Data Lineage in der Bioinformatik

Datenbank-Seminar
Sommer Semester 2005

Hintergründe, Datenbankkonzepte und Einsatz in GUS

Autor:	Gian Marco Laube
Matrikel:	02-710-010
Datum:	1. Juli 2005

Einleitung

Als Grundlage für diese Arbeit über Data Lineage in der Biologie / Biotechnologie habe ich die Arbeit „K2/Kleisli and GUS: Experiments in Integrated Access to Genomic Data Sources“ [7] gewählt. Diese konzentriert sich auf den Vergleich zweier verschiedener Ansätze für die Integration von biotechnologischen Informationen. Wie sich herausstellt, ist dabei Data Provenance va. für den zweiten Ansatz relevant, weshalb ich hier Kleisli auch nur sehr kurz beschreibe.

Ganz allgemein sind die Begriffe Data Lineage/Provenance in der Biotechnologie allgegenwärtig, aber nur selten wird genauer darauf eingegangen, wieso diese Datenbanktechnologie in der Biotechnologie entscheidend ist, oder wie diese konkret umgesetzt wird. Diese beiden Punkte möchte ich mit dieser Arbeit beantworten, womit ich das Thema weiter auffasse, als das beschriebene Paper, das wie die meisten Arbeiten in diesem Bereich, das Thema Data Provenance zwar aufgreift, aber dabei nur sehr oberflächlich behandelt.

Da das Gebiet der Bioinformatik ein sehr komplexes ist, möchte ich mich Schritt für Schritt an dieses Themengebiet heranarbeiten. Deshalb habe ich auch eine kleine Einleitung zu Biotechnologie, Bioinformatik und Datenbanken in der Bioinformatik geschrieben. Wobei ich trotz der Komplexität des Themas versucht habe, diese kompakt und kurz zu halten, um dann im zweiten Teil auf das eigentliche Kerngebiet „Data Lineage in der Biotechnologie“ einzugehen. Als Abschluss zeige ich anhand eines kleinen Beispiels des Datenbankschemas GUS, wie dieses in der Praxis Data Lineage umsetzt.

Inhaltsverzeichnis

1. Bioinformatik

1.1. Einleitung: Das Human Genome Projekt

Das Human Genome Projekt [1], 1985 vom Gesundheits- und Forschungsprogramm des US Department of Energy ins Leben gerufen, wurde in der Presse in den 90er Jahren als die Enthüllung eines der letzten Geheimnisse der Menschheit angepriesen, als die Entschlüsselung des Bauplans des Lebens. In Wahrheit war das Human Genome Projekt hingegen nur ein wichtiger erster Schritt in die heutige Biotechnologie und Bioinformatik Die Entschlüsselung der Gene ist dabei nur die grobe Aufbereitung der Grunddaten unseres Erbgutes. Das sind Daten, die wir brauchen um die Forschung in der Medizin und Biotechnologie einen wirklichen Schritt voranzubringen.

2.1.1. DNA, RNA, mRNA, Proteine und Stoffwechselvorgänge

Da diese Arbeit im Kontext der Datenbanktechnologie stattfindet und nicht im Rahmen einer Biologie-Vorlesung, will ich hier als kleine Auffrischung ganz kurz die Zusammenhänge zwischen DNA, RNA, mRNA und Proteinen aufzeigen. Mit dieser kleinen „Nachhilfestunde“ sollen die Erläuterungen zu den verschiedenen Datenbanken in der Biotechnologie und deren Einsatz in der Praxis auch denjenigen Lesern, deren letzte Biologie-Stunde ein paar Jahre zurückliegen, verständlich gemacht werden.

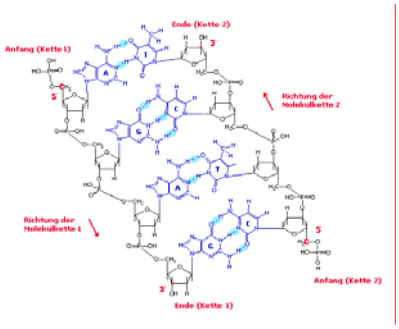

Illustration 1: Chemischer Aufbau der DNA

Deshalb können wir hier aber getrost jegliche chemischen Eigenschaften der DNA und den anderen noch zu besprechenden Baustoffen des Lebens vernachlässigen und die DNA einfach als eine zweisträngige Kette aus Informationen ausgedrückt durch ein Alphabet mit vier Buchstaben (A,T,G und C, welche für Adenin, Thymin, Guanin oder Cytosin stehen) ansehen. Wie wir später sehen werden, sind jeweils drei aufeinander folgende Base (ein Basentriplett) die entscheidende Information für die Herstellung verschiedener Aminosäuren, die wiederum die Proteine bestimmen. In einem DNA Strang können bis zu mehreren Millionen von diesen Buchstaben aneinander gereiht sein.

Für das weitere Verständnis ist die Benutzung der Informationen der DNA entscheidend. Das heisst, wie die Lebewesen auf die Daten in den Genen zugreifen. Um die DNA zu „lesen“ wird sie zuerst in RNA und dann in m-RNA (messenger RNA) übersetzt. Die RNA ist chemisch ein wenig anders aufgebaut, benutzt aber fast das gleiche Alphabet wie die DNA (Anstatt Thymin benutzt die RNA/mRNA für ihre Zwecke das ein wenig unstabilere Uracil). Die m-RNA bildet die Grundlage zur Bildung von den Proteinen. Die m-RNA muss aber für diesen nächsten Übersetzungsschritt zuerst aus dem Zellkern ins Cytoplasma gelangen (deshalb auch „messenger“-RNA), wo sie dann zur weiteren Verarbeitung zu Proteinen auf Ribosomen (selber ein Protein) triff.

Proteine sind aus einem Alphabet von 20 Aminosäuren aufgebaut, wobei jeweils drei Basen der ursprünglichen DNA/RNA eine Aminosäure bestimmt. Proteine sind die eigentlichen „Arbeiter“ in all unseren Zellen, sei es als Strukturproteine, Enzyme, Hormone oder in vielen anderen Rollen. Es gibt sehr viele verschiedene Proteine, da man einige hundert bis sogar tausende Aminosäuren miteinander verketten kann, ergeben sich bereits bei einem kleinen

Illustration 2: Zelle, Chromosom und DNA

Protein (mit „nur“ 100 Aminosäuren) rechnerisch 20^{100} Kombinationen. Zusätzlich spielt aber auch noch die räumliche Anordnung der Proteine eine wichtige Rolle.

Die Proteine sind wiederum massgebend für den ganzen Stoffwechselprozess in unserem Körper. Auf diesen möchte ich hier nicht weiter eingehen, da es den Rahmen dieser Arbeit definitiv sprengen würde. Einen kleinen Einblick in die Komplexität der Stoffwechselprozesse mag die Übersichtskarte des Pharmaunternehmens Roche über die biochemischen Zusammenhänge in Lebewesen geben.

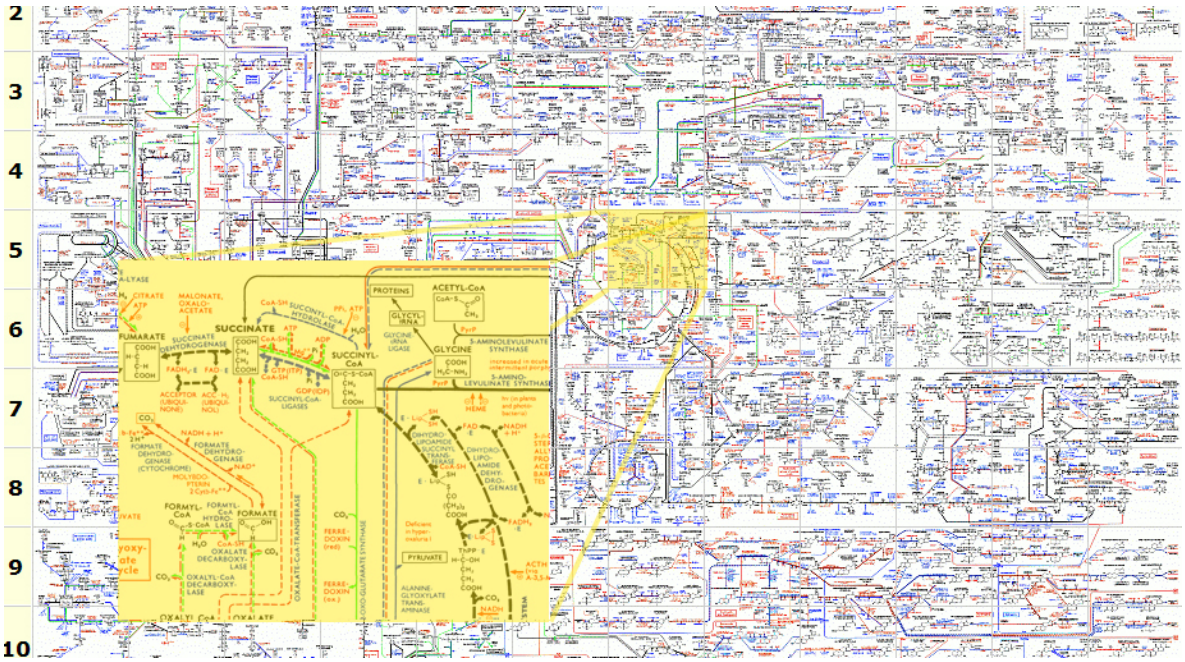

Illustration 2: Übersichtskarte von Roche über Stoffwechselpfade. In gelb: Detailausschnitt G5.

