# 4. Ergebnisse

## 4.1. Die Klon-Datenbank *CloneBase*

### 4.1.1. *CloneY* - cDNA-Klone und Oligomere für Expression-Microarrays

Die Sammlung von cDNA-Klonen und DNA-Oligomeren, die in der Abteilung verfügbar sind und hauptsächlich für die Erstellung von Microarrays für Expressionsstudien unterschiedlicher Tumorentitäten eingesetzt werden, umfasst ca. 77000 Proben aus unterschiedlichen Quellen (Tabelle 1). Es handelt sich dabei einerseits um zwei Kollektionen muriner cDNA-Klone. Die eine Klonsammlung stammt von der Firma LION Bioscience (Heidelberg) und umfasst 20172 exprimierten Sequenzen (*ArrayTAG*™), die andere wurde vom National Institute of Aging (NIA) am National Institute of Health (Bethesda, USA) mit ca. 22656 exprimierten Sequenzen der Embryonalentwicklung bereitgestellt. Andererseits sind 26827 70 basenpaarlangen Oligomer-Fragmente mit humanen Sequenzen (Operon Technologies Inc., Alameda/USA) verfügbar. Schließlich gibt es eine Sammlung von 6773 ausgewählten, Projektspezifischen humanen cDNA Klonen. Diese Klone wurden hinsichtlich verschiedener Kriterien selektioniert. In Abhängigkeit von der jeweiligen Fragestellung wurden sie z.B. auf Grund einer vermuteten oder bestätigten Relevanz für Leukämien oder für eine Funktion beim Aufbau des Mitoseapparates der Zelle ausgewählt. Sie wurden vom Ressourcenzentrum des Humangenom-Projekts Deutschlands (RZPD GmbH, Berlin) bezogen.

Sämtliche Klone, die auf den Expressions-Microarrays der Abteilung vorhanden sind, wurden in der Tabelle *CloneY* der Datenbank *CloneBase* erfasst und katalogisiert. Umfassende Daten für die Identifikation und Charakterisierung der Fragmente sind hier archiviert. Als verlässliche Ausgangsinformation (Schlüssel) gilt jeweils die Information, die als Identifikation aus den öffentlichen Datenbanken von den jeweiligen Klon-Lieferanten bereitgestellt worden war. Dies sind zum einen Accession-Nummern der GenBank-Datenbank (http://www.ncbi.nlm.nih.gov/), die sich auf eindeutige Sequenzen beziehen (Benson *et al.,* 2004) oder auch Image-IDs,

welche vom IMAGE-Konsortium für jeden dort erfassten Klon vergeben werden (Lennon *et al.,* 1996).

| Herkunft der Sammlung | Zielorganismus und Fokus der Sammlung | Art der Fragmente | Anzahl der Frag-mente | Ausgangs-ID |
|---|---|---|---|---|
| LION Bioscience AG Heidelberg | Maus, allgemein | 200-600 kb cDNA-Fragmente in *Bluescript*- Vektoren[1] | 20172 | LocusLink-ID |
| NIA/NIH USA | Maus, Embryonalentwicklung | 1,5 kb cDNA-Fragment in *pSPORT1*- Vektoren[2] | 22656 | LocusLink-ID |
| RZPD GmbH, Berlin | Mensch, Onkogene | cDNA-Fragmente verschiedener Größe in verschiedenen Vektoren | 1928 | IMAGE-ID |
| RZPD GmbH, Berlin | Mensch, Mitose-Apparat | ~ | 853 | IMAGE-ID |
| RZPD GmbH, Berlin (über Böhringer Ingelheim, Wien/AU) | Mensch, Hämatologie (Stratowa *et al.,* 2001) | ~ | 1379 | IMAGE-ID |
| RZPD GmbH, Berlin | Mensch, verschiedene Schwerpunkte | ~ | 2956 | IMAGE-ID |
| Operon Technologies Inc., Alameda/USA | Mensch, allgemein | 70mer Oligomer-Fragmente | 26827 | Ensembl-ID |
| Universität Düsseldorf und Charité Berlin | Tumorsuppressoren (im 1p36 und 19q13-Contig) | cDNA-Fragmente verschiedener Größe in verschiedenen Vektoren | 2161 | IMAGE-ID |

**Tabelle 2: Zusammensetzung der Klonsammlung in der Tabelle *CloneY***

[1] *pBluescript II KS*-Vektoren: BD Biosciences Clontech, Heidelberg.

[2] Ampizillin-resistente *pSPORT1*-Vektoren: Life Technologies, USA

Für die weitere Verwendung und eindeutige Identifikation erhält jeder Klon eine interne *CloneY-ID*, die sich aus einem Code von drei Buchstaben und einer jeweils fortlaufenden fünfstelligen Nummer zusammensetzt. Die Buchstaben geben einen Hinweis auf den Zielorganismus (H/human, M/murin) und die Herkunft der Bibliothek (z.B. BO für die Sammlung von Böhringer); eine Beispiel-ID wäre HBO*00192*.

Darüber hinaus wurden weitere Informationen zu den einzelnen Klonen möglichst standardisiert und nach gleichen Mustern gesammelt. Dies umfasst die Lokalisierung im Genom, enthaltene Gene und Informationen zu diesen wie Datenbank-Kennnummern, Funktionen, u.a. Des Weiteren stehen z.B. für die murinen Klone der Sammlung von LION Bioscience weitere Funktions-Informationen aus der *ArrayBase*-Datenbank von LION zur Verfügung. Die Generierung möglichst umfassender Zusatzinformationen wurde durch Abfragen aus öffentlichen Datenbanken generiert und in der Tabelle *Genes* gespeichert. Die Verbindung zwischen den Tabellen *CloneY* und *Genes* wird über eine dritte Tabelle realisiert, entsprechend der Theorie von relationalem Datenbank-Design mit *Entity-Relationship*-Paaren (ER-Modell; Chen 1976). Dieses Modell beschreibt Daten als Objekte (*Entities*), die in definierten Beziehungen (*Relationships*) zueinander stehen. Das Schema der Datenbank *CloneBase* ist in Abbildung 3 gezeigt. Das relationale Design von Datenbanken ist darauf ausgerichtet, Redundanzen in den verwalteten Daten zu reduzieren, indem Einträge singulär gespeichert, aber mehrfach vernetzt werden. In der *CloneBase* werden daher Gene und dazugehörige Annotationen singulär in der *Genes*-Tabelle gespeichert – sowohl innerhalb der exprimierten und genomischen Klone, als auch in der Schnittmenge beider Gruppen. Die Vernetzung der Klone- und Gen-Daten erfolgt in der *Clones-Genes*-Tabelle.

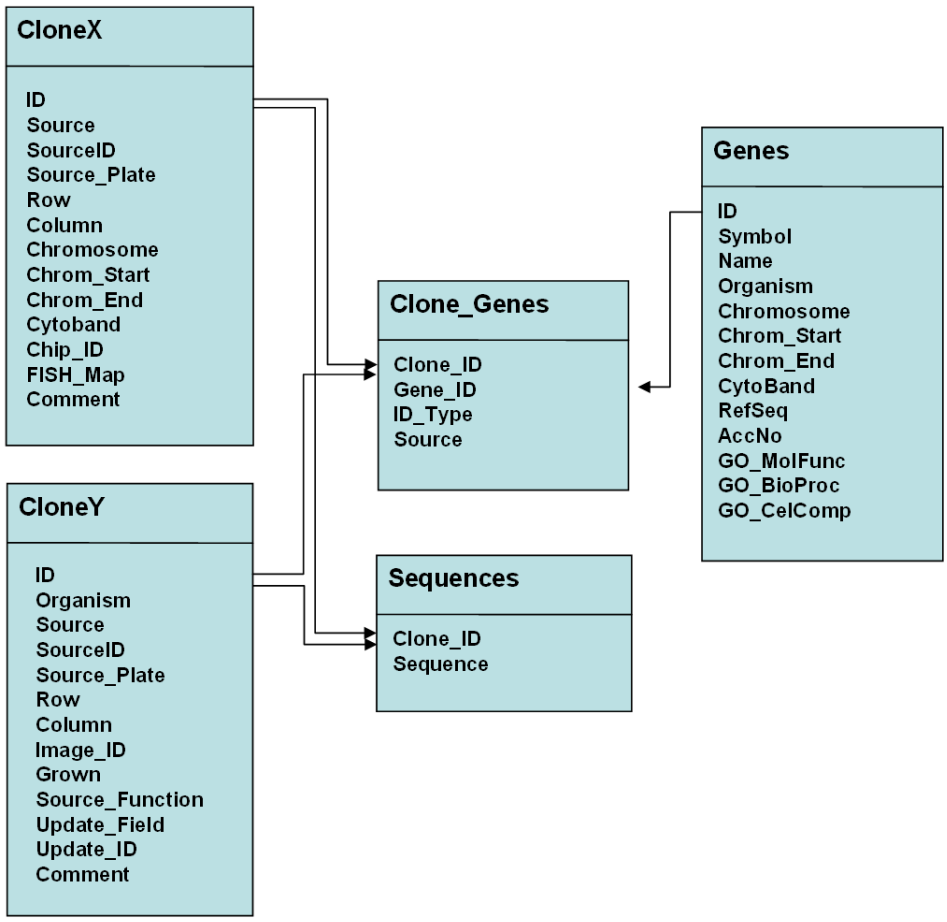



Abbildung 3: Schema der Datenbank *CloneBase*

Entsprechend dem Entity-Relationship-Modell sind Klon- und Gen-Informationen in getrennten Tabellen gespeichert und über eine dritte Tabelle (*Clones_Genes*) verknüpft. Sequenzinformationen sind gesondert abgelegt (*Sequences*). *CloneY* enthält exprimierte Sequenzen, *CloneX* genomische Fragmente (Kapitel 4.1.2.) und *Genes* die Gen-Informationen.

Die in den Datenbank-Tabellen enthaltenen Datentypen und deren Beschreibungen sind in Tabelle 2 zusammengefasst. Die Datentyp-Bezeichnungen sind die folgenden:

- **Varchar(x)** Feld zur Speicherung von x alphanumerischen Zeichen (*Character with variable length*)
- **Int(x)** Feld für Ganzzahlwerte der binären Länge x (*Integer*).
- **Boolean** Feld zur Speicherung von Wahr / Falsch bzw. 1 / 0 –Werten
- **Blob** Feld zur Speicherung von der maximalen Kapazität z.B. alphanumerischer Zeichen im Rohdatenformat (*binary large object*).

| Feldname | Datentyp | Beschreibung |
|---|---|---|
| CloneY_ID | Varchar(9) | 8-stellige Datenbank-interne ID [Primärschlüssel] |
| Organism | Varchar(10) | Humane oder murine Herkunft |
| Source | Varchar(20) | Lieferant der Klone |
| Source_ID | Varchar(50) | ID des Lieferanten |
| Source_Plate | Varchar(20) | Name der Original Klon-Platte des Lieferanten |

| | | |
|---|---|---|
| BarCode | Varchar(11) | Eigener Barcode der Original Klon-Platte |
| PlateName | Varchar(15) | Eigener Name der Platte |
| RowNo | Int(3) | Nummer der Reihe auf der Original Platte |
| RowChar | Char(2) | Buchstabe der Reihe auf der Original Platte |
| Col | Int(4) | Nummer der Zeile auf der Original Platte |
| Chromosome | Int(3) | Lokalisierung des Klons: Chromosom |
| Grown | Boolean | Wachstum des Klons |
| GenePixID | Int(3) | Funktion des Klones |
| Mapping | Varchar(20) | Lokalisierung des Klons: Chromosom, Arm und Bande |
| Image_ID | Varchar(10) | ID des IMAGE-Konsortiums des Klons |
| AccNo | Varchar(200) | ID der Sequenz in der GenBank (*) |
| Process | Blob | GO-Eintrags-Beschreibung: Funktion des Genprodukts im Organismus (*) |
| Source_Function | Varchar(200) | ArrayBase-Eintrag: Funktion des Gens |
| Source_Keywords | Varchar(200) | ArrayBase-Eintrag: Funktionsstichworte (*) |
| Source_Tissue | Varchar(30) | ArrayBase-Eintrag: Gewebe, aus dem die Klon-RNA isoliert wurde |
| Source_Gene | Varchar(50) | ArrayBase-Eintrag: Name des Gens |
| Remark | Varchar(20) | Zusätzliche Kommentare |
| Storage | Varchar (16) | Lagerungsort der Originalplatte |
| SeqVer | Boolean | Erfolgte Sequenzierung des Klons |
| SeqOK | Boolean | Sequenz-Bestätigung des Klons |
| UpdateField | Varchar(20) | Name des Datenbank-Feldes, welches zum Update verwendet werden soll |
| UpdateID | Varchar(30) | Wert des Datenbank-Feldes, welches zum Update verwendet werden soll |

**Tabelle 3: Beschreibung der Felder der *CloneY*-Tabelle**

Feldname entspricht der Benennung einer Tabellenspalte, Datentyp entspricht der Art von Daten, die in dieser Spalte gespeichert werden können.

(*) = Es können mehrere, durch Kommata getrennte Einträge vorhanden sein.

| Feldname | Datentyp | Beschreibung |
|---|---|---|
| Gene_ID | Varchar(20) | Ensembl-ID des Gens |
| Gene_Symbol | Varchar(100) | Offizieller Gen-Kurzname |
| Gene_Name | Varchar(200) | Offizieller Gen-Name |
| Organism | Varchar(60) | Organismus |
| Mapping | Varchar(20) | Chromosomale Lokalisierung (Chromosome, Arm, Bande, Subbande) |
| Chromosome | Varchar(20) | Chromosomale Lokalisierung (Chromosome) |
| MB_Start | Int(6) | Chromosomale Lokalisierung (Start in MB) |
| MB_End | Int(6) | Chromosomale Lokalisierung (Ende in MB) |
| RefSeq | Varchar(200) | ID der Referenz-Sequenz (NCBI) |
| AccNo | Varchar(200) | Accession-Nummern (*) |
| LokusLink | Varchar(200) | LokusLink-ID |
| Process | Varchar(200) | GO-Eintrags-Beschreibung: Funktion des Genprodukts im Organismus(*) |
| Function | Varchar(200) | GO-Eintrags-Beschreibung: Funktion des |

| | | |
|---|---|---|
| | | Genprodukts in zellulären Prozessen (*) |
| Cell_Localization | Varchar(200) | GO-Eintrags-Beschreibung: Zelluläre Lokalisierung des Genprodukts (*) |
| Process_ID | Varchar(200) | GO-Eintrags-ID: Funktion des Genprodukts im Organismus (*) |
| Function_ID | Varchar(200) | GO-Eintrags-ID: Funktion des Genprodukts in zellulären Prozessen (*) |
| Cell_Loc_ID | Varchar(200) | GO-Eintrags-ID: Zelluläre Lokalisierung des Genprodukts (*) |

**Tabelle 4: Beschreibung der Felder der *Genes*-Tabelle**

Feldname entspricht der Benennung einer Tabellenspalte, Datentyp entspricht der Art von Daten, die in dieser Spalte gespeichert werden können.

(*) = Es können mehrere, durch Kommata getrennte Einträge vorhanden sein.

Zur Generierung und Aktualisierung der Annotationsdaten wurde zu Beginn des Projektes das Programm *DB-Updater* in Java entwickelt und angewendet (Kapitel 4.1.3.) und in Folge Perl-Skripte programmiert, welche eine stärkere Automatisierbarkeit erlauben (Kapitel 4.1.4.).

### 4.1.2. *CloneX* - genomische Klone für Matrix-CGH Experimente

In Analogie zu den cDNA- und Oligo-Fragmenten existiert eine umfangreiche Sammlung an genomischen Klonen in der Arbeitsgruppe, welche in der Hauptsache für die Methode der Matrix-CGH erworben und genutzt werden. Die Speicherung und Verwaltung umfasst teilweise andere Informationen, als sie für das *CloneY*-System beschrieben wurden. Die genomischen Klone wurden daher in einer eigenen Tabelle (*CloneX*), jedoch ebenfalls in der Datenbank *CloneBase* gespeichert. Die implementierten Funktionen, die auf die jeweiligen Tabellen zugreifen, sind größtenteils identisch und werden für beide Systeme verwendet. Es sei hier daher lediglich der Aufbau der *CloneX*-Tabelle der Datenbank vorgestellt (Tabelle 5).

*CloneX* enthält zurzeit etwa 6400 Klone mit jeweils 26 Datenpunkten (Tabelle 6).

| Feldname | Datentyp | Beschreibung |
|---|---|---|
| CloneX_ID | Int(11) | Datenbank-interne ID [unique, Schlüssel] |
| Clone_Name | Varchar(20) | internationaler Klon-Name |
| Source | Varchar(20) | Name des Klon-Lieferanten |
| Source_ID | Varchar(20) | Klon-ID des Klon-Lieferanten |
| Source_Plate | Varchar(20) | Platten-Name des Klon-Lieferanten |
| SourcePlate_Row | Varchar(2) | Position: Reihe auf der „Source_Plate“ |
| SourcePlate_Col | Int(2) | Position: Spalte auf der „Source_Plate“ |
| Plate | Varchar(20) | Platten-Name im DKFZ |
| Row | Varchar(2) | Position: Reihe auf der „Plate“ |
| Col | Int(2) | Position: Spalte auf der „Plate“ |
| Clone_AccNo | Varchar(20) | GenBank-Accession Nummer(n) des Klons |
| Chrom | Varchar(5) | Genom. Position des Klons: Chromosom |
| Chr_Start | Int(11) | Genom. Position des Klons: Start-Position in MB |
| Chr_End | Int(11) | Genom. Position des Klons: End-Position in MB |
| Chr_Midpoint | Int(11) | Genom. Position des Klons: Berechneter Mittelpunkt |
| Clone_Length | Int(11) | Länge des Klons in MB |
| Contig | Varchar(20) | GenBank-Accession Nummer des “*Contigs*” |
| GenePixID | Int(20) | Kontroll-Nummer |
| Target | Varchar(20) | Ursprünglicher Grund zur Klonwahl |
| FISH_Map | Varchar(80) | Genom. Position des Klons: *FISH-Banding* |
| FISH_Center | Varchar(80) | Lieferant des FISH-mappings |
| Mapped | Varchar(20) | Kriterium des FISH-mappings |
| BAC_End_1 | Varchar(20) | Accession-Nummer für 5’ Sequence |
| BAC_End_2 | Varchar(20) | Accession-Nummer für 3’ Sequence |
| ensembl_stat | 1 oder 0 | Indikator ob Klon in *Ensembl* gefunden wurde |
| Comment | Varchar(200) | Weitere Kommentare |

**Tabelle 5: Beschreibung der Felder von der *CloneX*-Tabelle**

Feldname entspricht der Benennung einer Tabellenspalte, Datentyp entspricht der Art von Daten, die in dieser Spalte gespeichert werden können.

| Herkunft der Sammlung | Zielorganismus und Fokus der Sammlung | Art der Fragmente | Anzahl der Fragmente | Ausgangs-ID |
|---|---|---|---|---|
| Wellcome Trust Sanger Centre, Hinxton/GB (Fiegler *et al.,* 2003) | Mensch, 1MB Genom-Abdeckung | Genomische PACs (*RP11* und *CalTech* Klone) | 4797 | Internationaler Klon-Name |
| RZPD GmbH, Berlin | Mensch, allgemein | ~ | 2914 | Internationaler Klon-Name |

**Tabelle 6: Zusammensetzung der Klonsammlung in der Tabelle *CloneX***

### 4.1.3. Datenbank-Aktualisierung mit dem *DB–Updater*

Um experimentelle Ergebnisse umfassend analysieren zu können, muss zu möglichst vielen Fragmenten, die auf einem Microarray fixiert sind, eine vollständige Annotation zur Verfügung stehen. Als Annotation bezeichnet man hier sämtliche Zusatzinformationen, die eine Ausgangsinformationseinheit weiter beschreiben. Dies beinhaltet für die Klonsammlung sowohl Identifizierungsnummern von Sequenzen und kodierten Genen, als auch Angaben zu Funktion und Lokalisierung des Gens und des Genprodukts, sofern diese bekannt sind. Diese umfangreichen Informationen können aus unterschiedlichen Quellen – hauptsächlich den öffentlichen Datenbanken – stammen, und müssen regelmäßig aktualisiert werden.
Ein Ziel dieses Teilprojekts war daher, eine Möglichkeit zu schaffen, einerseits neue Klon-Einträge in der Datenbank *CloneBase* mit Annotationsdaten zu vervollständigen und andererseits sämtliche Einträge mit veränderten Informationen zu aktualisieren. Diese Funktionen sollen möglichst robust und leicht bedienbar sein.
Als Annotationsbasis wurde die Datenbank *euGene* von der Universität von Indiana gewählt (Gilbert, 2002). Es handelt sich hierbei um eine so genannte *Meta-Datenbank*, welche Informationen Gen-basiert aus anderen *Primärdatenbanken* zusammenstellt und anbietet. Es wurden außerdem GeneOntology Informationen aus der Datenbank des GeneOntology-Konsortiums genutzt (GeneOntology Consortium 2001), um eine Klassifizierung der Gene in funktionelle Gruppen zu ermöglichen. GeneOntology bezeichnet ein hierarchisches System von Annotationsbegriffen der Kategorien *Molekulare Funktion*, *Biologischer Prozess* und *Zelluläre Lokalisierung*, welches in Form eines azyklischen, gerichteten Graphen aufgebaut ist und sich seit seiner Einführung 2000 als Standard etabliert hat. Als Verifizierung von Gennamen (Name, Symbol, Lokalisierung) wurden Basisinformationen vom Humanen Genom-Projekt (*human genome project, hugo*) genutzt (http://www.gene.ucl.ac.uk /nomenclature).

