Von Daten zur Information - ein System für Mikroarray-Analyse und Datamining

Xiaolei Yu
Zentrum für Medizinische Forschung, Universitätsklinikum Mannheim, Universität Heidelberg
Theodor-Kutzer-Ufer
68167 Mannheim
xiaolei.yu@zmf.ma.uni-heidelberg.de

Li Li
Zentrum für Medizinische Forschung, Universitätsklinikum Mannheim, Universität Heidelberg
Theodor-Kutzer-Ufer
68167 Mannheim
li.li@zmf.ma.uni-heidelberg.de

Serge Luke
Department of Urology of Wellington Public Hospital
Riddiford Street
Wellington, New Zealand
Serguei@em.co.nz

Norbet Gretz
Zentrum für Medizinische Forschung, Universitätsklinikum Mannheim, Universität Heidelberg
Theodor-Kutzer-Ufer
68167 Mannheim
norbert.gretz@zmf.ma.uni-heidelberg.de

Zusammenfassung

Die Mikroarray-Technologie ermöglicht eine schnelle, simultane Expressionsanalyse tausender Gene. Die Analyse der Mikroarrays kann aber aufgrund der dadurch erzeugten riesigen Datenmengen nicht mehr mit herkömmlichen Programmen, wie Excel, bewältigt werden. In Zentrum für Medizinische Forschung wurde ein System zur Mikroarray-Analyse und zum nachfolgenden Datamining mit Hilfe von *SAS microarray, SAS/BASE und SAS/STAT* etabliert. In diesem Beitrag wird anhand eines Beispiels vorgestellt, wie man mittels dieser SAS-basierten Routine zu biologisch relevanten Informationen gelangen kann.

Schlüsselwörter: Microarray, SAS *microarray*, Data Mining, Pathway-Analyse

1 Einleitung

Im Zentrum für Medizinische Forschung (ZMF) der Medizinischen Fakultät Heidelberg wird seit 2002 eine Core-Einrichtung für Mikroarray-Analytik betrieben. Unter verschiedenen Mikroarray Formaten werden im ZMF hauptsächlich Affymetrix Chips be-

nutzt. Bei *Affymetrix*-Chips wird ein Gen als *probe set* definiert. Laut Definition *von Affymetrix* enthält jedes *probe set* 11 bis 20 Sondenpaare. Auf einer Fläche eines Fingernagels sind mehrere hunderttausend Sondensequenzen vorhanden, die wiederum mehr als zehntausend Gene detektieren können. Die dadurch erzeugten Datenmengen sind zu groß, um sie mit herkömmlichen Programmen, wie z.B. Excel, auswerten zu können. Um diese Daten auszuwerten, braucht man ein spezielles Programm. Aufgrund seiner Fähigkeit zur statistischen Analyse und Verwaltung großer Datenmengen, eignet sich SAS hervorragend für diese Aufgabe. Außerdem basiert die statistische Analyse bei *SAS microarray* auf dem Probe-Level, was die statistische Leistung um mehr als das 10-fache erhöht. Im ZMF wurde eine Routine zur Mikroarray-Analyse und zum nachfolgenden Datamining mit Hilfe von *SAS microarray*, *SAS/BASE* und *SAS/STAT* etabliert.

Diese Routine soll anhand eines Beispiels dargestellt werden. *Polycystic kidney Diease* (PKD) ist eine polyzystische Nierenerkrankung, die durch eine langsam voranschreitende Vergrößerung flüssigkeitsgefüllter Zysten in beiden Nieren gekennzeichnet ist. PKD ist eine der häufigsten monogenetischen Erbkrankheiten. Die Prävalenz der Krankheit liegt bei 1:400 bis 1:1000. Interessanterweise ist die Ausprägung der Krankheit abhängig vom Geschlecht und von der genetischen Ausstattung der Erkrankten. Es ist bekannt, dass bei Männern die Krankheit früher eintritt und schneller fortschreitet als bei Frauen. Der genaue Mechanismus dafür ist noch unklar. In dieser Studie wurden zwei unterschiedliche Rattenlinien als PKD-Modelle verwendet. Um herauszufinden, wie die Zysten entstehen und wie das Geschlecht an der Ausprägung der PKD beiträgt, wurde ein Genexpressionsexperiment mit gesunden und kranken Ratten durchgeführt. Dabei wurden beide Geschlechter beider Rattelinien verwendet. In diesem Beitrag wird anhand dieses Beispiels vorgestellt, wie man mittels unserer Routine von Mikroarray-Daten zu biologischen Informationen gelangen kann.

2 Ergebnisse

SAS microarray ist ein SAS-Paket, das speziell für die Mikroarray-Analyse entwickelt wurde. In diesem Paket ist die *JMP statistical discovery* Software integriert, was die Visualisierung der Mikroarray-Ergebnisse in Form dynamischer Graphiken ermöglicht.
Die Mikroarray-Auswertungsroutine besteht aus dem Dateninput, der Qualitätskontrolle, der statistischen Analyse und einem Datamining. Durch die Mikroarray-Auswertung kann eine statistische Aussage darüber getroffen werden, ob ein Gen zwischen zwei unterschiedlichen Zuständen oder Versuchsbedingungen signifikant unterschiedlich exprimiert wird. Um den dahinter stehenden biologischen Zusammenhang zu verstehen, ist im Anschluss ein Datamining notwendig, wie z.B. eine Pathway-Analyse, eine Transkriptions-Faktor-Analyse (TF) oder Literaturrecherche. Dadurch können die Kandidaten-Gene identifiziert werden, die sowohl statistisch signifikant als auch biologisch sinnvoll sind. Solche Gene können dann im Labor mit Hilfe weiterer Methoden, wie z.B. qualitativer PCR, verifiziert werden. Der Ablauf ist in Abbildung 1 dargestellt.

