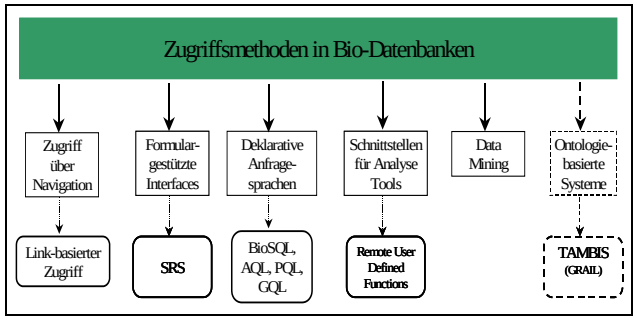
Kapitel 7: Zugriffsmethoden in Bio-Datenbanken

- Navigation
- Stichwortsuche
  - Formulargestützte Interfaces
  - SRS
- Deklarative Anfragesprachen
- Schnittstellen für Analyse-Tools
- Data Mining

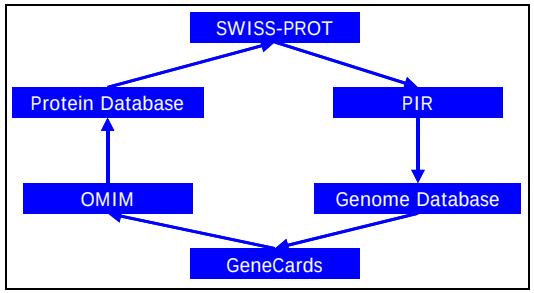


Navigation

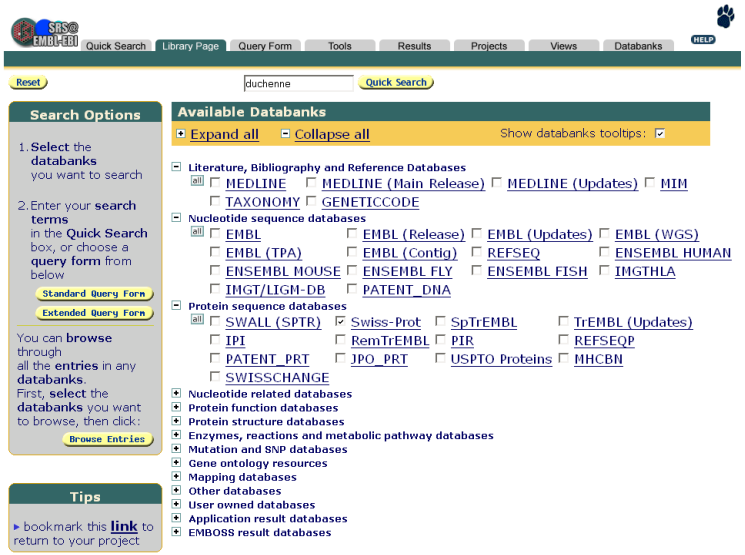
- Ansatz
  - Browsen in den Datenbeständen; Zugriff auf benötigtes (Zusatz)-Wissen über html-Links
- Vorteile
  - Einfach zu realisieren
  - Für Standardfälle effizient
- Problematik
  - Wenig flexibel, kein Muster-basierter Zugriff
  - Verlinkung unterliegt der "Willkür" der DB-Anbieter
  - "Lost in Hyperspace"-Phänomen
  - Referentielle Integrität schwierig zu wahren
- Wesentliche größere Flexibilität erst durch Bio-Ontologien
  - Ontologie: Explizite begriffliche Formalisierung eines Anwendungsbereiches (Fachsprache)
  - Dazu mehr in Kapitel 8 (Integration von Bio-Datenbanken)

Navigation: Beispiel

- Stichwort: Duchenne Muskeldystrophie
- Startpunkt:
  - Swiss-Prot (EBI) mit Stichwort "Duchenne" (DMD_HUMAN; menschliche Duchenne Muskeldystrophie)