Zur Verarbeitung der Informationen wurde das Programm *DB-Updater* in der Programmiersprache JAVA entwickelt. Es bietet als grundsätzliche Funktionen zur Aufbereitung der Daten und Einspeicherung in die Klon-Datenbank: den

elektronischen Dateitransfer von einem entfernten Computer (*FTP-Client*), die automatische Text-Verarbeitung zur Informationsextraktion (*Parser*) und die Datenbank-Abfrage.

Dem Benutzer stehen auf einer Maske, die nach Programmstart gezeigt wird, grundsätzlich folgende Möglichkeiten zur Verfügung (Abb. 4):

1. Einlesen einer Datei mit Informationen aus der *ArrayBase* von LION Bioscience. Extraktion der relevanten Informationen und Umwandeln des Formats.
2. Einlesen einer Datei der *euGene*-Datenbank, um die lokale euGene-Version (in der Datenbank *GeneInfo*) zu aktualisieren, die auch ein Ausgangspunkt für die *CloneY*-Annotation ist.
3. Abfragen der für *CloneY* relevanten Informationen von der lokalen *euGene*-Version.
4. Abfragen der den Genen zugeordneten GeneOntology-Nummern von der lokalen GO-Annotations-Installation. Zuordnen der GO-Beschreibungen zu den gefundenen Nummern.

Die erste und die dritte Funktion können verknüpft werden, wodurch die Verarbeitung der enthaltenen Informationen der umfangreichen Klonsammlung von LION auf einen Zwei-Schritt-Prozess reduziert wird. Die Schritte sind dann vergleichbar mit dem Aktualisieren der anderen Klon-Bibliotheken, deren Inhalt in Form von GenBank-Accession-Nummer oder Image-Identifizierern vorliegt (euGene-Abfrage + GO-Abfrage).

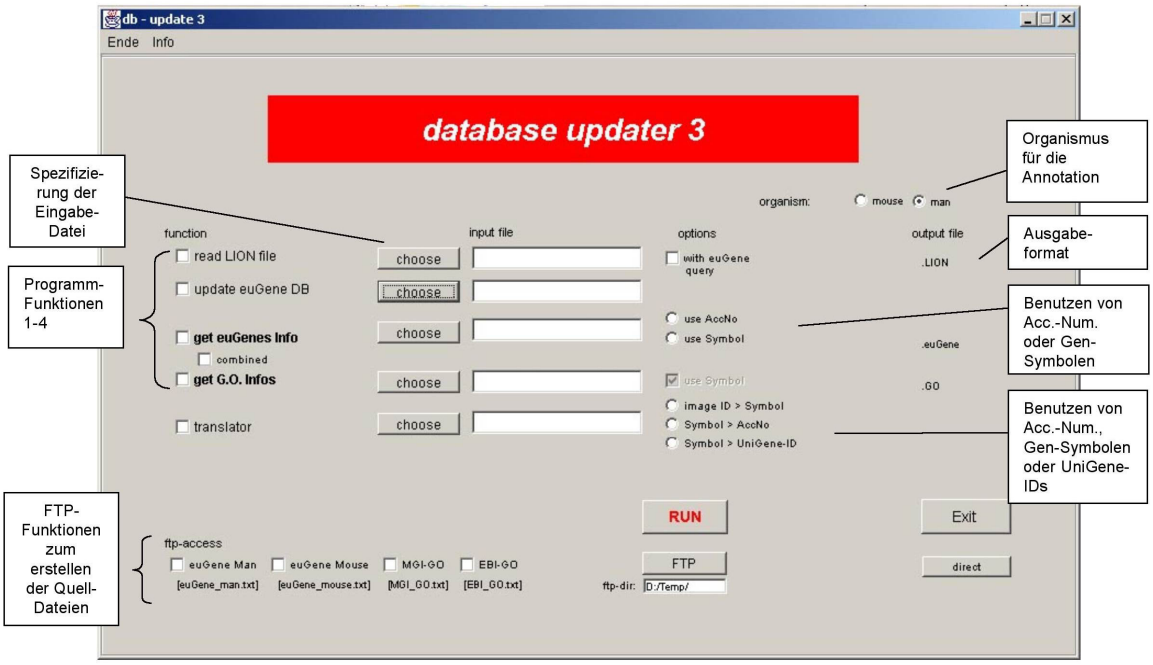



Abb. 4: Ansicht der Programm-Oberfläche vom *DBUpdater*

Das Programm kann sämtliche Ausgangsdaten, die zur Klon-Annotation genutzt werden, bearbeiten und in die Klonsammlung einbringen. Ausgehend von spezifizierten Dateien mit Rohdaten können sehr unterschiedliche Datenbanken (*euGene, MGI-GeneOntology, EBI-GeneOntology*) lokal aktualisiert werden. Dazu existieren FTP- und Parser-Funktionen, die aufeinander aufbauend genutzt werden können.

Die einzelnen Ergebnisse werden als Dateien gespeichert, welche die auf der Maske spezifizierten Endungen besitzen und welche die Daten in Form von Semikolon-getrennten, mit einfachen Anführungszeichen umschlossenen einzeiligen Einträgen beinhalten. Sie können dadurch gut von Tabellenkalkulationsprogrammen wie Microsoft Excel™ gelesen oder direkt in Datenbanken geladen werden.

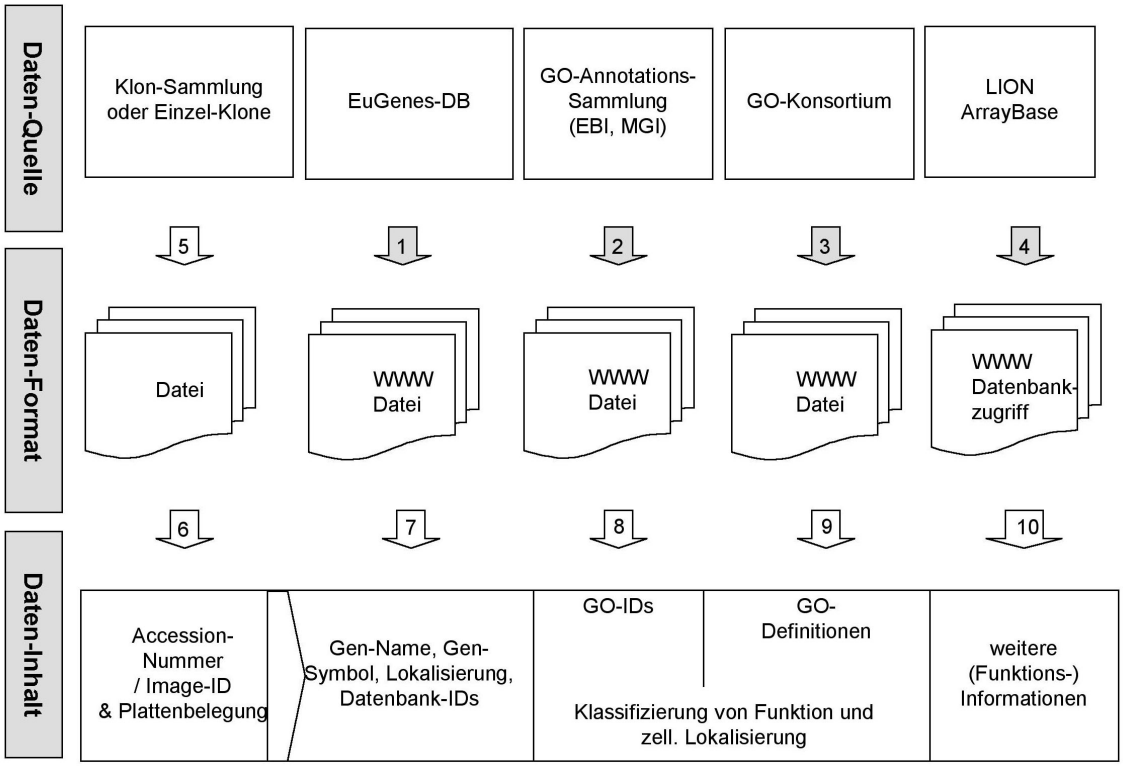



Abb. 5: Die unterschiedlichen Datenquellen der *CloneY*-Tabelle (Stand 2003)

Aus unterschiedlichen Datenquellen (lokalen Dateien oder Informationen aus dem Internet) werden Inhalte durch den *DB-Updater* in die Datenbank integriert und mit den Klonen verknüpft.

## Erläuterung der einzelnen Abläufe

(mit Bezug auf die Nummerierungen 1 bis 10 in den Pfeilen der Abbildung 5)

**A. Aktualisierung der Ausgangsdaten**

1. Der komplette Datensatz der euGene-Datenbank wird vom *DB-Updater* per FTP-Funktionen von der Universität von Indiana (USA) über das Internet abgerufen, die erhaltenen Dateien werden aufbereitet (*geparsed*) und in die lokale Datentabelle *GeneInfo* eingelesen.
2. Als GO-Annotations-IDs werden Daten vom EBI (Europäisches Bioinformatik Institut Hinxton, GB; humane Gene) und MGI (Mouse Genenome Informatics, USA, Jackson; murine Gene) mittels der FTP-Funktionen lokal gespeichert.
3. Die GO-Annotations-Bezeichnungen (Namen zu den IDs aus 2) werden aus einer GO-Version des Jackson-Labors generiert, welche ebenfalls lokal als Tabelle nachgebildet wird.
4. Aus der ArrayBase-Datenbank, die im Passwort-geschützten Internet-Bereich von LION erreichbar ist (http://arraybase.lionbioscience.com), können sämtliche dort zu einer Klon-Sammlung verfügbaren Informationen mittels einer Speicherfunktion

abgerufen werden. Die Dateien müssen zur Nutzung mittels spezieller Parser-Funktionen des *DB-Updater* umformatiert werden (s.u).

**B. Einspeichern neuer Klondaten, bzw.**
**Aktualisierung der vorhandenen Annotations-Daten**

5. / 6. Neue Klone werden mittels einer möglichst eindeutigen Bezeichnung (Accession-Nummer, GenBank-ID, Image-ID, UniGene-Cluster-ID oder behelfsweise Gen-Symbole) aufgenommen. Diese grundlegenden Informationen können für die folgenden Funktionen zur Annotationsdaten-Generierung genutzt werden.
7. Die lokale Version der euGene-Datenbank (in *GeneInfo*) kann mittels einer Funktion nach weiteren Annotationsdaten abgefragt werden, indem eine Liste mit Gen-Symbolen, UniGene-IDs oder GeneBank Accession-Nummers abgearbeitet wird.
8. Über die Gen-Symbole wird dann eine Annotation mit den zutreffenden GeneOntology-Identifizierungsnummern erreicht.
9. Im zweiten Schritt werden zu jeder der gefundenen GO-Nummer die Beschreibung des Gene-Ontology-Konsortiums aufgenommen und zusammen mit der Nummer in einer Datei mit der Bezeichnung *Name.GO* gespeichert.
10. Das *GeneUpdater*-Programm kann aus einer Datenliste der LION Arraybase (mit üblicherweise ca. 10.000 Klon-Einträgen) die relevanten Informationen heraussuchen (*parsen*). Dabei wird die Hauptinformation in eine Datei mit der Bezeichnung *Name.LION* geschrieben, zugleich erzeugt das Programm zwei Dateien *Name.Sym_temp* und *Name.UniGene_temp*, in denen alle gefundenen Gen-Symbole, bzw. UniGene-Cluster-IDs gespeichert werden. Letztere können für die bereits beschriebenen weiteren Funktionen genutzt werden.

### 4.1.4. Datenaktualisierung mit automatisierten Skripten

Als weitere Datenquelle zur Annotation und Aktualisierung der Klondatenbank bieten sich die *Ensembl*-Datenbanken des Europäischen Bioinformatik Instituts, bzw. des Wellcome Trust Sanger Institutes an (Hubbart *et al.,* 2002). Sequenzdaten von sämtlichen bekannten Genomen werden hier mit computergestützten Methoden annotiert und mit weiteren Daten aus öffentlichen Datenbanken ergänzt. Zu den Datenbanken ist durch eine definierte Schnittstelle eine direkte Verbindung mittels eines lokalen mySQL-Datenbankmanagementsystems möglich, benötigt werden lediglich die Netzwerk-Adresse (*ensembldb.ensembl.org*) und der „Benutzername" (*anonymous*). Es wird außerdem eine eigene Programmierumgebung zur Verfügung gestellt, welche die Abfrage und Manipulation der Datenbank in einer Art Meta-Sprache sowohl in Java als auch in Perl ermöglicht (Perl, bzw. Java-API, *Application Programmer Interface*; Stabenau *et al.,* 2004).
In der Sprache Perl wurden, unter Einbeziehung der *Ensembl* Perl-Module, Skripte geschrieben, welche durch die Nutzung von Informationen der *Ensembl*-Datenbank und der *UCSC*-Datenbank (Karolchik *et al.,* 2003) automatisiert sämtliche gespeicherten Informationen der *CloneBase*-Datenbank aktualisieren und ergänzen. Sie sind als cloneX_update.pl (genomsche Klone, siehe Kapitel 4.1.2.), cloneY_update.pl (Expressions-Klone, siehe Kapitel 4.1.1.) und genes_update.pl auf dem Server-Computer abgelegt und können als so genannter *cron-job* selbsttätig vom System zu definierten Zeitpunkten aufgerufen und bearbeitet oder vom Datenbank-Verwalter ausgeführt werden. Als Zeitintervall ist ein mindestens monatlicher Turnus sinnvoll. Das Ergebnis bzw. aufgetretene Fehler werden automatisch in einem Protokoll aufgezeichnet. Der Ablauf des Aktualisierungsvorganges der Datenbank ist in Abb. 6 dargestellt.

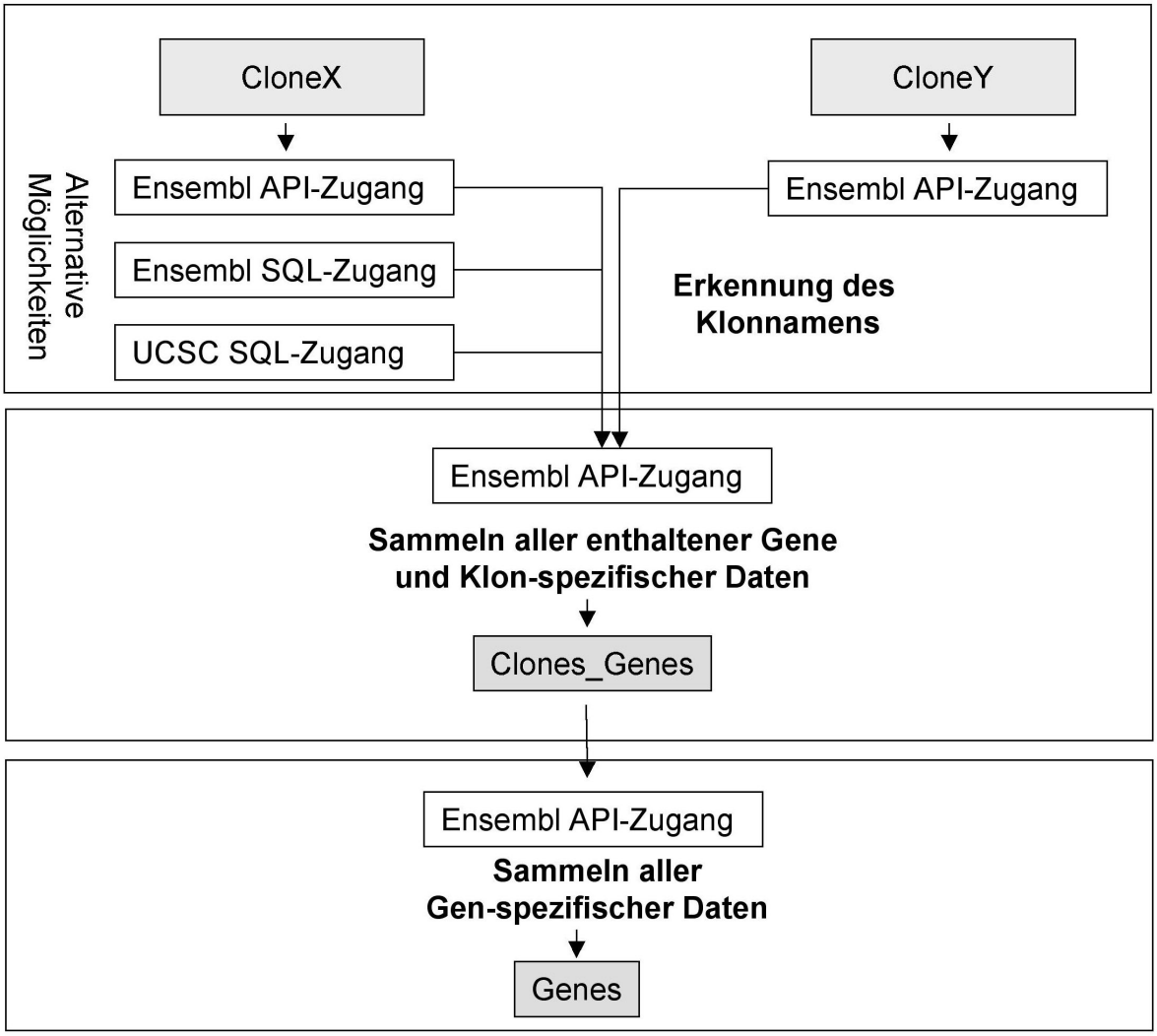



Abb. 6: Aktualisierungsvorgang der *CloneBase* Datenbank mittels automatisierten Perl-Skripten (Stand 2004)

Im ersten Schritt werden Grundinformationen über die Klone basierend auf den individuellen Ausgangs-Identifikationswerten gesammelt. Dies wird für die Expressions-Klone der Tabelle *CloneY* und die genomischen Klone der Tabelle *CloneX* getrennt, aber analog durchgeführt. In den folgenden Schritten werden die evt. im jeweiligen genomischen Bereich lokalisierten Gene ermittelt und weiterführende Annotationen zu den gefundenen Genen gesucht.