2.1.2. Bestimmung des menschlichen Genoms

Das Ziel des Human Genome Projektes war es die menschliche DNA zu entziffern, das heisst alle Basenpaare zu bestimmen und falls möglich Genen zuzuordnen. Dies ist deswegen wichtig weil die menschliche DNA zu 97%[1] aus „Junk-DNA“ besteht. (Diesem „Junk“ wird heute aber immer mehr Beachtung in der Forschung geschenkt.)

Um die riesige Menge an Gen-Daten in unseren Zellen zu entschlüsseln, mussten zuerst neue Sequenzierungs-Methoden gefunden werden, da von Hand diese Aufgabe nicht in vernünftiger Zeit zu bewältigen gewesen wäre. Beim so genannten „Shotgun sequencing“ wird die DNA in kleine Einheiten zerstückelt. Nur so kann die Reihenfolge der einzelnen Basen effizient bestimmt werden. Dieser Schritt ist technisch noch relativ einfach. Schwieriger ist es die einzelnen Puzzle-Stücke wieder zusammenzusetzen und zu bestimmen welche davon eine Funktion in einem Gen haben und welche nicht.

Dieser Schritt wird durch das „Sequence allignment“ angegangen. Hochleistungsrechner werden dabei benötigt, um überlappende Enden in den DNA-Stücken festzustellen und so Schritt für Schritt die DNA-Daten wieder

[1] Gemäss Wikipedia: http://en.wikipedia.org/wiki/Junk_DNA, Abfragedatum [20.05.05]

zusammenzustellen. Dieser Prozess birgt natürlich auch eine gewisse Fehleranfälligkeit in sich. Die sich daraus ergebende DNA-Daten sind heute in verschiedenen Datenbanken (die wichtigsten davon werden wir später nennen) für jedermann online frei abrufbar.

2.1.3. Proteom, Genome und Gen Therapien

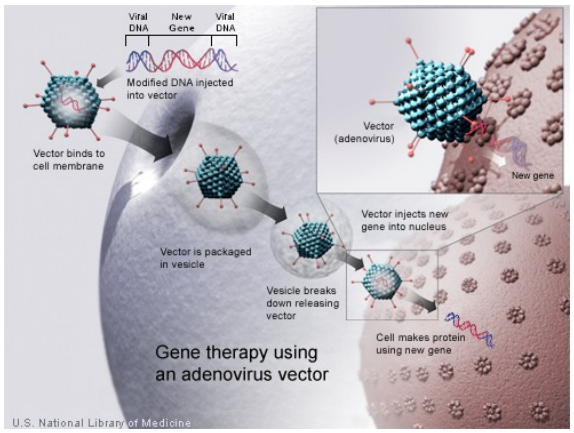

Illustration 3: Gen-Therapie: Transport von Genen in den Zellkern mit Vektoren

Die Überwindung der Brücke zwischen Genom und Proteom ist heute ein zentrales Forschungsgebiet in der Biotechnologie. Viele Krankheiten kann die Medizin einem gewissen Stoffwechsel zuordnen und deshalb auch einem gewissen Protein, das seine Funktion nicht oder nicht mehr richtig erfüllt. Dies ist oft durch fehlerhafte DNA-Grunddaten verursacht worden. Heute ist es durch die neuen Datenquellen möglich, immer mehr Stoffwechselvorgängen DNA-Daten zuzuordnen. Dies wird teilweise bereits für die Gen-Therapie benutzt. Ziel von Gen-Therapie ist es Patienten zu behandeln, indem ihnen DNA-Stücke injiziert werden. Die Injektion eines Gen-Stückes in einen Zellkern (meistens durch Viren als Vektoren) ist bisher nur sehr umständlich möglich. Einige Beispiele von bereits angewandten Gen-Therapien findet man bspw. in [2].

1.2. Bioinformatik

Gen-Therapien mögen eines der ultimativen Ziele der Gentechnologie sein, aber momentan beschäftigen sich die Forscher va. noch mit den Grundlagen, wie mit den Stoffwechselvorgängen und mit der Zuordnung von diesen zu Gen-Daten. Diese Integration von biologischen Daten ist wohl neben der eigenen technischen Weiterentwicklung, die wichtigste Aufgabe der Bioinformatik heute.

Während viele kleine Projekte und Forschungsteams sich mit der genauen Analyse einzelner Bruchstücke von Gen-Informationen auseinandersetzen, liefern andere High-Tech-Labore in einer Art Massenproduktion berechnete riesige Volumen an oft auch unpräzisen Daten. Manuelles Recherchieren in dieser Informationsmenge ist kaum mehr möglich [3]. Dank der heutigen Vernetzung von biotechnologischen Datenbanken, kann man aber bspw. innert wenigen Minuten einem gefundenen DNA-Bruchstück, eine Raumstruktur eines Proteins oder ganze Stoffwechselinformationen zuordnen. Für den Vergleich von Baseninformationen (ein Mensch hat z.B. über 2 Milliarden Nukleotiden in seinen Genen) wurden spezielle Vergleichsalgorithmen erfunden, wie z.B. BLAST (**B**asic **L**ocal **A**lignment **S**earch **T**ool) [4], welcher eine Hash-Indexierung benutzt. BLAST wird heute in verschiedensten Implementationen für alle mögliche Vergleiche in der Biotechnologie gebraucht. Viele dieser Tools sind für jedermann per Web-Interface frei zugänglich, was Sequenzvergleich von DNA-Daten kinderleicht erscheinen lässt.

1.3. Datenbanken in der Bioinformatik

Je nach Quelle spricht man heute von ca. 500-1'000 relevanten Datenbanken in der Bioinformatik, welche nach Datenherkunft, Spezialisierung und Funktionen charakterisiert und kategorisiert werden können [5]. Eine wichtige Unterscheidung ist diejenige, ob eine Datenbank als primäre Datenquelle dient oder als sekundärer

Informationsmediator. Die Unterscheidung ist in diesem Kontext hier sehr schwierig, da auch die grössten Datenbanken wie GenBank ihre Informationen zu einem grossen Teil aus anderen Datenbanken sammeln und nicht direkt von Experimentaldaten erheben.

1.3.1. GenBank/EMBL/DDBJ

GenBank (U.S. National Center of Biotechnology Information), DDBJ (DNA Data Bank of Japan vom Japan National Institute of Genetics) und das EMBL (European Molecular Biology Laboratory) bilden zusammen die grösste DNA-Datenbank der Welt (International Nucleotide Sequence Database Collaboration).

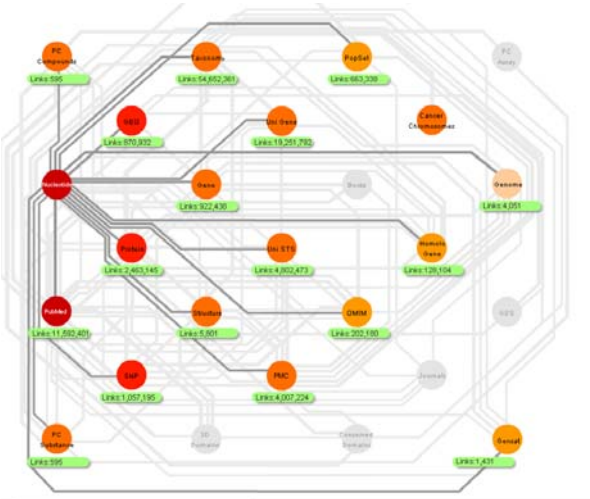

Illustration 4: GenBank beruht auf einem komplexen Link-Netz. So gibt es alleine zwischen den DNA Einträgen und der PubMed Bibliothek über 11 Mio. Links.

GenBank legt die Gen-Sequenzen in so genannten „flat files" ab. Flat Files sind ASCII-Textdateien die Datensätze enthalten, ohne eine implizite Beziehungsstruktur, wie sie Datenbankkonzepte liefern. Gemäss [6] benutzen heute ca. 40% aller Bioinformatik-DB Flat Files, während nur ca. 35% von Relationalen Modellen gebrauch machen. Ca. 5% benutzen Objektorientierte Datenbankansätze. GenBank benutzt dafür das Datenaustausch-Format ASN.1. Deshalb müssen diese Files auch mit einem bestimmten Anfragesystem zugegriffen werden. Im Fall von GenBank ist dies das Anfragesystem ASN.1-Entrez[2].

1.3.2. UNI-PROT / SWISS–PROT und TrEMBL

Die bekannteste Datenbank der Welt für die Sequenzierung von Proteinen, und eine der meist benutzten Biotech-Datenbank überhaupt, kommt aus der Schweiz. In SWISS-PROT, einer Datenbank des Swiss Institute of Bioinformatics, findet man heute bereits 181'571 experimentell nachgewiesene Proteine[3] . Noch viel mehr Proteine findet man im Geschwister-Projekt TrEMBL (1'714'475), diese sind aber nicht wirklich nachgewiesen/erforscht, sondern nur anhand von DNA-Sequenzen aus dem EMBL (European Molecular Biology Laboratory) theoretisch berechnet. UNI-PROT ist die Vereinigung von SWISS-PROT, TrEMBL und PIR, eine weitere Protein-DB ähnlich zu SWISS-PROT.

