

## Übungen zur Vorlesung „Molekularbiologische Datenbanken“

### Aufgabenblatt 6: Gene Ontology

---

Abgabetermin: **13.07.2003** bis **17 Uhr** über e-mail oder in RUD25 IV.104

Abzugeben: siehe Aufgaben

Maximal: 20 Punkte, Erforderlich: 15 Punkte.

Namen nicht vergessen!

---

Nachdem wir die Informationen über die aus der RNA resultierenden Proteine in die Datenbank gespeichert haben, wollen unsere Partner auch noch wissen, mit welchen biologischen Prozeß, welcher molekulare Funktion und welcher zellulären Komponente diese Proteine annotiert. Kurz gesagt, sie wollen auch noch die Information der GeneOntology integrieren.

Dazu stellen Sie uns 2 Dateien zur Verfügung:

term.txt mit

- internal\_id int
- GO-name char
- term\_type char (molecular function, ... )
- acc char (GO:00030434, ...)
- is\_obsolete int (nicht mehr aktuell)
- is\_root int (gibt nur eine root)

und für die Beziehung zwischen einem parent und einem child

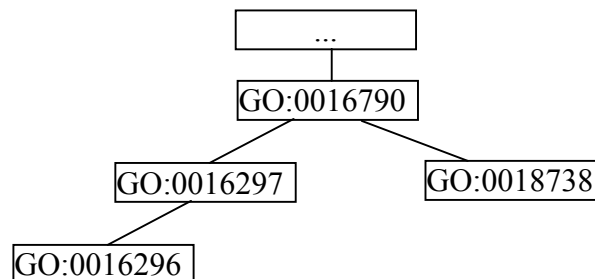
parent\_child.txt

- id int
- relationship\_type\_id int (partof, isa)
- parent\_id int (id von term.internal\_id)
- child\_id int (id von term.internal\_id)

Die einzelnen Werte in den Dateien sind durch tabs voneinander getrennt.

1. Ihre Aufgabe ist es nun, diese Daten in das bestehende Datenbank-Schema zu integrieren und dazu die entsprechenden Tabellen zu erstellen.  
Abgabe: SQL Scripts der neuen Tabellen **2 P**
2. Wie auch bei QuickGO und anderen Retrieval Systems sollen wir in der Lage sein, mit einer Anfrage durch den Nutzer alle ancestors (Vorgänger) eines bestimmten GO-terms abrufen zu können. Ermitteln Sie die Vorgänger mit Hilfe einer Datenbank-Funktion(PL/pgSQL). Innerhalb der Vorgänger soll auch eine Ordnung herrschen, das bedeutet, der Abstand für jeden GO-Term zur ursprünglichen Anfrage soll mitgeliefert werden.  
Abgabe: Funktion zur Ermittlung der Vorgänger **4 P**
3. Für BIR4\_HUMAN, ein baculoviral IAP repeat-containing protein 4 mit der SWISSPROT-Accession Number P98170 sind in den Datenbank-Referenzen GO-Accession Nummern gespeichert. Zu diesen sollen Sie die Namen, und für jede GO-Accession Nummer eine Liste von deren Vorgängern ausgeben. Es sollen dabei die GO-Accession Nummer, der Name und der Term Typ ursprünglichen GO-Accession Nummer auch noch die Namen der Vorgänger und deren Abstand zu der gegebenen GO-Accession ausgegeben werden (order by term\_type, dist!).  
Abgabe: Query und Ergebnis **4 P**
4. Wir bekommen eine Liste von GO-Accession Numbers (GO\_Acc.txt). Speichern sie diese Liste als Tabelle in der Datenbank, um darauf mit einer Datenbank-Funktion zugreifen zu können. Die GO-Accession Nummer GO:0016296 hat mit jeder GO-Accession Nummern von GO\_Acc.txt mindestens einen gemeinsame Knoten. Berechnen Sie den/die kleinsten gemeinsamen Knoten zwischen GO:0016296 und jeder GO-Accession Nummern in GO\_Acc.txt mit Hilfe einer Datenbank-Funktion. Geben Sie neben der kürzesten Distanz auch die GO-Accession Nummer und den Namen des gemeinsamen Knotens an.

Beispiel:



das bedeutet, die Entfernung zwischen GO:0016296 und GO:0018738 ist der kleinste gemeinsame Knoten ist GO:0016790 mit einer kürzesten Pfadlänge von 3.

Abgabe: Funktion/SQL und deren Ergebnis für jede GO Accession in dem File **5 P**

5. Von dieser Liste (GO-Acc.txt) sollen alle gemeinsame Knoten aller mit Hilfe einer Datenbank-Funktion errechnet werden und diese mit GO-Accession, dem Namen und dem Term Type ausgegeben werden. Es soll außerdem erkennbar sein, welcher der kleinste gemeinsame Knoten ist.

Beispiel: GO:0030371 -> GO:0045182 -> GO:0003674 -> GO:0003673

GO:0030533 -> GO:0003674 -> GO:0003673

dann ist der kleinste gemeinsame Knoten GO:0003674, ein weiterer GO:0003673

Abgabe: Funktion/SQL und das Ergebnis

**5 P**

Die Berechnungen der Vorgänger soll möglichst schnell sein (Tip: Beziehungen vorberechnen und speichern)