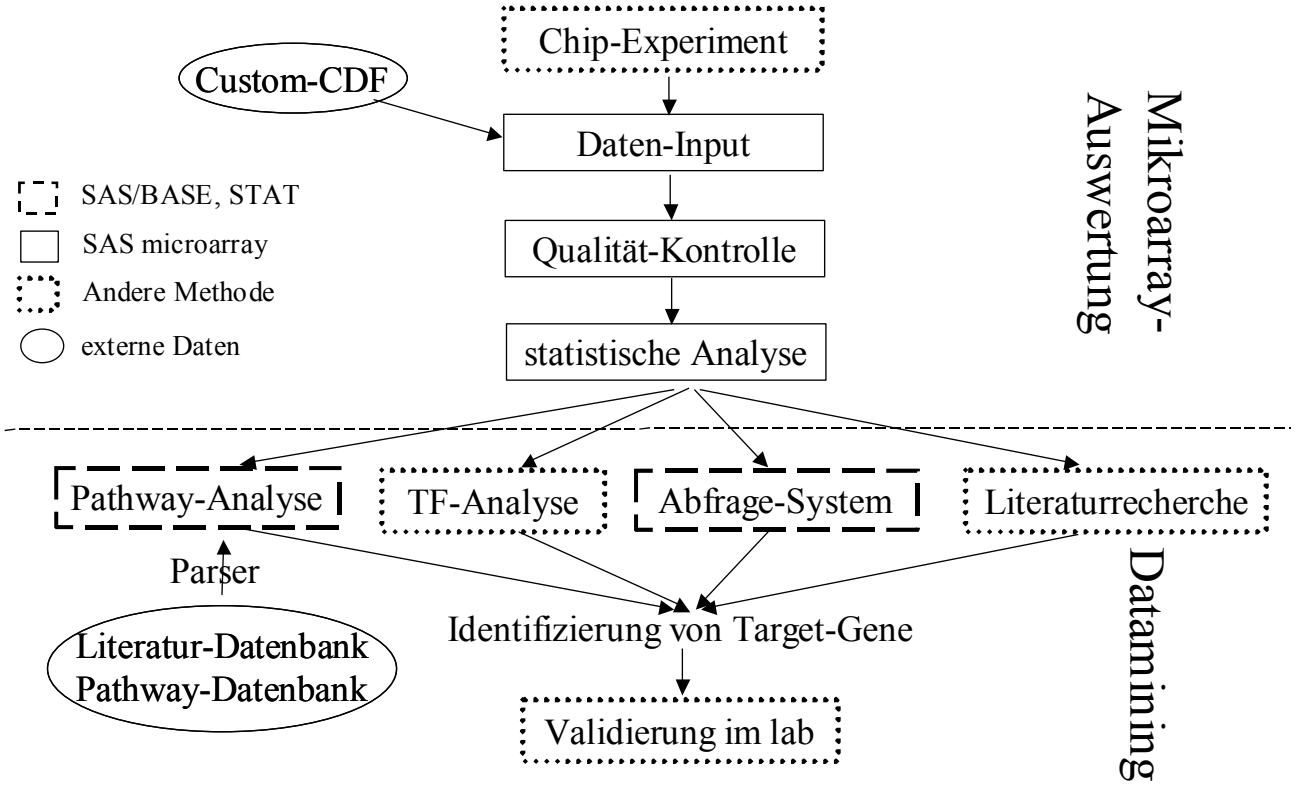

Abbildung 1: Ablauf des Systems

2.1 Dateninput

Der erste Schritt der Chipanalyse ist der Dateninput. Drei Dateien werden benötigt: CEL-Datei, CDF-Datei und eine Experiment-Design-Tabelle. Die CEL-Datei beinhaltet die Koordinaten und Intensitäten der Spots. Die CDF-Datei enthält die Information über die Zugehörigkeit der Sonden zum Gen. Bei uns werden an dieser Stelle die *Custom*-CDF-Datei anstatt der *Affymetix*-CDF-Datei verwendet, die alle drei Monate aktualisiert werden [1]. Die Experiment-Design-Tabelle beschreibt die experimentellen Bedingungen des einzelnen Chips (Tabelle 1). In Tabelle 1 ist jeweils einer der drei Chips pro Gruppe dargestellt.

Tabelle 1: Experiment-Design-Tabelle

ChipID	Sex	Line	PKD	File
1	female	mhm	krank	Chip1.cel
2	female	mhm	gesund	Chip2.cel
3	female	us	krank	Chip3.cel
4	female	us	gesund	Chip4.cel
5	male	mhm	krank	Chip5.cel
6	male	mhm	gesund	Chip6.cel
7	male	us	krank	Chip7.cel
8	male	us	gesund	Chip8.cel

Es gibt für verschiedene Array-Formate jeweils eine eigne *Input Engine*. Da im beschriebenen Beispiel *Affymetrix*-Chips im Einsatz waren, wurde die *Affymetrix Input Engine* benutzt (Abbildung 2). Hier kann man verschiedene Parameter eingeben und diese Einstellung kann auch für eine spätere Anwendungen gespeichert werden. Die SAS *Input Engine* sorgt dafür, dass die Informationen aus CEL-Dateien, CDF-Dateien sowie die experimentellen Bedingungen in einer großen SAS-Datei zusammengefasst werden, die als Basis für die weitere Analyse dient.

