

Projektauswertung für das Modul
Projektmanagement im Softwarebereich in der
SeqAn Arbeitsgruppe

Annkatrin Bressin

24. Mai 2013

Inhaltsverzeichnis

1	Implementierung	2
2	Unterschiede zum Projektplan	2
3	Implementierungsprobleme	2

1 Implementierung

Eine genaue Dokumentation unserer Implementierung befindet sich als HTML Datei im SVN Dateipfad *gruppe5/apps/blastX/Dokumentation/doku* oder im Anhang. Um den unten stehenden Ausführungen folgen zu können, befindet sich alternativ in Abbildung 1 ein Überblick aller Funktionen mit kurzen Beschreibungen.

2 Unterschiede zum Projektplan

Die Implementierung der BlastX Variante orientiert sich im Allgemeinen an den Projektplan vom 24.04.2013. Es gibt keine Veränderungen im Aufbau und an den Aufgaben der Kernfunktionen (*groß geschriebene Funktionen*).

Unterschiede zeigen sich bei der Erstellung der Alphabete. Hier wird in der Funktion *make_cluster()* keine Nächste-Nachbarn-Klassifikation (*k-Nearest-Neighbor*) durchgeführt sondern ein Hierarchisches Cluster Verfahren.

Das Programm besitzt im Gegensatz zum Projektplan mehrere Hilfsfunktionen wie *append_to_match_found()*, *get_read_position()*, *verify_seed_match()* und *get_position_in_prot()*. Diese machen das Programm übersichtlicher und übernehmen zur Planungszeit unbekannte Funktionalitäten.

Zusätzlich ist die Funktion *known_position()* implementiert, um überflüssige Durchläufe beim Verifizieren zu vermeiden und das Programm hinsichtlich der Laufzeit zu optimieren. Die *hash()*-Funktion ist jetzt ein Templateaufruf. Zuerst ist sie für Integer Werte konzipiert worden, später haben wir uns für den platzsparenderen SeqAn-Datentyp *AminoAcid* entschieden und den Eingabetyp variabel gehalten.

3 Implementierungsprobleme

Allgemeine Probleme bei der Implementierung waren Unklarheiten bei der Benutzung der Seqan Bibliothek durch unvollständige Dokumentationen. Außerdem gab es Probleme, die betriebssystemabhängig sind, und daher schwer zu lokalisieren waren.

Spezifische Probleme gab es bei der Anwendung der SeqAn-Funktion *Backtracking*. Der Grund war zum einen die fehlende Dokumentation, zum anderen Fehler in der Implementation. Bei der Erstellung der Auswertung ist aufgefallen, dass die BlastX Variante Probleme mit großen Datenmengen hat, was wahrscheinlich auf Speicherprobleme zurückzuführen ist. Für zukünftige Projekte ist es ratsam früher mit der Auswertung zu beginnen, damit Probleme rechtzeitig erkannt und behoben werden.

Trotz der aufgeführten Komplikationen ist das Projekt erfolgreich beendet worden. Die Implementierung verlief innerhalb des aufgesetzten Zeitplanes und in ständiger Absprache mit dem Betreuer Jochen Singer.

Tabelle 1: Implementierungsübersicht und kurze Beschreibungen

QUELLTEXT	FUNKTIONEN	BESCHREIBUNG
own_functions.h	GET_DATAS() <i>template</i> hash() <i>template</i> Variable <i>Klasse</i> Match_found <i>Klasse</i>	Einlesen der Datenbanken Hash-Funktion Kommandozeilen Parameter Speicherung Treffer
blastX.cpp	main()	Roter Faden
alphabet.cpp	GET_ALPHABET() make_cluster()	Erstellung der Alphabete Hierarchisches Cluster
translate.cpp	TRANSLATE_READS() TRANSLATE_DATABASE() translate() get_translate_from_codon() get_Amino_Acid_Pos()	Übersetzung aller Reads Übersetzung aller Proteine Übersetzung Read Übersetzung Codon Adresse zum Alphabet
find.cpp	FIND_MATCHES() append_to_match_found() get_read_position()	Subsequenz Suche Speicherung in Match_found Zuordnung zum Read
verify.cpp	VERIFY_ALL() verify_seed_match() known_position() get_position_in_prot() append_to_match_found()	Überprüfung aller Teiltreffer Bewertung und Auswertung Aussortierung von Intervallen Proteinposition aus Alignment Speicherung in Match_Found
output.cpp	WRITE_TO_FILE()	Ausgabe der Treffer