Bioinformatics I
WS 15/16
Tutor: Alexander Seitz

1	2	3	$\sum$

Jonas Ditz
& Benjamin Schroeder

Blatt 7

(Abgabe am 19. October 2015)

Exercise 1 - *Smith Waterman today*

The 30 years old Smith-Waterman (SW) algorithm is one of the fundamental algorithms in bioinformatics. This dynamic programming (DP) algorithm allows in a biological sense for example the detection of Domains in a protein or a protein in a larger sequence. But the algorithm has to face the problem of time and space complexity of $O(n \times m)$. This complexity is getting a problem when starting to compare large sequences or even whole genomes. In the paper of Sandes et al., there is a model, which describes the problems of the space complexity as follows:" For instance, in order to compare two 33 MBP (Million Base Pairs) sequences, we would need at least 4.3 PB of memory." [Edans Flavius de O. Sandes, Alba Cristina M.A. de Melo] which was shown by D. Hirschberg [D.S. Hirschberg,]. This amount of space is at least today a major problem.

But the SW-algorithm has still a high importance in bioinformatics, which can be seen in the following examples. The classical example for todays usage of the Smith Waterman algorithm is the BLAST algorithm. Which uses the algorithm for seed refinement. Thereby the algorithm is accelerated with a vectorization process. Basically this is an improvment on the hardwarelevel of a computer, which can be adressed by the software for parallelizing the alignment. As a result, the Smith-Waterman algorithm is accelerated by a factor of 10 in comparison to the standard algorithm Novoalign (http://novocraft.com). Also an advantage in complexity matters is constraining the algorithm around seeds from the seeding step. [Li, H., Homer, N.]

Parallelizing is general a very modern topic for the Smith-Waterman Algorithm. Some research groups try to accelerate the algorithm by parallelizing the algorithm via the usage of the graphics processing unit (GPU). Some of the prominent algorithms are the CUDAlign1.0 [Sandes, E.F. de and Melo A.C.M.A. de] or the Weiguo Liu algorithm [W. Liu, B. Schmidt, G. Voss et al].
The CUDAlign Algorithm for example can align up to 32 MBP x 47 MBP. [Edans Flavius de O. Sandes, Alba Cristina M

Exercise 2 - *DP Algorithm for finding motifs*

```
for i = 0 to length(X)
  if X[i] == Y[0]
    Score = 0       for j = 0 to length(Y)
      if (X[i+j] == Y[j]) Score += 1
    if (Score > T)          print(motif found at position i)          i += length(Y)
```

Exercise 3 - *Programming assignment: Needleman-Wunsch alignment*

One task was to handle command line options. We wrote a new class to have a modular and easy way to provide this functionality. The java package *args4j* has everything we need. So we used that package to write our own option handler class. The code can be found in *CommandLineParser.java*. We did the same with gap penalties. A gap penalty class gives one the advantage of modular gap handling. If a gap penalties are needed, just reuse this class.
A new member function of our NeedlemanWunsch class takes care of file writing. If a user wants the resulting alignment in a FastA file, this function takes care of it.

References

[Li, H., Homer, N.] Li, H., Homer, N. (2010). A survey of sequence alignment algorithms for next-generation sequencing. Briefings in Bioinformatics, 11(5), 473–483. http://doi.org/10.1093/bib/bbq015

[Chen, B. and Xu, Y. et al] Chen, B. and Xu, Y. and Yang, J. and Jiang, H. (2010). A New Parallel Method of Smith-Waterman Algorithm on a Heterogeneous Platform. Springer. Algorithms and Architectures for Parallel Processing.79–90

[Edans Flavius de O. Sandes, Alba Cristina M.A. de Melo] Edans Flavius de O. Sandes, Alba Cristina M.A. de Melo (2013), "Retrieving Smith-Waterman Alignments with Optimizations for Megabase Biological Sequences Using GPU", IEEE Transactions on Parallel & Distributed Systems, vol.24, no. 5, pp. 1009-1021, doi:10.1109/TPDS.2012.194

[D.S. Hirschberg,] D.S. Hirschberg, “A Linear Space Algorithm for Computing Maximal Common Subsequences,” Comm. ACM, vol. 18, no. 6, pp. 341-343, 1975

[Sandes, E.F. de and Melo A.C.M.A. de] E.F. de, O. Sandes, and A.C.M.A. de Melo.(2010) “CUDAlign: Using GPU to Accelerate the Comparison of Megabase Genomic Sequence,”. Proc. 15th ACM SIGPLAN Symp. Principles and Practice of Parallel Programming (PPoPP). pp. 137-146

[W. Liu, B. Schmidt, G. Voss et al] W. Liu, B. Schmidt, G. Voss, A. Schroder, and W. Muller-Wittig. (2006). “Bio-Sequence Database Scanning on a GPU”. Proc. 20th Int’l Conf. Parallel and Distributed Processing (IPDPS).