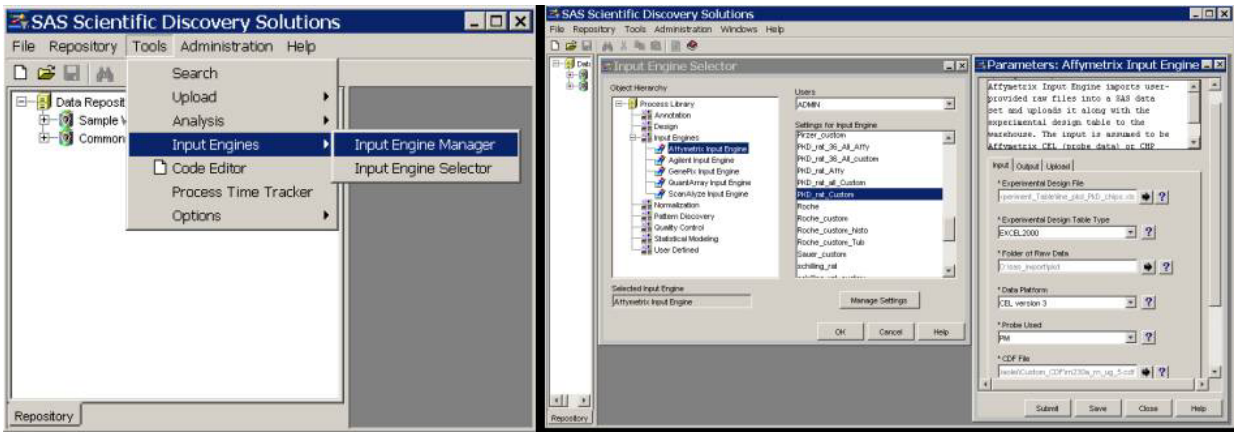

Abbildung 2: Screenshot des *Input Engine*

2.2 Qualitätskontrolle

Die Qualitätskontrolle der Chipdaten wird mit Hilfe einer Korrelationsanalyse (*array group correlation*) durchgeführt. Nur die Chips mit guter Qualität werden für weitere Analysen zugelassen. Die Chips mit schlechter Qualität werden entweder von der Analyse ausgeschlossen oder das Experiment wird wiederholt. Bei der *array group correlation* werden die *Pearson* Korrelations-Koeffizienten für alle Chips einer Experimentbedingung berechnet. Die Korrelation wird mit Hilfe einer *Scatterplot Matrix* dargestellt. Ein Korrelationskoeffizient nahe 1 bedeutet eine gute biologische und technische Übereinstimmung der Chipdaten. Ein Korrelationskoeffizient nahe -1 bzw. +1 besitzt eine Zigarrenform. Ein Korrelationskoeffizient nahe 0 besitzt eine Kreisform. Die Korrelationskoeffizienten unseres Experiments liegen zwischen 0.97 und 1 (Abbildung 3), was

zeigt, dass die einzelnen Chips innerhalb ihrer Gruppe eine hohe Korrelation untereinander besitzen.

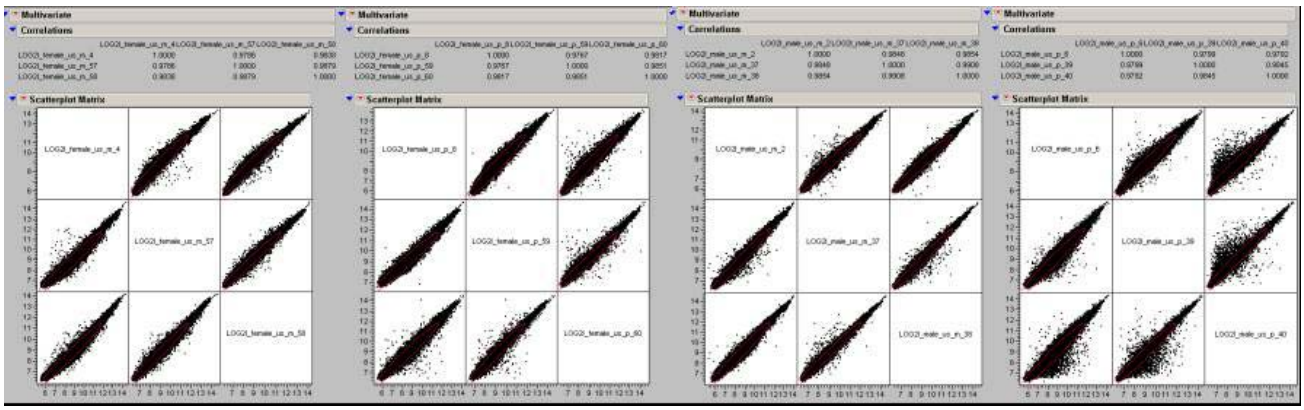

Abbildung 3: array group correlation

2.3 Pseudoimage

Eine andere Möglichkeit die Qualität der Chips zu überprüfen ist, mit dem SAS *array Pseudo Image* Prozess das Abbild des Chips anzuschauen. In Abbildung 4 ist ein Chip mit schlechter Qualität auf der linken Seite und ein Chip mit guter Qualität auf der rechten Seite dargestellt. Diese zwei Chips stammen aus einem anderen Experiment. Wenn der linke Chip bei der Analyse miteinbezogen wurde, gab es 278 signifikante Gene. Wenn dieser Chip jedoch ausgeschlossen wurde, wurden 967 Gene signifikant.

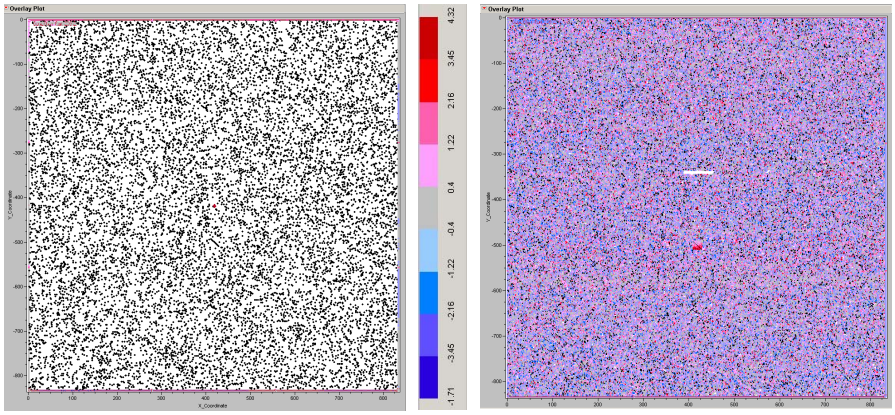

Abbildung 4: *Pseudoimage* ein guter und ein schlechter Chip

2.4 Statistische Analyse

Um einen Vergleich zwischen den unterschiedlichen Bedingungen zu ermöglichen, werden zunächst die log-transformierten Daten durch ein gemischtes Linear-Model normalisiert. Anschließend wird ein *gene-by-gene* gemischtes Linear-Model für die Analyse der Varianz eingesetzt. Die zugrunde liegende statistische SAS Prozedur für diese zwei Schritte ist *PROC MIXED*.

2.4.1 *Mixed Model* Normalisierung

Bei der *Mixed Model* Normalisierung wurden in unserem Beispiel Geschlecht, Rattenlinie, Phänotyp der Tiere und ChipID als Klasse definiert. Die ersten 3 Parameter sind *fixed* Effekte und der letzte ist der *random* Effekt. Bei diesem Model wurde sowohl der einzelne Parameter als auch die Wechselwirkung zwischen den Parameter berücksichtigt. Auf der unteren linken Seite in Abbildung 5 sind die dazu verwendeten SAS-Anweisungen gezeigt.