Auch SWISS-PROT legt ihre Protein-Sequenzen nicht in einer relationalen oder Objektorientierten DB ab, sondern benutzt ebenfalls Flat Files. Um sich darunter ein wenig mehr vorstellen zu können, hier ein kleines Beispiel (aus [7]), wie ein solcher SWISS-PROT Eintrag aussieht:

```
ID EF1A_CAEEL STANDARD; PRT; 463 AA.
AC P53013;
DT 01-OCT-1996 (Rel. 34, Created)
```

[2] ASN.1 Entrez http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Nucleotide, Anfragedatum [21.5.2005]

[3] Swiss Institute of Bioinformatics, http://au.expasy.org/sprot/, Anfragedatum [20.5.2005]

```
DT 01-OCT-1996 (Rel. 34, Last sequence update)
DT 15-DEC-1998 (Rel. 37, Last annotation update)
…
RN [1]
RP SEQUENCE FROM N.A. (EFT-3).
RC STRAIN=BRISTOL N2;
RA Favello A.;
RL Submitted (NOV-1995) to the EMBL/GenBank/DDBJ databases.
…
DR EMBL; U51994; AAA96068.1; -.
DR HSSP; P07157; 1AIP.
…
KW Elongation factor; Protein biosynthesis; GTP-binding;
FT NP_BIND 153 156 GTP (BY SIMILARITY).
SQ SEQUENCE 463 AA; 50668 MW; 12544AF1F17E15B7 CRC64;
        MGKEKVHINI VVIGHVDSGK STTTGHLIYK CGGIDKRTIE KFEKEAQEMG KGSFKYAWVL
        DKLKAERERG ITIDIALWKF ETAKYYI…
```

Wichtige Elemente dieser Datenstruktur :

- SQ: Neben der ID wohl das wichtigste Feld. Hier wird die ganze Sequenz von Aminosäuren aufgelistet. M steht z.B. für die Aminosäure Methionine.
- DT: Jedes File kann bis zu drei Zeitstempel haben. Das Entstehungsdatum und evtl. Daten von Updates der Sequenz oder von Annotationen.
- DR: Referenzen zu anderen DB (Kurzbezeichnung der DB und ID zum entsprechenden Eintrag).
- R_: Referenzen zu Forschungsliteratur

Diese Dateistruktur wurde gewählt, weil sie so für Menschen einfach lesbar ist. Wobei dies heute natürlich nicht mehr optimal ist, da auf Grund der riesigen Datenmenge jede Abfrage sowieso über irgendein Tool läuft, das die Syntax für den Benutzer aufarbeiten kann.

1.3.3. Zwei weitere wichtige primäre Biotechnologie-Datenbanken

KEGG:
Die „**K**yoto **E**ncyclopedia of **G**enes and **G**enomes" beschäftigt sich mit den Stoffwechseln in den Zellen und wie diese hauptsächlich durch Enzyme beeinträchtigt werden.

dbEST:
Beinhaltet so genannte „Expressed Sequence Tags", das sind kleine DNA-Schnippsel die gebraucht werden um schnell und einfach dafür aber auch ungenau und unvollständig die Gene eines Lebewesens zu finden. Diese Methode funktioniert so, dass aus mRNA durch eine künstliche Rückwärts-Übersetzung DNA-Schnippsel gewonnen werden.

2. Datenbankkonzepte in der Bioinformatik

2.1. Drei verschiedene Konzepte und ihre Nutzung in der Praxis

Wie bereits erwähnt, ist der zentrale Aspekt der Bioinformatik die Integration von verschiedenen Informationen aus unterschiedlichen Quellen. Nach [8] kann man in der Bioinformatik drei Konzeptarten von Integrationsstrategien in Bioinformatik-Datenbanken unterscheiden:

a. Informations-Verlinkung (Information Linkage)
b. Informations-Integration
c. Data Warehousing

Andere Quellen (z.B. [3]) benutzen andere Einteilungen, die sich aber grundsätzlich alle sehr ähneln. Die ersten beiden Methoden arbeiten mit einem Sichten-Konzept, das heisst das sie die Daten, die sie aus verschiedensten Quellen beziehen nicht in einer eigenen Datenbank materialisieren.

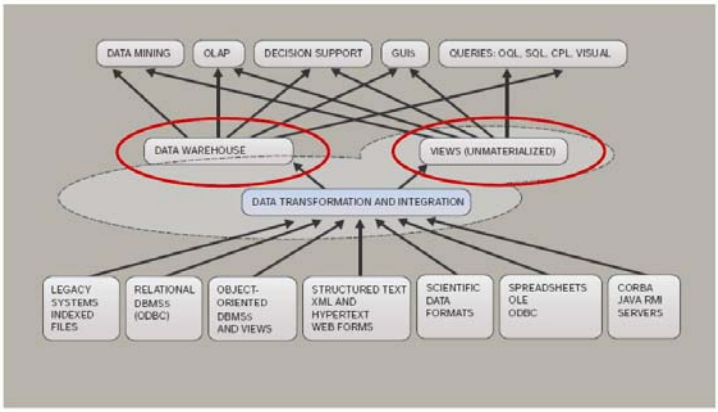

Illustration 5: Data Warehouse und Sichten-Modelle als Informations-Mediatoren

2.2. Information Linkage

Dem Konzept der Information Linkage liegen statische Links zwischen Einträgen in verschiedenen Datenbanken zugrunde. So sind alle Protein-Einträge in Expasy (das Web-Interface zu SWISS-PROT) mit fixen Links zu diversen anderen Datenbanken (wie zum Beispiel zu den Forschungsberichten von PubMed) verlinkt. Daraus ergeben sich Vor- und Nachteile:

+ Einfach für unerfahrene Benutzer zu benutzen (einfache Web-Interfaces)
+ Auch für kleine Labore/Forschungsstellen umsetzbar, da sie ihre Forschungsberichte/Einträge ohne grosses technisches Know-how und ohne riesige Informationsbasis zu weiterführenden Informationen verlinken können.
- Skalierbarkeit stark eingeschränkt: Jede neue Datenbank muss sich mit allen anderen Datenbanken verlinken (Problem eines quadratischen Aufwands bei wachsender Datenbankanzahl)
- Joins zwischen Einträgen auf verschiedenen Datenbanken unmöglich

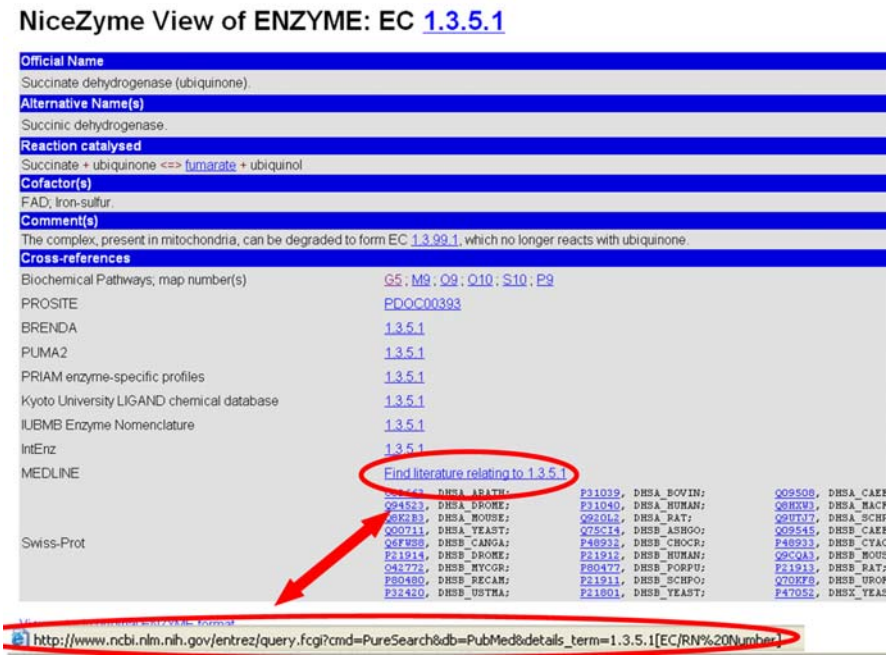

Illustration 6: Expasy, das Web-Interface für SWISS-PROT, baut auf Information Linkage. Der eingekreiste Link ist direkt und statisch mit einem PubMed Eintrag beim NCBI verlinkt.

2.3. Information Integration

Auch im Ansatz der Information Integration bleiben die Daten bei den ursprünglichen Datenquellen (im Gegensatz zu Data Warehouses). Hinter dem Integrations-Ansatz steht ein semantisches Verständnis von Objekten/Einträgen aus anderen Datenbanken. Anfragen an diese „Informations-Mediatoren" werden je nach den benötigten Informationen aufgeteilt und an einzelnen Quelldatenbanken weitergeleitet. Die dann daraus erhaltenen Resultat müssen wieder rückübersetzt und mit anderen Sub-Abfragen integriert/vereint werden. Um die Schnittstellen zu den Quelldatenbanken (alle mit meistens völlig verschiedenen Datenbankkonzepte und anderen Anfragesprachen) kümmern sich so genannte „Wrappers". Informations-Integration Systeme benutzen im Normalfall eine höhere Anfragesprache wie SQL, OQL oder im Fall von Kleisli CPL. Diese Systeme ermöglichen somit Anfragen, die durch das statische Information Linkage nicht möglich sind. Unter anderem va. auch Joins über verschiedene Datenbanken hinweg.