API - *Application Developer Interface*, Schnittstelle eines Programms für andere Anwendungen.

### 4.1.5. Internet-Abfrageseiten für den Benutzer

Die enge Anbindung an das Internet (bzw. Intranet) soll den Zugang zur Datenbank von sämtlichen Computern der Abteilung ermöglichen. Vorraussetzung hierzu ist die Installation eines Webservers und die Bereitstellung von Skripten, die auf die Datenbank zugreifen und Benutzeranfragen entgegennehmen können. Die Skriptsprache PHP wird auf dem Server-Computer benutzt, um die Datenbank-Kommunikation zu ermöglichen, und um dem Benutzer die Ergebnisse der Suchanfrage in Form von im Webbrowser darstellbaren dynamischen Webseiten zu präsentieren. Weitergehende Funktionen dieser HTML-Seiten werden über (*Client-sided*) *JavaScript*-Funktionen ermöglicht. Hierzu zählt z.B. eine Reaktion auf die Zeigegerät- (Maus-) Bewegung des Benutzers. Diese Zusammenhänge sind in Abb. 7 graphisch dargestellt.

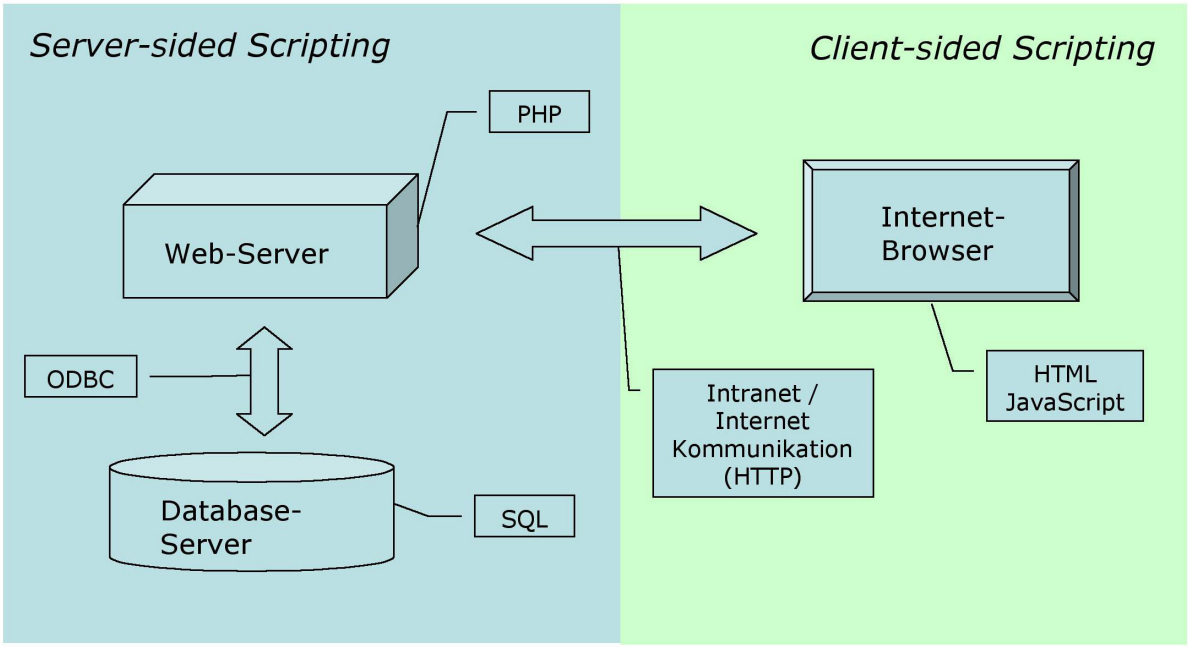



Abb. 7: Eingesetzte Client-Server-Architektur und Technologie

Der Web-Server stellt das Bindeglied dar, um Benutzeranfragen zu bearbeiten und die Inhalte der Datenbank darzustellen. Auf den *Server-Computer* wird die Benutzeranfrage von Webserver verarbeitet. Er sucht über den Datenbankserver die benötigten Daten heraus und liefert sie zurück. Der *Client-Computer* erhält die für ihn formatierte Antwort.

Implementierte Funktionen im Einzelnen:

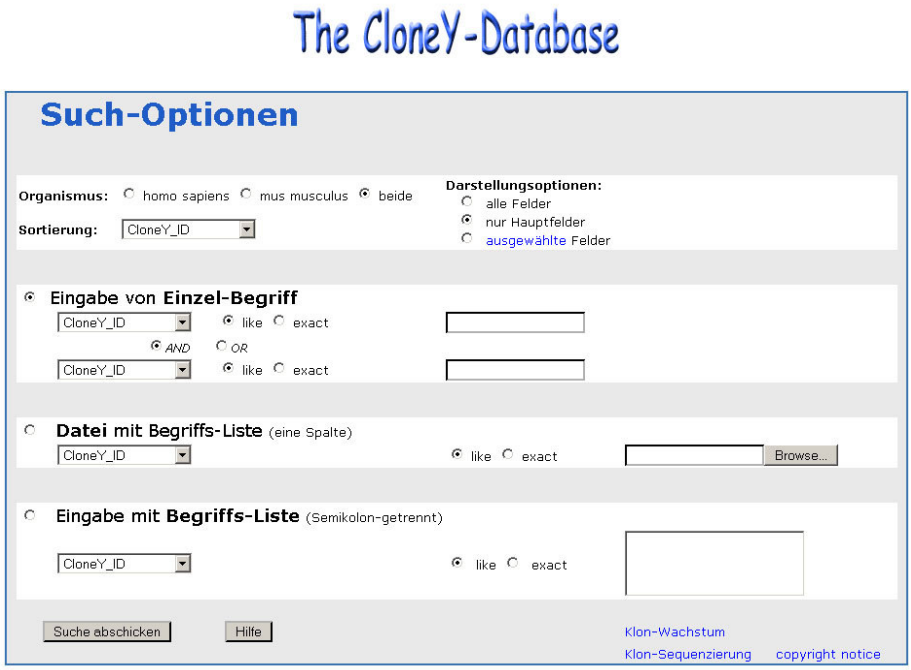
Zur Formulierung einer Anfrage an die Datenbank wird ein Steuerungs-Fenster (*control-page,* Abb. 8.a.) genutzt. Hier werden dem Benutzer in einer Maske sämtliche Möglichkeiten zur Abfrage geboten. Im Einzelnen können folgende Optionen gewählt werden:

A. Suchkategorie: Sämtliche Felder der Datenbank-Tabelle sind abfragbar, z.B. *Gen-Kurzbezeichnung.*
B. Suchbegriff: „Nach was soll im Zusammenhang mit A. gesucht werden“? (z.B. *CDK4*).
C. Kombination von zwei dieser beiden Suchoptionen mit den Logik-Operatoren der additiven (OR) oder der obligatorischen (AND) Verknüpfung (z.B. *GenSymbol = ´CDK4´ AND SeqOK = ´1´*).
D. Optionale Einschränkung der Suche auf humane oder murine Klone.
E. Sortierung des Ergebnisses nach einem der Felder.
F. Abfrage sämtlicher vorhandener Informationen oder Einschränkung auf die grundlegenden Informationen (erhöhte Geschwindigkeit der Antwort). Alternativ können die gewünschten Felder interaktiv ausgewählt werden.
G. Parallele Abfrage von mehreren Klonen, von denen z.B. die *Gen-Kurzbezeichnung* als Semikolon-getrennte Liste in die Maske eingegeben (kopiert) werden oder die als Textdatei automatisch auf den Server geladen werden kann.
H. Verringerung der Stringenz der Suche durch den *like*-Operator (Suchbegriff kann dann Teil eines längeren Eintrags sein).

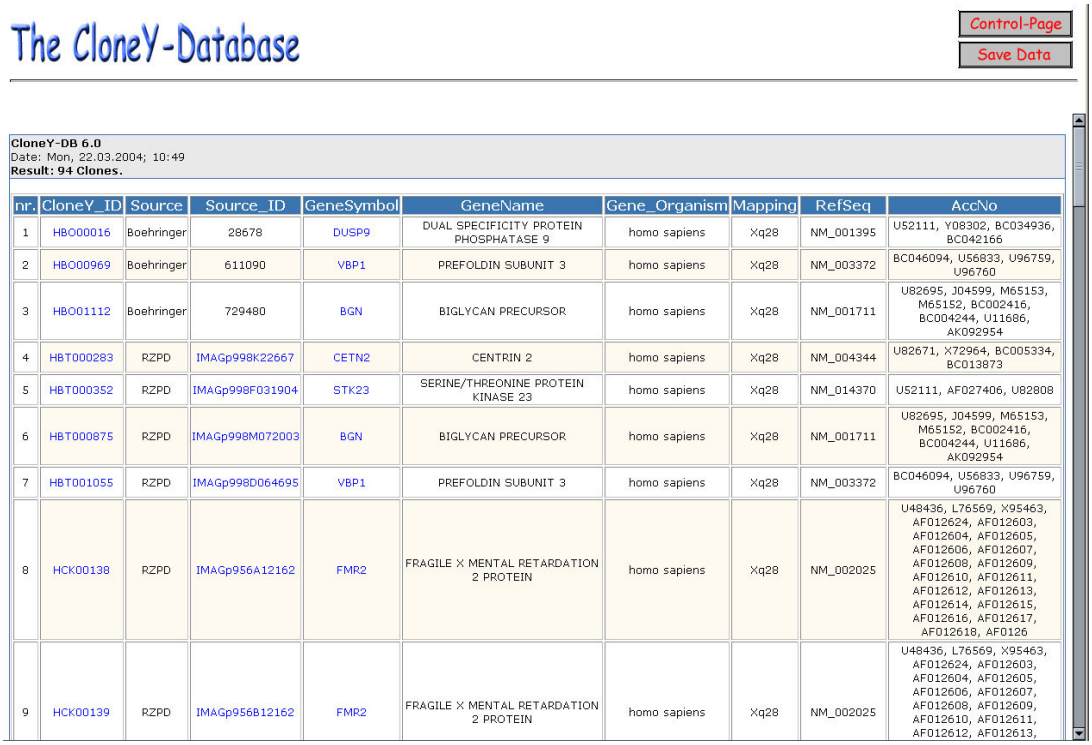
Auf dem Steuerungsfenster sind ferner diese Funktionen anwählbar:

A. Starten der Abfrage.
B. Zurücksetzen sämtlicher Einstellungen der Maske auf die Ursprungswerte.
C. Aufrufen einer Hilfeseite, auf der grundlegende Funktionen erläutert werden.
D. Aufrufen der Seiten zum direkten Eintragen von Bemerkungen und Klon-Wachstum.
E. Aufrufen der Seiten zum direkten Eintragen von Sequenzierungs-Informationen.

a.




b.




| nr. | CloneY_ID | Source | Source_ID | GeneSymbol | GeneName | Gene_Organism | Mapping | RefSeq | AccNo |
|---|---|---|---|---|---|---|---|---|---|
| 1 | HBO00016 | Boehringer | 28678 | DUSP9 | DUAL SPECIFICITY PROTEIN PHOSPHATASE 9 | homo sapiens | Xq28 | NM_001395 | U52111, Y08302, BC034936, BC042166 |
| 2 | HBO00969 | Boehringer | 611090 | VBP1 | PREFOLDIN SUBUNIT 3 | homo sapiens | Xq28 | NM_003372 | BC046094, U56833, U96759, U96760 |
| 3 | HBO01112 | Boehringer | 729480 | BGN | BIGLYCAN PRECURSOR | homo sapiens | Xq28 | NM_001711 | U82695, J04599, M65153, M65152, BC002416, BC004244, U11686, AK092954 |
| 4 | HBT000283 | RZPD | IMAGp998K22667 | CETN2 | CENTRIN 2 | homo sapiens | Xq28 | NM_004344 | U82671, X72964, BC005334, BC013873 |
| 5 | HBT000352 | RZPD | IMAGp998F031904 | STK23 | SERINE/THREONINE PROTEIN KINASE 23 | homo sapiens | Xq28 | NM_014370 | U52111, AF027406, U82808 |
| 6 | HBT000875 | RZPD | IMAGp998M072003 | BGN | BIGLYCAN PRECURSOR | homo sapiens | Xq28 | NM_001711 | U82695, J04599, M65153, M65152, BC002416, BC004244, U11686, AK092954 |
| 7 | HBT001055 | RZPD | IMAGp998D064695 | VBP1 | PREFOLDIN SUBUNIT 3 | homo sapiens | Xq28 | NM_003372 | BC046094, U56833, U96759, U96760 |
| 8 | HCK00138 | RZPD | IMAGp956A12162 | FMR2 | FRAGILE X MENTAL RETARDATION 2 PROTEIN | homo sapiens | Xq28 | NM_002025 | U48436, L76569, X95463, AF012624, AF012603, AF012604, AF012605, AF012606, AF012607, AF012608, AF012609, AF012610, AF012611, AF012612, AF012613, AF012614, AF012615, AF012616, AF012617, AF012618, AF0126 |
| 9 | HCK00139 | RZPD | IMAGp956B12162 | FMR2 | FRAGILE X MENTAL RETARDATION 2 PROTEIN | homo sapiens | Xq28 | NM_002025 | U48436, L76569, X95463, AF012624, AF012603, AF012604, AF012605, AF012606, AF012607, AF012608, AF012609, AF012610, AF012611, AF012612, AF012613, |

Abb. 8: *CloneBase* Web-Oberfläche (nächste Seite)

a. Kontroll-Seite, auf der Suchanfragen spezifiziert werden können. Der Benutzer kann festlegen, nach was er in welchem Feld sucht und was in welcher Reihenfolge im Ergebnis gezeigt werden soll. Es gibt außerdem die Möglichkeit, mit mehreren Suchbegriffen gleichzeitig oder mit einer Suchliste zu arbeiten.

b. Daten-Seite, auf welcher das Ergebnis der Anfrage aufgelistet wird und als Datei gespeichert werden kann; Hyperlinks führen zu zusätzlichen verknüpften Informationen.

Das Ergebnis der Datenbankanfrage wird auf einem zweiten Fenster des Browser-Programms (*Datapage*, Abb. 8.b.) dargestellt. Im Kopfbereich sind hier Funktionen möglich, die erstens das erzielte Ergebnis speichern. Dies erfolgt in Form einer Semikolon-getrennten Textdatei, welche z.B. in Microsoft Excel™ importiert und weiterverarbeitet werden kann. Zweitens kann das Steuerungsfenster aufgerufen werden, um eine neue Anfrage zu formulieren. Im Datenfenster selbst wird das Ergebnis der Datenbankanfrage in Form einer HTML-Tabelle angezeigt. Es wird angegeben, wie viele Klone in der Datenbank von einer Einzelbegriffsuche betroffen waren. Bei der parallelen Abfrage von mehr als 1000 Klonen wird die Datenbankantwort direkt als Textdatei erzeugt und dem Benutzer zum Abspeichern angeboten. In einem gesonderten Fenster können weiterführende Informationen abgerufen werden, die von den angezeigten Klonen, bzw. Genen in anderen Internet-Datenbanken existieren und die als Hyperlink angezeigt werden. Folgende Informationen werden dazu angeboten:

- Das Gen-Symbol führt zu einer Anfrage bei der GeneCards Datenbank (Rebhan *et al.,* 1997)
- Die *Ensembl*-ID verweist direkt auf den entsprechenden Eintrag in der *Ensembl*-Datenbank (Hubbart *et al.,* 2002)
- Die OMIM-ID führt zur OMIM-Datenbank (OMIM, 2000)
- Die Klon-ID verweist auf die Sequenz, die für diesen Klon lokal gespeichert wurde.

Zusätzlich existiert eine *Cascading-Stylesheet*-Datei, die das Erscheinungsbild sämtlicher Internetseiten definiert und vereinheitlicht. Sämtliche Dateien des Teilprojekts sind in der Abbildung 9 im Zusammenhang dargestellt.

Das *CloneBase*-System ist auf einem *Linux*-Servercomputer installiert. Das Betriebssystem *Linux* ist für den Einsatz von Computersystemen in Netzwerken und für Server-Applikationen aufgrund von Sicherheitstechnik (z.B. vollständiger Mehrbenutzerbetrieb mit genau definierbaren Rechten) und Stabilität (z.B. Trennung von Prozessräumen) besonders gut geeignet.

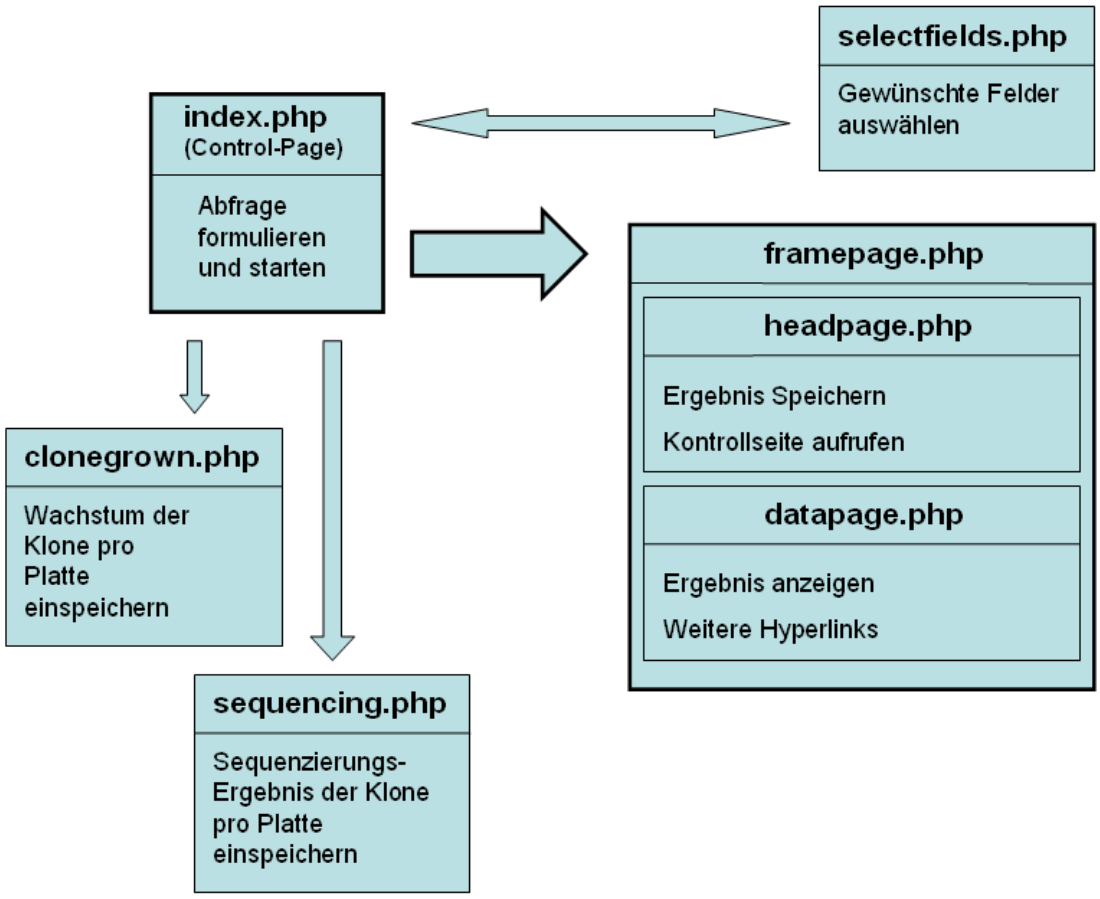



Abb. 9: Internet-Abfrageseiten des *CloneBase*-Projekts

Das Schema zeigt die unterschiedlichen Dateien zur Abfrage der Datenbank und ihre Interaktion. Ausgehend von der Kontroll-Seite (*index.php*) können Skripte genutzt werden, welche die Zielfelder definieren, das Suchergebniss darstellen oder die Eingabe von Informationen zum Wachstum und zur Sequenzierung der Klone ermöglichen.