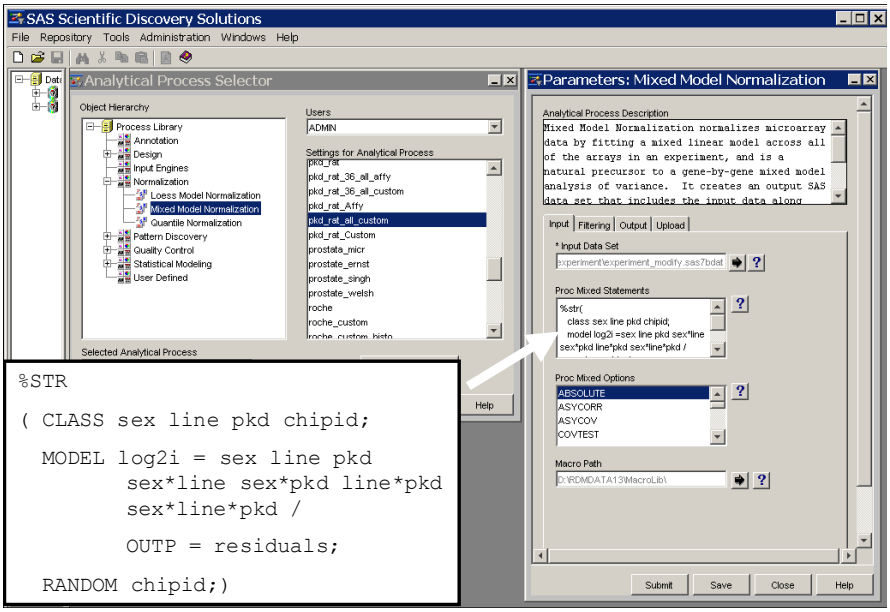

Abbildung 5: *Mixed Model* Normalisierung

2.4.2 *Mixed Model* Analyse

Mit den normalisierten Werten wurde die *mixed model ANOVA* durchgeführt. Die SAS-Anweisungen dafür sind ähnlich wie bei der Normalisierung (siehe unten). Die Sonde wurde aber bei der Klasse als *fixed* Effekt definiert. Außerdem wurde bei der *estimate*-Anweisung angegeben, welche Bedingungen miteinander verglichen werden sollten. Dabei wurden männliche Tiere mit weiblichen Tieren, kranke Tiere mit gesunden Tieren sowie Mhm-Linie mit US-Linie verglichen. Man kann auch nur einen Teil der Daten analysieren, indem man eine *where*-Anweisung wie `where line="mhm";` schreibt.

```
%str
( CLASS sex line pkd probe chipid;
  MODEL log2in=sex line pkd sex*line sex*pkd line*pkd
                sex*line*pkd probe / outp=generesiduals;
  RANDOM chipid;
  LSMEANS sex*line*pkd;
  ESTIMATE "female_male" sex 1 -1;
  ESTIMATE "pkd_gesund" pkd -1 1;
  ESTIMATE "mhm_us" line 1 -1; )
```

Die Ergebnisse werden sowohl tabellarisch als auch graphisch darstellbar. Alle Tabellen und Diagramme sind miteinander verknüpft, was eine interaktive Datenverarbeitung ermöglicht. Es gab zwei Tabellen, „*mixedmodelresults*“ und „*significant genes*“. Die erste enthält alle Gene und die zweite beinhaltet nur die signifikant regulierten Gene. Als *cutoff* wurde die *Bonfferoni* Korrektur verwendet. Allerdings kann der *cutoff*-Wert manuell verändert werden.

Auf der linken Seite des Verteilungsdiagramms (Abbildung 6) ist *Rsquared* dargestellt. Mit diesem Wert kann man überprüfen, ob das statistische Modell geeignet ist. Dieser Wert soll zwischen 0 und 1 liegen, wobei ein höherer Wert für ein gutes Model steht. Auf der rechten Seite ist die restliche Varianz dargestellt. Je kleiner der Wert ist, desto besser ist das Modell. In unsrem Fall liegen *Rsquared* bei 0.96 (Mean) und die restliche Varianz bei 0.03, was auf eine gute Modellbildung hinweist.