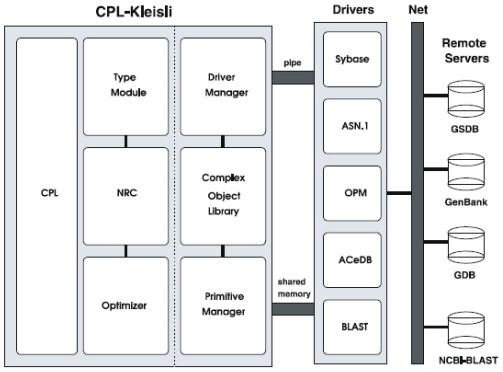

Illustration 7: Die Architektur von CPL/Kleisli (ein Information Integration System). Wichtig für die Informations-Integration Prinzip sind die "Wrappers" oder auch Data Drivers welche die Verbindung zwischen Kleisli und den Datenquellen bewahrt.

Betrachtet man das folgende Beispiel, wird es offensichtlich, dass die manuelle Durchsuchung mittels Links seine Grenzen hat: „Finde alle Gene die Teilstücke besitzen, deren Länge grösser als 40 Basenpaare sind, die zu mindestens 95% mit beliebigen gleich langen Ausschnitten aus der DNA-Übersetzung des Proteins X übereinstimmen!“. Es ist wohl offensichtlich, dass diese Anfrage durch manuelles Durchsuchen von Information Linkage Datenbanken fast unmöglich ist. Diese Anfragen sind aber auch trotz der höheren Anfragesprachen wie SQL/OQL/CPL für einen typischen Benutzer aus der Medizin oft zu komplex, um sie selber zu definieren.

```
sybase-add (#name:"gdb", ...);
readfile locus from "locus_cyto_location" using gdb;
readfile eref from "object_genbank_eref" using gdb;
{(#accn: g.#genbank_ref, #nonhuman-homologs: H)
| \c <- locus, c.#chrom_num = "22",
  \g <- eref, g.#object_id = c.#locus_id,
  \H == { u
   | \u <- na-get-homolog-summary(g.#genbank_ref),
     not(u.#title string-islike "%Human%"),
     not(u.#title string-islike "%H.sapien%")},
  not (H = { })}
```

Illustration 8: Anfragen für Informations-Integrations-Systeme wie hier am Beispiel von Kleisli mit CPL, können einen Wissenschaftler ohne Programmierkenntnisse schnell überfordern

Genau wie bei dem heute allherschendem SQL wurde hier der Benutzer überschätzt. Die Lösung für dieses Problem ist es vordefinierte Sichten und parametrisierbare Anfragen zu generieren und diese per Web-Schnittstellen bereitzuhalten.

Das Ziel der Informations-Integrations Systeme war es, eine komplette virtuelle Vernetzung der Biodatenbanken zu erreichen: Eine Art „Biomatrix“. Ein Beispiel für ein solches System ist Kleisli, welches seit ca. 1994 am Penn Center für Bioinformatik entwickelt wird. [7],[9],[11]. Diese Idee der vollkommen vernetzten und integrierten Welt aller Biodatenbanken ist heute aber eher eine Utopie. So wird Kleisli heute zwar von verschiedenen Systemen genutzt, aber so richtig durchgesetzt hat es sich nicht.

2.3.1. Rolle von Data Lineage bei Information Integration

Wie bereits angesprochen, brauchen Informations-Integrations Systeme wie Kleisli keine separate physische Kopie der integrierten Datenbasis, weil diese jeweils bei einer Anfrage aus den zugrunde liegenden Datenquellen ermittelt wird. Ob die benötigte Datenquelle nun auf dem gleichen Server liegen oder auf einer externe Datenbank angefragt werden, spielt kaum eine Rolle, da diese Arbeit die „Wrapper“ übernehmen. Damit ergeben sich drei Vorteile:

+ Kleinerer Initialaufwand um das System zu erstellen
+ Tiefe Wartungskosten (die Daten werden von den jeweiligen Datenbanken gewartet). Das eigene System muss sich nur um die korrekte Anbindung und semantisch korrekte Interpretation der erhaltenen Daten kümmern.
+ Die Resultate sind immer „up to date“

Gerade der letzte Punkt ist für den Kontext dieser Arbeit sehr wichtig. Die Data Lineage/Provenance der Daten ist in diesem Fall weniger ein Problem des Systems auf der obersten Ebene, da es nur direkt und nicht zeitlich versetzt auf bestimmte Quellen zurückgreift. Somit muss man sich nicht darum kümmern, ob und wann sich die Basisdaten verändert haben, wie man diese Änderungen am effizientesten übernimmt und auch die Herkunft der einzelnen Datenelemente ist sehr einfach zu bestimmen.

Wenn man dies betrachtet, muss man sich natürlich fragen, wieso man sich überhaupt die Mühe machen sollte, ein eigenes Data Warehouse für all die komplexen Daten aufzubauen, wenn es doch solch praktische Mediatoren dafür bereits gibt. Dafür betrachten wir die sich ergebenden Nachteile durch den Verzicht auf eine eigene physische Kopie :

- Effizienz und Zuverlässigkeit: Sichtenkonzepte sind weitaus weniger effizient als Data Warehouses (eine empirische Untersuchung dazu ist in [7] zu finden). Zudem können ganze Datenbanken durch Netzwerkausfälle betroffen oder überlastet sein.
- Integrierende über verschiedene Bereiche gehende Anfragen sind sehr komplex
- **Keine Bereinigung der Daten möglich (Data Cleansing)!**
- **Kein zusätzliches eigenes Kommentieren der Daten möglich!**

Die letzten beiden Punkte sind in der Biotechnologie wichtiger als man denkt, und erklären wieso Sichten-Konzepte oft nicht ausreichen. In diesem Forschungsgebiet ist es entscheidend auf Grund von Basisdaten einen Added Value zu generieren, indem man Inputdaten kommentiert, verbessert, bereinigt, etc [12]. Dies weil die meisten dieser Daten bisher nur berechnet, also „in silico" erstellt wurden, womit eine Nachbearbeitung und manuelle Erforschung durch andere Labore unabdingbar ist. Die folgende Illustration aus [13] zeigt ein typisches Beispiel wie eine Datenmenge die ursprünglich aus der Datenbank PRISM[4] und GenBank exportiert, in vielen Einzelschritten kommentiert, verändert und schliesslich auch noch mit Information aus KEGG erweitert wurde. Dies ist für ein Forschungsteam nur mit physischen Kopien der ursprünglichen Daten in einem Data Warehouse möglich.

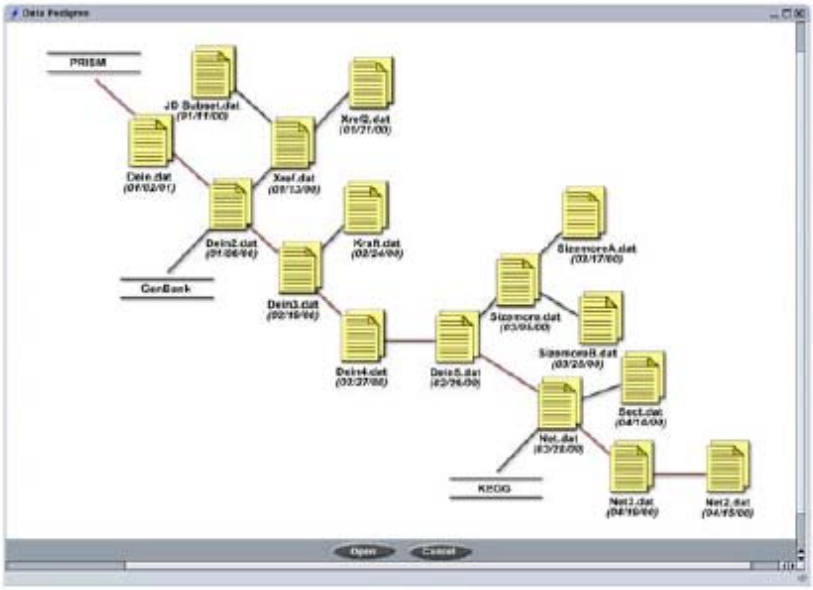

Illustration 9: Data Provenance (Pedigree) - Ansicht im Kollobarations Tool: Biological Sciences Collaboratory. Daten aus PRISM, GenBank und KEGG wurden benutzt und zusätzlich kommentiert oder korrigiert.

2.4. Data Warehousing

Data Warehouses speichern eine physische Kopie der zu benutzenden Daten ab. Somit werden die integrierenden Anfragen im Gegensatz zur Information Integration im Voraus ausgeführt und nicht erst angefragt, wenn diese wirklich benötigt werden. Diese zumeist periodische Datenintegration wird in der Biotechnologie oft genutzt, um ein automatisches Data Cleansing durch vordefinierte Algorithmen oder durch einen Abgleich mit den bereits bestehenden Daten durchzuführen. Dies bringt natürlich aber auch ein paar Probleme mit sich:

[4] http://www.ecsprism.com/prism.htm, Abfragedatum [23.5.2005]

- Die Datenquellen müssen laufend auf Veränderungen analysiert werden (Push- oder Pull-Technologien).
- Der Datenübernahmeprozess sollte möglichst automatisiert werden, da dies manuell zu aufwendig ist.
- Die „Herkunft“ (Provenance/Lineage) der Daten muss verfolgt werden.

Die folgenden Ausführungen konzentrieren sich auf den letzten Punkt, aber die Themen sind sehr eng verwandt miteinander. So kann man nur Änderungen feststellen und Abgleiche ausführen, wenn man auch weiss woher ein bestimmtes vorhandenes Daten-Objekt gekommen ist, und wie es verändert wurde.

