Approximate String Overlap Finder

Introduction

This is an implementation for a solution for approximate all-pairs suffix prefix using Pigeon Hole principle. We called our solution Approximate String overlap finder (ASOF). This solution uses OpenMp to support multithreading.

To compile: make

Running the program

Apsp filename

The program has one parameter and four optional parameters:

filename: is the name of the file. Here is an example for the contents of the file:

```
ACCCCAT
TTTCCAGG
TTTGGCCAAA
```

where '\n' (new line) is the separator between input strings. The separator can be changed.

Optional parameters:

-p the number of threads which are used. (The maximum is the default)

-o Output. 0 : no output

1 : outputting all suffix prefix matches (default).

-m Minimal match length. (The default is 1).

-h Number of mismatches (The default is 3).

Examples

This command will find approximate overlaps using 4 threads. Minimal length is 10. The number of allowed mismatches is 2:

```
Apsp  test.txt -p 4  -m 10 -o 1 –h 2
```

Run the code sequentially

To run the code sequentially:

```
Apsp test.txt -p 1
```

Important

- You can generate random cases to test the code. The program 'gen' will generate random strings. The user specifies 3 parameters:
  1- K (number of strings)
  2- N (total length of all strings)
  3- If the generated strings have equal sizes.

The resulting file, test.txt, includes a string with the appropriate format.

- If you have a fasta file, please use the program 'converter' to convert a fasta file to a file with the right format. To run:

```
converter t1.fasta t1.txt
```

- You may supply your own file. An example:

```
AACCCCAAAA
CCCGGTTTAAAAAA
AAGTCCCC
```

- In Apsp.cpp, there is a constant MAX_K which determines the maximum number of strings which the program can accept. Please feel free to increase it and run make again.

- If you have any problem, please contact us:

Maan Haj Rachid
Qatar University
mh1108047@qu.edu.qa