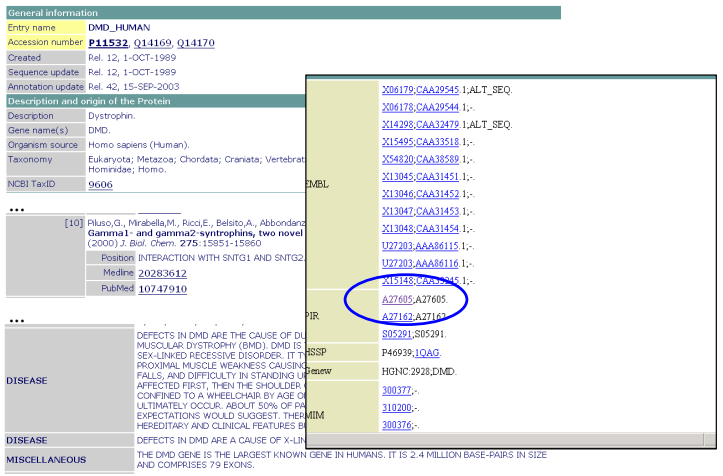


Swiss-Prot: Eingabe



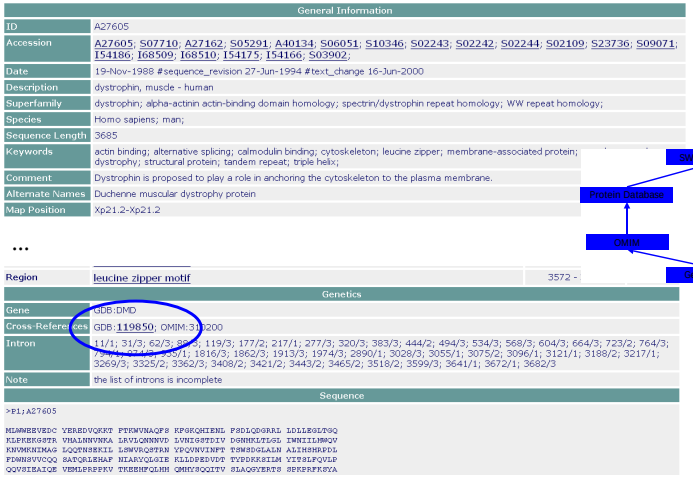
Swiss-Prot: Ausgabe (Auszug)

- Kurze Beschreibung (mit Lit.-Referenzen etc.)
- Link u.a. zu Datenbank PIR (Protein Information Resource): A27605



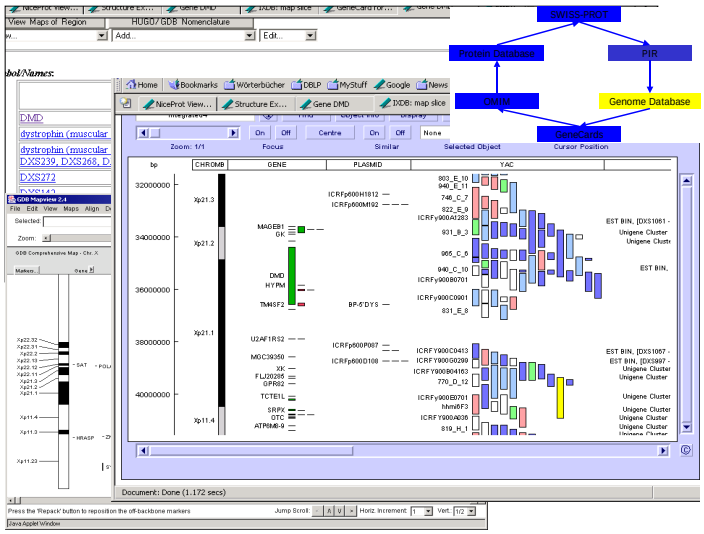
PIR-Eintrag

- Link u.a. zu Datenbank GDB (Genome Database): GDB:119850



GDB-Eintrag

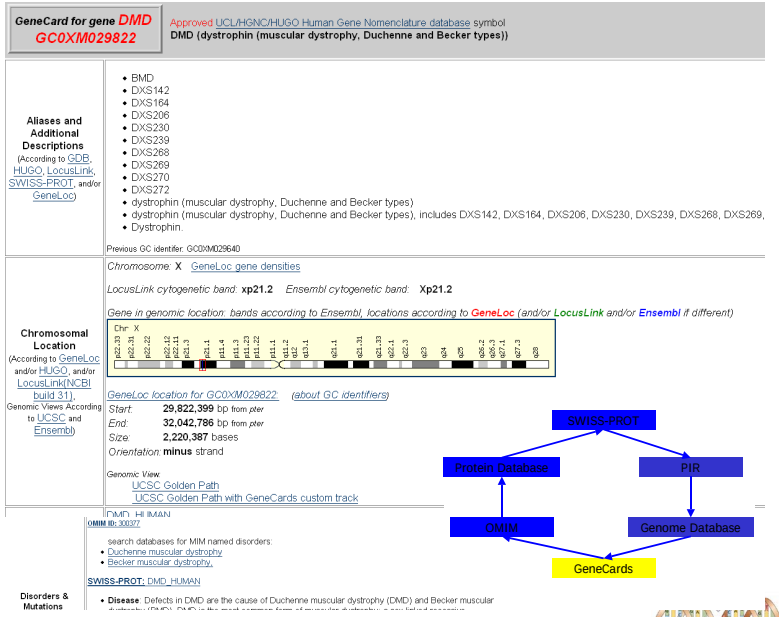
- Genome Database*
- Aliase, Clone, Loci
- Karten
- Link to GeneCard DMD



