392177
Bioinformatische Strategien zur Identifikation von Pathogenitätsdeterminanten in Metagenom-Sequenzdaten

A. Fust, B. Osterholz, W. Pätzold

13. April 2015

Inhaltsverzeichnis

1 Einleitung **3**

2 Vorbereitung **3**
2.1 Erstellung pHMMs 3

3 Benutzung der Pipeline **4**
3.1 Datenbank aus pHMMs erstellen und durchsuchen 4
3.2 Blastp Suche 4
3.3 Overview 5
3.4 Pubmedsuche und Aktualisierung der Overview 5
3.5 Sortieren der Overview 6

1 Einleitung

Alle hier benutzen Skripte müssen auf Linux Maschinen ausgeführt werden um ihre Funktionalität zu gewährleisten.

Die hier vorgestellte Pipeline dient zur Identifizierung möglicher Antibiotika-Resistenzgene. Dafür wird eine selbst erstellte Datenbank aus profile hidden Markov models (pHMMs) mit einem Metagenomdatensatz durchsucht. Die Anzahl möglicher Treffer kann durch das Setzen eines E-Value Cutoffs weiter gefiltert werden. So gefundene Treffer werden mittels einer NCBI-Blastp Suche verifiziert um eine Entscheidung über die Glaubwürdigkeit des Treffers machen zu können. Falls der beste Blastp Treffer einen Pubmed Eintrag besitzt, wird der Abtract von diesem nach vorher festgelegten Schlüsselwörtern durchsucht. Die so erhaltenen Ergebnisse werden in einer Tabelle gespeichert. Zusätzliche Angaben wie die Coverage der einzelnen Biogasanlagen, die zugehörige Klasse, Proteinsequenz usw. können ebenfalls in der Tabelle abgelesen werden.

2 Vorbereitung

2.1 Erstellung pHMMs

Um ein pHMM zu erstellen müssen die Namen der Sequenzen, welche für ein spezifisches pHMM genutzt werden sollen, in einer Datei gespeichert werden. Überflüssige Zeichen werden mit dem Pythonskript IDSforGrepFormat.py entfernt, dies ist für die spätere Funktion des Skrips grep_ids notwendig.

Aufruf : python IDSforGrepFormat.py <input> <output>

Anschließend können die zu den Namen gehörigen Sequenzen mit dem Pearlskript grep_ids aus der entsprechenden Datenbank extrahiert und in eine Datei gespeichert werden.

Aufruf: ./grep_ids <input> <Datenbank> read > <output>

Die so erhaltenen Sequenzen werden zu einem multiplen Muscle Alignment aligniert und im FASTA Format gespeichert.

Aufruf: muscle <input> <output>

Um ein pHMM zu erstellen muss sich das Alignment im Stockholm Format befinden, hierfür kann das Muscle Alignment mittels eines Online Tools konvertiert werden (z.B.
http://sequenceconversion.bugaco.com/converter/biology/sequences/fasta_to_stockholm.php).
Das pHMM wird nun durch hmmbuild erstellt (Version: 3.8.31).

Aufruf: hmmbuild <output> <input>

Wurden mehrere pHMMs erstellt kann durch einen Phylogenetischen Baum überprüft werden, ob die verwendeten Sequenzen den korrekten pHMMs zugeordnet wurden und es keine Überschneidungen zwischen verschiedenen pHMMs gibt (Überprüfung z.B. mit Fastree) .

3 Benutzung der Pipeline

3.1 Datenbank aus pHMMs erstellen und durchsuchen

Aus den generierten pHMMs wird durch das Shellskript HMMFolderScan.sh eine Datenbank generiert, die anschließend mit dem Metagenomdatensatz durchsucht wird. Beim Aufruf des Skripts kann ein Cutoff mit angegeben werden der bei der Suche in der Datenbank somit nur Treffer über dem gewünschten Wert liefert. Sollen alle Treffer ausgegeben werden, wird ein leerer String angegeben.

Innerhalb des Skriptes wird das Shellskript createFolder.sh aufgerufen, welches dafür sorgt, dass alle benötigten Ausgabe Ordner erzeugt werden. Dafür ist es nötig, dass sich dieses Skript im selben Ordner wie das Shellskript HMMFolderScan.sh befindet.

Aufruf: sh HMMFolderScan.sh <EValue> <input Folder> <output Folder>

Aufruf alle Treffer: sh HMMFolderScan.sh " " <input Folder> <output Folder>

Bei der pHMM Suche kann es vorkommen, dass ein Gen mehrere Treffer für verschiedene pHMMS liefert. Um für die weitere Analyse nur das pHMM zu verwenden, welches den Besten Score für ein Gen liefert, müssen die erhaltenen Daten noch unique sortiert werden. Dies geschieht mit dem Shellskript uniquer.sh. Hier werden die gefundenen Treffer im ersten Schritt zuerst nach Gennamen und dann nach Score sortiert, anschließend werden die besten Treffer behalten und der Rest verworfen. Sollte ein pHMM an mehreren Stellen des Gens treffen, so wird die erste Domäne behalten und alle weiteren verworfen.