Nur wenige Biotech-Datenbanken bieten schon einen Dienst an, um alle oder nur bestimmte spezifisierte Änderungen/Neuerungen direkt an Benutzer zu versenden (Push-Technologien für die View Maintenance). SWISS-PROT bietet mit SWSS-SHOP[5] [14] einen solchen Dienst per Mail an, ist damit aber noch eine Ausnahme.

Heute wünscht man sich zudem detailliertere Auskunft, wenn Updates in den Datenquellen geschehen. So muss man sich veranschaulichen dass z.B. das Schema von GUS (ein Data Warehouse-Schema, das wir später noch genauer betrachten werden) einen einzelnen SWISS-PROT Eintrag mit all den benötigten Meta-Infos auf 15 verschiedene Relationen verteilt. Es wäre also sehr umständlich, alle 15 Relationen auf Änderungen abzusuchen, anstatt nur in einer einzigen Relation einen Eintrag hinzufügen zu können, weil z.B. ein zusätzlicher Forschungsbericht zu einem bestimmten Gen erschienen ist.

Man muss hier aber zwischen dem allgemeinen und bereits gut erforschten View Maintenance Problem (welches wir oben beschrieben haben) und dem Auffrischen oder Korrigieren von Daten und Annotationen in Data Warehouses unterscheiden. Für diese Zusatzdienste braucht es Methoden aus dem Bereich von Data Lineage und Provenance. Schliesslich wollen wir nicht alle Annotationen zu einem DNA-Eintrag verwerfen, nur weil zu diesem einzelne neue medizinische Fakten erforscht wurden. Viel mehr liegt es in unserem Interesse, diese Annotationen, bei einer solchen Änderung der zugrunde liegenden Daten neu aufzuwerfen, und zur Überarbeitung zu markieren. Oder wir würden gerne automatisch ermitteltes Data Cleansing anhand der neuen Informationen neu berechnen lassen.

Das Ziel ist es, Provenance Services so in den Workflow einer Biotech-Datenbank einzubauen, dass möglichst jedes biotechnische Datenobjekt bis zu seinem Ursprung in einem Experiment oder Forschungsbericht verfolgt werden kann. Folgende Informationen wären z.B. für einen Datensatz in einem integrierten Biotech Data Warehouse relevant:

- Von welcher Quelle stammt ein gewisser Datensatz (Quelldatenbank, Versionsinfo, Identitäten von benutzen Einträge aus der Quelldatenbank, etc.)?
- Aus welchem Anfrage-Algorithmus wurde dieser Datensatz gebildet (Anfrage, Parameter, Zeitpunkt)?
- Wie wurde der Datensatz verändert, bereinigt, kommentiert (Algorithmus, Parameter, Datum)?
- Falls der Datensatz aus der aktiven Datenbank gelöscht wurde: Durch wen oder welchen Algorithmus?
- Wenn der Datensatz experimentell eruiert wurde, durch was für ein Experiment, Parameter, Ursprung?

[5] http://au.expasy.org/swiss-shop/, Abfragedatum [23.5.2005]

3. Data Lineage / Provenance

Die „Herkunft“ (aufgrund der weiten Verbreitung hier entweder mit Provenance oder Lineage bezeichnet) eines Datensatzes besteht aus seiner ganzen Verarbeitungsgeschichte. Dies beinhaltet Ursprung (anderer Datensatz, Experiment, Algorithmus, etc.) und alle weiteren Bearbeitungsschritte (Autoren von manuellen Mutationen, oder falls „in silico“ die verantwortlichen Algorithmen und Parameter) [15].

Data Lineage ist ein breit diskutiertes Thema im Bereich der Bioinformatik. Die Bioinformatik ist wahrscheinlich das wichtigste Anwendungsgebiet von Data Lineage Methoden überhaupt bis heute. Nur mit Hilfe dieser Informationen kann ein Forscher entscheiden, wie vertrauenswürdig ein Datensatz ist, und somit selber die Qualität seiner Arbeit steuern. Auch haben wir bereits den Punkt angesprochen, dass Data Provenance benötigt wird, um Annotationen wieder neu aufzuwerfen, wenn die dafür zugrunde liegenden Daten verändert wurden, oder um daraus erfolgte Berechnungen/Joins nochmals nachzuberechnen.

Grund für diesen breiten Einsatz in der Biotechnologie sind wohl die riesigen Datenvolumen aus verschiedensten Quellen, die Wichtigkeit von effektiver Integration um Zusammenhänge zu erkennen, oder der z.T. sehr unterschiedlichen Qualität von Daten auf Grund von verschiedenen Entwicklungsmethoden (nur berechnet oder experimentell manuell ermittelt). Innerhalb der Naturwissenschaften ist Data Provenance auch noch in der Geologie, Meterologie oder in der Astronomie wichtig. Denn Satelliten „scannen“ respektive untersuchen die Erdoberfläche oder des Universum in einer ähnlich datenlastigen Art, wie das menschliche Genom entschlüsselt wurde (Beispiele: Astrologie [16], Meterologie [15], Geologie [17]).

Leider stellen heute noch die wenigsten Datenquellen wirklich Provenance-Daten über den Ursprung ihrer Daten zur Verfügung, obwohl auch grosse Datenbanken wie KEGG ihre Informationen aus vielen Quellen integrieren. Gewisse Ansätze sind sicher vorhanden, so wurde ein Standard entwickelt um die wichtigsten Informationen von DNA-Chip Experimenten festzuhalten: MIAME (Minimal Information About a Microarray Exeriment) [13]. DNA-Chips werden gebraucht um rechnerischen schneller und dafür unpräziser Gene zu ermitteln und analysieren (so genannte expressed sequence tags). Mithilfe der MIAME Meta-Daten lässt sich u.a. die Relevanz/Qualität eines Experiments erahnen. Es gibt jedoch bisher noch keine Standards um die Herkunft eines Datensatzes über verschiedene Datenbanken hinweg zu ermitteln.

Wir wollen hier kurz auf drei Data Lineage Verfahren eingehen, welche in der Biotechnologie oder zumindest in anderen Naturwissenschaftlichen Feldern - ob es eine Biotech-Datenbank gibt die „Weak Inversion“ benutzt, ist mir nicht bekannt - eingesetzt werden: Meta-Daten, Schwache Inversion und Präzisere Umkehrung von relationalen Anfragen.

3.1. Meta-Daten

Bis jetzt ging man in den Naturwissenschaften meistens vom Meta-Daten-Ansatz aus, um die Provenance eines Datensatzes zu bestimmen. Das heisst, dass für jeden Datensatz Meta-Daten über dessen Herleitung (Derivation/Provenance) gespeichert wird. Dies ist sicherlich der natürlichste und einfachste Ansatz, und heute im Biotech-Bereich bisher auch noch das am weitesten verbreitete Mittel für die Lineage Bestimmung.

Die Meta-Daten können auf diverse Methoden gespeichert und verwaltet werden. Ein Beispiel dafür werden wir im Anschluss mit dem GUS Schema, welches speziell für die Bioinformatik zugeschnitten wurde, besprechen. Eine andere Umsetzung von Data Lineage mit Metadaten findet man bspw. in [18].

Falls die Meta-Daten über verschiedene Datenbanken hinweg genutzt werden sollten, ist es wichtig, dass man sich über deren Semantik einig ist. So ist die Forschung im Bereich von Ontologien oder Semantic Web hier entscheidend, und XML ein bereits jetzt in der Bioinformatik breit eingesetztes Mittel (welches oft aber auch überschätzt wird, da es nur den Syntax nicht aber die Semantik eines Datensatzes definieren kann).

3.2. „weak" Inversion

Da es im Vergleich zu anderen Methoden relativ simpel ist Meta-Daten zu speichern, um die Herleitung eines Datensatz festzuhalten, kann man sich fragen, wozu es andere Ansätze braucht. Meta-Daten sind zumeist kleinere Zusätze zu ganzen Datensätze. So ist eine Sequenz eines GenBank Eintrags bis zu mehreren 10'000 Basen lang. Um die Herleitung der Daten zu bestimmen, braucht es aber in einem Minimalfall nur ein paar wenige Informationen als Meta-Daten abzuspeichern (wobei wir noch sehen werden, dass z.B. GUS relativ viele Meta-Infos anlegt). Was wäre aber, wenn man zu einzelnen Ausschnitte (im Extremfall einzelne Basenpaare) die Provenance feststellen möchte? Es wäre ja durchaus denkbar, dass bestimmte Autoren, durch ein bestimmtes Experiment, einzelne Ausschnitte einer Gen-Sequenz neu berechnet und überarbeitet haben. Das ganze hört sich vielleicht nach Zukunftsmusik an, wird aber bereits in der Meteorologie angewendet [15]. Man spricht von einer fein-maschigeren Provenance-Ermittlung.

So verfolgen Meteorologen in den Vereinigten Staat Tiefdruckgebiete und somit potentielle Wirbelstürme durch Hochauflösende Satellitenbilder. Dabei werden die Pfade der Tiefdruckzentren anhand Vergleiche der zeitlich sich folgenden einzelnen Bilder und den dazu gemessenen Luftdruckwerten festgehalten. Um die Qualität einer Berechnung zu testen, möchten nun Meteorologen einen einzeln Pfadpunkt auf seinen Ursprung in einem oder mehreren Satellitenbilder zurückführen. Dies ist mit einem herkömmlichen Metadatenansatz unmöglich, weil so rein theoretisch im Extremfall für jedes ursprünglich aufgenommene Satellitenbild pro Bildpunkt diese Meta-Daten festgehalten werden müssten. Bei einer hohen Auflösung, wie sie in diesem Bereich üblich ist (z.B. 6'000 x 6'000 Pixel) würde zu mehrere Millionen einzelnen Meta-Daten-Sätze für ein einziges aufgenommenes Satellitenbild führen.