* http://gdbwww.dkfz-heidelberg.de/

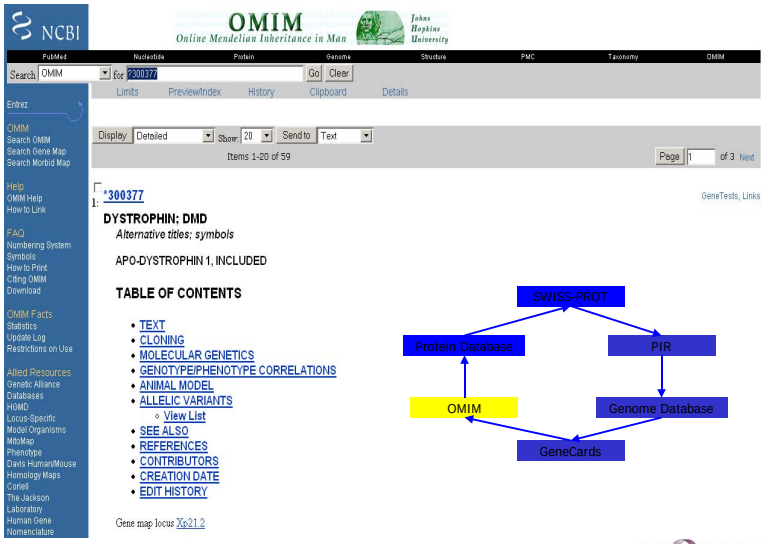
Gene-Card DMD

- SNPs, Mutationen, ...
- Link u.a. zu OMIM ID: 300377

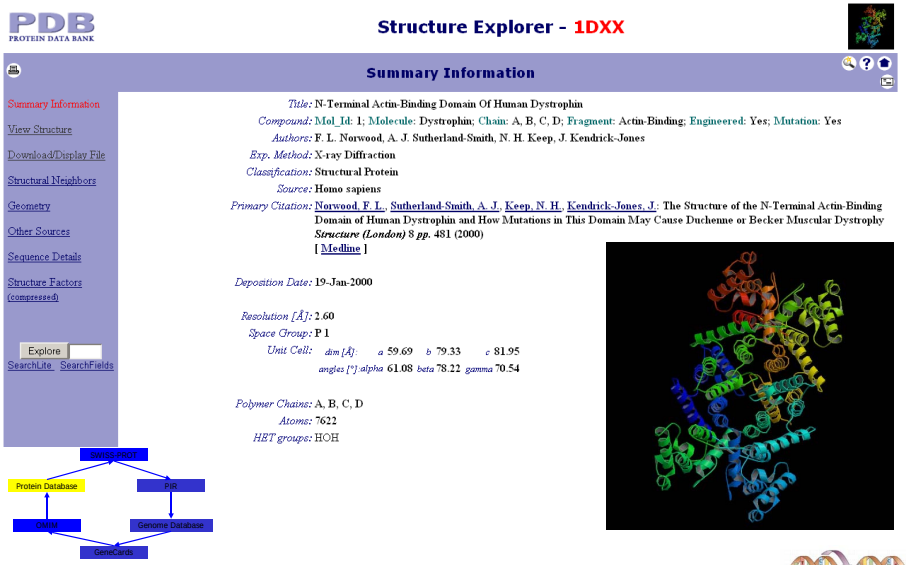


OMIM

- OMIM 300377 DMD
  - Beschreibung
  - Klinischer Verlauf
  - Molekulare Grundlagen
  - Populationsgenetik
  - Mutationen und Häufigkeiten
  - Links u.a. zu PDB

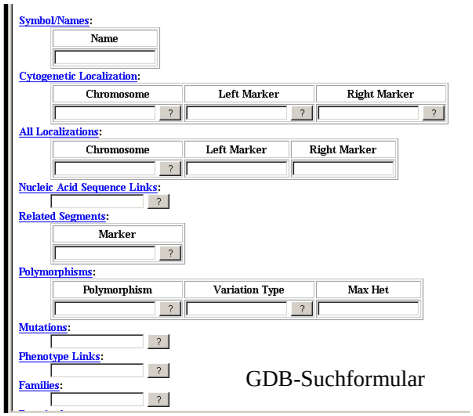


PDB



Stichwortsuche / Suchformulare

- Typische Zugriffsmöglichkeit im Web (Google, Altavista, Internet-Shopping etc.)
  - Einfach, schnell, verständlich, bekannt
- Vorstrukturierte Suchformulare
- Verwendung von Methoden des IR
  - Ranking der Ergebnisse
  - Operatoren: AND, OR, NOT, + / –
- Probleme
  - Ergebnis nicht zwingend Treffer
  - Wortformen: Zeiten, Sing. / Plural, Casus, ...
  - Synonym / Homonymprobleme
  - Treffer sind Dokumente, nicht Attribute

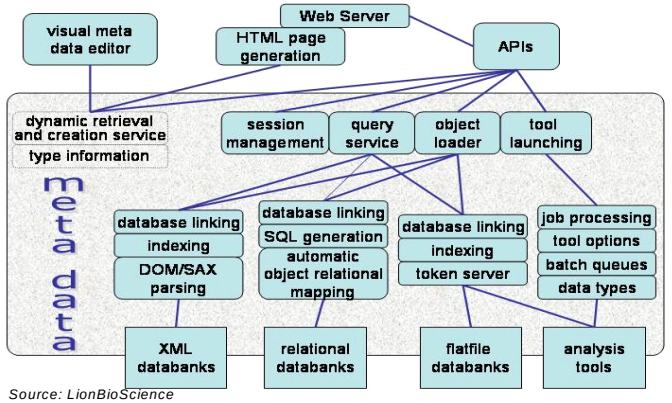


GDB-Suchformular

- Nachteil: Starke Einschränkung der Expressivität, keine Unterstützung vom komplexen Anfragen

SRS (Sequence Retrieval System)

- Ursprünglich als Zugriffstool für Sequenzdatenbank EMBL entwickelt
  - Mehr als 400 wissenschaftliche Datenbanken ansprechbar
  - Einheitliche graphische Nutzerschnittstelle
  - Formularbasiert
  - Fokus auf Flatfile-Datenbanken, Weiterentwicklung zur Unterstützung relationaler Datenquellen
  - Übernahme und kommerzieller Vertrieb durch LION BIOSCIENCE*