## 4.2. Die Prozessdatenbank *QuickLIMS*

### 4.2.1. Grundsätzlicher Aufbau

Zur Steuerung und Archivierung des Herstellungsprozesses der innerhalb der Gruppe produzierten Microarrays wurde in Zusammenarbeit mit Dr. Gunnar Wrobel das Labordatensystem *QuickLIMS* entwickelt. Es stellt die Datenbasis für die Roboter-Skripte dar, welche die Aktionen der Maschine steuern. Der Minitrak-Roboter identifiziert eine Platte über deren individuellen Barcode, „fragt“ über die Steuerungs-Skripte das LIMS nach Informationen zu dieser Platte und führt dementsprechend die nächsten Schritte aus. Es leitet Mensch und Maschine durch das definierte Protokoll und archiviert währenddessen sämtliche relevanten Daten.
*QuickLIMS* ist ein Protokoll-basiertes System, d.h. der Programmablauf orientiert sich am tatsächlichen experimentellen Verlauf der Chipproduktion.
Das Protokoll wird aus einer eigenen Tabelle (*Master*-Tabelle) gelesen, in welcher zusätzlich das Format der Parameter (Ganzzahl, Text, usw.), Zugehörigkeit zu einem bestimmten Prozess-Schritt, usw. definiert werden. Daten zu einer Platte können nur in der Reihenfolge des Protokollverlaufs eingegeben und bestimmte Werte müssen zwingend eingetragen werden, bevor der nächste Schritt erreicht werden kann. Bei bestimmten Schritten werden die Datenpunkte vom Roboter direkt in die

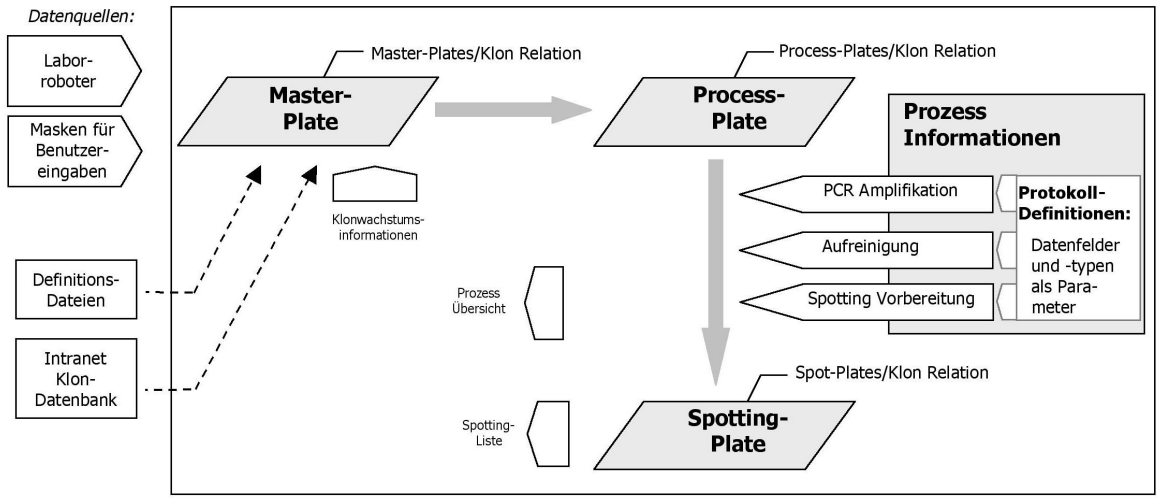



Abb. 10: Platten-orientierter Aufbau von *QuickLIMS*

Daten werden vom Benutzer oder vom Laborroboter in das System geschrieben. Die Klone werden in 96-Loch Mikrotiterplatten als *Master-Plates* im System registriert, gehen in *Process-Plates* den Verarbeitungsprozess ein und werden schließlich auf *Spotting-Plates* zusammengeführt.

Datenbank geschrieben und können vom Benutzer nur nach Passwort-Abfrage editiert werden. Andere Werte werden vom Benutzer über Programm-Masken eingetragen. Diese Formulare sind entweder statisch vordefiniert oder werden dynamisch Protokoll-basierend generiert.

Das Programm ist durch eine dreistufige Passwortabfrage gegen unbefugten und unkontrollierten Zugriff gesichert:

1. Benutzer-Passwort: Zum regulären Bedienen (Anlegen von Platten, Eingeben von Daten, Suchen nach Informationen).
2. Hauptbenutzer-Passwort: Zum Korrigieren von Daten, die der Roboter geschrieben hat.
3. Entwickler-Passwort: Zum Ein-/Ausschalten der Sichtbarkeit des Programmcodes. Im ausgeschalteten Modus kann das *QuickLIMS* durch Drücken der SHIFT-Taste im *Debug-Modus* gestartet werde, was die Überprüfung und Änderungen des Programm-Codes erlaubt.

*QuickLIMS* ist wie im Folgenden beschrieben im Ablauf auf die Bearbeitung von „Platten“ fokussiert (Abb. 10). Die Klone werden in 96-Loch Mikrotiterplatten gelagert (*Source Plates*). Davon werden im Regelfall drei Replikate erstellt (*Master Plates*). In den Bearbeitungsprozess gehen die Klone in Form von sogenannten *Process Plates* ein, die als Abbild der *Master Plates* – virtuell oder tatsächlich – erstellt werden. Zum Übertragen auf Glasobjektträger (*Spotten*) werden die Proben aus vier 96er *Process Plates* zu einer 348-Loch Mikrotiter *Spotting Plate* zusammengefasst.

Sämtliche Platten sind mit einem Barcode markiert, alle Löcher der Platten sind mit einer Koordinate (A1 bis H12) identifizierbar. Datenbankintern ist jede Platte über ihren Code als *Master*-, *Source*- oder *Spotting Plate* gekennzeichnet. Die Prozessplatten sind außerdem mit einer Nummer verbunden, die ihre Position im Produktionsprozess und die nächsten auszuführenden Schritte festlegt (*step count*).

### 4.2.2. Spezifische Funktionen

A. Anlegen neuer Platten

Sämtliche Arten von Platten (Master-, Prozess- und Spotting-Platten) können über mehrere Methoden angelegt werden. Standardmäßig werden sie direkt von Roboter erzeugt, welcher einen bisher unbekannten Barcode liest und kontextbezogen diesen einer neuen Platte der richtigen Art zuweist.

B. Abspeichern von Prozessdaten

Schritte, die vollständig vom Pipettier-Roboter durchgeführt werden, erfordern keine weitere Benutzer-Interaktion. Die Steuerungssoftware der Maschine „fragt“ die Prozessdatenbank nach der aktuellen Platte und leitet daraufhin den nächsten Protokollschritt ein. Beginn und Endzeit werden in der Datenbank gespeichert, ebenso vorhandene Parameter des jeweiligen Schrittes. Möchte der Benutzer Daten dieser Schritte ändern, kann dies nur nach der Abfrage eines weiteren (Hauptbenutzer-) Passwortes erfolgen.

C. Definition von benutzten Chemikalien

Es existieren unterschiedliche Protokolle für den Herstellungsprozess von Microarrays. Außerdem variieren die Hersteller, Oberflächenarten, Chemikalienlösungen oder insgesamt die experimentellen Fragestellungen. Die Änderungen in der Zusammensetzung von einzelnen Lösungen können in einem eigenen Formular eingetragen werden (Bezeichnung und Menge der einzelnen

Chemikalien). Ferner kann es für den Erfolg des Experimentes von entscheidender Bedeutung sein, ob während der Array-Produktion die Charge einer bestimmte Chemikalie gewechselt werden musste. Daher wird dies für die Produktion ebenfalls erfasst.

D. Abspeichern von PCR- und Prozessfehlern

Um das Ergebnis einer PCR-Amplifikation in die Datenbank eintragen zu können, generiert das LIMS eine visuelle Repräsentation ein 96er Mikrotiterplatte. Der Benutzer kann für jede Klon-Position einen Wert abspeichern. Mit der gleichen Methode lassen sich allgemeine Prozessfehler Klon-spezifisch festhalten.

E. Erzeugung von Spotting-Listen

Nach Abschluss der Probenaufbereitung werden die DNA-Lösungen in 384-Loch Mikrotiter-Platten per Pipettier-Roboter auf die Glasoberflächen aufgebracht. Dies erfolgt nach einem genau definierten Schema, welches sich aus der Zusammensetzung und Anordnung der Spotting-Platten ergibt. *QuickLIMS* kann daher als letzten Protokollschritt eine so genannte Spotting-Liste erstellen, eine Textdatei mit sämtlichen Positions- und Klonangaben.

F. Datenbank-Suchfunktionen

Im Labordatensystem können sämtliche Klon-spezifischen Prozessdaten zunächst über die jeweiligen Platten angegeben werden. Es besteht außerdem die Möglichkeit gewisse Such- und Darstellungsfunktionen zu nutzen, welche den Gesamtbestand oder einzelne Bereiche anzeigen. Hier kann z.B. nach der Produktion in bestimmten zeitlichen Fenstern gesucht werden.

G. Weitere Funktionen

Beim Start von *QuickLIMS* erscheint ein Anmelde-Dialog, bei dem der Benutzer ausschließlich mit einem Passwort zum Hauptprogramm gelangt. Zum Sichern im laufenden Betrieb kann durch einen Dialogknopf das Hauptfenster gesperrt und wiederum nur durch Passwort entsperrt werden. Es kann jederzeit eine Seite mit allgemeinen Informationen und Hilfestellungen zum Programm aufgerufen werden.

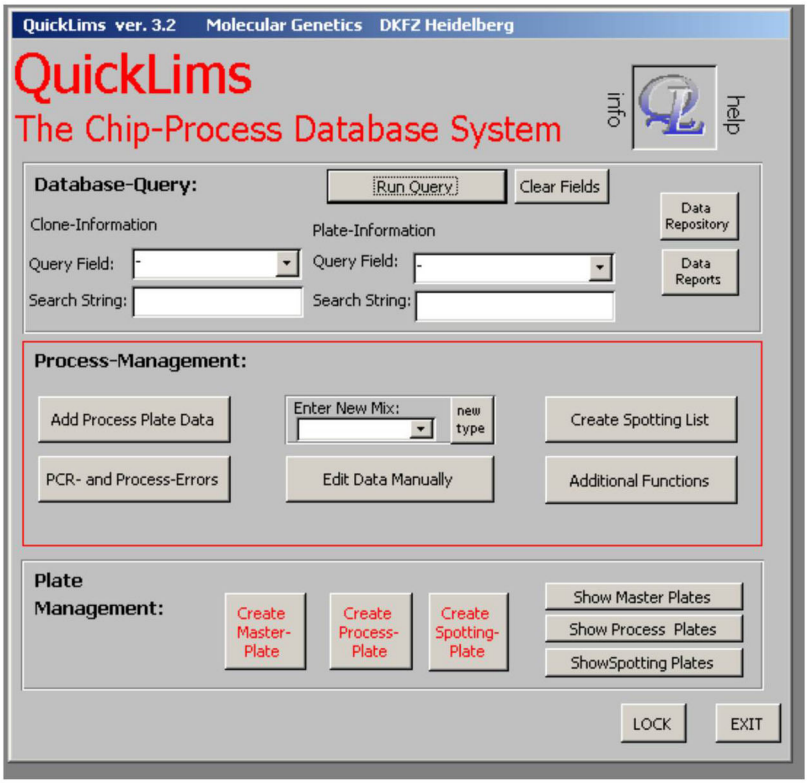



Abb. 11: *QuickLIMS*-Hauptmaske für Benutzer-Interaktionen

Im oberen Bereich (*Database-Query*) können einfache Anfragen nach den archivierten Daten formuliert werden. In der Mitte (*Process-Management*) werden manuelle Schritte der Verarbeitung eingetragen und spezielle Funktionen ausgeführt. Im unteren Drittel (*Plate-Management*) können neue Platten erzeugt und vorhandenen Platten angezeigt werden.

### 4.2.3. Interaktion mit dem Pipettier-Roboter und der Klondatenbank

Der Laborroboter Minitrak wird von Skripten gesteuert, welche in einem Dialekt der Programmiersprache VisualBasic geschrieben sind. Sie wurden im Wesentlichen von Dr. Gunnar Wrobel entwickelt und sind modular aufgebaut. Dies heißt, dass alle unterscheidbaren Funktionen in eigenen Dateien formuliert sind und der Gesamtablauf durch das Zusammenfügen der Einzelkomponenten entsteht. Neben einem Hauptprogramm kontrollieren ein Konfigurations- und ein Ressourcen-Teil die Maschine, die Prozessschritte Animpfen, Fällen, Waschen, Umverteilen, Rücklösen und Trocknen werden von eigenen Modulen durchgeführt (Wrobel, 2004). Aus dem Hauptprogramm erfolgt die Anbindung an das *QuickLIMS*-System.

Bei Bedarf kann *QuickLIMS* Anfragen an die Klondatenbank *CloneBase* stellen. Dies erfolgt einerseits bei der Definition neuer Platten, die Belegung kann dadurch automatisch eingetragen werden. Der Benutzer kann andererseits von jeder Platte ausge-

hend Informationen zu den darauf enthaltenen Klonen erfragen, es werden die Haupt-Annotationen der *CloneBase* dargestellt.

## 4.3. Das *AutoPrime* Programm zur automatisierten Primergenerierung

Die *Quantitative Real-Time Polymerase-Kettenreaktion* (RQ-PCR, Wittwer *et al.,* 1989) wird unter anderem für die Verifizierung von Microarray-Ergebnissen eingesetzt. Die Generierung von Primern für diese Methode wurde durch die Entwicklung des Programms *AutoPrime* in Zusammenarbeit mit Dr. Gunnar Wrobel vereinfacht. Es stellt ein Bindeglied zwischen der Sequenzdatenbank *Ensembl* (Hubbart *et al.,* 2002) und des Programms *Primer3* zur Überprüfung der Primerqualität (Rozen und Skaletsky, 2000) dar.

Um Kontaminationen der RNA durch genomische DNA zu erschweren, können mit *AutoPrime* Sequenzen automatisch so gewählt werden, dass sie eine Exon-Exon-Grenze überspannen. Da diese Basenabfolge ausschließlich in der gespleissten mRNA auftritt, werden genomische intronhaltige Abfolgen unterdrückt. Alternativ kann der Benutzer wählen, dass die Primer-Sequenzen nicht auf den Exon-Exon-Grenzen, sondern auf verschiedenen Exons liegen, sodass das dazwischen liegende Intron

durch ein verlängertes PCR Produkt von der genomischen DNA wiederum die Gefahr der Kontamination durch genomische DNA verringert (Abbildung 12). Es kann außerdem eine so genannte *Mispriming-Library* genutzt werden, eine Sammlung von Sequenz-Fragmenten, die in den Primern nicht enthalten sein darf, sodass unspezifische Amplifikationsprodukte vermieden werden. Hierfür werden *Repeat-Libraries* mit genomischen Wiederholungseinheiten der jeweiligen Organismen genutzt (Jurka 2000), welche mit Genehmigung vom Genetik Information Research Institute (http://www.girinst.org) bezogen worden sind.

Die möglichen Eingabewerte für *AutoPrime* sind die folgenden:

- Gen-Kurzname (Gensymbol) oder *Ensembl*-ID
- Qualitätsparamter der Primer, welche denen von *Primer3* entsprechen
- Auswahl des gewünschten Organismus
- Option, Primer innerhalb von Exon-Bereichen oder ausschließlich an Exon-Übergängen zu suchen
- Option, eine zusätzliche interne Sequenz zur Herstellung eines Hybridisierungs-Oligomers für die RQ-PCR zu finden.
- Wahl des Ausgabeformats: HTML (Internet-Darstellung), Text (vereinfachte Darstellung) oder XML (maschinelle Weiterverarbeitung)

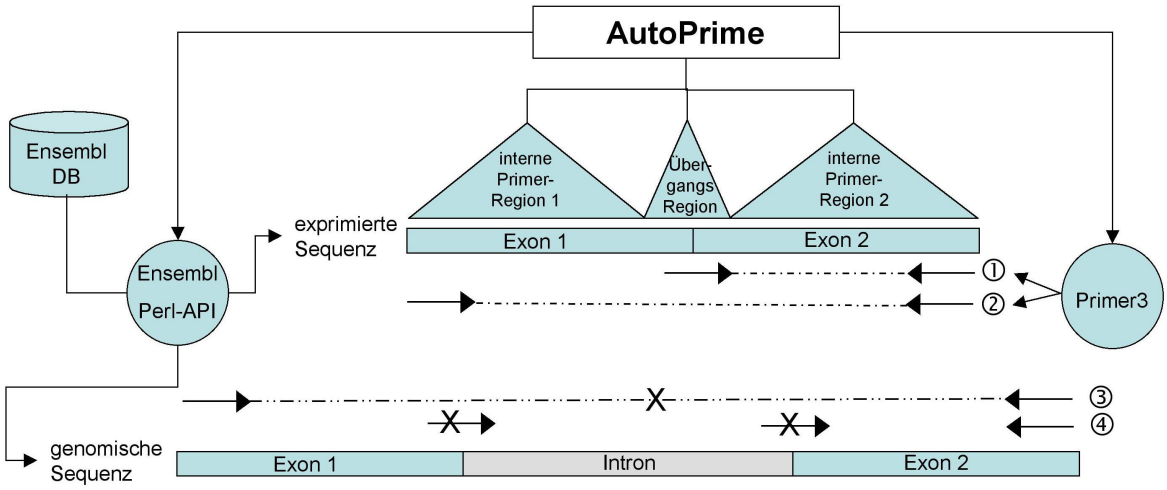



Abb. 12: Funktionsweise von *AutoPrime*

Über Funktionen der Perl-API werden Informationen zur genomischen Sequenz und zu Exon/Intron-Grenzen aus der Ensembl-Datenbank gesucht.

Liegt ein Primer auf einer Exon-Exon-Grenze und der zweite innerhalb des Exons, werden ausschließlich exprimierte Sequenzen amplifiziert (1). Die Primer können nicht auf der genomischen Sequenz binden (4). Liegen die Primer-Sequenzen in unterschiedlichen Exons, würde von der

genomischen Sequenz ein zu langes Produkt (mit Intron) entstehen, die Wahrscheinlichkeit ist sehr gering (3). Stattdessen wird die exprimierte Sequenz amplifiziert (2).

