SEG Filter

Robert Bakarić
rbakaric@irb.hr
bakaric@evolbio.mpg.de
13.04.2015
SegFilt-1.0

Abstract

This is a C++ implementation of SEG program written by Wootton and Federhen created for identifying and masking low complexity segments in amino acid sequences.

Contents

1 Installation **2**

2 Input files **2**

3 Program options **3**

4 Functions and classes **3**

5 Example **3**

5.1 SegFilt.cpp 3

5.2 SegFilt.hpp 4

6 Acknowledgement **4**

7 Future work **5**

1 Installation

The simplest way to compile this program is to:

1. Unpack the SegFilt package (`segfilt-XXX.tar.gz`):

   ```
   tar -xvzf segfilt-XXX.tar.gz
   ```

2. Change the current directory to `segfilt-XXX`:

   ```
   cd segfilt-XXX/
   ```

3. Configure the program for your system (–bindir is optional):

   ```
   ./configure --bindir=/absolute/directory/path/segfilt-xxx/bin
   ```

4. Compile the program:

   ```
   make
   ```

5. Install the program:

   ```
   make install
   ```

Your binaries should be located in your local bin directory if `--bindir` option has been set. Otherwise installation needs to be carried out with root privileges in order to be installed into `/usr/local/bin` directory.

2 Input files

The SegFilt takes a regular (mulit-)fasta file as input. The example can be found in `./segfilt-xxx/demo` and it should look like this:

`hox.fa`:

```
>gi|500757|gb|AAA86954.1| HOX A1 homeodomain protein [Homo sapiens]
MDNARMNSFLEYPILSSGDSGTCSARAYPSDHRITTFQSCAVSANSCGGDDRFLVGRGVQIGSPHHHHHH
HHHHPQPATYQTSGNLGVSYSHSSCGPSYGSQNFSAPYSPYALNQEADVSGGYPQCAPAVYSGNLSSPMV
QHHHHHQGYAGGAVGSPQYIHHSYGQEHQSLALATYNNSLSPLHASHQEACRSPASETSSPAQTFDWMKV
KRNPPKTGKVGEYGYLGQPNAVRTNFTTKQLTELEKEFHFNKYLTRARRVEIAASLQLNETQVKIWFQNR
RMKQKKREKEGLLPISPATPPGNDEKAEESSEKSSSSPCVPSPGSSTSDTLTTSH
>Rand_Seq_Prt
NWELYLYDPAGHRIRSWVSPNPVHFYADHCPYYPIFPRNVTTQWSPDTAGWDFEAPHTKHCTTVMMRRCALPDVIRSCSG
SSFRYRKYITKAHWICVMIWNHLSANKMKMGDQPWKECHYFKHVSCMANFAHPPVGGHKECVQCMFAWGCKNWFFNHVMP
ALKCWMKPGSEFCHHHHHHHHHHHHHHHHHHHHHHHHHHHERVCYMHNIHRNWYHGDQSYGDECILKPGIILEYVYRCVD
DCFHWWFCAKDEPHKLMSTSFPRIMCTPLMPGCIEARIMPLCWYADWLHRRQYDCWSLCKFCANNVTPHMYKLYNPQYLW
YIASNINVTDHRKLICKRWVDDPERNFGKLVSYWSGSDLSVVPRSQYPDWRNHHMNPYTNCANFYWILNYVDCNVLHRMI
YPSDMFMSAKIETRQQDDLIENLLAWWYKQKTAWMTGPRTPHFRNSTWWFHVWIYAPTRNDPANLMVCNWYGQYVYDILW
RNEGLNTVAILEKTDQMCGWAHCFPQHMCSKQSEQHNHIIIIIIIIIIIIIIIIIIHHHHHDHRKTASYENKSFVISPCQ
KNGRHRKQPTQFGHCVNSMEHSGYGLVTKFVINCHRNSMWNTKWTFIWADRAPRSWSKILGVFLNYATDDERKGSDGRLW
WKELVTFLRHKAQSCWHPVWECTADQCGRTNWQGQYLMNVGVCVHHFVSDVCMLQYPFVVNGTCAVMSKWKRHRYWVCFH
MDPYMEEYHKYWSRFPEWKAFPCNRPYKSRICRMMQMWTLAVQCQVITRFHGHWAESPKTILTFHCQFGKEHVQACVDKY
HFGAKLRPTLELQLWMKVAEAKISYFKRGMAATQHDNYYCEMNSPWLSLWHFIVFVHCINWEK
>Test|test|
RWVDDPERNFGKLVSYWSGSDLSVVPRSQYPDWRNHHMNPYTNCANFYWILNYVDCNVLHRMIFHCQFGKEHVQACVDKY
YPSDMFMSAKIMSAKIMSAKIMSAKIMSAKIMSAKIGPRTPHFRNSTWWFHVWIYAPTRNDPANLMVCNWYGQYVYDILW
LEILEILEILEILEILEILEILEFPQHMCSKQSEQHNHIFRGRHFGHKTFVKPQTDDCETTDHRKTASYENKSFVISPCQ
KNGRHRKQPTQFGHCVNSMEHSGYGLVTKFVINCHRNSMWNTKWTFIWADRAPRSWSKILGVFLNYATDDERKGSDGRLW
WKELVTFLRHKAQSCWHPVWECTADQCGRTNWQGQYLMNVGVCVHHFVSDVCMLQYPFVVNGTCAVMSKWK
```