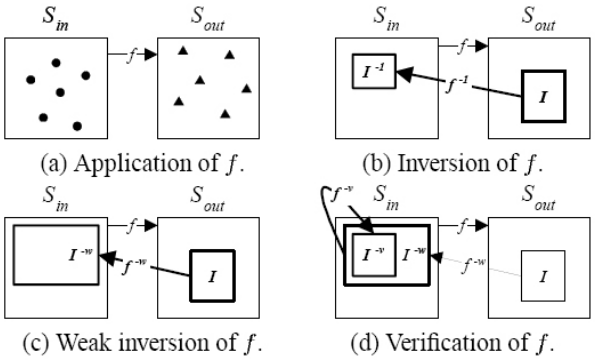

Illustration 3: (a) Daten aus S(in) werden durch eine Anfragefunktion f in Ergebnisdaten S(out) überführt. (b) Zeigt die theoretische richtige Inversion der Funktion f auf einem bestimmten Bildbereich I. (c) Da f oft nicht invertierbar ist, versucht man mit der schwachen Inversion eine Annäherung zu finden. (d) Diese Menge an möglichen Input-Datensätzen kann mit einer Verifikationsfunktion nochmals eingeschränkt werden.

Woodruff und Stonebraker [15] schlagen deshalb eine „weak" (schwache) Inversion der Anfrage, die zu einem bestimmten Ergebnis geführt hat, vor. Die Idee dahinter ist, dass zwar die tatsächliche Umkehrung einer Anfragefunktion anzustreben wäre, dass tatsächlich aber nur eine sehr eingeschränkte Menge von Funktionen vollständig invertierbar ist. Somit beschränken sie sich auf die schwache Inversion, welche nur annähernd die Ursprungsdaten liefert.

Zusätzlich braucht es aber auch noch eine Verifikation der Resultate der schwachen Inversion, weil wie schon angesprochen, die meisten Anfragefunktionen nicht perfekt umkehrbar sind. Im Gegensatz zu der schwachen Inversion, welche anhand der Anfragefunktion eine Umkehrfunktion berechnet und diese auf die Inputdaten anwendet, vergleicht die Verifikation, die damit berechneten Inputwerte direkt mit den ihnen dadurch zugeordneten Werte im Output. Somit wird das Resultat verfeinert, und die gewünschte Provenance-Qualität erzielt.

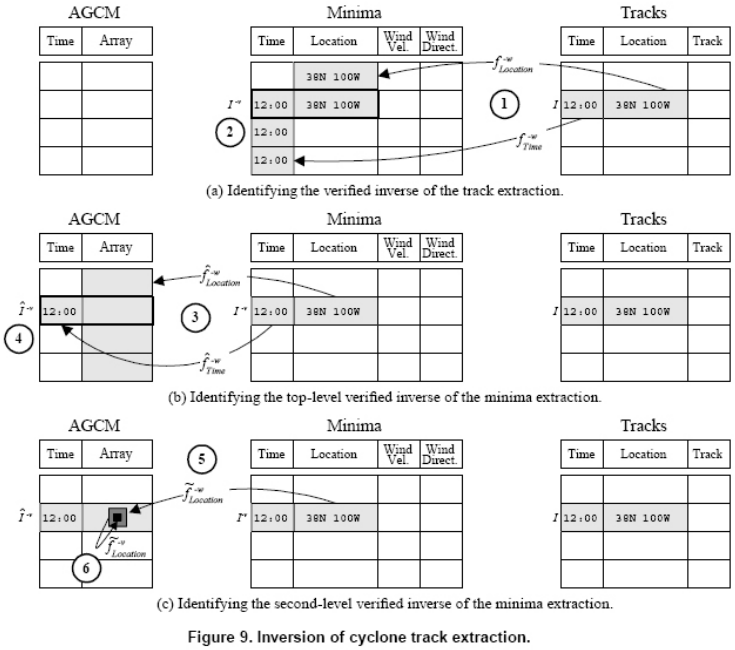

Figure 9. Inversion of cyclone track extraction.

Illustration 4: Beispiel einer Bestimmung der Daten Provenance bei der Wirbelsturmverfolgung mit weak Inversion und Verification.

3.3. Präzise Algorithmen für relationale Ansätze

Cui, Widom und Wiener [19] kritisieren am Ansatz der schwachen Inversion vom Woodruff und Stonebraker, dass dieser nur funktioniere, wenn die Datenbank-Administratoren diese inversen und Verifikations- Funktionen definieren, und dass man dies oft von den jeweiligen Datenbank-Designern nicht erwarten kann. Sie propagieren deshalb ihren eigen Ansatz, welcher sich aber ausschliesslich auf relationale Funktionen bezieht. Dabei unterscheiden sie zwischen einfacheren SELECT-PROJECT-JOIN (SPJ) Anfragen (ohne Aggregation) und ASPJ-Anfragen (mit Aggregation). Je nach Fall vermitteln sie für alle Anfragequeries den passenden inversen Algorithmus, wobei bei Aggregationen Zwischenresultate benötigt werden, während bei „einfachen“ Anfragen, dies nicht möglich ist. Dabei überführen sie jeweils die relationale Anfrage in eine kanonische Form, um mit den gleichen Algorithmen für jede Anfrage die Datenquelle herzuleiten (Provenance). Auch unterscheiden sie zwischen Set- und Bag-Semantics, also ob Duplikate vorkommen dürfen oder nicht.

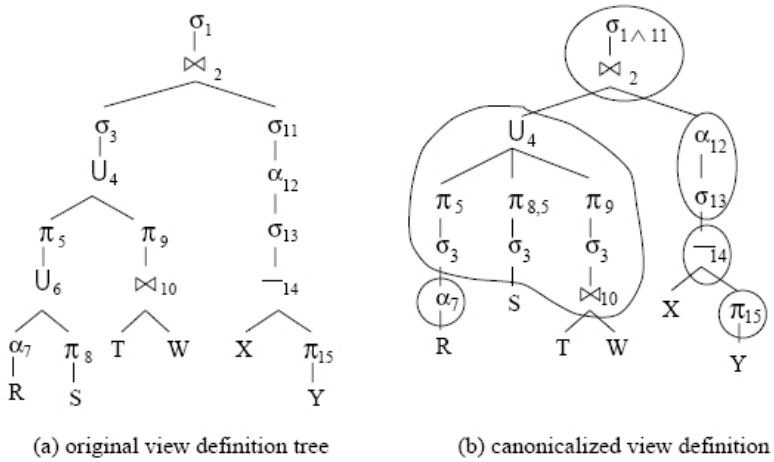

Illustration 5: Eine Sichten-Definition als Baum (a) und deren kanonische Form (b)

Das Data Lineage Konzept von Cui, Widom und Wiener wird bereits in einer anderen Arbeit im Rahmen dieses Seminars behandelt, und deshalb hier nicht weiter erläutert. Es ist aber festzuhalten, dass nur ein Teil der Biotech-Datenbanken relationale Anfragesprachen benutzen, und dass man deshalb diese Algorithmen auf andere Anfragesprachen und Datenhaltungskonzepte erweitern müsste. Eine andere Möglichkeit wäre, allen wichtigen Biotech-Datenbanken eine relationale Schnittstelle zu ihrem System zur Verfügung zu stellen.

4. GUS: Schema für Data Warehousing in der Bioinformatik

Nachdem wir auf die Grundlagen der Bioinformatik, auf den Einsatz von Datenbanken in dieser und auf verschiedene Data Lineage Konzepte theoretisch eingegangen sind, folgt nun schlussendlich noch ein kurzer Einblick auf den Einsatz von Data Lineage in der Praxis. Dafür betrachten wir GUS (Genomics Unified Schema) einem Schema für Data Warehouses in der Biotechnologie.

4.1. Genomics Unified Schema

GUS ist ein kollobaratives Projekt zwischen verschiedenen amerikanischen universitären Instituten und der Universität von Glasgow[6]. GUS ist nur ein Schema und keine Datenbank an sich, wie z.B. SWISS-PROT oder GenBank. Das Ziel von GUS ist es, eine Architektur für eine effiziente und umfassende Integration von Biotech-Daten zu bieten. GUS wird in verschiedenen Projekten benutzt. Die bekanntesten Datenbanken, die GUS einsetzen sind AllGenes[7], PlasmoDB[8] und EPConDB[9]. Während erstere noch eine relativ allgemeine Datenbank für expressed sequence tags und mRNA ist,

[6] CBIL - The Computational Biology and Informatics Laboratory at The University of Pennsylvania's Center for Bioinformatics PSU, The Pathogen Sequencing Unit of the Wellcome Trust Sanger Institute, The Kissinger Lab, CTEGD and the Department of Genetics, University of Georgia , Terry Clark and The Laboratory of Daphne Preuss, The University of Chicago. Fidel Salas, Sucheta Tripathy and Tejal Karkhanisat the Virginia Bioinformtics Institute Andy Jones at the University of Glasgow. [Siehe auch (19)]

[7] http://www.allGenes.org [Abfragedatum: 26.5.2005]

[8] http://plasmodb.org/ [Abfragedatum: 26.5.2005]

[9] http://www.cbil.upenn.edu/EPConDB/ [Abfragedatum: 26.5.2005]

sind die beiden anderen hoch spezialisierte Datenbanken zu sehr speziellen Themengebieten (bestimmte Malaria-Ereger, resp. Daten die durch einen bestimmten DNA-Chip erhoben wurden).