Für folgende Organismen können von *AutoPrime* RQ-Primer generiert werden:

- *Homo sapiens* (Mensch)
- *Mus musculus* (Hausmaus)
- *Rattus norwegicus* (Ratte)
- *Caenorapdidis elegans* (Fadenwurm)
- *Caenorapdidis briggsae* (Fadenwurm)
- *Danio regio* (Zebrafisch)
- *Fugu rupripes* (Kugelfisch)
- *Drosophila melanosgaster* (Fruchtfliege)
- *Anopheles gambiae* (Anopheles-Mücke)

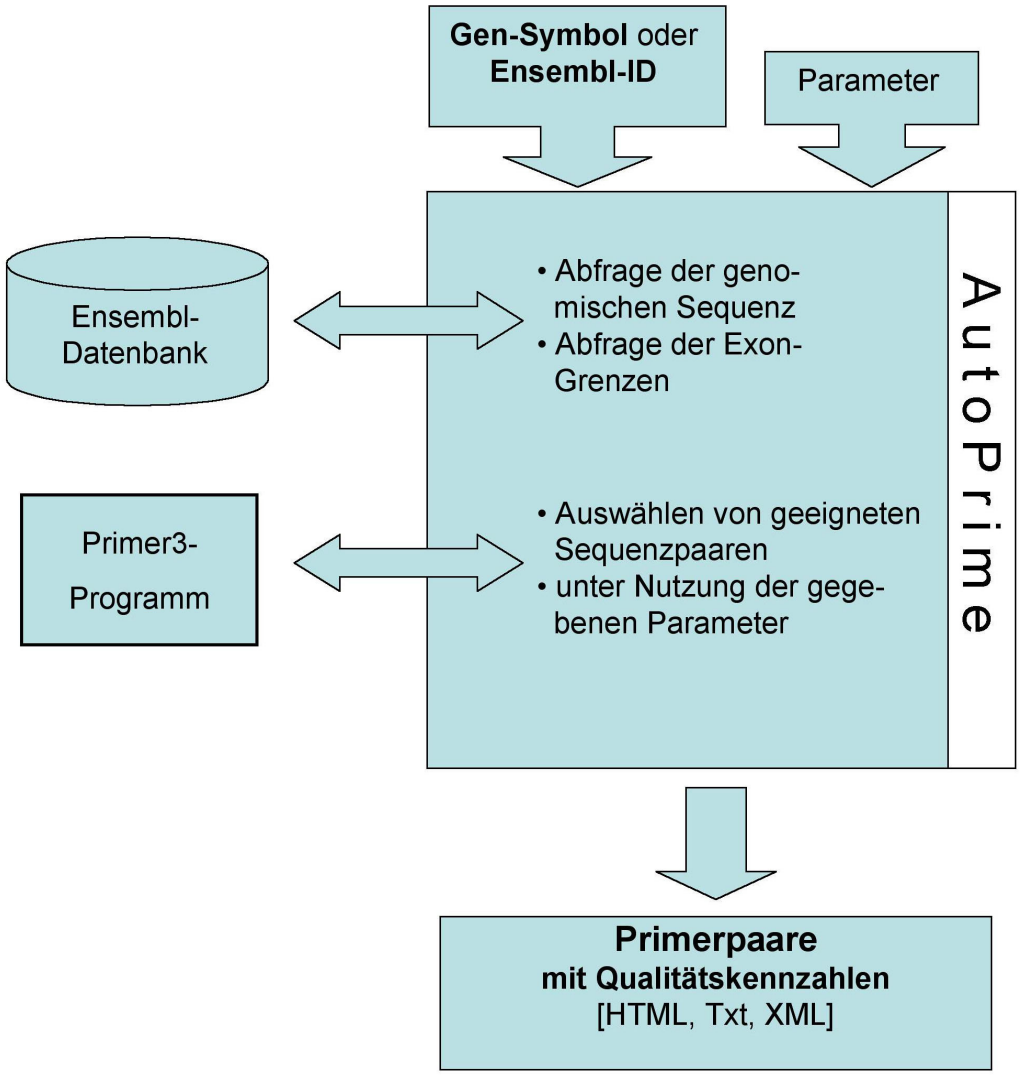



Abb. 13: Aufbau des *AutoPrime*-Programms

Die Angabe eines Gen-Symbols ist ausreichend, um mit den Standard-Parametern nach Primern zu suchen. Die Sequenzinformationen werden automatisch in der *Ensembl*-DB gesucht und an *Primer3* weitergeleitet.

Der Ablauf des Programms umfasst die folgenden Schritte (Abbildung 13):

- Für das gewählte Gen werden die genomische Sequenz und Informationen über die Exon/Intron-Grenzen von *Ensembl* bezogen.
- Die vom Benutzer gestellten Parameter und
- wenn gewünscht, die *Misspriming-Library* des jeweiligen Organismus werden genutzt.
- Das *Primer3*-Programm wird über einen Systembefehl gestartet.

- *Primer3* überprüft alle möglichen Sequenzabschnitte und liefert valide Primer-Paare mit deren Qualitätsmerkmalen und Sequenzen an das Hauptprogramm zurück.
- Ausgegeben werden außer den positiven Ergebnissen auch die Anzahl der Primer, welche aufgrund der gewählten Parameter ausgefiltert wurden, sowie die exprimierte und die genomische Sequenzen.

Das *AutoPrime*-Programm kann über die Systemzeile mit einem einzelnen Aufruf gestartet werden. Um die Bedienung zu vereinfachen, bzw. von entfernten Rechnern aus zu ermöglichen, wurde außerdem eine Internet-Oberfläche in der Sprache Perl programmiert (Abbildung 14). Die Benutzeranweisungen werden von dem Skript, welches die Suchmaske generiert, mit einem *fork*-Befehl an das *AutoPrime*-Hauptprogramm übergeben. Dies bedeutet, dass dessen Ausführung von der Maske und von weiterer Benutzerinteraktion abgekoppelt wird. Das Hauptprogramm wiederum schreibt seinen Verarbeitungsfortschritt in eine *Log-Datei*. Indem das Internet-Skript diese Log-Datei in regelmäßigen Abständen abfragt, können so die Schritte des Programms überwacht werden (Verbindung mit der *Ensembl*-Datenbank, Abrufen der Sequenz, Überprüfung der einzelnen Exons und Generierung der Primersequenzen).

Die Ergebnisse werden in Form von maschinenlesbarem XML-Code abgelegt. Zur Darstellung im Internet wurde ein XML-Parserskript geschrieben, welches daraus für den Menschen mit Internet-Browserprogrammen leichter lesbaren HTML-Code generiert. Ein zweites Parser-Skript kann einfachen Text in ASCII-Zeichen erzeugen.

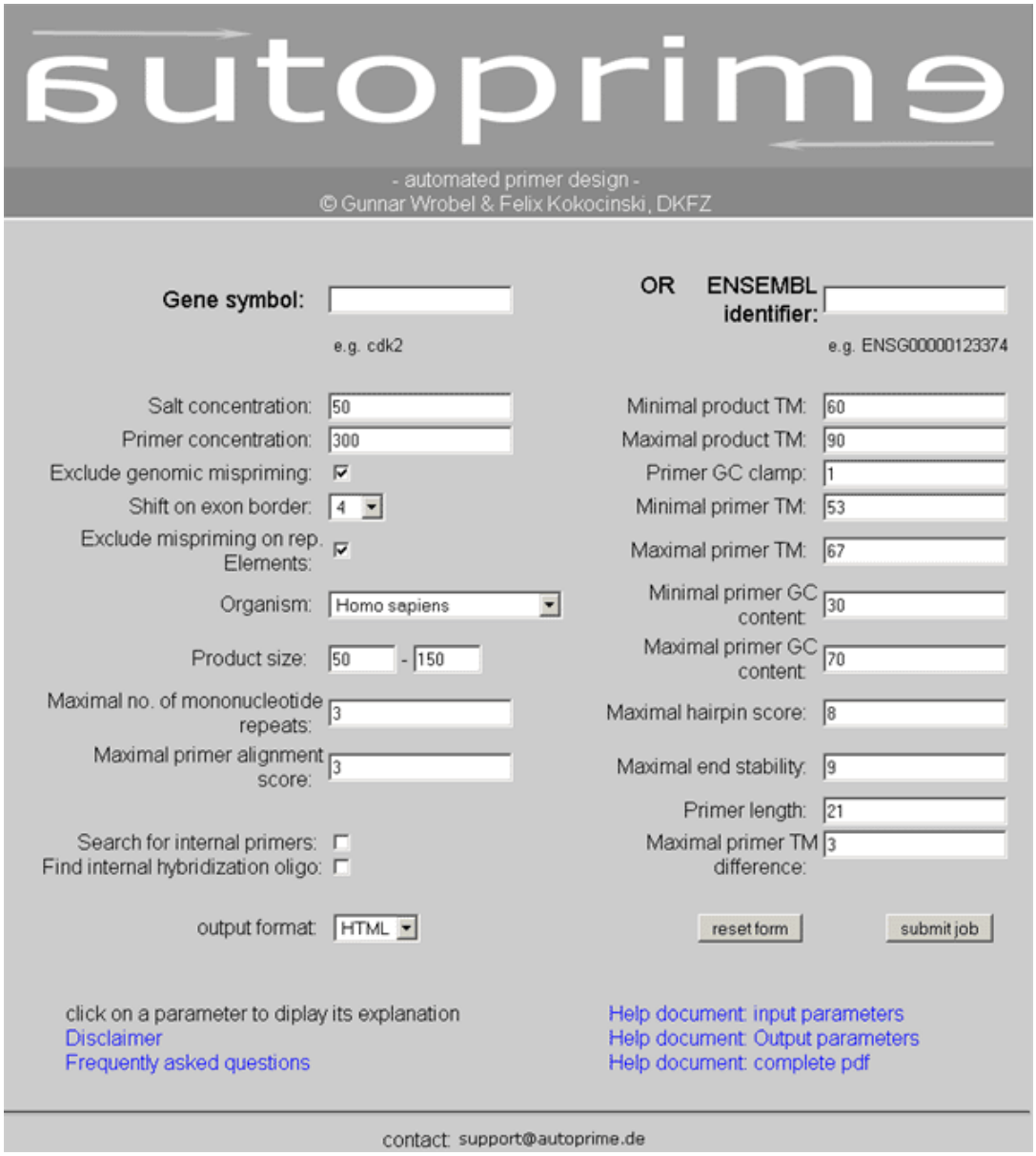



Abb. 13: Internet-Oberfläche von *AutoPrime*

Für alle Parameter sind Standardwerte vorhanden, der Benutzer muss lediglich das gewünschte Gen benennen und die Suche starten. Von der Maske aus sind außerdem Texte mit weiteren Erklärungen erreichbar. Das Programm kann unter der Adresse http://www.AutoPrime.de benutzt werden.

# 4.4. Funktionelle Analyse von Experimenten (*FACT*)

## 4.4.1. Grundsätzlicher Aufbau

Die Anforderungen an das *Flexible Annotation and Correlation Tool* sind die Interpretation von experimentellen Ergebnissen mit Hilfe von Annotationsdaten und die Zusammenführung von relevanten Informationen aus heterogenen Datenquellen. *FACT* kann mit den unterschiedlichsten Datenquellen arbeiten, d.h. verschiedenartige experimentelle Ergebnisse können eingelesen und eine Vielzahl von Annotationsquellen genutzt werden (Abb. 15). Desgleichen können unterschiedliche Analysemethoden angewendet werden. Diese Flexibilität wird durch den modularen Aufbau des Systems ermöglicht: Jede Daten- oder Analyse-Quelle wird durch ein eigenes Software-Modul (Adapter, *data source adaptors*) verwaltet, welches auf die jeweiligen Spezifikationen zugeschnitten ist. Es führt eine Transformation der Informationen in das gemeinsame Schema durch und nutzt dann generelle Funktionen zum Abspeichern und Verwalten der Daten. Diese generellen Funktionen sind als „Software-Bibliothek“ (API) zusammengefasst und bilden das Hauptprogramm von *FACT* (Abb. 16).

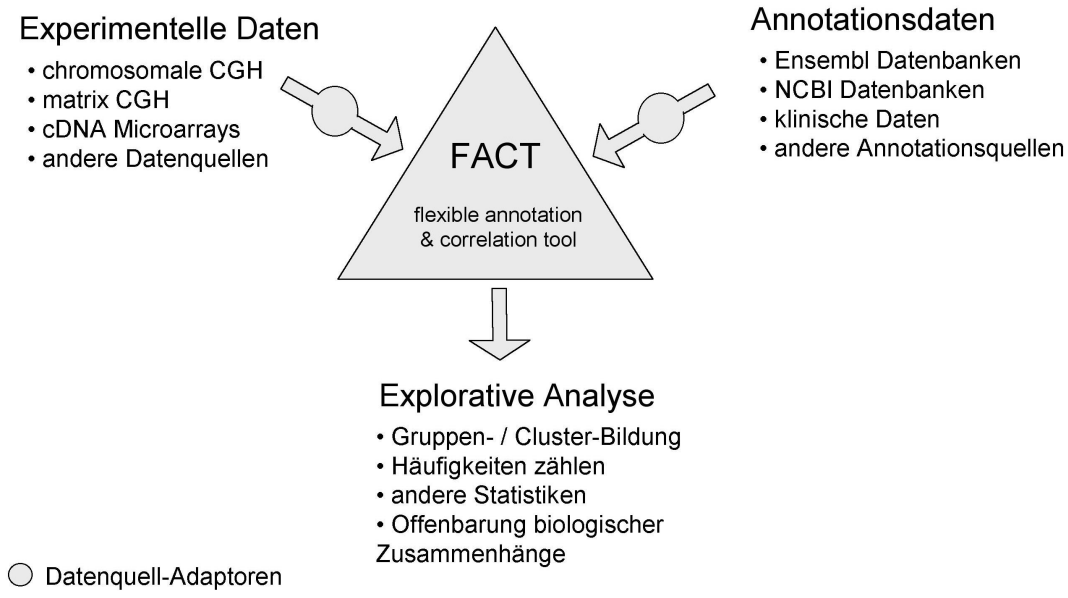



Abb. 15: Verwendung von heterogenen Datenquellen mit *FACT*

Durch spezifische Module können verschiedenste Arten von Werten eingelesen werden (Experimentelle Daten), unterschiedlichste Quellen zur Annotation (Annotationsdaten) und zur Analyse, bzw. Visualisierung (Explorative Analysefunktionen) herangezogen werden. Sie sind „Datenquell-Adaptoren“, welche die Transformation vom spezifischen zum abstrahierten Datenlayout durchführen und in *FACT* einspeisen.

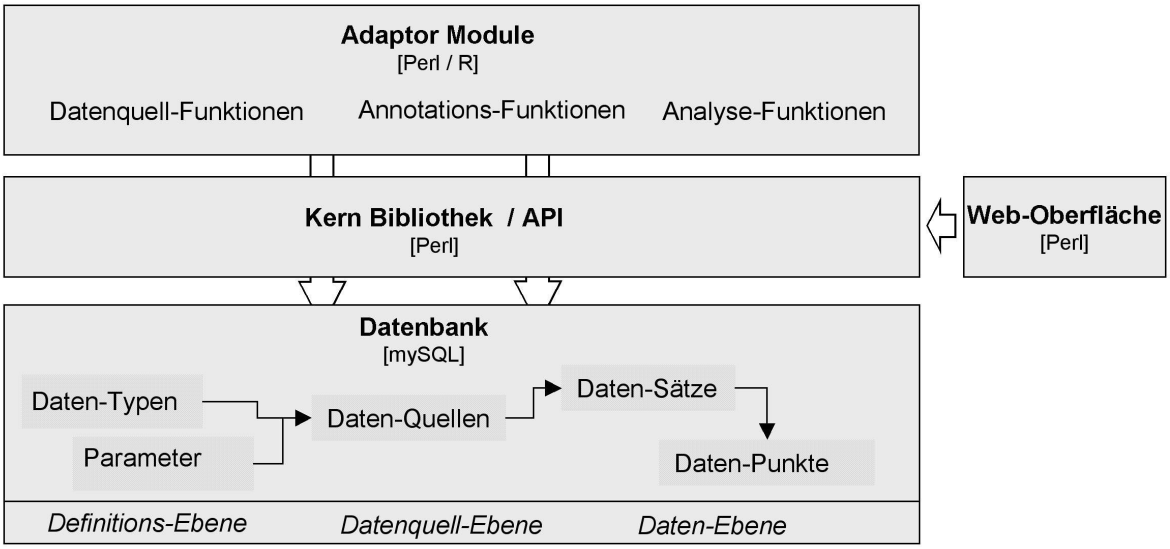



Abb. 15: Aufbau des *FACT*-Systems

Der modulare Aufbau und die Trennung in Datenbank-, Kernbibliothek- und Adaptor-Module erlaubt ein hohes Maß an Flexibilität. Es besteht außerdem eine Trennung von Programmfunktionen und GUI (*Graphical User Interface*: Web-Oberfläche).

## 4.4.2. Die Datenbank-Basis

Die Datenbasis des *FACT*-Systems ist eine mySQL-Datenbank, welche einerseits die Daten (experimentelle Werte und Annotationen) und andererseits Meta-Informationen (d.h. „Informationen über Informationen“) darüber speichert. Die Werte werden hierbei zu einem hohen Maß abstrahiert, um sie in einem gemeinsamen Schema speichern und nutzen zu können. Dies bedeutet, dass das System grundsätzlich nicht zwischen verschiedenen Arten von Datensätzen unterscheidet. Die verschiedenen experimentellen und Annotations-Daten werden in gleicher Art abgelegt. Die Meta-Daten über die einzelnen Quellen und Typen ermöglichen wiederum eine definierte Zuordnung. Das Datenbank-Schema (Abb. 17.a) zeigt die (inhaltliche) Unterscheidung von drei Ebenen in der Datenbank. In der Definitions-Ebene (*Data Definition Layer*) werden vorhandene Datentypen definiert, es existiert zum Beispiel der Datentyp „Gen-Name“, welcher eine Identifikationsnummer hat, und aus alphanumerischen Zeichen besteht. Hier werden außerdem die drei Modultypen als Experimentelle Datenquelle, Annotations-Datenquelle oder Analyse-Modul genannt. Auf der nächsten Ebene werden die unterschiedlichen Datenquellen beschrieben und identifiziert (*Data Source Layer*). Zusätzlich werden dazu mögliche Parameter und

Datentypen gespeichert. Ein Beispiel ist die Datenquelle „*Ensembl*“, welche mit ID, dem Namen des Skriptes, der für spezifische Datenbankabfrage bei *ensembldb.ensembl.org* zuständig ist, eine Beschreibung und dem letzten Aktualisierungsdatum abgelegt ist. Hinzu kommen hier die Informationen, dass ein Gensymbol oder eine Accession-Nummer als Datentypen an die Funktion übergeben werden sollen (*SourceDataType*). Von der Funktion zurückgegeben werden dagegen der vollständige Genname, die Lokalisierung in Chromosom und MB-Positionen, SwissProt-IDs, InterPro-IDs und anderes (Tabelle 7). Die Unterscheidung in Eingangs- und Ausgangs-Datentypen wird durch das Flag „Relevance“ gesetzt. Als Parameter kann hier z.B. „*homo sapiens*“ als gesuchter Organismus genannt werden. Die Datenebene (*DataSet Layer*) speichert die eigentlichen Daten als Datenpunkte (*DataFeatures*), welche einzelne Informationseinheiten als Name/Wert-Paar (experimentelle Daten) oder als Beschreibung derselbigen sind. Alle Datenpunkte eines Experimentes oder eines Annotationschrittes für ein Experiment werden zu Datensätzen (*DateSets*) zusammengefasst. So wird z.B. ein Microarray-Experiment als *Dataset* der Quelle „Expressions-Microarray Genliste“ mit allen Messpunkten (z.B. Gen-Name und Hybridisierungs-Ratio) als *Datafeatures* abgespeichert. Um den Zusammenhang zu verdeutlichen zeigt Abbildung 17.b eine Beispiel-Belegung der Datenbank-Werte. Zu der Hauptdatenbank *FACT* gehört eine Hilfs-Datenbank *FACT-Modules*, welche Informationen, die die einzelnen Annotations-Module nutzen, speichern kann. Jede Funktion kann in einer eigenen Tabelle Daten aus entfernten Quellen lokal nachbilden.

