Übung 1
Algorithmische Bioinformatik

WS 15/16

Yvonne Mayer

Allgemeines

Ablauf der Übung

- Insgesamt 7 Übungszettel
- Abgabe in Gruppen von 2 Personen
- Pro Übung ~zwei Wochen Bearbeitungszeit

- 7 Pflichttermine (alle zwei Wochen)
  - Ausgabe neuer Übungszettel
  - Vorstellung der Lösungen letzter Übungszettel durch 2-3 Gruppen

- Termine dazwischen
  - Klärung von Fragen
  - Übungen nach Wunsch (nach Möglichkeit vorher Email)

- Webseite:
  https://www.informatik.hu-berlin.de/de/forschung/gebiete/wbi/teaching/archive/ws1516/ue_algbio/

Termine im Einzelnen

- Pflichttermine/Themen:
  - 22.10.2015, Naives Stringmatching, alle Gruppen
  - 03.11.2015/Do 05.11.2015, Boyer Moore
  - Di 17.11.2015/Do 19.11.2015, q-gram Index
  - Di 01.12.2015/Do 03.12.2015, Suffix Arrays
  - Di 15.12.2015/Do 17.12.2015, Globales Alignment
  - Di 12.01.2016/Do 14.01.2016, Approximatives Stringmatching
  - Di 26.01.2016/Do 28.01.2016, Neighbor Joining

- Ansonsten jeden Dienstag/Donnerstag Klärung von Fragen
  Ausnahme nächste Woche: keine Übung am 29.10.2015
  (stattdessen 4. VL Molekularbiologie), ein Übungstermin für alle am 27.10.2015

Übungsschein

- Schein: Voraussetzung für die Prüfung

- Abgabe der Übungszettel in Gruppen von 2 Personen
  - Mindestens 51% der Punkte von jedem Zettel benötigt
  - Gruppen bestehen/scheitern nur als Ganzes

- Vorstellung der Lösungen der letzten Übung durch 2-3 Gruppen
  - Wir losen vorstellende Gruppen aus
  - Ein Student der Gruppe muss Lösung vortragen
    → immer einen Vortrag parat haben
  - Wir behalten uns vor den Student zu bestimmen
  - Ziel: Jeder Student trägt einmal vor

Aufgaben

- Übungen sind implementationsintensiv
- Implementationen in JAVA

- I.d.R. sind Eingabe, Ausgabe und Aufrufform vorgegeben

- Textaufgaben
  - Abgabe als PDF
  - Müssen Gruppennnamen beinhalten

- Code-Abgaben
  - Abgabe als Jar (class und source files!)
  - Abgaben ohne Quellcode werden ignoriert
  - Dateiname: AssignmentX_GrXY.jar
  - Kompilierung unter Java 1.7 (oder niedriger)
  - Jar Datei auf gruenau2 testen!

- Nichteinhaltung der Regeln: Punkteabzug

Wettbewerb

- Einige Aufgaben (4-7) sind als Wettbewerb konzipiert
- Punkte gibt es (unabhängig von den Punkten für korrekte Lösungen) für die schnellste Lösung
- Messungen auf gruenau2:
  - `java version "1.7.0_85"`
  - `OpenJDK Runtime Environment (IcedTea 2.6.1) (suse-24.21.1-x86_64)`
  - `OpenJDK 64-Bit Server VM (build 24.85-b03, mixed mode)`
- Wir messen mehrmals mittels Linux „`time`“
  → Parallelisierung lohnt sich nicht
- Beste Gruppe bekommt am Ende der Veranstaltung eine kleine Überraschung

Gruppeneinteilung

(1) Aufteilung in zwei Gruppen (Di/Do)

- wenn > 20 angemeldete Teilnehmer
- ansonsten: Abstimmung Übung Di oder Do

(2) Aufteilung in Gruppen von 2 Personen zur Abgabe der Übungszettel

- Gruppen bitte bis Dienstag in Goya eintragen als „GruppeX“

  Gruppe1: X, Y
  Gruppe2:
  Gruppe3:
  Gruppe4:
  Gruppe5:
  Gruppe6:
  Gruppe7:

Übung 1

Übung 1.1

(1) Sequenzen lesen (5 Punkte)

- Für die Aufgaben müssen Sie die Dateien einlesen, die DNA Sequenzen enthalten
- Die Dateien sind im Fasta Format
  - Zitat: „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 indicatesthe start of another sequence“
  - Beispiel:

```
> gi|5524211|gb|AAD44166.1| cytochrome b
LCLYTHIGRNIYYGSYLYSETWNTGIMLLLITMATAFMGYVLPWGQMS
EWIWGGFSVDKATLNRFFAFHFILPFTMVALAGVHLTFLHETGSNNPL
LLLLLALLSPDMLGDPDNHMPADPLNTPLHIKPEWYFLFAYAILRSVP
GLMPFLHTSKHRSMMLRPLSQALFWTLTMDLLTLTWIGSQP
>gi|5454351|gb| cytochrome x
LLLITMATAFMGYVLPWGQMSLCLYTHIGRNIYYGSYLYSETWNTGIM
LLLITMATAFMGYVLPWGQMS
```