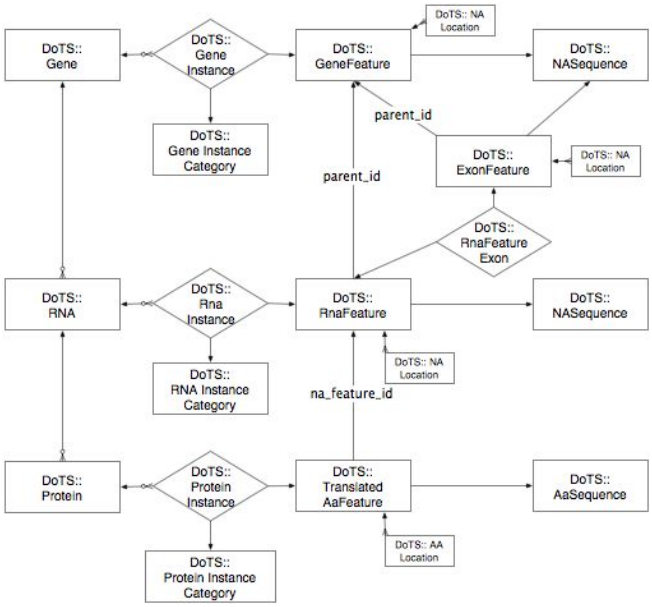

Illustration 6: Der Kern des GUS Schemas zeigt die Beziehungen zwischen Protein, RNA und DNA-Daten (links).

Zentral im Schema von GUS ist, dass kommentierte Sequenzen von DNA, RNA und Proteinen gespeichert werden können. Wobei die natürliche biologische Ordnung eingehalten wird: So ist die DNA der RNA und diese den Proteinen übergeordnet (Die Protein Relation bezieht sich mit einem Fremdschlüssel auf die RNA Relation). Die Datenbank ist so angelegt, dass die Informationen von den bekanntesten Biotech-Datenbanken bezogen werden können, wie z.B. von der International Nucleotide Sequence Database Collaboration (GenBank, DDBJ und EMBL) von SWISS-PROT und dbEST.

GUS legt viel Wert darauf, dass alle diese Daten ausführlich kommentieren und nachbearbeiten werden können. So glauben die Entwickler von GUS, dass nicht die grundlegenden Daten selber, sondern va. deren manuelle Überarbeitung in der Zukunft die entscheidenden Informationen sind[19].

GUS hat eine erweiterbare Architektur. Das Basis-Datenbankschema in GUS ist relational, aber darüber wurde eine Objektorientierte Schicht mit Pearl gelegt. So können die Relationen vom User ähnlich wie Objekte behandelt werden. Diese Objektorientierte Schicht unterstützt zudem das Data Lineage, in dem es Änderungen an den Tupeln verfolgt und in den entsprechenden Relationen festhält.

CBILBLD: DoTS::Protein

180. CBILBLD..DoTS::Protein (standard table)

A central dogma (GENE->RNA->PROTEIN) table containing Protein information for GUS proteins

column	nulls?	type	parent table	description
protein_id	no	number(10)		primary key
name		varchar2(255)		"The name of a protein (e.g., P450 monooxygenase)"
rna_id	no	number(10)	DoTS::RNA	The identifier of the rna which results in the protein entry
review_status_id	no	number(10)	SRes::ReviewStatus	The identifier of the review status
description		varchar2(500)		A description of the protein
reviewer_summary		varchar2(4000)		Annotator comments regarding the Protein
modification_date	no	date(7)		

Illustration 7: Ausschnitt aus dem Schema einer Relation in GUS (zentrale Protein Relation). Hier sieht man auch gut die Beziehung zu einem übergeordneten RNA-Eintrag (ID eines DoTS::RNA-Eintrages im GUS)

4.2. Data Lineage in GUS

In der Biotechnologie ist es aus zwei Gründen sehr wichtig, die „Provenance“ der Daten festzuhalten. Dabei geht es nicht nur um die Herkunft der Daten (aus welcher externen Datenbank wurde diese bezogen), sondern fast relevanter ist es, die Entstehung/Entwicklung, der in dem Date Warehouse integrierten Daten zu verfolgen. Wie wir bereits mehrmals angesprochen haben, ist ein grosser Teil der Daten in der Biotechnologie bisher nur durch ungenaue Berechnungen von Hochleistungsrechner entstanden (wie z.B. bereits erwähnt mit EST-Daten). Und trotzdem „masst“ es sich die Forschung an, auf der Basis dieser nicht fundierten Daten, allerlei weitere Berechnungen und neue Annahmen zu treffen, und so aus potentiell ungenauen Informationen noch ungenauere Resultate zu liefern. Deshalb müssen laufend die bestehenden Daten weiter experimentell validiert werden, und falsche Annahmen korrigiert werden. GUS sollte deshalb, so viele Informationen über die Entstehung/Herkunft von Daten zu liefern, dass die Experten, welche diese kommentieren und/oder nutzen, eine genügend gute Schätzung über die Qualität ihrer Datenbasis machen können.

GUS hat ein praktisches Schema, welches gleichermassen die Herkunft beim Laden aus externen Quellen und die Entwicklung beim manuellen oder automatischen Bearbeiten innerhalb des Data Warehouse verfolgt. GUS speichert diese Informationen als Meta-Daten in eigene Relationen. Hauptsächlich sind folgende Klassen von Relationen (Menge von Relationen) für die Data Provenance verantwortlich[10]:

- **Algorithm**: Erhält bei jedem Computerunterstützten Kommentierungsfunktion (nicht manuelles kommentieren) und beim Laden von Daten aus externen Quellen, das heisst jedes Mal wenn irgendein Algorithmus die Datenbasis verändert, ein Update. Dazu werden folgende Eigenschaften des Algorithmus-Aufrufs gespeichert:
    - Welcher Algorithmus?
    - Die Version des Algorithmus
    - Runtime-Informationen
    - Parameter des Algorithmus-Aufrufs

  Damit könnten, wie wir im vorhergehenden Kapitel über schwache Inversion und präzisere Umkehrungsalgorithmen erkannt haben, sehr viele Informationen im Nachhinein eruiert werden. Tatsächlich werden diese Informationen in GUS aber kaum dazu gebraucht – zumindest ist dies in den Dokumentationen zu GUS nicht beschrieben. Viel mehr werden die dazu notwendigen Informationen als „statische“ Meta-Daten in anderen Relationen festgehalten (siehe unten). In diesem Bereich könnte man sich in Zukunft noch weitere Optionen denken, wie z.B. feinmaschigere Provenance-Bestimmung ähnlich zu [15].
- **External / Data Source:** Darin werden die externen Datenquellen (z.B. Name, ID des benutzten Datensatz, Version der Basisdatenbank) festgehalten. Zusammen mit den Informationen aus den Algorithmen-Relationen ist somit die „klassische“ Provenance bestimmt.
- **Evidence**: Dient als Verbindung zwischen Daten in einer Ziel-Relation und Fakten (Facts). Dabei wird für die Kommentierungen (sei es manuell oder automatisch) von Datensätzen festgehalten, auch welche Fakten sich diese beziehen.
- **Facts**: Wenn Algorithmen auf der Datenbasis ausgeführt wird (z.B. ein Vergleich mit einem BLAST Algorithmus), werden diese als Fakten festgehalten. In dem genannten Beispiel erfolgt dies in einer „Similarity“-Relation.

[10] http://www.gusdb.org/wiki/index.php/DataProvenance

- **Version**: Zentral um die Verarbeitungsgeschichte von Tupeln festzuhalten. Alle Relationen von GUS werden in Versionen-Relationen gesichert. Dabei ist es nicht das Ziel eine Sicherungskopie der Daten zu erstellen (dies muss ausserhalb des GUS-Schema erledigt werden). Damit können aber z.B. auch Sequenzen von DNA gefunden werden, die aus bereits wieder aus der Datenbank gelöscht wurden.

Da ein einzelner SWISS-PROT Eintrag wegen dessen verschachtelter Struktur (z.B. Literaturverweise), auf 15 Relationen aufgeteilt wird, und dies auch für die meisten anderen Quell-Datenbanken gilt, resultieren sehr viele Relationen. In der dritten Version von GUS (3.0) habe ich über 400 Relationen gezählt. Hinzu kommt, dass diese durch die Versionisierung jeder einzelnen Relation in mehrfacher Ausführung existieren.

4.3. Illustration des Einsatzes von Data Lineage in GUS

Folgend wollen wir die Zusammenhänge zwischen Ziel-Relation, Facts-Relation und Evidence anhand der Illustration auf der nächsten Seite kurz illustrieren. Die Relation DoTS::Evidence enstammt dem GUS-Schema, die anderen beiden Screenshots stammen von einer Anfrage an Allgenes.org, welches GUS verwendet. (2: TARGET) zeigt wie Funktionen eines Gens einem bestimmten Gen in Chromosom 20 zugeordnet werden. Dabei wird die Qualität dieser Zuordnung mit „Gut“ bezeichnet (Was das heisst, wird auf einer weiteren Info-Page genauer spezifiziert). Auf welchen Daten diese Charakterisierung beruht, erhält man, wenn man den Link rechts anwählt. Dadurch wird man auf (3: FACT) verwiesen, welche diesen Vergleich (Alignment) graphisch aufzeigt und mit genauen Daten unterlegt. Dahinter steckt ein Evidence-Eintrag, wie wir in schematischer Form in (1: EVIDENCE:) sehen.

