Assembly Pipeline

Projektmanagement im Softwarebereich – SeqAn

David Weese
April 2009

Inhalt

- Fragmente erzeugen**| read simulator**
- SWP Teilprojekte
- Projektplan

FRAGMENTE ERZEUGEN

readsim.r

Parameter:

```
# Parameters for a tiny example
seqLength = 100              # Source sequence length
numOfReads = 10              # Number of reads to simulate
avgReadLength = 10           # Average read length
numOfRepeats = 2             # Number of copies (repeats,haplotypes, ...)
amountOfSNPs = 1             # Number of mismatches per copy
amountOfIndels = 0           # Number of indels per copy
indelBaseRange = 4:6         # Indel size range
errorPerBaseCall = 0.01      # Sequencing error rate for each base in a read
simulateMatePairs = 1        # 0 = no mate pairs are simulated,
                             # 1 = all reads are in a mate pair
librarySizes = c(30, 40)     # Mean library sizes for mate pairs
librarySd = c(5, 10)         # Standard deviations for library sizes
avgRepeatLength = 20         # 0 = Overcompressed repeat is simulated,
                             # otherwise the repeat is implanted into the sequence
```

Erzeugung starten:

R CMD BATCH readsim.r

Ausgabe

Vier Dateien:

1. source.fasta — simuliertes Genom (1 Contig mit Repeats)
2. reads.fasta — simulierte Reads mit Sequenzierfehlern
3. repeats.fasta — eingebaute Repeats
4. library.txt — Mate-Pair Libraries (Means und Standardabw.)

reads.fasta:

>9471,9432[id=1,mateId=4287,libraryId=2]
TGCTACACTAAAGCAATCCCCCTAACCCCCAGCCGCGCG
>7335,7374[id=4287,mateId=1,libraryId=2]
GCGTAGATCACCTGAGGGAGTTTCGGATGGCCAGCCAGA
>5268,5232[id=2,mateId=4288,libraryId=2]
GTATCTTATTATCTTAATTAGGTATGTGATCTAATT

repeats.fasta:

>Repeat1
TTTTATCCTATGAGTGTTTTCGGTGCGAAGTAGCAACATGACTCTGGGCA
>Repeat2
TTTTACCCTAATAATGTCACGATTGAGAACCAGGACTGTAGGCA

library.txt:

>libraryId=1
(1000,100)
>libraryId=2
(2000,200)

SWP TEILPROJEKTE

Assembly Pipeline

(siehe Webseite, assembly.pdf)

1. **Fragment Store** | Datenstruktur für Fragmente, Mate-Pairs, Libraries
2. **Repeat Screener** | Finde Fragmente aus bekannten/de novo Repeats
3. **Overlapper** | q-gram basiertes Filtern + Overlap-Alignments
4. **Unitigger** | Overlap-Graph + konsistentes Layouten
5. **Scaffolder I** | Contig-Mate-Graph + edge bundling
6. **Scaffolder II** | greedy Path-Merging
7. **Repeat Resolution** | Heuristiken um Lücken im Scaffold zu schließen
8. **Consensus** | Multi-Alignment, Consenus, Profile, N50-score

1. _____ 2. _____

3. _____ 4. _____

5. _____ 6. _____

7. _____ 8. _____

PROJEKTPLAN

Projektplan

20min Vortrag bestehend aus:

- Benutzte Schnittstelle von vorderen Modulen
- Vorstellung der eigenen erforderlichen Datenstrukturen
- Schnittstellen zu hinteren Modulen
- Algorithmen die benutzt werden
- Welche davon gibt es schon in SeqAn, welche werden neu implementiert

Termin:

- Montag 27.4., 13-17 Uhr
- A6 Raum 017

Hinweise

SVN Repository:

- https://www.seqan.de/svn/trunk/teaching/swp09/

Kommunikation/Kooperation:

- Nutzt Plattformen zur projektweiten Kommunikation (Foren, Wikis, …)
- Baut ein gemeinsames Gerüst für die Pipeline
- Sprecht euch regelmäßig mit Modulnachbarn ab

Deployment:

- Nicht mit dem Testen auf die ersten echten Ergebnisse des/der Vordermanns/-frau warten
- Erst mit kleinen Probleminstanzen beginnen und auf Korrektheit testen
- Vor dem Einchecken auf Kompilierfähigkeit prüfen
- Regelmäßig einchecken