);

```

```

61     printf("%s_ _", chomp(seq->id));
    if(args->r)
        printf("REVERSE");
    if(args->c)
        printf("COMPLEMENT");
66     if(!(args->c || args->r))
        printf("REVERSECOMPLEMENT");
    printf("\n");
    while(i < residues){
        if(c == LINELENGTH){
71             printf("\n");
            c = 0;
        }
        printf("%c", seq->seq[i]);
        i++;
76         c++;
    }
    printf("\n");
    freeSequence(seq);
}
81 }

```

4 Change Log

- Version 1.6 (January 6, 2015)
 - First version with standardized interface. valgrind still flags one lost memory block:


```
sh valgrind.sh
```