Abb. 17: Datenbank-Schema von *FACT* (nächste Seite)

a. Layout des Schemas

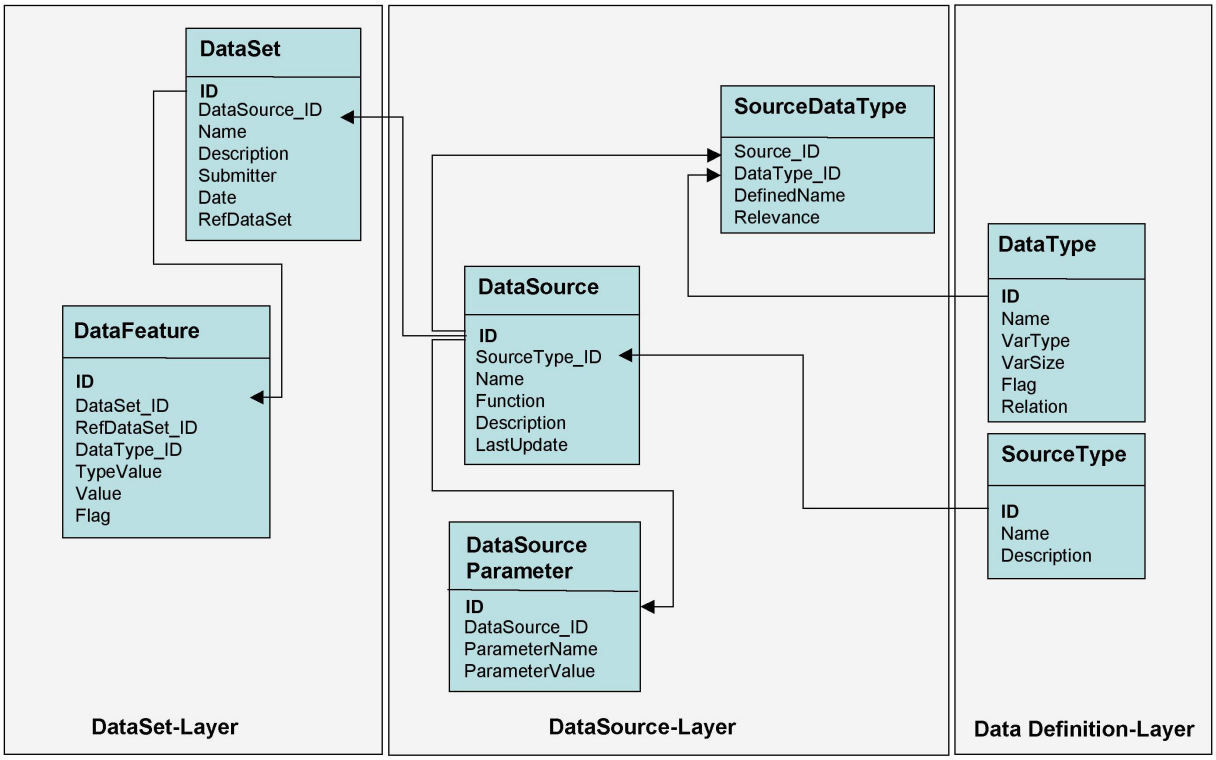
Während die eigentlichen Daten abstrahiert als *DataSets* mit *DataFeatures* in der Daten-Ebene gespeichert werden, sind in den Daten-Quell- und Daten-Definitions-Ebenen Meta-Informationen über Herkunft und Art dieser Daten abgelegt.

b. Beispiel-Daten im Datenbank-Schema

Als Beispiel wurde ein Datentyp „Gen-Symbol“ definiert, der von der Datenquelle „Ensembl“ genutzt wird. „Ensembl“ von Datenquellen-Typ 2 (Annotationen), benutzt das Modul *Ensembl_Parser*, erwartet als Parameter den gesuchten Organismus und bezeichnet intern den Datentyp als „Genesymbol“. Es wurde von einem Benutzer die Datenquelle benutzt, um einen eigenen Datensatz (ID 22) mit

Annotationen zu versehen. Dazu wurde der Datensatz 23 erstellt, der unter anderen das Daten-Feature Nr. 499 beinhaltet, welches von Typ 2 (Gen-Symbol) ist und CDK5 beinhaltet.

a.




b.

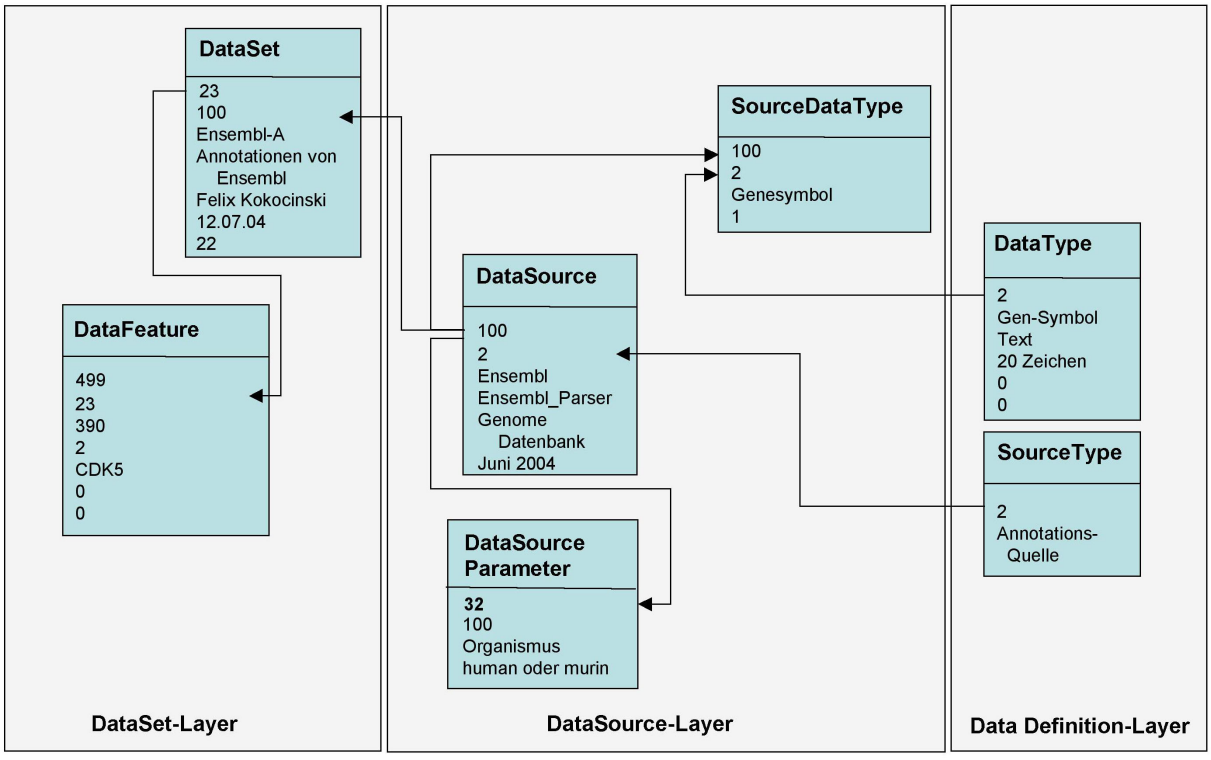



Abb. 16: Legende vorhergehende Seite

### 4.4.3. Flexible Integration heterogener Datenquellen

Die für *FACT* gewählte und oben beschriebene Struktur resultiert in einem extrem modularen Programm, das jederzeit durch neue Funktionen erweiterbar ist. Neue Datentypen und –quellen können durch das Einbinden eigener „Datenparser“ erschlossen werden. Dazu wird der, entsprechend einem vordefinierten Prototyp geschriebene, spezielle Parser im Programm angemeldet und kann dann direkt aufgerufen werden. Es wird dabei als *DataSource* mit seinen eigenen Daten-Typen und Parametern in der Datenbank gespeichert und die Funktion wird in das entsprechende Verzeichnis kopiert. Das dynamische Laden der Module erfolgt in einer iterativen Initialisierung sämtlicher gefundener Funktionen. Ist eines der Module fehlerhaft, wird es nicht geladen und eine Fehlermeldung protokolliert. Dadurch kann ein System-ausfall vermieden werden.
Annotationsfunktionen können natürlich neben den experimentellen Anfangswerten auch Annotationsdaten von diesen Werten als Ausgangspunkte benutzten, wodurch geschachtelte Annotationen entstehen (Abb. 18). In der Datenbank werden zu allen Datensätzen die jeweiligen Referenz-Datensätze vermerkt und zu jedem Datenpunkt wird der ursprünglichste Referenz-Punkt gespeichert.

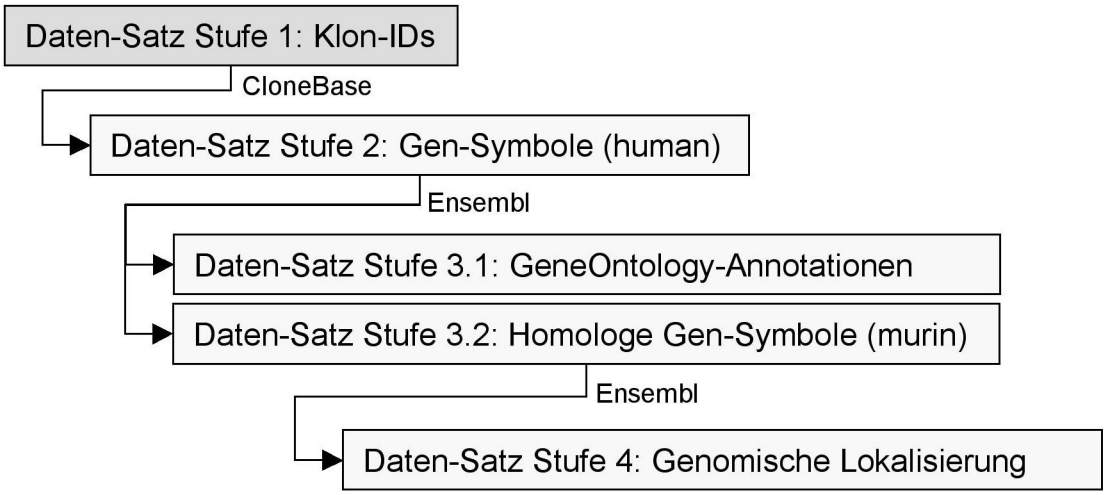



Abb. 18: Beispiel für Annotations-Stufen

Der ursprüngliche Datensatz (Stufe 1, experimentelle Ergebnisse) wird annotiert. Der daraus resultierende Datensatz (Stufe 2, Annotationsdaten) wird mit neuen Quellen annotiert, usw.

Als Annotationsquellen kommen sowohl lokale Datenbanken (*Data-Warehouse* Konzept), als auch Datenbanken auf entfernten Rechnern (Datenbank-Föderations-Konzept) in Frage, außerdem können Text-, XML- und andere Dateien genutzt werden. Die zur Verfügung stehenden Annotations-Module werden im Folgenden erläutert (Tabelle 7).

| **Datenquelle, Daten-Zugangsmethode** | **Datenherkunft** | **Art der Annotation** |
|---|---|---|
| *Ensembl*, Perl API-Zugang zu lokaler oder entfernter DB | European Bioinformatics Institute and Wellcome Trust (GB) | Ensembl ID, Gen-Symbol, Gen-Name, chromosomale Lokalisierung, homologe Gene, Interpro Domänen, RefSeq Accession Nummer, Affymetrix ID |
| *Image Consortium*, Datei als DB | Lawrence Livermore National Laboratory | Klon Image ID, Accession-Nummer |
| *Mouse Genome Database*, Datei als DB | Jackson Laboratory (USA) | MGI ID, Gen-Symbol |
| *Biological Biochemical Image Database*, HTTP-Parser | National Institute of Aging, NIH (USA) | Stoffwechselweg-Name und Bild-Verweis |
| *GeneOntology*, lokale DB | GeneOntology Konsortium | ID und Name des GO-Terms (Biologischer Prozess, Molekulare Funktion, Zelluläre Lokalisierung) |
| *Cancer Genome Anatomy Project*, Datei als DB | National Cancer Institute, NIH (USA) | Biocarta Name, Biocarta Kurzname, KEGG Name, KEGG ID, PFAM ID |
| *LocusLink*, Datei als DB | National Institute of Health (USA) | A. LocusLink ID, Gen-Symbol, Gen-Name, Genomische Lokalisierung, GO-Term, OMIM ID<br>B. Haupt-Literaturreferenz (PubMed Verweise) |
| *euGenes*, Datei als DB | University of Indiana (USA) | euGene ID, Gen-Symbol, Gen-Name, GDB ID, OMIM ID, Genomische Lokalisierung, GO-Term, Protein Accession Nummer |
| Interne *CloneBase*, direkter DB-Zugang | Deutsches Krebsforschungszentrum, Abt. Molekulare Genetik (D) | Allgemeine Informationen über Klone |

| | | |
|---|---|---|
| *CpG*, Datei als DB | National Institute of Health (USA) | Berechneter relativer *CpG*-Gehalt einer genomischen Region |
| *STRING*, Datei als DB | EMBL (D) | Protein-Interaktions-Daten |

Tab. 7: Zur Verfügung stehende Datenquellen für die Annotation in *FACT*

Als Hauptannotations-Quelle wurde die *Ensembl*-Datenbank gewählt. Die Abfrage von Daten erfolgt mittels der Perl-API vom Ensembl-Projekt (Stabenau *et al.,* 2004) direkt von *Ensembl*-Datenbankserver (ensembldb.ensembl.org). Unter Nutzung von Gen-Symbolen oder Accession-Nummern können in einem Modul umfangreiche Annotationen parallel abgefragt werden (*Ensembl-Basic*-Modul). Dies umfasst offizielles Gen-Symbol, Gen-Name, genomische Lokalisierung, OMIM-ID, SwissProt-ID und InterPro-Proteindomänen.

Liegen Kloninformationen als Image-IDs vor, so kann ein Modul entsprechende Accession-Nummern der NCBI-Datenbanken ausgeben. Die entsprechenden Informationen werden von der Internetseite des IMAGE-Konsortiums (Lennon *et al.*, 1996) als Datei geladen, entschlüsselt (*geparst*) und auf dem Server-Computer (lokal) in eine eigene Tabelle der FACT-Modules Datenbank gespeichert. Diese Methode wird von verschiedenen Modulen zur Daten-Zwischenspeicherung genutzt.

Auf ähnliche Weise könne Informationen von der *Mouse Genome Database* über murine Gene erhalten werden (Blake *et al.*, 2003). Hier liegen die Informationen als MGI-IDs vor.

Die *Biological Biochemical Image Database* (Becker *et al.*, 2000) wird genutzt, um für ein Gen oder Protein Informationen über die Zugehörigkeit zu einem bestimmten biologischen Stoffwechselweg zu erhalten. Zusätzlich wird ein Verweis zu einer graphischen Darstellung des Stoffwechselweges ausgegeben. Die Technik, die hierbei verwendet wird, bezeichnet man als *Screen-Grabbing*: Da die Daten nicht zum direkten Download geeignet sind, werden für jedes Gen oder Protein Anfragen über einen virtuellen Web-Browser generiert. Die Antwort des entfernten Webservers wird jedoch nicht dargestellt, sondern direkt nach Schlüsselwörtern gefiltert, und die gewünschten Informationen abgespeichert.

Das GeneOntology-Projekt (GeneOntology Consortium 2001) bietet seine Daten dagegen als Datenbank-Auszug an, der auf dem lokalen System direkt nachgebildet werden kann. Für tausende von Genprodukten sind hier Informationen in den drei Kategorien *Biologische Funktion*, *Zellulärer Prozess* und *Zelluläre Lokalisierung* gespeichert.

Das *Cancer Genome Anatomy Project* (Strausberg *et al.*, 2000) stellt eine Sammlung von Daten zur Verfügung, welche unter anderem Informationen zu Stoffwechselwegen in Form von *Biocarta* Name, *Biocarta* Kurzname, *KEGG* Name, *KEGG* ID, *PFAM* ID bietet.

Aus der LocusLink-Datenbank des NCBI (Pruitt *et al.*, 2000) können mit einem Modul die wichtigsten Literatur-Referenzen zu den einzelnen Genen gesucht werden.

*euGenes* (Gilbert, 2002) ist eine klassische Meta-Datenbank. Sie stellt Informationen aus anderen (Primär-) Datenbanken zusammen und ist dabei auf Eukaryonten fokussiert. Aus ihr kann eine alternative Annotation von Genen mit den Grundinformationen (Gen-Name, Gen-Symbol, Lokalisierung, usw.) erfolgen.

Die zuvor beschriebene Klon-Datenbank *CloneBase* wird in einem spezialisierten Modul abgefragt. Sie stellt das Bindeglied von den internen Klon-Bezeichnungen und der vollständigen Annotation mit öffentlichen Daten dar.

Vom FTP-Server des NCBI sind eine Reihe von Datensammlungen erhältlich. Ein Modul nutzt Teile davon, um den Gehalt von *CpG-Inseln* einer gegebenen Sequenz, bzw. eines genomischen Bereiches im Vergleich zum Gesamtgenom zu berechnen. Das Modul entstand in Zusammenarbeit mit Nicolas Delhomme.

In der *STRING*-Datenbank am EMBL (Europeen Molecular Biology Laboratory, Heidelberg) sind Daten über potentielle Protein- und Nukleinsäure-Interaktionen berechnet worden (von Mering *et al.*, 2003). Diese können ebenfalls von einem *FACT*-Modul als Annotation genutzt werden.

### 4.4.4. Analyse mit Annotationsdaten

Je nach vorliegendem Datentyp (Gen-Symbol, Klon-ID, usw.) stehen unterschiedliche Analysefunktionen zur Verfügung. Die zurzeit in *FACT* eingebundenen Module werden im Folgenden beschrieben (Tabelle 8).

Das Modul *SimpleCount* kann auf alle Daten-Typen angewendet werden. Es zählt das Vorkommen der einzelnen Annotations-Begriffe und zeigt die Häufigkeiten als Balken- Diagramm in der Übersicht über sämtliche Daten, als Kuchen-Diagramm für jeden Daten-Typ separiert und als Tabelle an. Es können mehrere Datensätze

vereinigt werden und es kann ein Schwellwert für die Anzeige definiert werden (Abb. 19.a).

| Methoden-Name | Referenz | Methoden-Beschreibung |
|---|---|---|
| *Simple Count* | *FACT* | Zählen und Darstellen von Häufigkeiten der Annotationsbegriffe |
| *Hypergeometric Tail* | In Teilen von *GeneMerge* (Castillo-Davis *et al.*, 2003) | Detektion von signifikant überrepräsentierten Begriffen aller Datentypen (Hypergeometrische Verteilungsfunktion) |
| *GO-Term Comparison* | In Teilen von *GO::TermFinder* (G. Sherlock, Stanford und E. Boyle, MIT) | Detektion von signifikant überrepräsentierten GO-Termen (Hypergeometrische Verteilungsfunktion) |
| *K-Means - EASE* | *EASE* (Hosack *et al.*, 2003) und Wrobel *et al.*, unveröffentlicht | Detektion von signifikant überrepräsentierten GO-Begriffen in einer Genliste (Fishers Exakt Test) in *K-Means*-Klustern |
| *MedLiner* | *Bio::Biblio* (M. Senger, EBI) | Finden von Publikationen mit mehrfach auftretenden Begriffen |
| *Chromosomal Plot* | *FACT* | Darstellung von Werten oder Häufigkeiten im genomischen Kontext |
| *CGH – Expression Comparison* | *FACT* | Detektion von direkten Korrelationen zwischen genomischen und Expressions-Datensätzen (2-seitige T-Tests) |
| *CGH database* | Deutsches Krebsforschungszentrum, Abt. Molekulare Genetik (D) | Vergleich von CGH Daten zu archivierten Datensätzen |

Tab. 8: Analysefunktionen, die in *FACT* genutzt werden können

Das *Hypergeometric Tail*-Modul untersucht, ob in einer Annotationsliste bestimmte Begriffe im Vergleich zu einer Hintergrundsliste signifikant überrepräsentiert vorkommen und damit Hinweise über die biologische Bedeutung der Gruppe zulassen. Die Berechnung erfolgt auf Grundlage der hypergeometrischen

Verteilungsfunktion, welche die Wahrscheinlichkeit dafür angibt, dass zwei unabhängige Teilmengen genau X Elemente gemeinsam besitzen.