3 Program options

I order to see program options type:

```
./bin/SegFilt -h
```

Expected output:

```
Usage: ./program [options]

***************************************************************************************
                                 SegFilt - SEG Filter
                                          by
                                    Robert Bakaric

CONTACT:
 Code written and maintained by Robert Bakaric,
 email: rbakaric@irb.hr , bakaric@evolbio.mpg.de

ACKNOWLEDGEMENT:
   Wootton, J. C. and S. Federhen (1993).  Statistics of local complexity in amino
   acid sequences and sequence databases.  Computers and Chemistry 17:149-163.

LICENSE:
 The program is distributed under the GNU General Public License. You should have
 received a copy of the licence together  with this software. If not, see
 http://www.gnu.org/licenses/
***************************************************************************************


Allowed options:
  -h [ --help ]                         produce help message
  -v [ --version ]                      print version information
  -i [ --input-file ] arg               Fasta input file
  -w [ --window ] arg (=12)             SEG window size
  -L [ --locut ] arg (=2.2)             Low complexity cutoff (starter)
  -H [ --hicut ] arg (=2.5)             High complexity cutoff (starter)
  -x [ --maxXes ] arg (=0)              Maximum number of xxx  symboles
                                        tolerated (dynamically defined if left
                                        unchanged)
  -t [ --maxTrim ] arg (=100)           Maximum trimming of raw segment
```

It should be noted that default values are set as in the original version of the program (seg).

4 Functions and classes

SEG class :

PUBLIC :

`SEG` : Constructor. It takes a set of predefined cut-offs and values as $unordered\_map < string, int|... >$ object, or it can be initialized without any parameters in which case constructors default values are assumed.

`SegFilt` : Function takes a standard STL string as an input and returns back the filtered version.