5. Fazit

Hier wollen wir nur noch einmal die allerwichtigsten Erkenntnisse dieser Arbeit auflisten:

- Grosse vernetzte Datenbanken sind das zentrale Mittel für die weitere Entschlüsselung des Aufbaus der Lebewesen und somit der weiteren Entwicklung in der Biotechnologie und Medizin.
- Sichten-Konzepte (Views) reichen nicht aus, um integrierte Informationen weiterzuverarbeiten und Data Warehouses sind deshalb oftmals nötig.
- Durch die komplexen Zusammenhänge zwischen den einzelnen Informationen und der riesigen Datenmenge sind Data Lineage-Methoden in Biotech-Datenbanken ein wichtiger Faktor.
- Bisher wird in diesen Datenbanken zumeist noch mit Flat-Files und Data Lineage durch Meta-Daten gearbeitet.
- Hier ist eine Entwicklung von feinmaschigeren und automatisierten Provenance-Methoden wünschenswert.
- Zusätzlich könnte die Integration vereinfacht werden, wenn alle Datenbanken auf relationale oder Objektorientierte Datenbankkonzepte umsteigen.
- Eine standardisierte Ontologie durch ein Semantic Web würde dies zusätzlich unterstützen.
- Da die Informationsbasis durch immer mehr Grunddaten auch in Zukunft exponentiell wachsen wird, und die DNA-Daten von einzelnen Individuen früher oder später auch miteinbezogen werden wird, muss der Data Provenance in der Biotechnologie heute noch mehr Beachtung geschenkt werden.

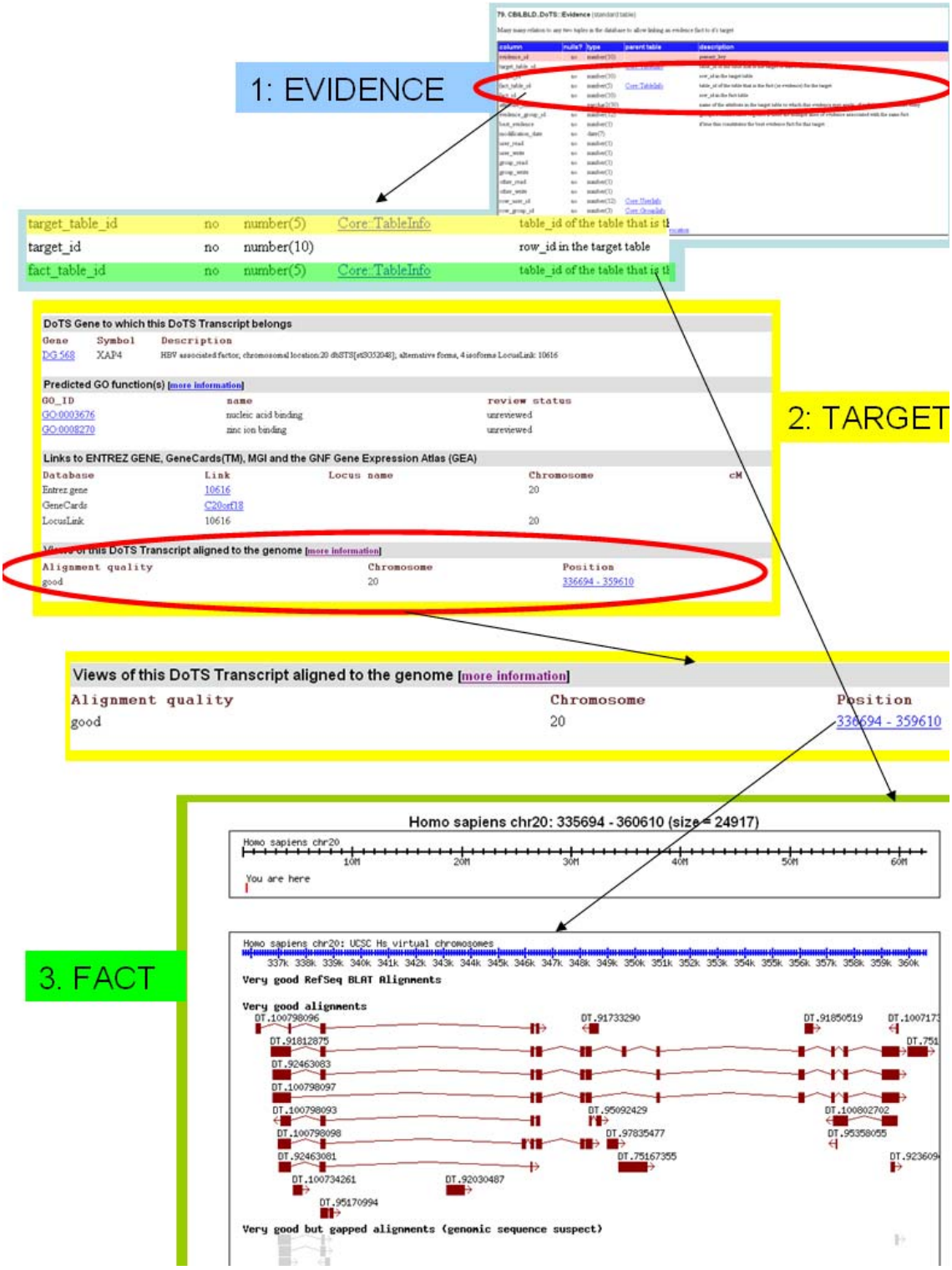

Illustration 8: Data Lineage in GUS: Eigene Darstellung, zusammengesetzt aus Screenshots vom GUS Schema und von Allgenes.org. Erklärung zu der Grafik siehe oben.

Literaturverzeichnis

[1] U.S. Department of Energy, Office of Health and Environmental Research: “Sequencing the Human Genome”, Summary Report of the Santa Fe Workshop, Santa Fe, NM, 3 to 4 March 1986

[2] W.F. Anderson: “Human gene therapy”, Nature 392, p. 25-30, 1998

[3] U. Leser, P. Rieger, „Integration molekularbiologischer Daten“, Datenbankspektrum, Ausgabe 6, 2003

[4] SF. Altschul, W. Gish, W. Miller, EW. Myers, DJ. Lipman: “Basic local alignment search tool”, J Mol Biol 215(3), 1990

[5] A. Baxevanis: “The Molecular Biology Database Collection: an updated compilation of biological database resources”, Nucleic Acids Research, Vol. 29, No. 1, Seiten 1-10, 2001

[6] G. Lirk: „Bioinformatics for Biotechnology“, Vorlesungsskript Einleitung, Fachhochschule Krems, Sommersemester 2005

[7] Davidson, S., Crabtree, J., Brunk, B., Schug, J., Tannen, V., Overton, C., Stoeckert, C., “K2/Kleisli and GUS: Experiments in Integrated Access to Genomic Data Sources”, IBM Systems J., 40, 512–531, 2001

[8] N. Miled, G. Li, G. M. Kellett, B. Spies, O. Bukhres: “Complex life science multidatabase queries”, Proc. IEEE, vol. 90, Seiten 1754–1763, Nov. 2002

[9] S. B. Davidson, C. Overton, V. Tanen, L. Wong: BioKleisli: “A digital library for biomedical researchers”, J. Digital Libraries, vol. 1, no. 1, Seiten 36–53, Nov. 1997.

[10] L. Wong: Kleisli: “Its Exchange Format, Supporting Tools and an Application in Protein Interaction Extraction”, Proceedings of the IEEE International Symposium on Bio-Informatics and Biomedical Engineering, Arlington, VA, Seiten 21–28, 2000

[11] G. Wiederhold: “Mediators in the architecture of future information systems”, IEEE Computer, 18, Seiten 38-48, Nov 1992

[12] P. Buneman, S. Khanna, W.-C. Tan: “Data Provenance: Some Basic Issues”, Foundations of Software Technology and Theoretical Computer Science (2000)

[13] G. Chin, C. Lansing, “Capturing and supporting Contexts for Scientific Data Sharing via the Biological Sciences Collaboratory”, Proceedings of the 2004 ACM conference on Computer supported cooperative work, 2004

[14] A. Bairoch, R. Apweiler: “The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000”, Nucleic Acids Res., 28, Seiten 45–48, 2000

[15] A. Woodruff, M. Stonebraker: "Supporting Fine-Grained Data Lineage in a Database Visualization Environment", In Proceedings of the 13th International Conference on Data Engineering, Birmingham, England, April 1997

[16] D. Liu, M. Franklin: "GridDB: A Data-Centric Overlay for Scientific Grids", Proceedings of the 30th VLDB Conference, Toronto Canada, 2004

[17] R. Boss: "A conceptual framework for composing and managing scientific data lineage", 14th International Conference on Scientific and Statistical Database Management, Seiten 47-55, 2002

[18] N. Hachem, I. Qiu, K. Gennert, M. Ward: „Managing derived data in the Gaea scientific DBMS", In Proceedings of the Conference on Very Large Data Bases (VLDB, Dublin, Ireland). Morgan Kaufmann Publishers Inc., San Francisco, CA, 1–12. 1993

[19] Y.Cui, J.Widom, J.L. Wiener: "Tracing the lineage of view data in a warehousing environment" ACM Tods, 25(2): Seiten 179-227, 2000

Abbildungsverzeichnis / Bildquellen