Als Wahrscheinlichkeitsfunktion gilt dazu:

$$P(m) = \frac{(M,m)(N-m,n-m)}{\binom{N}{n}}$$

Mit der Grundgesamtheit N und dem Stichprobenumfang n, ist die Wahrscheinlichkeit P, genau m Elemente mit der Ausprägung M zu erhalten. Für die Ausprägungen M oder „nicht M“, lässt sich für jeden Annotationsbegriff eine relative Wahrscheinlichkeit im Vergleich zu der Hintergrundliste berechnen.

Das *GO-Term*-Modul annotiert Gen-Symbole, bzw. SwissProt-IDs mit GeneOntology-Begriffen und sucht damit Kategorien, die im Vergleich zu einer Hintergrundliste überrepräsentiert vorkommen heraus. Es nutzt dabei die hypergeometrische Wahrscheinlichkeitsfunktion. Wird kein Hintergrund definiert, wird das komplette Genom als Vergleich herangezogen. Das Modul nutzt die Funktionalität von *GO::TermFinder* (G. Sherlock, Stanford and E. Boyle, MIT), welches frei über CPAN (Öffentliches Archiv von Perl-Modulen, http://cpan.org) erhältlich ist. Ein Beispiel der Ausgabe vom *Go-Term*-Modul zeigt Abbildung 19.b.

Das Modul *K-Means–EASE* ist die Kombination von zwei Methoden und wurden in Zusammenarbeit mit Dr. Gunnar Wrobel in der Sprache R implementiert. In einem ersten Schritt wird der K-Means-Clustering Algorithmus verwendet, um innerhalb einer Name/Wert-Liste Untergruppen zu identifizieren. Im zweiten Schritt wird mittels des Fishers Exakt-Tests berechnet, ob in diesen Gruppen bestimmte Annotationsbegriffe überrepräsentiert sind.

$$P = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{n!} \sum_i \frac{1}{a_i!b_i!c_i!d_i!}$$

Ausgehend von Gen-Kurznamen kann das Modul *MedLiner* Puplikationen in der öffentlichen Datenbank für medizinisch-wissenschaftliche Literatur *PubMed* (NCBI, USA) heraussuchen, welche mit zwei oder mehr Genen in der Suchliste assoziiert sind. Es werden also Berichte gefunden, die beide (oder mehr) Gene in Zusammenhang bringen. Es greift dabei auf Funktionen des Skriptes *Bio::Biblio* (M. Senger, EBI) zurück. Die Ergebnisse werden mit Autoren, Titeln und Hyperlink zur PubMed-Internetseite ausgegeben.

*Chromosomal Plot* produziert für eine Liste mit Lokalisierungsdaten eine Darstellung im genomischen Kontext. Es stützt sich auf Banden-Informationen und einer statischen Graphik-Vorlage des *Ensembl*-Projektes. Es können Häufigkeiten von einzelnenen Banden gezählt und als Balken-Diagramm dargestellt werden. Die experimentellen Werte können direkt als Kurven-Diagramm oder als positive und negative Doppelbalken gezeichnet werden. Letzteres ist für die Dokumentation von Verlusten und Gewinnen von genomischem Material als Ergebnis von CGH oder matrixCGH anwendbar (Abb 19.c).

Der Vergleich von genomischen und Expressions-Informationen kann mit dem Modul *CGH–Expression-Comparison* durchgeführt werden, welches in Zusammenarbeit mit Dr. Gunnar Wrobel entstanden ist. Es teilt die Daten von den jeweils korrespondierenden genomischen und Expressions-Datensätzen in die Gruppen Amplifiziert / Überexprimiert und Deletiert / Unterexprimiert ein. Mit Hilfe von zweiseitigen T-Tests wird dann die Wahrscheinlichkeit berechnet, dass die Gruppen einer gemeinsamen Grundgesamtheit entstammen. Hiermit wird gezeigt, ob es eine signifikante, d.h. nicht zufällige Korrelation zwischen den Paaren gibt. Die Ausgabe der Informationen erfolgt als Datentabelle mit den berechneten Wahrscheinlichkeiten und als Diagramme, die signifikante -Abweichungen direkt erkennen lassen.

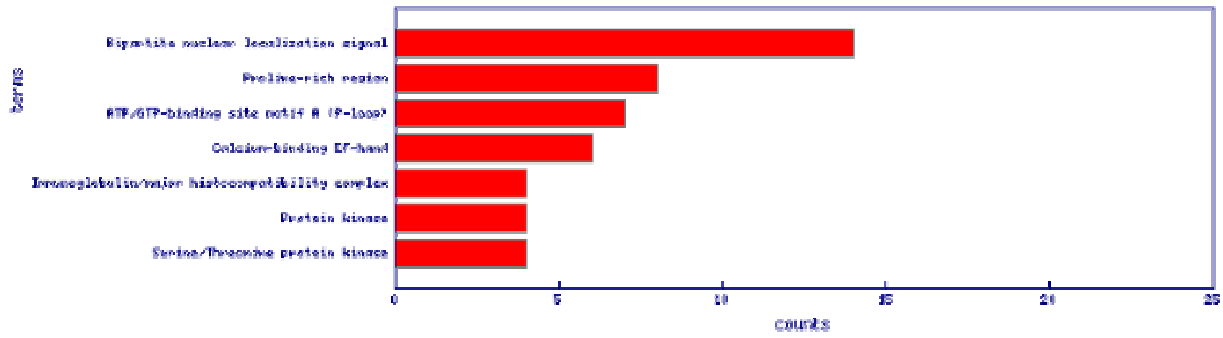
Die *CGH-DB*-Funktion greift auf Daten zu, die in der CGH-Datenbank der Abteilung gespeichert sind (Berrar *et al.*, 2001) und vergleicht sie mit vorliegenden Ergebnissen.

### 4.4.5. Ausgabe von Ergebnissen

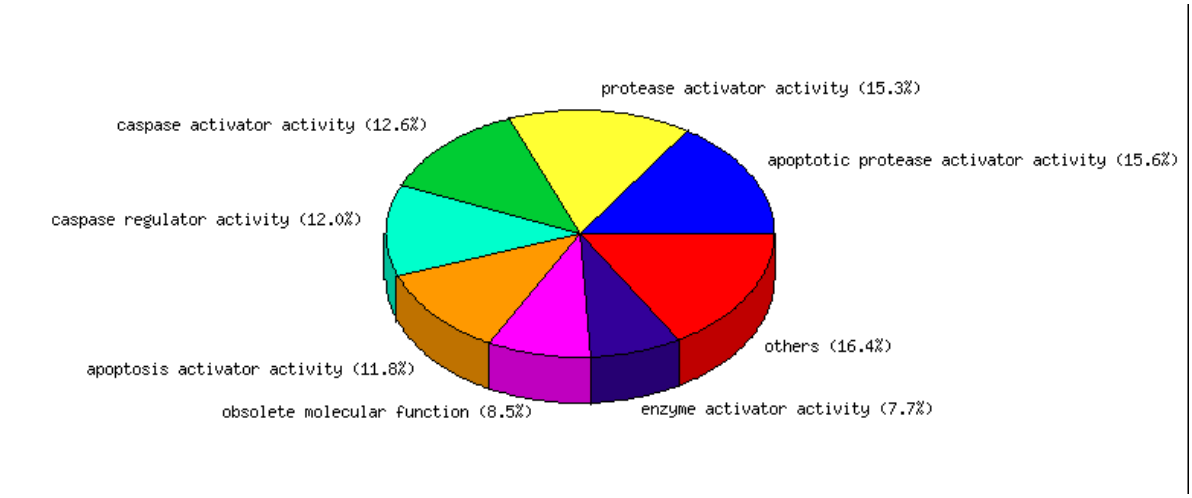
Die ursprünglichen Datensätze (z.B. Klone mit Ratio-Werten) können mit sämtlichen gefundenen Annotationdaten – auch bei geschachtelten Annotationen – in unterschiedlichen Formaten ausgegeben werden. Für die Darstellung im Internet-Browser kann HTML-Code generiert werden, für maschinelle Weiterverarbeitung kann XML oder auch unformatierter Text erstellt werden. Die Informationen können über die Web-Oberfläche auch per E-Mail an den Benutzer gesendet werden.

Ergebnisse von Analyse-Modulen liegen im Regelfall als HTML- und Graphik (gif) Datei vor (Abb. 19). Sie stehen 30 Tage lang zum Abruf für den Benutzer bereit und werden dann vom System automatisch gelöscht.

a.




b.




c.

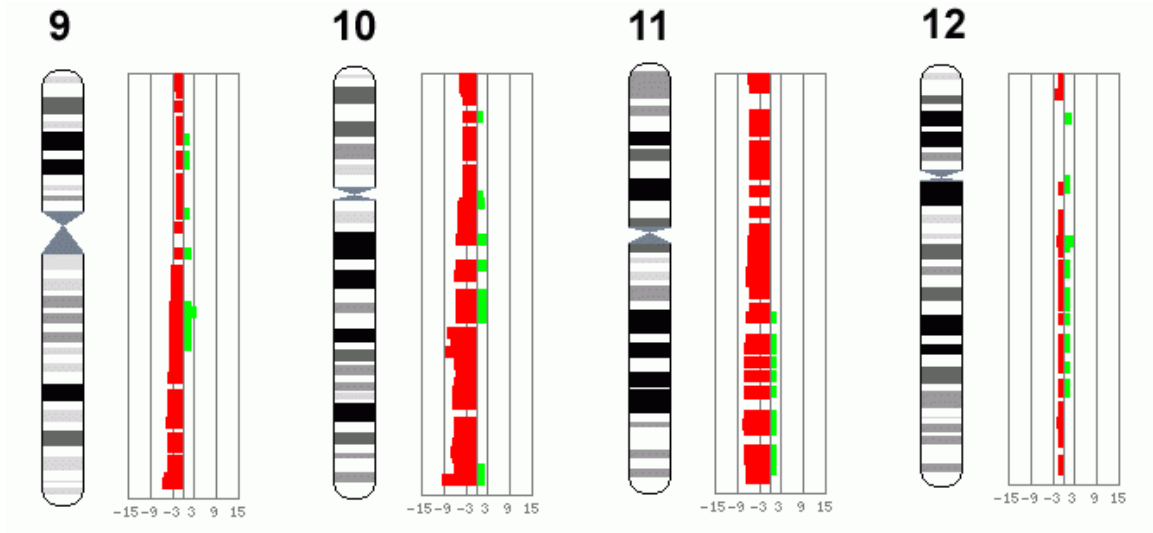



Abb19: Beispiele für von *FACT*-Modulen produzierte Ergebnis-Graphiken

a. Die Funktion *SimpleCount* produziert unter anderem eine Graphik, die häufig auftretende Annotationsbegriffe als Balkendiagramm darstellt, hier InterPro-Proteindomänen. b. *GO-Term-Comparison* zeigt hier die Verteilung der signifikantesten GO-Annotationen als Kuchendiagramm. c. Mit *Chromosomal Plot* können Anhäufungen in bestimmten genomischen Bereichen schnell visualisiert werden, hier Gewinne und Verluste genomischen Materials.

### 4.4.6. Internet-Oberfläche

Sämtliche Funktionalität des Hauptprogramms kann von einer Internet-Oberfläche, die in Perl programmiert wurde, aufgerufen werden. Es wurde ein Authentifizierungssystem integriert, welches ermöglicht, dass sich neue Benutzer registrieren und mit persönlichem Passwort anmelden. Die experimentellen und Annotations-Daten sind ausschließlich dem Eigner der Daten selbst sichtbar. Es gibt außer dem normalen Benutzer-Status einen „Super-Benutzer"-Status, welcher das dynamische Hinzufügen von neuen Daten-Typen und Funktionen erlaubt. Nach dem Anmelden erhält der Benutzer allgemeine Hinweise und Neuigkeiten zum System. Er kann aus einem Menü folgende Funktionen aufrufen (Abb. 20):

- Hochladen von neuen Datensätzen
- Löschen von bestehenden Daten
- Aufruf von Annotations-Funktionen
- Aufruf von Analyse-Funktionen
- Anzeigen von eigenen Daten als einzelne Datensätze oder als Überblick
- Anzeigen von eigenen Analyse-Ergebnissen

Auf der entsprechenden Internet-Seite können die einzelnen Module ausgewählt werden, welche dann dynamisch geladen werden und - entsprechend ihrer Definition aus der Datenbank - Parameter und Daten-Typen angezeigt werden. Ferner wird zu jedem der Module eine kurze Erklärungs-Seite geladen, die aus dem Programm-Code generiert wird. Der Aufruf der modularisierten Funktionen wird dann an das *FACT*-Hauptprogramm weitergeleitet. Dort wird der Prozess von der Benutzer-Oberfläche entkoppelt (*fork*) und ausgeführt.

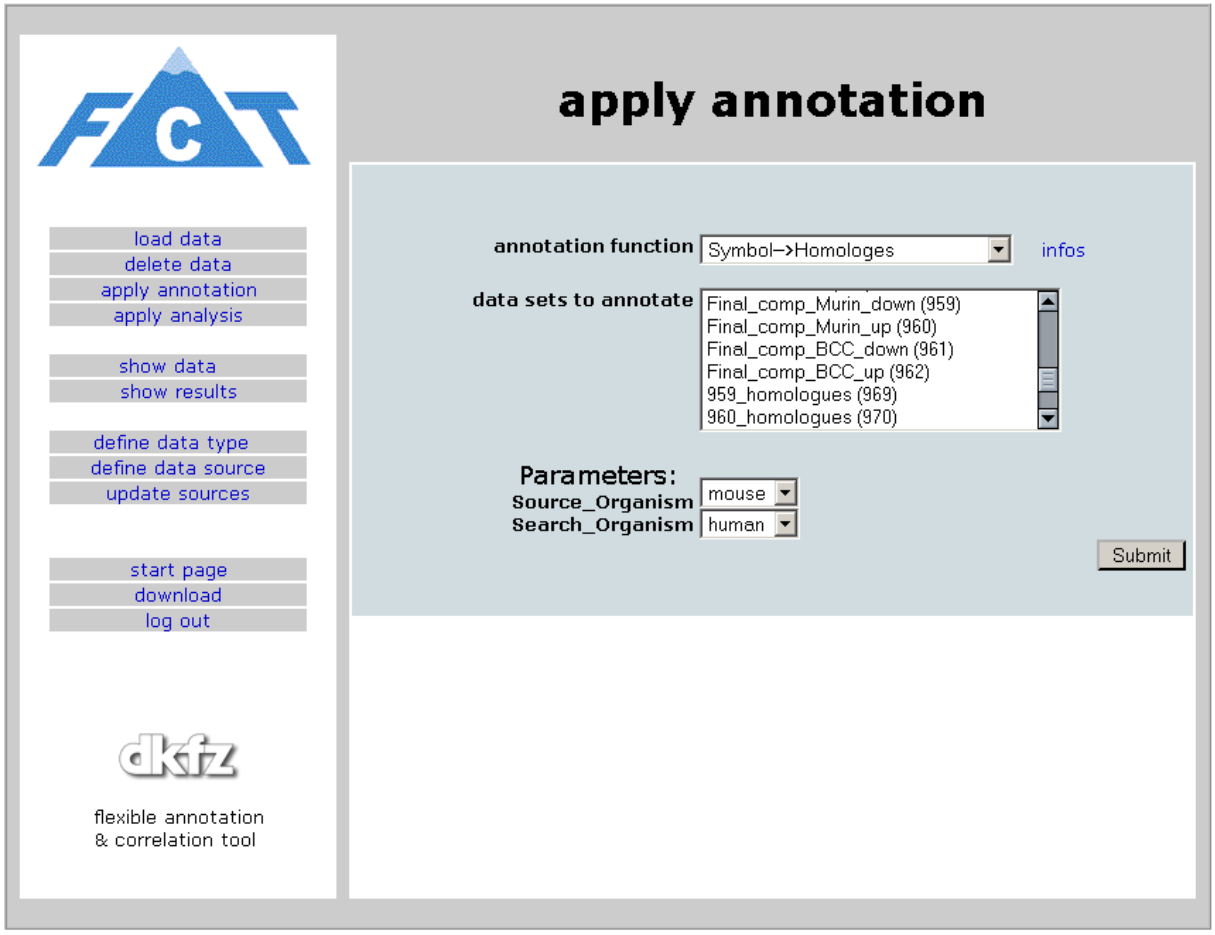



Abb. 20: Webinterface von *FACT*

Sämtliche Funktionen des Systems können über die Web-Oberfläche genutzt werden. Auf der linken Seite können Skripte zum i) Daten-Laden und –Löschen, zum Annotieren und Analysieren ii) Definieren von Daten-Typen und –Quellen iii) Anzeigen von Datensätzen, Ergebnissen und allgemeinen Beschreibungen aufgerufen werden. Auf der rechten Seite werden die gewählten Funktionen dargestellt.

# 4.5. Untersuchung der Entstehung und Progression von *Non-Melanom* Hautkrebs

## 4.5.1. Durchführung der Experimente

Die hier beschriebenen Experimente wurden in Zusammenarbeit mit Diplom-Ökotrophologen Lars Hummerich unter Einsatz des beschriebenen Systems zur Microarray-Produktion und -Analyse durchgeführt und sind zur Publikation eingereicht (Hummerich *et al.*, eingereicht).

Zur Untersuchung der Expressionsprofile wurden zwei unterschiedliche Microarrays mit murinen cDNA-Sequenzen produziert. Der erste Array beinhaltete sämtliche 20172 Fragmente der ArrayTAG™ Klonsammlung (LION Bioscience, Heidelberg), der zweite enthielt 15303 Fragmente der Sammlung vom National Institute of Aging (siehe Kapitel 4.1.1). Sämtliche Klone wurden in die Datenbank *CloneBase* aufgenommen und umfassend annotiert. Die einzelnen Fragmente wurden über PCR amplifiziert, anschließend mit Hilfe des *MiniTrak*-Roboters aufgereinigt und in Spotting-Puffer rückgelöst. Alle Prozessschritte wurden vom Labordatensystem *QuickLIMS* gesteuert und protokolliert. Sämtliche Mikrotiterplatten sind durch ihren Barcode eindeutig identifizierbar und die zugehörigen Prozessdaten sind in der Datenbank archiviert.

Die für die Microarray-Versuche benötigte mRNA wurde zu verschiedenen Zeitpunkten jeweils aus Gewebeproben der dorsalen Rückenhaut der Mäuse extrahiert. (I) TPA-induzierte Rückenhaut (6h), (II) Papillome (10 Wochen), (III) SCC (~50 Wochen). Die entsprechende Kontroll-Haut von Mäusen aus dem gleichen Wurf wird zeitgleich extrahiert. Das Umschreiben der mRNA in die entsprechende Menge cDNA, das Markieren mit Fluoreszenzfarbstoffen und die anschließende Hybridisierung wurde nach Wrobel *et al.* (2003) durchgeführt.