Übung 1.2

(2) Stringmatching (8 Punkte)

- Implementieren Sie einen String-Matching-Algorithmus, der alle Vorkommen eines Patterns (kurzer String) in einem Template (langer String) findet
- Sie müssen diesen Algorithmus selbst implementieren; STRING.indexof(), Pattern.matches(), etc. reicht nicht!
- Beispiel GATATC:

```
>Some FASTA Sequence
aaaattattctaaggagatacgcgagagggcttctcaaatttattcagagatggatgtttttagatggtggtttaagaaaagcagta
ttaaatccagcaaaactagaccttaggtttattaaagcgaggcaataagttaattggaattgtaaaataattcttcttcatttgttg
gaggaaaactagttaacttcttaccccatgcaggGATATCccatagggtcgaatacgatctgtcactaagcaaaggaaaatgtgagt
gtagactttaaaccatttttattaatgactttagagaatcatgcatttgatgttactttcttaacaatgtgaacatatttatgcgat
taagatgagttatgaaaaaggcgaatatattattcagttacatagagattatagctggtctattcttagttataggacttttgacaa
gatagcttagaaaataagattatagagcttaataaaagagaacttcttggaattagctgcctttggtgcagctgtaatggctattgg
tatggctccagcttactggttaggttttaatagaaaaattccccatgattgctaattatatctatcctattgagaacaacgtgcgaa
gatgagtggcaaattggttcattattaactgctggtgctatagtagttatccttagaaagatatataaatctgataaagcaaaatcc
tggggaaaatattgctaactggtgctggtagggtttggggattggattatttcctctacaagaaatttggtgtttactGATATCctt
ataaataatagagaaaaaattaataaagatgatat
```

Übung 1.1 und 1.2: Programmaufruf und Ausgabe

- Auf der Übungs-Homepage sind zwei Dateien bereitgestellt, die das Template und mehrere Patterns enthalten
  - Template: Chromosom 20 des Menschen (50MB)
  - Patterns: Schnittstellen von Restriktionsenzymen

- Programm muss wie folgt aufrufbar sein:
  - `java -jar Assignment1_GrXY.jar file1 file2`
  - file1: Dateiname Patterns, file2: Dateiname Template
  - file1 und file2 sind unkomprimiert
  - Ausgabe auf STDOUT
  - Pro Paar (Template/Pattern) muss das Programm ausgeben:
    - Patternlänge
    - Anzahl Fundstellen
    - Startpositionen der ersten 10 Fundstellen

Übung 1.3

(3) Patternlänge und Laufzeit (7 Punkte)
(nicht Teil des Wettbewerbs)

- Erstellen Sie einen Plot (PDF), der die Laufzeit in Abhängigkeit der Patternlänge (5-50bp) darstellt.
  Diskutieren Sie das Ergebnis.

- Optional: Testen sie wie sich unterschiedliche Alphabetgrößen auf die Laufzeit auswirken.

Wettbewerb

- Umfasst Aufgaben 1.1 und 1.2
- Wir messen die Gesamtlaufzeit Ihres Verfahrens über alle Pattern (mittels Linux „`time`“)
- Wur verwenden 20 neue Patterns, deren Länge zwischen 5 und 50 Zeichen liegt (Alphabet E = {ACGTN})
- Wettbewerbspunkte:
  - Platz 1: 3 Punkte
  - Platz 2: 2 Punkte
  - Platz 3: 1 Punkt

Zur Orientierung

Anzahl Vorkommen der Pattern im Template

- tccgga: 2506
- gctacc: 6799
- taataa: 28279
- cctcagc: 17520
- cctgcagg: 2425
- ggcgcgcc: 141
- ccccccccccc: 140
- aaaaaaaaaaa: 52695
- aaaaaaaaaaaa: 44140
- aaaaaaaaaaaaaaa: 25063
- aaaaaaaaaaaaaaaaaaaa: 8571

Abgabe

- Abgabe bis Sonntag den 01.11.2015 um 23:59 Uhr
- Abgabe per Email an: mayeryvo@informatik.hu-berlin.de
  (gerne auch Fragen zur Übung per Email)
- Abgabe wie beschrieben (PDF mit Plot, Jar Datei mit Quellcode)
- Bitte die geschätze Bearbeitungszeit mitteilen
  (z.B. 10h für zwei Teilnehmer)