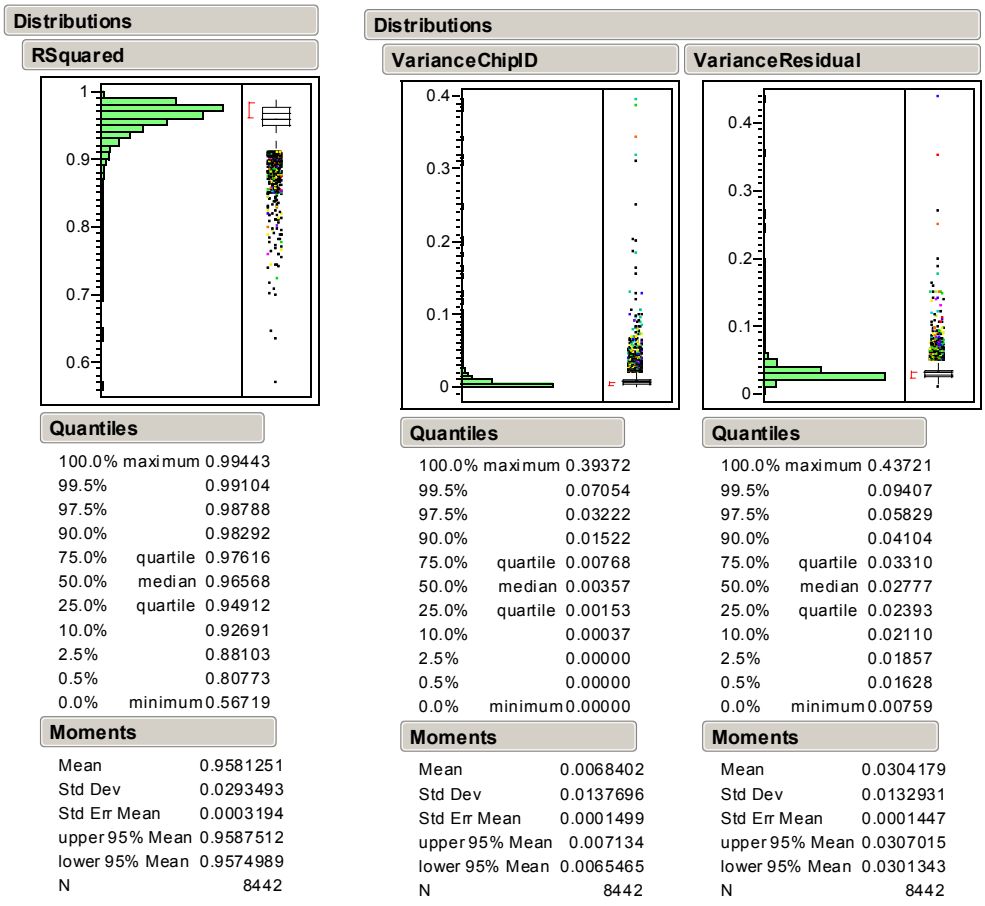

Abbildung 6: Verteilungsdiagramm

Im *volcano plot* (Abb. 7) wird die statistische Signifikanz gegen den log 2 des *fold changes* der Expressionswerte aufgetragen. Dieser Plot wird benutzt, um die unterschiedlich regulierten Gene darzustellen. Die interessantesten Gene liegen in der oberen linken und oberen rechten Ecke. Diese Gene zeigen sowohl größere *fold changes* als auch ein höheres Signifikanzniveau.

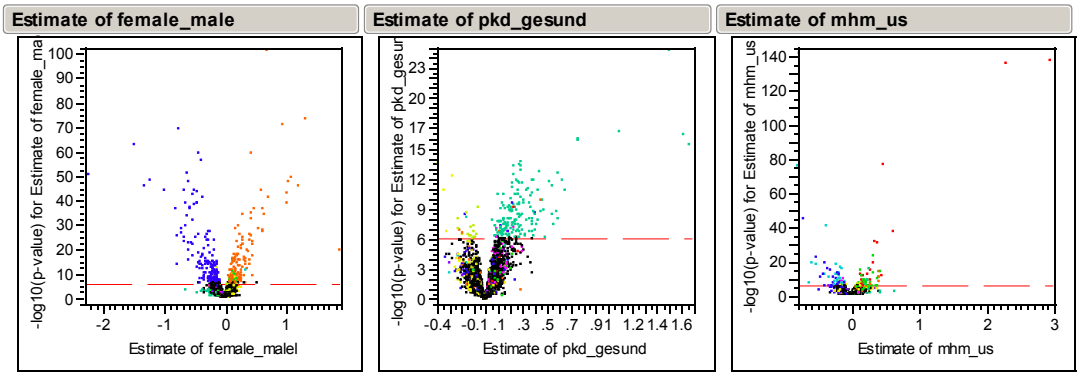

Abbildung 7: *volcano plot*

Das Diagramm auf der linken Seite der Abbildung 8 zeigt das hierarchische Clustering. Die ko-regulierten Gene werden zusammen geclustered. Sie könnten eventuell eine gemeinsame Funktion haben. Es gibt eine klare Trennung zwischen männlichen und weiblichen Tiere. Die Trennung lag bei den weiblichen Tiere an der Tierlinie, während sie bei den männlichen Tieren am Gesundheitszustand lag. Dies lässt sich auf das Fortschreiten der Nierenkrankheit zurückführen. Das rechte Diagramm zeigt einen *parallel plot*. Dabei wird der Expressionswert jedes einzelnen Gens unter verschiedenen Bedingungen dargestellt. Er ist sehr nützlich für die *time course* Analyse.

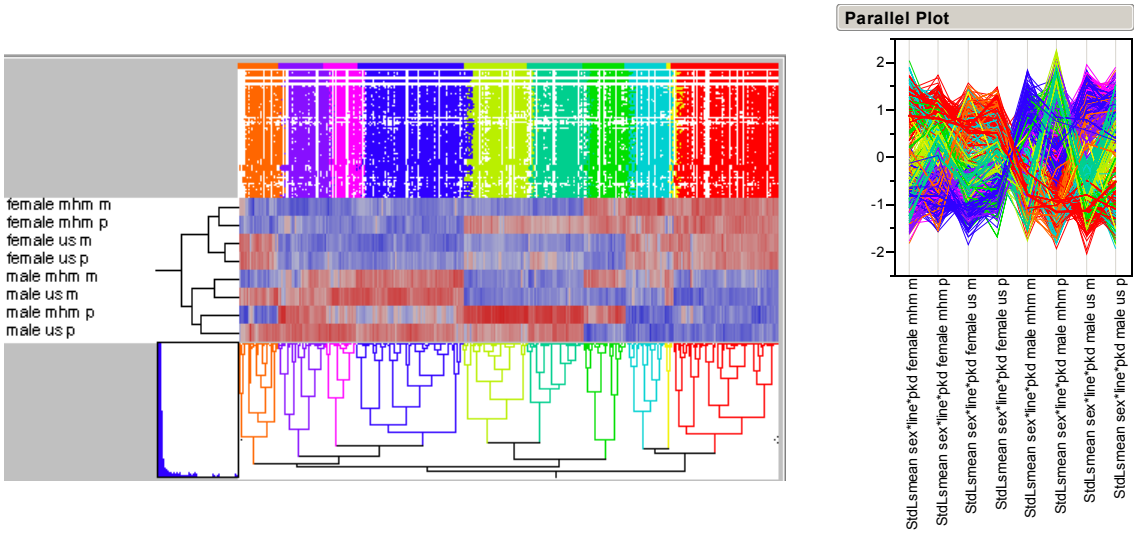

Abbildung 8: Hierarchisches Clustering und *parallel plot*

2.5 Datamining

2.5.1 Pathway-Analyse

Durch diese Mikroarray-Auswertung kann man die statistisch signifikant geregelten Gene herausfinden. Um den dahinter stehenden biologischen Zusammenhang zu verstehen, ist im Anschluss ein Datamining, wie z.B. eine Pathway-Analyse, notwendig. Für