Aufruf: sh uniqer.sh 1 <input > <output>

3.2 Blastp Suche

Die uniquen Treffer können nun mit dem Shellskript pipline_blast.sh gegen die NCBI Datenbank geblastet werden. Dies geschieht, um die Glaubwürdigkeit gefundener Treffer zu verstärken und zu überprüfen, ob das gefundenen Gen schon bekannt ist, oder ob es sich um ein neues ähnliches Gen handelt.

Für den Aufruf ist es wichtig, dass die Skripte createFolder.sh, grep_ids_to_files.pl und run_jobs.py in dem gleichen Ordner wie pipline_blast.sh liegen, da diese von dem Skript aufgerufen werden.

Aufruf: sh pipline_blast.sh <input >

Die Ausgabe dieser Blastp Suche erfolgt im *.html und *.txt Format. Das *.txt Format wird für weitere Schritte verwendet, während das *.html für den Benutzer gedacht ist, um direkt über die angegebenen Links zu den Ergebnissen des NCBIs zu gelangen. Des weiteren erstellt dieses Skript für jedes unique Gen, welches ein pHHM getroffen hat eine *.faa Datei, die die Proteinsequenz enthält.

3.3 Overview

Im nächsten Schritt wird mit dem Shellskript Overview.sh eine vorläufige Overview Datei erstellt, hierfür müssen alle Dateien aus der pipline_blast.sh Suche (*.html, *.faa, *.txt & Outputdatei Blastsuche) in einem Ordner liegen.

In dem selben Ordner wie dieses Skript müssen außerdem die Skripte createOverview.py und createFolder.sh liegen.

Aufruf: sh Overview.sh <input >

Die so erzeugte Datei, enthält die jeweilige Coverage der vier Biogasanlagen, den Gennamen, das pHMM (welches den Treffer gefunden hat), die Klasse der β-Lactamase, den Score des pHMMs, den Evalue des pHMMs, den besten blastp Treffer, den dazugehörigen Evalue, die Identität dieses Treffers zu dem blastp Treffer, die Subject Accession Nummern, die Subject Titels, die Subject Tay IDs, die Subject IDs, falls vorhanden Links zu Pubmed Treffern (zu diesem Zeitpunkt noch leer) und als letzten Eintrag die Gensequenz.
Dabei sind die Einträge im ersten Schritt nach den pHMMs sortiert und innerhalb dieser nach ihrem Score.

3.4 Pubmedsuche und Aktualisierung der Overview

Diese Overview bildet die Grundlage für die Pubmedsuche. In dieser wird für die besten Blastp Treffer der dazugehörige Abtract, falls vorhanden, nach vordefinierten Schlagworten durchsucht. Diese Schlagworte können in dem Skript UrltoPubmed.sh geändert werden. Das Pythonskript linksuche.py wird mit der Overview aufgerufen und erzeugt für jedes gefundene Gen eine GenID.acc Datei in der die URLs der NCBI Seiten gespeichert sind.

Aufruf: python linksuche.py <input: Overview>

Mittels des Shellskripts FoldertoPubmed.sh werden die in den GenID.acc Datein gespeicherten Links aufgerufen und die entsprechenden Seiten zwischengespeichert. Enthalten diese Verweise auf Pubmed Einträge, werden die entsprechenden Publikationen ebenfalls zwischengespeichert und nach den oben beschriebenen vordefinierten Schlüsselwörtern durchsucht. Bei erfolgreichen Treffern, werden Contig, Genname und Trefferlink in einer *.pubhit Datei gespeichert. Im letzten Schritt, werden alle *.pubhit Dateien sortiert und zu einer einzelnen *.pubhits Datei zusammengefügt.

In dem selben Ordner wie dieses Skript muss außerdem das Shellskript UrltoPubmed.sh liegen, da dieses von dem FoldertoPubmed.sh Skript aufgerufen wird.

Aufruf: sh FoldertoPubmed.sh <input Folder > <output Folder >

Die so gefundenen Pubmed Treffer können mit Hilfe des Pythonskripts linkzuordnung.py der Overview Datei hinzugefügt werden, dabei wird der entsprechende Link zu dem passenden Gen gespeichert.

Aufruf: python linkzuordnung.py <overview file > <pubhits file >

3.5 Sortieren der Overview

Das Shellskript sortOverview.sh kann dazu genutzt werden, um die Overview Datei nach gewünschten Spalten zu sortieren um sich einen besseren Überblick zu verschaffen.

Aufruf: sh sortOverview.sh <Spalte > <input > <output>