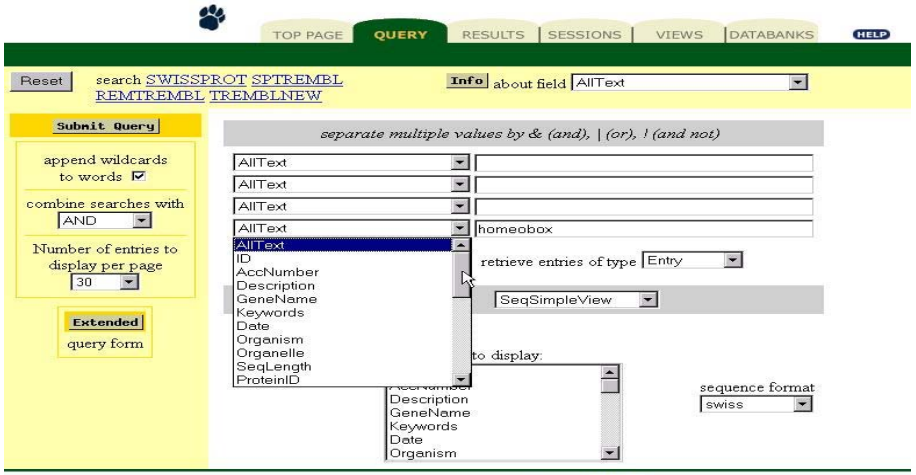


Source: LionBioScience

* http://www.lionbioscience.com

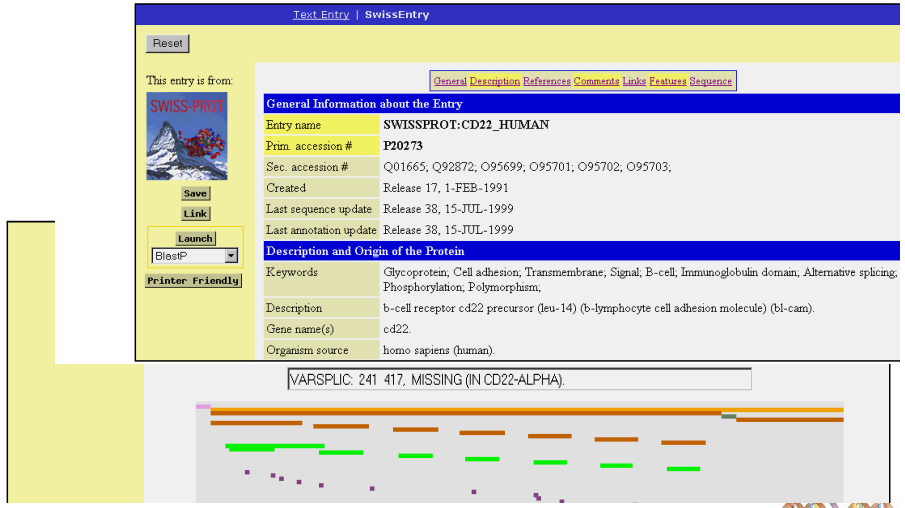
SRS: Konzept (1)

- Ein "Konzept" wird über eine Anzahl von Feldern (Entries, Attributen) beschrieben
  - Beispiel: Konzept *Gen*



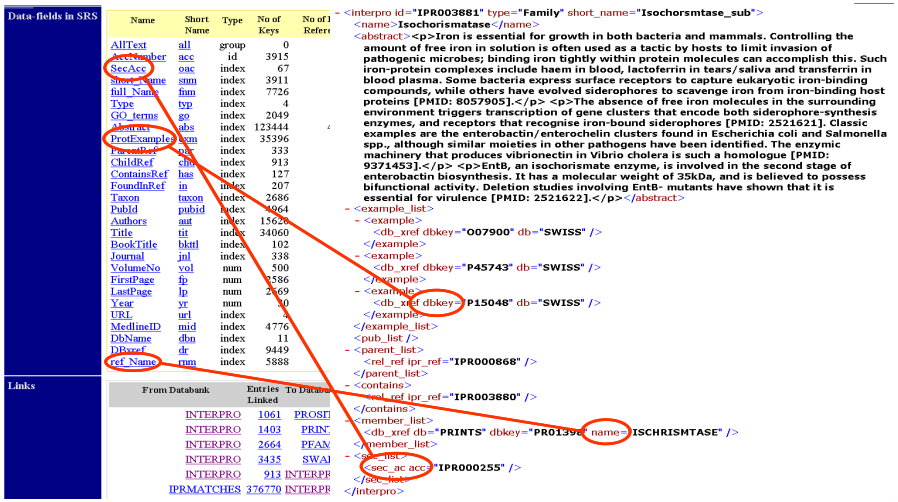
SRS: Konzept (2)

- Ein Konzept kann als html-Seite visualisiert werden



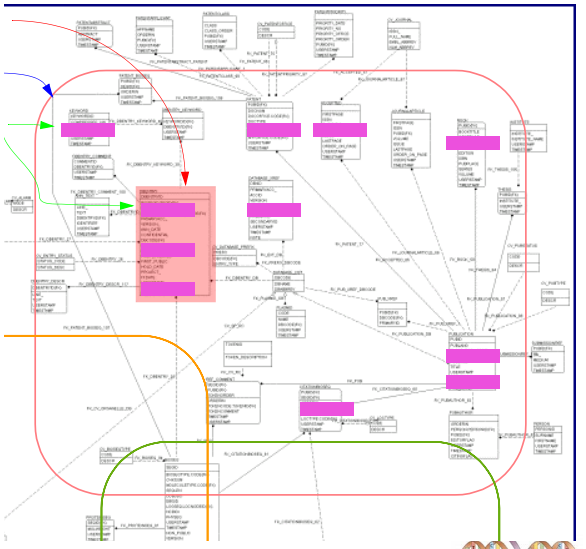
SRS: Konzept (3)