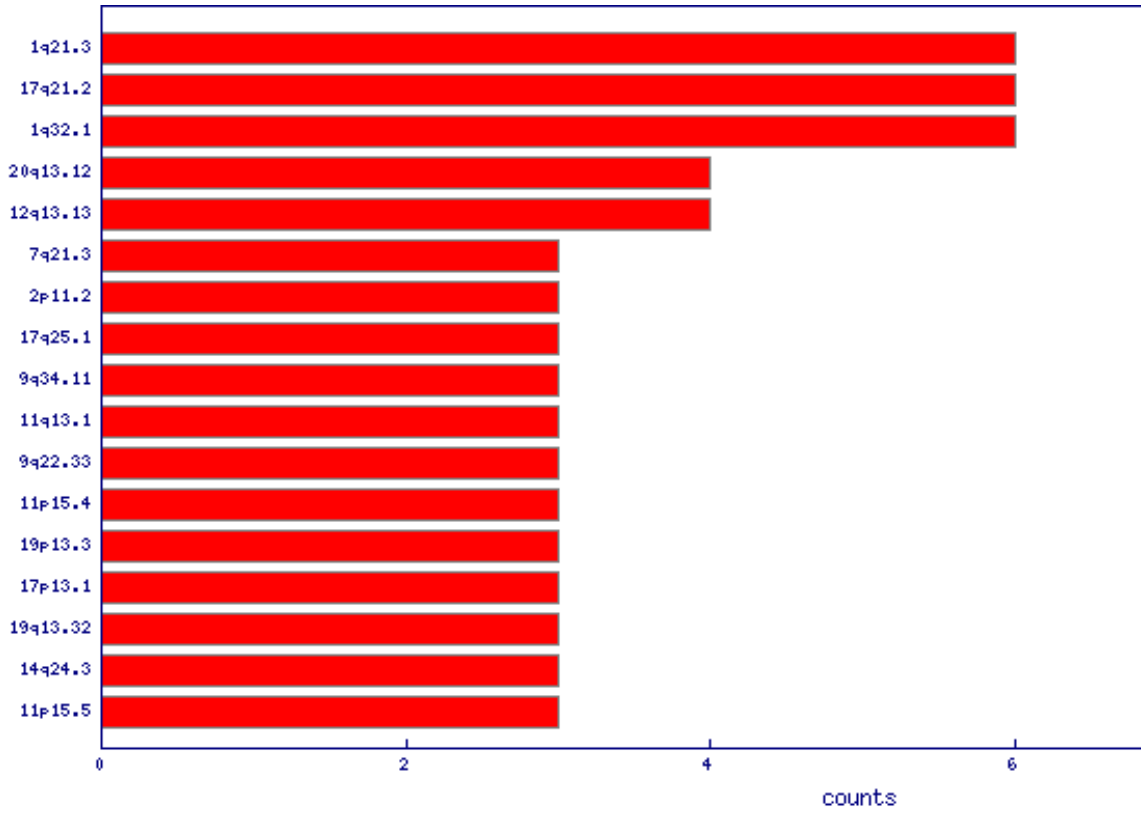
### 4.5.2. Datenanalyse

Die Signale der hybridisierten Microarrays wurden mit Hilfe eines Scanners (*GenePix 4000B*, Axon Instruments, USA) und der zugehörigen Software (*GenePix Pro 4.0*, Axon Instruments, USA) quantifiziert. Die Vorverarbeitung der Daten (Filterung, Normalisierung) erfolgte mit Hilfe von Skripten in der Sprache R (Wrobel, 2004; Hummerich *et al.*, eingereicht). Zur Validierung der Microarray Ergebnisse wurden einerseits *in situ*-Hybridisierungen mit murinen und humanen Tumor-Geweben durchgeführt. Andererseits wurden zusätzlich 35 Gene mittels semi-quantitativer RT-PCR oder mit quantitativer Real-Time PCR (RQ-PCR) überprüft. Die hierfür notwendigen Primersequenzen wurden mit Hilfe von *AutoPrime* automatisiert herausgesucht. Zur weiterführenden Analyse wurden die Programme *GeneSpring* (Silicon Genetics, USA), *EASE* (Hosack *et al.,* 2003) und *FACT* benutzt. Im Folgenden sind die mit *FACT* erzielten Ergebnisse beschrieben.

Aus den verrechneten Ergebnissen wurden Listen von Genen erstellt, welche im jeweiligen Stadium mindestens um 1,0 auf logarithmischer (ln) Skala verändert (über- oder unterexprimiert) vorlagen.

Mit Hilfe von *FACT* wurden die biologische Relevanz bzw. die zugrunde liegenden Mechanismen der Tumorentstehung näher charakterisiert. Hierzu erfolgte eine Erweiterung der Genannotation durch GeneOntology-Kategorien und die Suche nach signifikanten Vorkommen einzelner Kategorien. *FACT* nutzt hierzu die *Go-Term-Comparison*-Funktion (siehe Kapitel 4.4.4.). Die Annotation der Listen muriner Gene wurde ferner um Informationen zu homologen humanen Genen und deren chromosomaler Lokalisierung ergänzt (*Homology-Modul*). Die *SimpleCount*-Analysefunktion lieferte daraufhin den Hinweis, dass es für die untersuchten murinen SSCs eine erhöhte Anzahl von Gen-Überexpression in der humanen chromosomalen Bande 1q21 gab (Abb. 21a). Als Darstellung dieser genomischen Verteilung wurde das Modul *ChromosomePlot* genutzt (Abb. 21b). Die Analyse dieser Genliste mit der *MedLiner*-Funktion erleichterte schließlich die Selektion relevanter Veröffentlichung (Abb. 22). Es wurde gezeigt, dass Gene mit Zellwachstum- und Zellteilungsfunktionen, sowie Mitglieder der *S100*-Genfamilie eine entscheidende Rolle in der Karzinogenese der Haut tragen.

a.


1q21.3
17q21.2
1q32.1
20q13.12
12q13.13
7q21.3
2p11.2
17q25.1
9q34.11
11q13.1
9q22.33
11p15.4
19p13.3
17p13.1
19q13.32
14q24.3
11p15.5
0
2
4
6
counts

b.

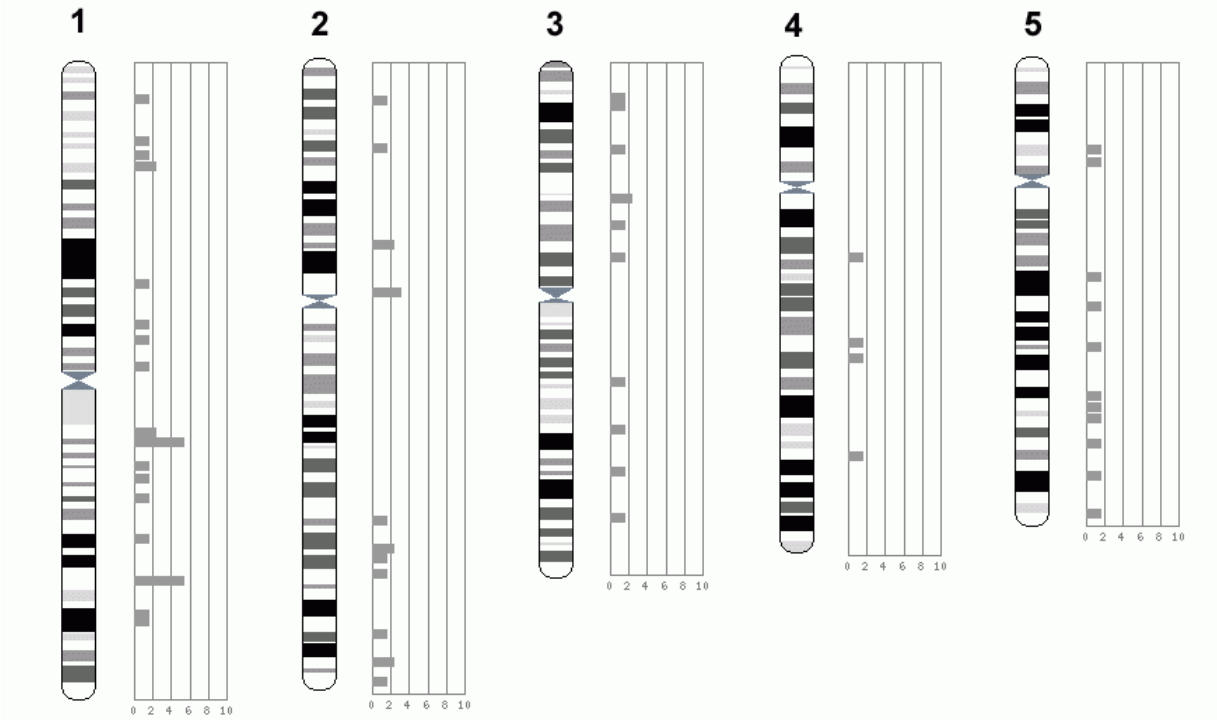



Abb. 21: *FACT*-Analysen zur Untersuchung des *Non-Melanom* Hautkrebses

a. Die *Count*-Funktion zeigt ein gehäuftes Vorkommen bestimmter genomischer Bande in den homologen Genen

b. Diese Verteilung kann mit der *Chromosome-Plot*-Funktion im genomischen Kontext dargestellt werden

- MEDLINE2004/11279127: Col1a2, Col1a1, Col3a1, Col5a2, Col6a2, Col6a3
  11279127: Verrecchia (2001)
  Identification of novel TGF-beta /Smad gene targets in dermal fibroblasts using a combined cDNA microarray/promoter transactivation approach.

- MEDLINE2004/11348466: Col1a2, Col1a1, Col3a1, Col5a2, Col7a1
  11348466: Verrecchia (2001)
  Blocking sp1 transcription factor broadly inhibits extracellular matrix gene expression in vitro and in vivo: implications for the treatment of tissue fibrosis.

- MEDLINENEW/14647409: S100a9, Sparc, Sprr2a, S100a8, Sprr1a
  14647409: Luo (2004)
  Discovery of Ca2+-relevant and differentiation-associated genes downregulated in esophageal squamous cell carcinoma using cDNA microarray.

- MEDLINE2004/12640676: Slpi,Saa3,Sprr1a,Gp38,Junb
  12640676: Schlingemann (2003)
  Profile of gene expression induced by the tumour promotor TPA in murine epithelial cells.

- MEDLINE2004/8618063: S100a9,S100a6,Sprr2a,S100a8
  8618063: Mischke (1996)
  Genes encoding structural proteins of epidermal cornification and S100 calcium-binding proteins form a gene complex ("epidermal differentiation complex") on human chromosome 1q21.

Abb. 22: *FACT*-Analysen zur Untersuchung des *Non-Melanom* Hautkrebses

Relevante Publikationen werden mit der *MedLiner*-Funktion ermittelt. Gezeigt sind jeweils die PubMed-ID, Gene, die in der Referenz gemeinsam zitiert werden, der Hyperlink zum Abstract und Autor und Titel der Publikation

## 4.6. Weitere Anwendungen im Bereich der Krebsforschung

Die etablierten Systeme zur Datenverwaltung und -analyse für Microarray-Experimente wurden in einer Vielzahl von Kooperationsprojekten eingesetzt, Tabelle 9 zeigt die bearbeiteten Fragestellungen.

### Hämatopoetische Fragestellungen

Die humane Zelllinie **HL60** ist ein gut etabliertes Modellsystem für Zelldifferenzierung innerhalb der menschlichen Hämatopoese. HL60-Zellen differenzieren aus dem promyelozytischen Stadium *in vivo* zu Granulozyton, *in vitro* durch Zugabe von chemischen Stimuli auch zu Makrophagen, Monozyten und eosinophile Granulozyten. Zur Untersuchung der Entwicklung von Promyelozyten zu Granulozyton bzw. zu

| Fragestellung / Untersuchte Krankheit | Experimentator | Genutzte Systeme * | Referenz |
|---|---|---|---|
| **Hämatopoetische Fragestellungen** | | | |
| Differenzierung der Zelllinie HL60 | G. Wrobel | 1,2 | Wrobel, unveröffentlicht |
| Veränderungen der Expression bei Akuter Myeloischer Leukämie | K. Neben | 1,2 | Neben *et al.*, 2003 b |
| **Hirntumoren** | | | |
| Veränderungen der Expression bei Ependymomen | A. Korshunov, K. Neben, G. Wrobel | 1,2 | Korshunov *et al.*, 2003 |
| Veränderungen der Expression bei Meningiomen | G. Wrobel | 1,2,3,4 | Wrobel *et al.*, eingereicht |
| Veränderungen der Expression bei Medulloblastomen | K. Neben | 1,2 | Neben *et al.*, 2004 |
| Genomische Veränderungen bei Medulloblastomen | F. Mendrzyk | 1,4 | Mendrzyk *et al.*, in Vorbereitung |
| ***Non-Melanom* Hautkrebs** | | | |
| Karzinogenese in der Haut am Mausmodell | J. Schlingemann | 1,2 | Schlingemann *et al.*, 2003 |
| Karzinogenese in der Haut am Mausmodell | L. Hummerich | 1,2,3,4 | Hummerich *et al.*, eingereicht |

Tab. 9: Anwendung der unterschiedlichen Systeme an konkreten Forschungsprojekten. * Systeme: 1-*CloneBase*, 2-*QuickLIMS*, 3-*AutoPrime*, 4-*FACT*.

Makrophagen wurde 12-O-Tetradecanoylphorbol-13-acetat (TPA) bzw. all-*trans*-Retinolsäure (RA) als Induktor eingesetzt und mit cDNA-Microarrays die Genexpression untersucht. Die für die Arrays benutzten Klone wurden in der Klondatenbank *CloneBase* gespeichert und umfassend annotiert. Die Produktion der Microarrays wurde mit Hilfe des Labordatensystems *QuickLIMS* durchgeführt. Die Ergebnisse der Hybridisierungen wurden einerseits zur Optimierung des Microarray-Produktionssystems genutzt und zeigten andererseits die unterschiedlichen Expressionsprofile definierter Gene in den beiden Entwicklungswegen (Wrobel, unveröffentlicht).

Bei der **Akuten myeloischen Leukämie** (AML), der häufigsten der akuten Leukämien, kommt es zu einer unkontrollierten Proliferation der Myelozyten. In

vorhergehenden Studien konnte eine Korrelation zwischen der Anzahl an Zellen mit Zentrosom-Abberationen und dem genetischen Risikoprofil der Patienten gefunden werden (Neben *et al.*, 2003 a). In einer Folgestudie wurden 29 Patientenproben auf cDNA-Microarrays mit 2800 verschiedenen Genen untersucht, um die molekularen Ursachen näher zu untersuchen. Die Klone wurden wiederum mittels der *CloneBase* annotiert und die Arrays mit Hilfe von *QuickLIMS* produziert. Es konnte eine molekulare Signatur identifiziert werden, welche die Patienten entsprechend des Ploidiestatus'und des Ausmaßes an Zentrosom-Abberationen einteilt. Es waren dabei Gene involviert, die Zellzylus-Regulatoren (*CCNA2*, *CCND3*, *CCNH*, *CDK6*, *CDKN2C*, *CDKN1A*, *PAK1*), bzw. Zentrosom-assoziierte Proteine (*PCNT1*, *TUBA*, *NUMA1*, *TUBGCP2*, *PRKAR2A*). kodieren (Neben *et al.*, 2003 b).

## Hirntumoren

**Ependymome** entwickeln sich aus der ependymalen Auskleidung der Ventrikel und können in allen Hirnkammern, dem Aquädukt und dem Spinalkanal anzutreffen sein. Um die molekularen Ursachen für die Tumorpathogenese näher zu charakterisieren, wurden mit 39 Ependymom-Proben *Expression Profiling*-Hybridisierungen durchgeführt. Dazu wurde ein Microarray benutzt, der 4211 humane cDNA-Fragmente als Replikate (2600 unterschiedliche Gene) enthielt. Die Klone wurden mit Hilfe des *CloneBase*-Systems annotiert. Zur Herstellung der Arrays wurde das Labordatensystem *QuickLIMS* genutzt. Die Ergebnisse der Experimente zeigen, dass es eine Korrelation der Expression bestimmter Gene mit der Tumorlokalisierung (z.B. Überexpression von HOXB5, PLA2G und CDKN2A in spinalen Ependymomen), dem Tumorgrad und dem Alter der Patienten (Überexpression von LDHB und STAM in Patienten unter 17 Jahren) gibt. Sie zeigen auch, dass Ependymome evtl. eine Gruppe molekular klar differenzierbaren Subtypen darstellt (Korshunov *et al.*, 2003).

**Meningiome** sind Tumoren, die von dem das Gehirn und Rückenmark umgebende Epithelgewebe ausgehen. Man unterscheidet zwischen benignem (WHO Grad I), atypischem (WHO Grad II) und anaplastischen (WHO Grad III) Meningiom. Um ein besseres Verständnis der Tumorpathogenese und eine gezielte Behandlung der Patienten erreichen zu können, ist es jedoch notwendig, molekulare Marker zu identifizieren, die diese Unterteilung besser an das tatsächliche klinische Bild

anpassen. Es wurden Hybridisierungen von 30 Tumoren auf den 2600-Gen Microarrays durchgeführt. Nach der Datenanalyse wurden im Vergleich von atypischen/anaplastischen zu benignen Tumoren 37 Gene als unter- und 27 Gene als überexprimiert identifiziert. Es wurde eine Gen-Signatur erstellt, welche anaplastische von benignen Meningiomen abgrenzen kann. Sie involvierte Gene der Zellzyklusregulation und der Proliferation. Die Ergebnisse wurden mit Hilfe von *FACT* umfassend annotiert und mit CGH-Experimenten korreliert, die zu identischen Fällen vorlagen (Wrobel *et al.*, eingereicht). Es konnte gezeigt werden, dass Verluste auf den Chromosomen 10 und 14 mit genau definierten Expressionsprofilen korreliert waren. Diese zeigten eine erhöhte Expression von Genen des *insulin-like growth factor*-Signalweges (*IGF2, IGFBP3* und *AKT3*) bzw. des *wingless/WNT*-Signalweges (*CTNNB1, CDK5R1*, *ENC1* and *CCND1)*.

**Medulloblastome** sind neuroektodermale Tumoren des Kleinhirns und sind unter den häufigsten Tumorerkrankungen des Zentralnervensystems bei Kindern. Trotz intensiver Therapiemaßnahmen liegt die 5-Jahres Überlebensrate bei nur 50-60%. Um Kandidatengene zu finden, die mit möglichen Therapieerfolgen korrelieren, wurden 35 Medulloblastom-Neoplasien mittels *Expression Profiling* untersucht. Als Ergebnis konnten 54 Gene identifiziert werden, deren Expressionsprofil mit einer schlechten Überlebensrate korrelieren. Es wurden außerdem immunhistochemische Untersuchungen und *In situ* Hybridisierungen auf Gewebe-Microarrays durchgeführt. Im Anschluss wurden mit matrixCGH-Microarrays mit teilweise identischen Patientenproben die genomischen Profile ermittelt. Auch die genomischen Klone sind in der *CloneBase* archiviert. Es wurde *FACT* benutzt, um über- und unterrepräsentierte Bereiche des Genoms als Ideogramm darzustellen und um eine Korrelation von Expressionswerten und genomischen Veränderung zu untersuchen. Die Überexpression der Gene STK6, STMN1 und CCND1 konnte mit einer schlechten Überlebensrate in Verbindung gebracht werden, STK6 erwies sich als starker unabhängiger Marker. Auch eine Erhöhung der genomischen Kopienzahl von MYC und STK6 korreliert mit einer verschlechterten Prognose (Neben *et al.*, 2004 und Mendrzyk *et al.*, in Vorbereitung).