Assignment 1: Substring Search

SS 2018
Überblick – Assignment 1 (20P)

(1) Analyse transcription factor GATA2 (4P)

(2) Substring search (10P)

(3) Properties of Boyer Moore Algorithm (6P)
(1) Transcription Factor \textit{GATA2} (4P)

- \textit{GATA2} is a transcription factor with established or assumed roles in a variety of different human cancers
- Search \textit{GATA2} in the JASPAR database

- \textit{JASPAR} contains a set of transcription factor DNA-binding preferences, modeled as matrices
- Profiles derived from published collections of TF-binding sites
- Profile can be used to scan query sequences for presence of potential binding sites
(1) Transcription Factor \textit{GATA2} (4P)

- Search human \textit{GATA2} in the JASPAR database

- Compute the \textbf{information content} of each position in the position specific weight matrix (PSWM, aka frequency matrix)

- Find the exact formula on the web
(1) Transcription Factor GATA2 (4P)

Submit

- URL to the JASPAR information on GATA2 ([Isoform .2](#), length 5)
- Formula for information content used in sequence logos (1P)
- Frequency matrix and information content (IC) of every position of the PSWM (2P)
- Show your IC-calculation for at least one position ([Rechenweg 1x zeigen](#))
- List of cancer types to which GATA2 is associated and supporting papers from PubMed, 8x(Cancer, Title, PMID) (1P)
(2) Substring Search (10P)

(a) Load a template string (~60MB) into main memory (3P)
   • File: sequence.fasta
   • Don't use the concatenation parameter +

(b) Load a set of patterns (0P)
   • File: patterns.fasta

(c) Search all exact occurrences of all patterns in the template
    and print first ten positions to STDOUT (7P)
2.a) Load a sequence (3P)

- You need to load sequences in FASTA Format

  "A sequence in FASTA format begins with a single-line description, followed by lines of sequence data. The description line is distinguished from the sequence by a greater-than symbol (">") in the first column. ... The sequence ends if another line starting with a "">" appears; this indicates the start of another sequence"

- Example:
  
  > gi|5524211|gb|AAD44166.1| cytochrome b
  LCLYTHIGRNIYYGSYLYSETWNTGIMLLLITMATAFGMYVLPWGQMS
  EWIWGGFSVDKATLNRFFAFHFILPFTMVALAGVHLTFLHETGSNNPL
  LLLLLALLSPDMLGDPDNHMPADPLNTPLHIKPEWYFLFAYAILRSVP
  GLMPFLHTSKHRSMLRPLSQAFLWTLTMDDLTLTWIGSQP
  >gi|5454351|gb| cytochrome x
  LLLITMATAFMGYVLPWGQMSLCLYTHIGRNIYYGSYLYSETWNTGIM
  LLLITMATAFMGYVLPWGQMS

- File: sequence.fasta
(2.b) Load a Set of Patterns (0P)

- You will get another file which contains a set of sequences in FASTA format. These should be used as patterns.
- File: pattern.fasta
- Format as on previous slide
(2.c) Substring Search (7P)

- Implement an algorithm of your choice to search all occurrences of all patterns in the template.

- Methods `indexOf(„AT“), equals(„AT“), etc. are not accepted`
  - Use `charAt()` (to access a string like an array)

**Submit:**
- A Java Archive including class files and source code
- Commandline:
  ```
  java -jar GdBioinf1_[Gruppe].jar pattern.fasta sequence.fasta
  ```
- Print pattern, number of occurrences and first ten positions to STDOUT:
  ```
  tccgga: 2506
  [29562, 30667, 134810, 244142, 276754, 315062, 318466, 330540, 344995, 347336]
  gctacc: 6799
  ...
  ```
- Runtime of the algorithm and a sentence on the implementation (naive, Boyer Moore, ...)
For Orientation

Number of occurrences:

- tccgga: 2506
- gctacc: 6799
- taataa: 28279
- cctcagc: 17520
- cctgcagg: 2425
- ggcgccg: 141
- cccccccccc: 140
- aaaaaaaaaaaa: 52695
- aaaaaaaaaaaaaa: 44140
- aaaaaaaaaaaaaaaa: 25063
- aaaaaaaaaaaaaaaaaaaaaa: 8571
(3) Properties of the Boyer Moore Algorithm (6P)

1) Give a template and a pattern such that the BM algorithm, as presented in the lecture, needs to calculate in the order of \(|T| \times |P|\) characters comparisons and explain why (3P)

2) Many implementations of the BM algorithm actually drop the good suffix rule, especially for larger alphabets. Give an argument why and when this can be useful (3P)
Wichtig

- .py/.R/.jar auf gruenau2 testen!
- Wir testen mit neuem Pattern (Länge: 4-50bp, Alphabet: E = \{acgtn\})
- Bei Fragen: raik.otto@hu-berlin.de
Abgabe

- Abgabe bis 16.05.2018 um 23:59 Uhr

Upload here

https://box.hu-berlin.de/u/d/bdd2ffa6c66b4de5b2a6/
Dateiname: GdBioinf_[ Assignmentnummer ]_[ Gruppennummer ].zip
(z.B. GdBioinf_1_Gruppe_X.zip)

- Abgabe als .zip hochladen
- PDF mit Antworten zu 1, 2 und 3 zusammen abgeben
- Sourcecode