- Export als SRS-"Objekt" (C++, Java, Perl, CORBA) oder Speicherung als XML



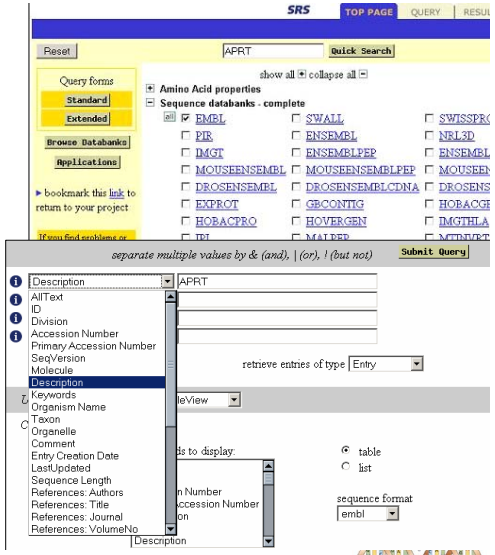
SRS: Zugriff auf relationale Datenbanken

- Auswahl einer *hub*-Tabelle (als "Aufhänger"-Konzept) (→)
- Angabe der Tabellen die zum Konzept "dazu gehören" (→)
- Angabe der abfragbaren Attribute (→)
- Interne Umsetzung via Views/ Joins



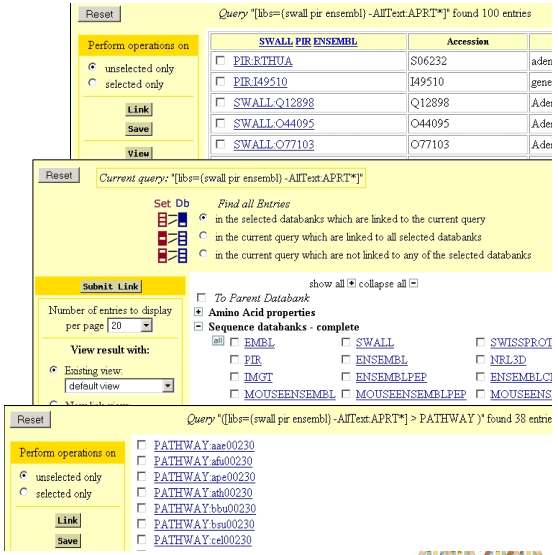
SRS: Datenzugriff (1)

- Objektsuche
  - Auswahl von Datenquellen
  - Spezifikation der Suchkriterien für abfragbaren Attribute
  - Schnittmenge der Attribute aller ausgewählten Quellen (z.B. ID)
- Unterstützte Query-Möglichkeiten
  - Textsuche, Bereichsuche für numerische/ Datumattribute
  - Regulär Ausdrücke
- Automatische Übersetzung von SRS-Queries nach SQL zum Zugriff auf relationale Datenbanken
- Ergebnis als Vereinigung der Suchergebnisse über einzelne Quellen



SRS - Datenzugriff (2)

- Querverweis-Suche
  - Für Objekte einer Ergebnismenge oder für eine ganze Quelle
  - Referenzierte Objekte in anderen Datenquellen
- Automatische Bestimmung der Pfade zwischen Quellen
  - Shortest-Path-Algorithmus
  - Gleiche Semantik der Beziehungen
- Tool-Integration
  - Anwendung auf Ergebnismengen der Abfragen
  - Anzeige direkt im Web-Browser
  - Große Anzahl von Tools bereits integriert

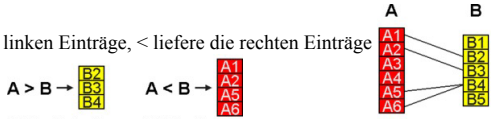


SRS Anfrage Sprache

- Anfragen können an vielen Stellen direkt formuliert werden, z.B. Results -> Search using a query expression
- Anfrage-Syntax
  - Stringsuche: [Menge-Mengenattribut:Suchmuster], Menge kann eine Datenbank, DB-Gruppe, Index, Index-Gruppe oder Suchausdruck sein
  - z.B. [pir-des:elestase], oder [swissprot-AllText:duchenne*]
  - Wildcards: [swissprot-aut:sanger,f*!coulson,a*]
  - Reguläre Ausdrücke: [swissprot-aut:/mue?ller/]
  - Zahlenbereiche: [swissprot-SeqLength#400:500]
  - mehrere DBs: [{swissprot swissnew sptrembl}-des:kinase], [dbs={swissprot swissnew sptrembl}-des:kinase] &[dbs-org:human]

SRS Anfrage Sprache

- Einfache Anfragen kombinieren:
  - operand operator operand …
  - z.B. verlinken: [swissprot-AllText:duchenne*] > pdb
- Operatoren
  - logisch: | oder, & und, ! aber nicht
  - Links:
    - > liefere die linken Einträge, < liefere die rechten Einträge



    - Komplexe Verweise: (q = [{swissprot swissnew}-des:kinase])!(q<swissnew)
  - Hierarchische Suche
    - >^ liefere Teilbaum-Einträge definiert durch linke Seite
    - >_ liefere Blatt-Einträge des Teilbaums definiert durch linke Seite