5 Example

5.1 SegFilt.cpp

A minimal example demonstrating the usage of SegFilt demo program:

```
./bin/SegFilt -i ./demo/hox.fa

>gi|500757|gb|AAA86954.1| HOX A1 homeodomain protein [Homo sapiens]
MDNARMNSFLEYPILSSGDSGTCSARAYPSDHRITTFQSCAVSANSCGGDDRFLVGRGVQIGSPxxxxxxxxxxPQPAT
YQTSGNLGVSYSHSSCGPSYGSQNFSAPYSPYALNQEADVSGGYPQCAPAVYSGNLSSPMVQHHHHHQGYAGGAVGSPQ
YIHHSYGQEHQSLALATYNNSLSPLHASHQEACRSPASETSSPAQTFDWMKVKRNPPKTGKVGEYGYLGQPNAVRTNFT
TKQLTELEKEFHFNKYLTRARRVEIAASLQLNETQVKIWFQNRRMKQKKREKEGLLPISPATPPGNDxxxxxxxxxxxx
xxxxxxxxxxTSDTLTTSH
>Rand_Seq_Prt
NWELYLYDPAGHRIRSWVSPNPVHFYADHCPYYPIFPRNVTTQWSPDTAGWDFEAPHTKHCTTVMMRRCALPDVIRSCS
GSSFRYRKYITKAHWICVMIWNHLSANKMKMGDQPWKECHYFKHVSCMANFAHPPVGGHKECVQCMFAWGCKNWFFNHV
MPALKCWMKPGSEFCxxxxxxxxxxxxxxxxxxxxxxxxxxxERVCYMHNIHRNWYHGDQSYGDECILKPGIILEYVYR
CVDDCFHWWFCAKDEPHKLMSTSFPRIMCTPLMPGCIEARIMPLCWYADWLHRRQYDCWSLCKFCANNVTPHMYKLYNP
QYLWYIASNINVTDHRKLICKRWVDDPERNFGKLVSYWSGSDLSVVPRSQYPDWRNHHMNPYTNCANFYWILNYVDCNV
LHRMIYPSDMFMSAKIETRQQDDLIENLLAWWYKQKTAWMTGPRTPHFRNSTWWFHVWIYAPTRNDPANLMVCNWYGQY
VYDILWRNEGLNTVAILEKTDQMCGWAHCFPQHMCSKQSEQHNxxxxxxxxxxxxxxxxxxxxxxxxDHRKTASYENKS
FVISPCQKNGRHRKQPTQFGHCVNSMEHSGYGLVTKFVINCHRNSMWNTKWTFIWADRAPRSWSKILGVFLNYATDDER
KGSDGRLWWKELVTFLRHKAQSCWHPVWECTADQCGRTNWQGQYLMNVGVCVHHFVSDVCMLQYPFVVNGTCAVMSKWK
RHRYWVCFHMDPYMEEYHKYWSRFPEWKAFPCNRPYKSRICRMMQMWTLAVQCQVITRFHGHWAESPKTILTFHCQFGK
EHVQACVDKYHFGAKLRPTLELQLWMKVAEAKISYFKRGMAATQHDNYYCEMNSPWLSLWHFIVFVHCINWEK
>Test|test|
RWVDDPERNFGKLVSYWSGSDLSVVPRSQYPDWRNHHMNPYTNCANFYWILNYVDCNVLHRMIFHCQFGKEHVQACVDK
YYPSDMFMSAKIMSAKIMSAKIMSAKIMSAKIMSAKIGPRTPHFRNSTWWFHVWIYAPTRNDPANLMVCNWYGQYVYDx
xxxxxxxxxxxxxxxxxxxxxxxxFPQHMCSKQSEQHNHIFRGRHFGHKTFVKPQTDDCETTDHRKTASYENKSFVIS
PCQKNGRHRKQPTQFGHCVNSMEHSGYGLVTKFVINCHRNSMWNTKWTFIWADRAPRSWSKILGVFLNYATDDERKGSD
GRLWWKELVTFLRHKAQSCWHPVWECTADQCGRTNWQGQYLMNVGVCVHHFVSDVCMLQYPFVVNGTCAVMSKWK
```

5.2 SegFilt.hpp

Adding the `SegFilt.hpp` header file to your program will allow you to include all the functions described in section 4. A minimal example:

```
    #include<string>
    #include<SegFilt.hpp>

    string ProtSeq = "\
VGRGVQIGSPHHHHHHHHHHPQPATYQTSGNLGVSYSHSSCGPSYGSQNFSAPYSPYAL\
NQEADVSGGYPQCAPAVYSGNLSSPMVQHHHHHQGYAGGAVGSPQYIHHSYGQEHQSLA\
LATYN";

/* Make object */

    /* Construction */
    SEG<int> seg;
    /*  OR  */
    SEG<int> seg(arg); // arg is : unordered_map<string, int|long|unsigned|double>


/*  Functions */


    string mask = seg.SegFilt(ProtSeq);
    /* mask = VGRGVQIGSPxxxxxxxxxxPQPATYQTSGNLGVSYSHSSCGPSYGSQNFSAPYSPYAL
     *       NQEADVSGGYPQCAPAVYSGNLSSPMVxxxxxxxGYAGGAVGSPQYIHHSYGQEHQSLA
     *       LATYN
     */
```

6 Acknowledgement

Wootton, J. C. and S. Federhen (1993). Statistics of local complexity in amino acid sequences and sequence databases. Computers and Chemistry 17:149-163.

Vakatov D, editor. The NCBI C++ Toolkit Book. Bethesda (MD):

National Center for Biotechnology Information (US); 2004-. Available from: *http : //www.ncbi.nlm.nih.gov/toolkit/doc/book/ch_getcode_svn*

7 Future work

Additional work is required in order to fully switch from current legacy code generic style c++. Therefore future work includes:

1. Increasing modularity
2. Rewrite legacy to STL
3. Pre checking : verify if the input sequence is protein or not and whether the file is in fasta format or not