`keywordMatcher`, v. 1.20: Match a set of motifs against one or more sequences

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

August 25, 2020

1 Introduction

The program `keywordMatcher` implements set matching based on the keyword tree data structure as described by **?**. The program takes as input one or more patterns and one or more sequences in FASTA format. Its output are the match positions of the patterns.

2 Getting Started

`keywordMatcher` 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 keywordMatcher_XXX.tgz
  ```

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

  ```
  cd KeywordMatcher_XXX
  ```

  and list its contents

  ```
  ls
  ```
- Generate `keywordMatcher`

  ```
  make
  ```
- List its options

  ```
  ./keywordMatcher -h
  ```
- Run program with two patterns entered on the command line

  ```
  ./keywordMatcher -p 'CCGGGC|GCGGCG' mgGenome.fasta
  ```
- Include reverse strand

  ```
  ./keywordMatcher -r -p 'CCGGGC|GCGGCG' mgGenome.fasta
  ```

- Run program with a set of patterns passed as a FASTA-file

  ```
  keywordMatcher -r -f patterns.fasta mgGenome.fasta
  ```

- Visualize a keyword tree:

  ```
  keywordMatcher -t kt.tex -p ’ACA|AC|ACG|GCG’ ../Data/test.fasta > /dev/null
  ```

  This generates the LaTeX file `kt.tex` containing the keyword tree for the keywords `ACA`, `AC`, `ACG`, `GCG`. To view it, either include `kt.tex` in your LaTeX document, or typeset the wrapper file `ktWrapper.tex`, which was also generated.

3 Listing

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

```
/***** keywordMatcher.c *******************************************
 * Description: Match a set of patterns against a text using a
 *    keyword tree.
 * Reference: Gusfield, D. (1997). Algorithms on Strings, Trees,
 *    and sequences; Computer Science and Computational Biology.
 *    CUP, p. 52ff.
 * Author: Bernhard Haubold, haubold@evolbio.mpg.de
 * File created on Fri Aug 27 19:30:32 2004.
 *****************************************************************/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <time.h>
#include <assert.h>
#include "eprintf.h"
#include "stringUtil.h"
#include "keywordMatcher.h"
#include "ktree.h"
#include "construct.h"
#include "drawTree.h"

char *getWrapperFileName(char *fileName){
  char *buffer;
  int i;

  buffer = (char *)emalloc(1024 * sizeof(char));
  i = 0;
  while(fileName[i] != ’.’ && fileName[i] != ’\0’){
    buffer[i] = fileName[i];
    i++;
  }
  buffer[i] = ’\0’;
  strcat(buffer, "Wrapper.tex");
  return buffer;
}

char *getDvipsFileName(char *fileName){
  char *buffer;
```

```
  int i;

  buffer = (char *)emalloc(1024 * sizeof(char));
  i = 0;
  while(fileName[i] != ’.’ && fileName[i] != ’\0’){
    buffer[i] = fileName[i];
    i++;
  }
  buffer[i] = ’\0’;
  strcat(buffer, ".dvi");
  return buffer;
}

char *getPsFileName(char *fileName){
  char *buffer;
  int i;

  buffer = (char *)emalloc(1024 * sizeof(char));
  i = 0;
  while(fileName[i] != ’.’ && fileName[i] != ’\0’){
    buffer[i] = fileName[i];
    i++;
  }
  buffer[i] = ’\0’;
  strcat(buffer, ".ps");
  return buffer;
}

int main(int argc, char *argv[]){
  int i;
  Args *args;          /* arguments */
  char *version;       /* program version */
  FILE *fp;
  KTree *ktree;
  char *wrapperFile, *dvipsFile, *psFile;

  version = "1.20";
  setprogname2("keywordMatcher");
  args = getArgs(argc, argv);
  if(args->v)
    printSplash(version);
  if(args->h || args->e)
    printUsage(version);
  ktree = getKeywordTree(args);
  if(args->t){
    wrapperFile = getWrapperFileName(args->t);
    dvipsFile = getDvipsFileName(wrapperFile);
    psFile = getPsFileName(wrapperFile);
    fp = efopen(wrapperFile, "w");
    printLatexHeader(fp);
    fprintf(fp,"\\begin{center}\n\\input{%s}\n\\end{center}\n\\end{document
       }",args->t);
    fclose(fp);
    fp = efopen(args->t,"w");
    drawTree(fp, ktree->root);
    fclose(fp);
    fprintf(stderr, "#␣The␣keyword␣tree␣is␣contained␣in␣%s;␣to␣view␣it,␣
       either␣include␣it␣in␣your␣document,\n",args->t);
    fprintf(stderr, "#␣␣␣␣or␣run␣latex␣%s;␣dvips␣%s␣to␣generate␣%s\n",
       wrapperFile,dvipsFile,psFile);
    fprintf(stderr, "#␣To␣print␣output-set␣instead␣of␣node␣labels:␣adjust␣
       commenting␣at␣end␣of␣%s\n",args->t);
  }
  if(args->numInputFiles == 0){
    fp = stdin;
    searchSingleSeq(fp, args, ktree);
  }else{
    for(i=0; i<args->numInputFiles; i++){
      fp = efopen(args->inputFiles[i], "r");
      searchMultipleSeq(fp, args, ktree);
      fclose(fp);
    }
  }
  free(args);
  free(progname());
  return 0;
}
```

4 Change Log

- Version 1.13 (March 31st, 2016)
  - Cleaned up compiler warnings
  - Changed email address from fh-weihenstephan to evolbio.mpg
  - Included genome of Mycolpasma genitalium in distrubution (`mgGenome.fasta`)
- Version 1.14 (April 15, 2017)
  - Brought interface in line with standard UNIX behavior; in particular, abolished the `-t` option.
  - Included this documentation.
- Version 1.15 (June 7, 2017)
  - Replaced `construct.initializeTree()` by `construct.getNode()`.
  - Added code for drawing keyword tree in LaTeX.
- Version 1.16 (June 9, 2017)
  - Now generates LaTeX wrapper whenever the keyword tree is written to LaTeX.
- Version 1.17 (June 9, 2017)
  - Improved interface to prevent run on null pattern.
  - Improved sizing of the pspicture.
- Version 1.18 (November 14, 2017)
  - LaTeX-output now includes output set. This needs to be activated by uncommenting the corresponding lines at the end of the included file.

  - Improved placement of the pspicture.
- Version 1.19 (December 15, 2018)
  - Fixed bug in `getArgs` in `interface.c`.
- Version 1.20 (August 25, 2020)
  - Fixed bug in reading of patterns from `argv` or file. In both cases one byte too little was allocated.