SRS Anfrage Sprache

- Mehrfache Links
  - [swissprot-AllText:duchenne*] >omim
    OMIM-Einträge zu denen man direkt von SwissProt-Einträgen mit Begriff "duchenne" gelangt
  - [swissprot-AllText:duchenne*] >pdb >omim
    wie oben, doch SwissProt und OMIM müssen über PDB verlinkt sein
  - [swissprot-id:acha_human] > prosite > swissprot
    Suche nach Eintrag "acha_human", Link nach Prosite (Protein Fam.) "neuronal acetylcholine receptors", Link zu Swissprot Einträgen => Sequenzen aus einer Proteinfamilie
  - [swissprot-id:gshr_caeel] > prodom > pdb
    kein direkt Link von „gshr_caeel" zur PDB, aber über ProDom (Protein Domainen), Ergebnis homologe Proteine zu „gshr_caeel"

SRS Anfrage Sprache

- Einträge und Untereinträge
  - [swissprot-keywords:transmembrane] alle Einträge mit Keyword Transmembr.
  - [swissprot-ftkey:transmem] Menge von Untereinträgen mit Typ transmem
- Einträge und Untereinträge können über Links kombiniert werden
  - [swissprot-org:human] > [swissprot-ftkey:transmem]
    Menge der Transmembransegemente in menschlichen Proteinen
  - [swissprot-org:human] < [swissprot-ftkey:transmem]
    Menge der menschlichen Proteine mit Transmembransegmenten
  - [swissprot-ftkey:transmem] > parent
    Konvertierung der Untereinträge zu den zugehörigen Einträgen
  - [swissprot-ftkey:transmem] > parent | [swissprot-key:transmembrane]
    Einträge mit Transmembransegementen oder mit dem Schlüsselwort "transmembrane"

SRS Anpassung

- Komplizierte Anfragen können vordefiniert werden
- Perl-ähnlich Sprache ICARUS
- Beispiel: Suche einen Swissprot-Eintrag nach Zugriffsnummer oder Beschreibung

```
$CannedQuery:[sampleQuery
prompt:|Sample canned query
options:{
        $AppOpt:[ac prompt:'Access number' defStr: 'Q1']
        $AppOpt:[des prompt:'Description' defStr:'cancer']
}
queryStr:
@"[swissprot-des:($des)*]|[swissprot-acc:($ac)*]"
]
```

- Das @ zeigt, das die Anfrage zur Laufzeit bearbeitet wird.
- Datei mit Anfragedefinition wird ähnlich wie ein HTML Link in SRS-Seite eingebunden.

Aufruf von SRS aus HTML Seiten

- Wegtz kann als CGI Programm direkt als HTML Link aufgerufen werden
- Einfaches Anzeigen von Einträgen
  - wgetz?-e+[embl-id:rnelas]
  - wgetz?-e+[{embl%20emblnew}-acc:X012345]
  - Die Option –e lässt wegtz volle Einträge anzeigen
- Anzeigen von Mengen
  - wgetz?[embl-all:elastase]
  - wgetz?swissprot
  - wgetz?swissprot+-lv+30+-bv+31 (-lv Anzahl Einträge pro Seite, -bv erster Eintrag
  - wgetz?swissprot+-lv+30+-bv+31+-view+SequenceSimple
    zeigt ZusatzInfos mit Hilfe eines vordef. Views
  - wgetz?swissprot+-lv+30+-bv+31+-view+SequenceSimple+-ascii+-rs+||+-cs+@@
    Wie oben aber als HTML Tabelle

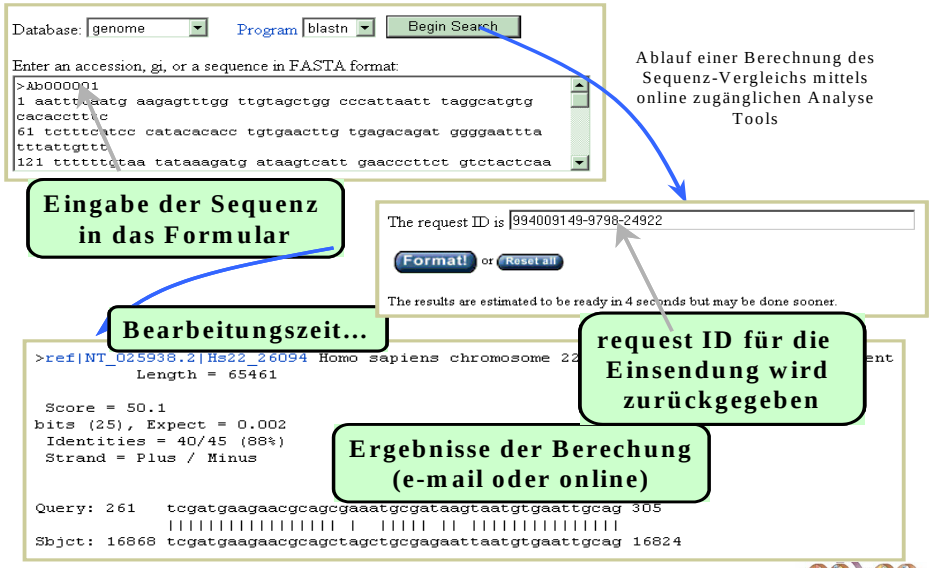
Kapitel 7 (Forts.): Schnittstellen für Analyse-Tools