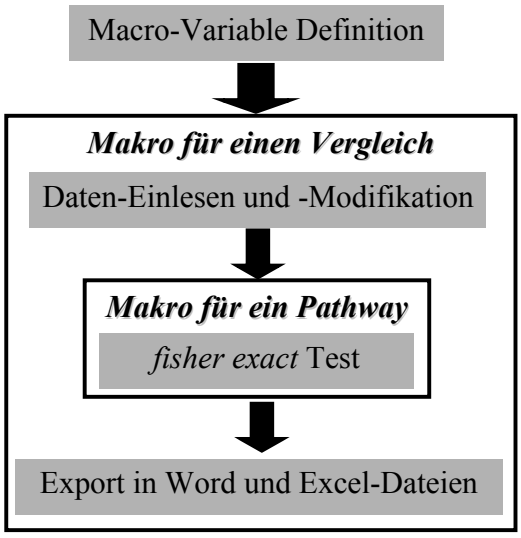

Abbildung 9: Pathway-

die Pathway-Analyse ist ein neues SAS-Programm entwickelt worden. Dieses Programm sagt aus, ob sich eine vordefinierte Gen-Liste in einem Mikroarray-Datensatz als Gruppe zwischen zwei Zuständen signifikant verändert. Die Gene in dieser Liste haben entweder eine gemeinsame Funktion oder beteiligen sich an einem gemeinsamen Prozess und werden daher als Pathway definiert. Sie sind sehr wahrscheinlich auch gemeinsam reguliert und haben ein ähnliches Expressionsmuster. Eine solche Analyse liefert im Vergleich zur Veränderung einzelner Gene stabilere Ergebnisse [2]. In diesem Programm wurden Pathways aus öffentlichen Datenbanken wie z.B. KEGG, aber auch Pathways aus eigener Literatursuche verwendet. Für die statistische Auswertung wurde die SAS Prozedur *PROC FREQ/Fisher's exact test* durchgeführt. Um die Prozedur für jeden Vergleich und für jeden Pathway anwenden zu können, wurden zwei verschachtelte Makros verwendet (Abbildung 9). Variable, wie die Spezies, der *cutoff* des P-Werts sowie die Input- und Output-Datei waren als globale Variablen definiert, wodurch das Programm für alle Chip-Typen einsetzbar ist.

```
/*Spezies*/
%LET species=rno;
/*SAS Microarray Analyse Output-Datei (mixedresults-Tabelle)*/
%LET filename=D:\MAS\mixedresults;
/*Name und Verzeichnis der Output-RTF-Datei*/
%LET out_file=D:\MAS\pathway_&Estimate..rtf;
/*Name und Verzeichnis der Output-Excel-Datei*/
%LET out_excel_file=D:\MAS\pathway_&Estimate..xls;
/*Signifikant cutoff neglog10p-value*/
%LET cutoff=3;
/*Variable-Name für Gene-Symbol in Mixedresults-Tabelle*/
%LET gene_var=Gene_Symbol;
/*Verzeichnis für Pathway-Text-Datei*/
%LET curated_lists_path=D:\pathway\kegg\&species.\txt\;
```

Im Marco für jeden Vergleich werden nur die Variablen, die für die Analyse notwendig sind, beibehalten. Durch *PROC SQL* wurde eine neue Variable „*Significant*" dazugefügt und mit 0 und 1 definiert, je nachdem ob ein Gen signifikant reguliert war oder nicht. Dieses Makro wird für jeden Vergleich aufgerufen.

```
/*Makro für jeden Vergleich*/
%MACRO One_Estimate (Estimate, p_variable, fold_variable);
...
     DATA chip;
```

```
          SET chip;
          KEEP UniGene_ID &gene_var &p_variable &fold_variable;
          WHERE &gene_var NE "---";
     RUN;
PROC SQL;
     ALTER TABLE chip ADD significant INTEGER;
     UPDATE chip SET significant=0;
     UPDATE chip SET significant=1 WHERE &p_variable>&cutoff;
...
%MEND One_Estimate;
/*Aufruf von Makro One_run für jeden Vergleich*/
%One_Estimate (female_male, NegLog10pEstimate1, Estimate1);
%One_Estimate (pkd_gesund, NegLog10pEstimate2, Estimate2);
%One_Estimate (mhm_us, NegLog10pEstimate3, Estimate3);
```

Innerhalb eines Vergleiches sollen alle Pathways mit einem SAS Makro überprüft werden. Die erste Anweisung dieses Makros ruft ein weiteres Makro auf. Der Pathway-Name wird als Parameter eingesetzt. Alle Pathways werden als Text-Datei gespeichert, in der die einzelnen Gene-Symbole aufgelistet werden. Mittels *PROC SQL* werden der Pathway-Name, der Pathway-Zähler und der Vergleich als Variablen definiert. Wenn ein Gen in einem Pathway vorhanden ist, wird 1 als Wert der Variable Pathway-Name zugewiesen, der Pathway-Name wird mit ein Kreuz markiert und der Pathway-Zähler wird um 1 erhöht.

```
/*Aufruf von Makro für jeden Pathway*/
% INCLUDE "D:\pathway\kegg\&species.\pathway_list.SAS";
/*Makro für jeden Pathway*/
%MACRO One_pathway (pathway_name=)
     /*Erzeugen curated_list von Pathway-Text-Datei*/
     %LET f=&curated_lists_path&pathway_name..txt;
     DATA curated_list;
     INFILE "&f" DELIMITER='09'x FIRSTOBS=2;
     INPUT gene_in_pathway $;
     RUN;
...
     PROC SQL;
       ALTER TABLE significant_genes ADD &pathway_name char(3);
       UPDATE significant_genes SET &pathway_name="X",

       pathway_count=pathway_count+1


       WHERE &gene_var IN(SELECT gene_in_pathway FROM curated_list);
...
     PROC SQL;
       UPDATE chip SET pathway_&Estimate=0;
```

```
        UPDATE chip SET pathway_&Estimate=1 WHERE &gene_var IN
        (select gene_in_pathway from curated_list);
...
%MEND One_pathway;
```

