

1	2	3	$\Sigma$

## 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 algorithm allows in a biological sense for example the detection of Domains in a protein or a Protein in a larger sequence. But the dynamic programmed algorithm has to face the problem of complexity of  $O(n \times m)$ . This complexity is getting a problem when starting to compare large sequences or even whole genomes. The classical example for todays usage of the Smith Waterman algorithm is the BLAST algorithm. [Li, H., Homer, N.]

### 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>