- Vielzahl an Analyseprogrammen für Verarbeitung von Bio-Daten, z.B.
  - CLUSTALX (Graphisches Tool für ClustalW multiple sequence alignment program)
  - FASTA
  - BLAST (Basic Local Alignment Search Tool)
  - Sacc3D (Structural Information for Yeast Proteins)
- Installation auf lokalem Rechner vs. Remote-Zugang über Web-Interface
- Installation auf lokalem Rechner: Effizient, aber Update-Problematik
- Zugang über Web-Interface
  - Via Formulare und ftp/e-mail: Einfach zu realisieren, aber oft umständlich und wenig flexibel
  - Via API von Remote User Defined Functions (RUDFs): Flexibler, aber komplexer
  - Realisierung von RUDFs als Internet Functions

Eigenschaften von Internet Functions (IFs)

- Kein Teil der Datenbank
- Werden von einem externen System gestartet
- Erreichbar über das Internet
- Beinhalten Kommunikationsprotokoll zum Datenaustausch mit der Datenbank
- Verschachtelungen mehrerer IFs beim Aufruf möglich

Beispiel-Szenario



Internet Function Definition Language (IFDL)

- Erweiterung der SQL DDL
  - *Kein* Teil des SQL-Standards
  - Nicht verwechseln mit der *Independent Form Definition Language* (Abk. auch IFDL)
- Definition für die Einbindung von IFs in SQL-Queries
- IF-Definition: 5-Tupel
  - Funktionsname der aufzurufenden IF
  - URL der Funktion
  - Liste von Eingabeparametern mit Typen
  - Typ des Rückgabewertes
  - HTQL*-Wert

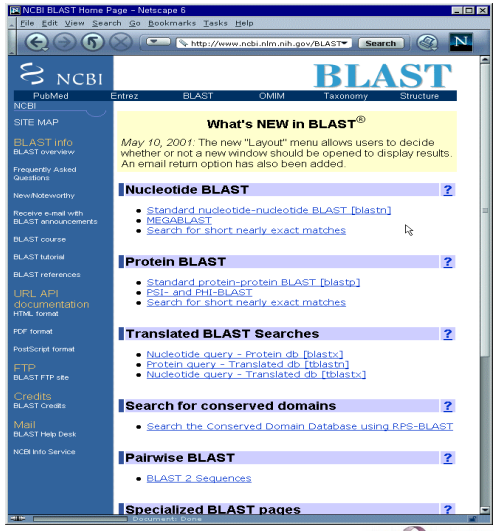
* Hyper Text Query Language

Hyper Text Query Language (HTQL)

- Unterstützt Suche in Web-Dokumenten
- Nutzung der Tags in HTML/XML-Dokumenten zur Navigation und Ausgabestrukturierung (ähnlich wie bei XPath, XQuery)
- Dient im Zusammenhang mit IFs als Schnittstelle zw. DB und IFs
  - HTQL-Wert in IFDL Function Definition informiert über Position des Rückgabewertes im (generierten) Web-Rückgabedokument
  - IF extrahiert für weitere Verarbeitung Rückgabewert aus dem (generierten) Web-Dokument

Beispiel für IF-Ausführung (GenBank)

- Vergleich von neuen, noch uncharakterisierten Gensequenzen (in Tabelle *local*) mit bekannten Sequenzen der Kenianischen Fruchtfliege
  - Nur solche Sequenzen von Interesse, die Ähnlichkeit von mind. 98 % zu geg. Sequenz aufweisen
  - Für Berechnung der Editierdistanz Verwendung des über Internet Schnittstelle zugänglichen Programms BLAST (http://ncbi.nlm.nih.gov/blast/blast.cgi)



Beispiel für IF-Ausführung (2)

- Verwendung von zwei IFs: *blast*, *get_seq*
  - *blast* verwendet als Eingabe eine Sequenz der Tabelle *local* und liefert "requestID"-Wert (rein "technischer" Parameter für weitere Referenz auf Anfrage, vor allem wg. oft langer Laufzeit von *blast*)
  - Ergebnis von *blast* befindet sich nach Ausführung in von *Blast.cgi* erzeugter HTML-Datei im Eingabefeld des 1. form-Tags
  - Dieser Wert wird extrahiert und dient als Parameter für *get_seq*, dessen Ergebnis nach dem 2. pre-Tag der erzeugten HTML-Datei steht
  - Ergebnis von *get_seq* ist Tabelle, welche die ermittelten Ähnlichkeitsscores, Herkunft und Art enthält

define function *blast*
href "http://www.ncbi.nlm.nih.gov/blast/Blast.cgi"
parameters *query* varchar(10000)
results *request_id* varchar(40)
htql value: <form>.<input>;

define function *get_seq*
href "http://www.ncbi.nlm.nih.gov/blast/Blast.cgi"
parameters rid varchar(40)
results *sequence* varchar(10000)
htql value: <pre>.<pre>;

Beispiel für IF-Ausführung (3)