Für die statistische Berechnung wird *Fisher's exact test* verwendet, wobei die Wahrscheinlichkeit für nicht zufällige Zusammenhänge zweier Variablen berechnet werden. Die zwei Variablen beschrieben, auf einer Seite, ob ein Gen in einem Pathway vorhanden war oder nicht, und auf der anderen Seite, ob dieses Gen in Mikroarray-Experiment signifikant reguliert war oder nicht. Ein Format wurde definiert und je nach Signifikanz die Gene mit verschiedenen Symbolen gekennzeichnet. Mit *ODS* werden die Ergebnisse als Word-Datei und Excel-Datei ausgegeben (Abb. 10).

```
/*Fisher´s exact test für jeden Pathway*/
PROC FREQ NOPRINT DATA=chip;
     TABLES pathway_&Estimate * significant/nocol exact nopercent;

     OUTPUT ALL OUT=fish;
RUN;
...
PROC FORMAT;
     VALUE stern low-0.00100='***'
     0.00101-0.01000='**'
     0.01001-0.05000='*'
     0.05001-0.10000='?'
     0.10001-1.00000='-';
RUN;
...
ODS RTF FILE="&out_file";
     PROC PRINT DATA=result;
          FORMAT star stern.;
     RUN;
ODS RTF CLOSE;
```

In der Word-Datei (Abb. 10) kann man Informationen, wie den Pathway-Namen, die Anzahl der signifikant regulierten Gene auf dem Chip in einem Pathways, die Anzahl der gesamten Gene auf dem Chip in einem Pathway und den p-Wert des *Fisher's exact test,* bekommen. Je nach Signifikanzniveau werden die p-Werte mit verschiedenen Symbolen codiert. In der Excel-Datei werden Informationen, wie der Pathway-Name, die AffyID, das Gen-Symbol, der -log10(p-Wert) und der *fold change,* dargestellt. Die Spalte „Pathway-Count" zeigt, in wie vielen Pathways jedes Gen involviert ist. Die letzte Zeile zeigt, wie viele Gene jeder Pathway enthielt. Die signifikant regulierten Gene werden mit einem Kreuz markiert.

Obs	pathway_female_male	hits	total	hits_percent	fisher	star
1	all	767	6488	11.822	.	.
2	Fatty_acid_metabolism	13	29	44.828	0.00001	***
3	Valine_leucine_and_isoleucine_degradation	10	24	41.667	0.00021	***
4	Fatty_acid_elongation_in_mitochondria	5	7	71.429	0.00039	***
5	Caprolactam_degradation	5	7	71.429	0.00039	***
6	Benzoate_degradation_via_hydroxylation	3	3	100.000	0.00165	**
7	Biosynthesis_of_steroids	5	10	50.000	0.00345	**
8	Propanoate_metabolism	6	16	37.500	0.00753	**
9	beta_Alanine_metabolism	5	12	41.667	0.00885	**
10	Folate_biosynthesis	4	8	50.000	0.00917	**
11	Pyrimidine_metabolism	6	18	33.333	0.01419	*
12	Nicotinate_and_nicotinamide_metabolism	4	9	44.444	0.01497	*
13	Tryptophan_metabolism	7	25	28.000	0.02229	*
14	Glutathione_metabolism	6	20	30.000	0.02414	*
15	Butanoate_metabolism	6	20	30.000	0.02414	*
16	Lysine_degradation	5	15	33.333	0.02483	*

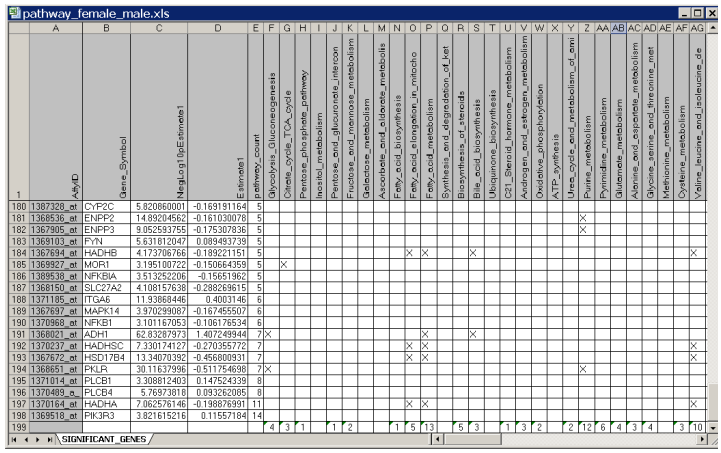

Abbildung 10: Output der Pathway-Analyse

2.5.2 Abfrage-System

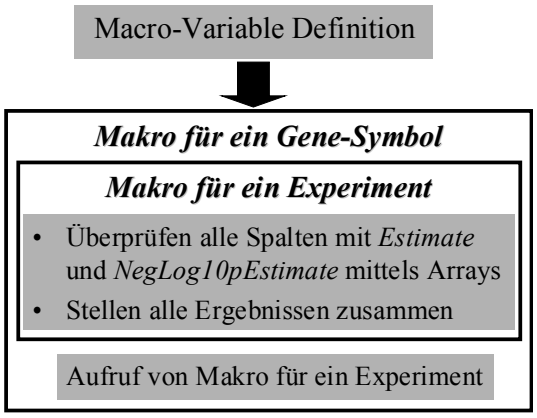

Abbildung 11: Abfrage-System

Wenn man sich für ein bestimmtes Gen interessiert und wissen möchte, in welchen anderen Experimenten sich die Expression dieses Gens ebenfalls verändert hat, kann man das Abfrage-Programm für diese Fragestellung benutzen. Dafür wurden zwei verschachtelte Makros verwendet. Das erste definiert den *cutoff* als globale Variable. Nur Gene mit einem bestimmten *fold change* und Signifikanzlevel werden gesucht. Darauf wurde das Makro für ein Gen definiert. Innerhalb dieses Makros wurde das Makro für ein Experiment definiert. *Array* wurde dabei benutzt um alle Spalten-Namen, die mit *estimate* angefangen, zu überprüfen. Beim Aufruf dieses Makros werden alle existierenden Mikroarray-Experimente durchgesucht. Diese Liste wird durch ein Skript automatisch aktualisiert. Als Output werden alle Chip-Experimente mit einer signifikanten Änderung des gesuchten Gens aufgelistet (Abbildung 12), wobei das Verzeichnis des Experimentes, der signifikante Vergleich, der *fold change* und der p-Wert angezeigt werden.

```
/*Cutoff für Veränderung*/
%LET Cutoff_fold_change=1;
/*Cutoff für P-Wert*/
%Let Cutoff_P_value=3;
...
ARRAY Estimate(*) Estimate: ;
ARRAY NegLog10pEstimate(*) NegLog10pEstimate: ;
n=DIM(Estimate);
```

```
DO i=1 TO n;
     IF (Estimate(i)>&Cutoff_fold_change or
     Estimate(i)<-&Cutoff_fold_change) and
     (NegLog10pEstimate(i)>&Cutoff_P_value)
     THEN DO;
          experiment=&Path;
          fold_change_name="Estimate"||left(i);
          fold_change=Estimate(i);
          P_value_name="NegLog10pEstimate"||left(i)
          P_value=NegLog10pEstimate(i);
          OUTPUT;
     END;
END;
...
%One_experiment ("D:\mas\Expriment1\MMA");
%One_experiment ("D:\mas\ Expriment2\MMA");
%One_experiment ("D:\mas\ Expriment3\MMA");
```

