

Übungen zur Vorlesung „Molekularbiologische Datenbanken“

Aufgabenblatt 5: Gene Ontology

Abgabetermin: **15.07.2003** bis **17 Uhr** über e-mail oder in RUD25 IV.102

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 integriert haben, wollen unsere Partner auch noch wissen, für welchen biologischen Prozeß, welche molekulare Funktion und in welcher zellulären Komponente diese vorkommen. 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

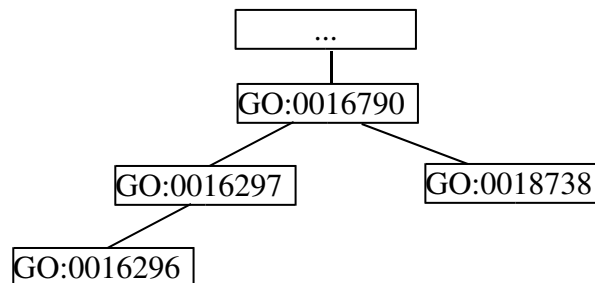
parent_child.txt

- 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 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: neue Tables **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 eines GO-terms muß eine Ordnung herrschen.
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 Numbers gespeichert. Zu diesen wollen Sie die Namen, und für jede GO-Accession-numbers eine Liste von deren Vorgängern. Es sollen neben der GO-Accession Numbers, des Namens und des Term typen der eigentlichen GO-Accession Numbers 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 von einer Datenbank-Funktion zugreifen zu können. Die GO-Accession number GO:0016296 hat mit den GO-Accession Numbers von GO_Acc.txt gemeinsame Knoten. Berechnen Sie den/die kleinsten gemeinsamen Knoten zwischen GO:0016296 und jeder GO-Accession number in GO_Acc.txt mit Hilfe einer Datenbank-Funktion. Geben Sie neben der kürzesten Distanz auch die GO-Accession number 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 und deren Ergebnis

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: Function und das Ergebnis

5 P

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