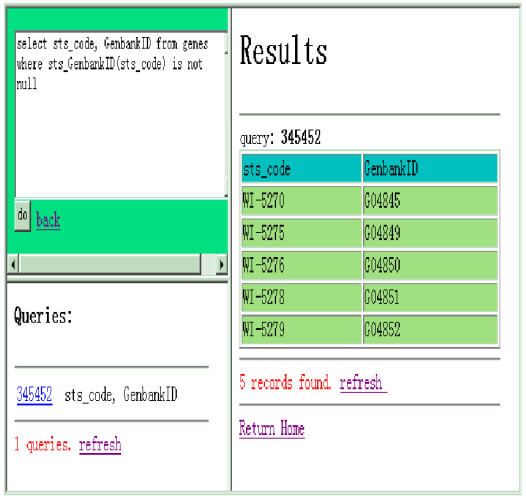
- SQL Query mit IF-Aufrufen

```
SELECT b.sequence
FROM (SELECT get_seq( blast( a.sequence ))
        FROM local as a) as b
WHERE b.organism = "Drosophila" AND b.source(country) = "Kenia"
      AND b.e-value >= 0.98
```

- Teile der Anfrage
  - Führe für alle Sequenzen der DB-Tabelle *local* (hier *a*) die BLAST-Suche in GenBank durch
  - Speichere die Ergebnisse in die Tabelle *b* ab und gib alle Sequenzen aus, die zur Kenianischen Fruchtfliege gehören und einen Ähnlichkeits-Score von mehr als 0.98 haben

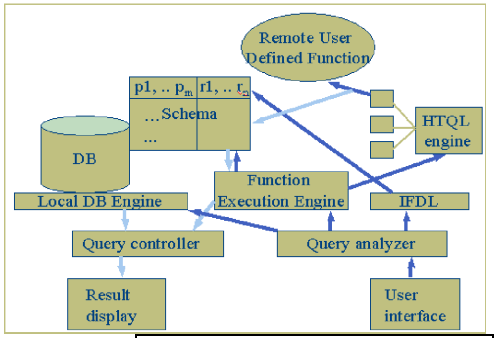
Anwendung: LifeDB

- Biologisches Datenbanksystem (Prototyp der Mississippi State University)
- Unterstützt das Konzept der IFs
- Dreiteiliges Interface
  - Textfeld links oben für Eingabe der IFs und SQL-Anfragen
  - Query-Ergebnisseite rechts
  - Feld links unten zur Auflistung bereits getätigter Anfragen



LifeDB Anfrageverarbeitung: Architektur

- Query Analyzer
  - Parsen von Eingaben im Query-Eingabefeld (stand-alone IF-Aufrufe sowie SQL-Anfragen mit oder ohne eingebundene IF)
  - Weiterleitung von SQL-Anfragen ohne eingebundene IF direkt an die lokale DB Engine
  - Weiterleitung von IF-Aufrufen an IFDL Modul



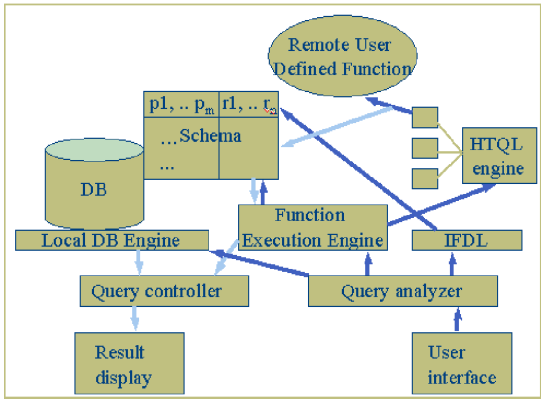
- IFDL Modul
  - Weiterleitung von eingebetteten IF an Function Execution Engine
  - Verfügt über Metadaten-Tabelle in lokalen Datenbank mit Strukturinformationen über die definierten IF (diese wird von Function Execution Engine benötigt, um IF über HTQL-Engine einzubinden und auszuführen)

IF-Metadaten-Tabelle

```
create table EDL_UDFS (
    function_name varchar(30),
    store_table_name varchar(30),
    address varchar(100),
    para_num number(2),
    para_name_list varchar(200),
    result_name_list varchar(200),
    result_htql varchar(200)
```


LifeDB Anfrageverarbeit.: Architektur (2)

- HTQL Engine
  - Nutzt http zur Übermittlung der Eingabedaten an IF
  - IF wird extern ausgeführt und legt Ergebnis in HTML-Seite ab
  - HTQL Engine extrahiert Ergebnis anhand des HTQL-Wertes der IF-Definition
  - Ergebnis wird in einer weiteren Tabelle in lokaler Datenbank abgelegt
- Query Controller führt SQL-Anfrage aus und verwendet dabei Ergebnistabelle aus der Ausführung der IF



- Schließlich wird Ergebnis dem Nutzer präsentiert

RUDF/IF: Zusammenfassung

- API / CLI Ansatz zur Einbindung externer Funktionalität bei der Auswertung von Bio-Daten
- Einbindung von RUDF / IFs innerhalb von SQL-Queries
- Rückgabe der Ergebnisse in html-Format mit Extraktionsinformation (wo ist gewünschte Information im Rückgabe-Dokument?)
- Vorteile: Umgehung lokaler Installation von externer Funktionalität
- Nachteile: Proprietäre Formate und Ausführungsarchitektur
- Weiter Informationen unter:
  - http://www.cse.msstate.edu/~cly/EDI/index.html
  - Chen, L.; Jamil, H. M.: Supporting Remote User Defined Functions in Heterogeneous Biological Databases. Proceedings IEEE International Conference on BIBE 2001: 144-152, 2001.
  - Chen, L.; Jamil, H. M.: On Using Remote User Defined Functions as Wrappers for Biological Database Interoperability; International Journal of Cooperative Information Systems (IJCIS),Special Issue on Data Management and Modeling Support in Bioinformatics, 12(2):161-195, 2003.