	experiment	fold_change_name	fold_change	P_value_name	P_value
1	D:\mas\chirurgie_gerstenbergk\horisberger\MMA\pra	Estimate1	1.2831053195	NegLog10pEstimate1	6.73621906
2	D:\mas\groene\kenzelmann_all_fibroblasts\mma\natu	Estimate1	1.0744127874	NegLog10pEstimate1	13.457453149
3	D:\mas\groene\kenzelmann_MOE430A_2\Xiaole\M	Estimate1	1.1822533453	NegLog10pEstimate1	15.028264385
4	D:\mas\med3\waldhof\n_vs_R_Projekt_7pairs\mma\	Estimate1	1.473577582	NegLog10pEstimate1	8.5816537716
5	D:\mas\med3\waldhof\n_vs_R_Projekt_7pairs\mma\	Estimate1	1.1098375078	NegLog10pEstimate1	6.1284787843
6	D:\mas\Schilling\MOE430_2_complete\mma	Estimate4	1.2620997838	NegLog10pEstimate4	11.235389023
7	.	Estimate6	-1.255519062	NegLog10pEstimate6	11.14655722
8	D:\mas\Schuetz\wolfgang schmid_Jan Tuckermann\mma\abchip\grdim	Estimate2	1.7530721036	NegLog10pEstimate2	17.024966268
9	.	Estimate3	1.4201637367	NegLog10pEstimate3	12.530711254
10	.	Estimate4	-1.764843941	NegLog10pEstimate4	26.695803781
11	.	Estimate5	1.4319355743	NegLog10pEstimate5	20.474344661
12	D:\mas\Schuetz\wolfgang schmid_Jan Tuckermann\mma\abchip\wt	Estimate2	1.9095205887	NegLog10pEstimate2	21.197020853
13	.	Estimate3	1.5023754906	NegLog10pEstimate3	15.21983758
14	.	Estimate4	-1.927262179	NegLog10pEstimate4	33.921198744
15	.	Estimate5	1.5201170809	NegLog10pEstimate5	25.806503439
16	D:\mas\Schuetz\wolfgang schmid_Jan Tuckermann\mma\achip\grdim	Estimate2	1.7530721036	NegLog10pEstimate2	17.024966268
17	.	Estimate3	1.4201637367	NegLog10pEstimate3	12.530711254
18	.	Estimate4	-1.764843941	NegLog10pEstimate4	26.695803781
19	.	Estimate5	1.4319355743	NegLog10pEstimate5	20.474344661
20	D:\mas\Schuetz\wolfgang schmid_Jan Tuckermann\mma\achip\wt	Estimate2	1.9095205887	NegLog10pEstimate2	21.197020853
21	.	Estimate3	1.5023754906	NegLog10pEstimate3	15.21983758
22	.	Estimate4	-1.927262179	NegLog10pEstimate4	33.921198744
23	.	Estimate5	1.5201170809	NegLog10pEstimate5	25.806503439

Abbildung 12: Output des Abfrage-Systems

2.6 Biologische Erklärung

Durch die Mikroarray-Analyse und das anschließende Datamining konnte in unserem Beispiel herausgefunden werden, dass die männlichen Tiere im Vergleich zu den weiblichen Tieren eine höhere Stoffwechsel- und Transport-Aktivität sowie eine erhöhte Stimulation für Zellproliferation zeigten [3]. Ebenso ausgeprägt war ein unbalanciertes ROS System. All das lässt sich auf die Geschlechtshormone zurückführen. Die Geschlechts-assoziierte Überaktivität begleitet mit einem erhöhten oxidativen Stress könnte der entscheidende Faktor für den schnellen Verlust der Nierenfunktion bei PKD in männlichen Tiere sein.

3 Zusammenfassung

Die Mikroarray-Technologie hat einen breiten Anwendungsbereich, von der Krebsdiagnostik bis hin zur Identifizierung von Virulenzfaktoren. In diesem Beitrag ist nur ein Beispiel gezeigt. Unser System ist jedoch für alle Anwendungsgebiete einsetzbar. In der Referenz sind die Publikationen aufgelistet [3-8], in denen unser System im Einsatz war. Sie zeigen ein breites Spektrum an Anwendungen.

Literatur

[1] Dai, M., et al., Evolving gene/transcript definitions significantly alter the interpretation of GeneChip data. Nucleic Acids Res, 2005. 33(20): p. e175.

[2] Manoli, T., et al., Group testing for pathway analysis improves comparability of different microarray datasets. Bioinformatics, 2006. 22(20): p. 2500-6.

[3] Li, L., et al., Gender-related over-activity accompanied by increased oxidative stress particularly in puberty as a key factor for the accelerated aging in males. BMC Genomics, 2006. [submitted]

[4] Frank, O., et al., Gene expression signature of primary imatinib-resistant chronic myeloid leukemia patients. Leukemia, 2006. 20(8): p. 1400-7.

[5] Fruehauf, S., et al., The CXCR4 antagonist AMD3100 releases a subset of G-CSF-primed peripheral blood progenitor cells with specific gene expression characteristics. Exp Hematol, 2006. 34(8): p. 1052-9.

[6] Gassler, N., et al., Molecular characterisation of non-absorptive and absorptive enterocytes in human small intestine. Gut, 2006. 55(8): p. 1084-9.

[7] Maier, P., et al., Overexpression of MDR1 using a retroviral vector differentially regulates genes involved in detoxification and apoptosis and confers radioprotection. Radiat Res, 2006. 166(3): p. 463-73.

[8] Zheng, C., et al., Gene expression profiling of CD34+ cells identifies a molecular signature of chronic myeloid leukemia blast crisis. Leukemia, 2006. 20(6): p. 1028-34.