Bioconductorの概要とその利用方法

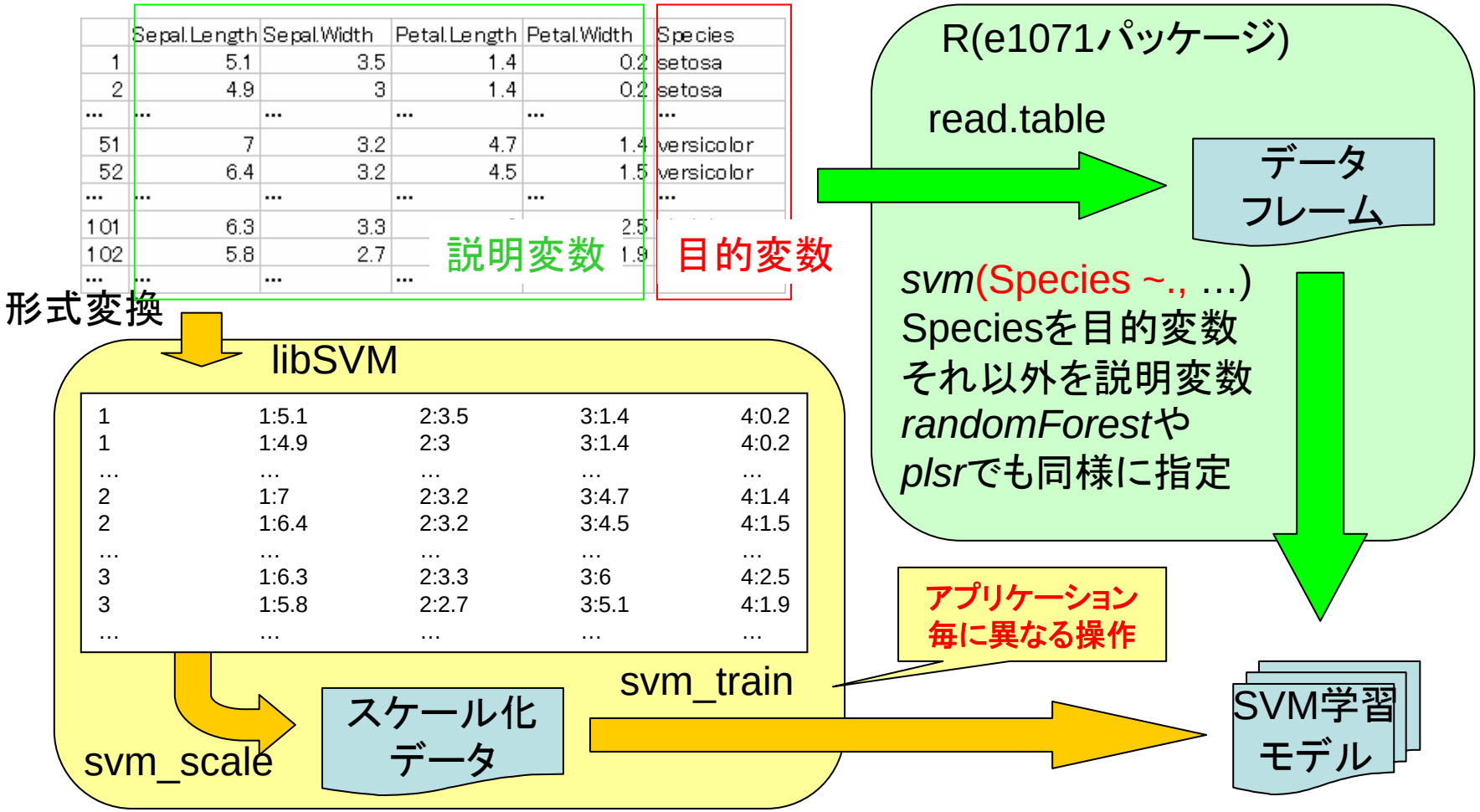
大日本住友製薬株式会社
ゲノム科学研究所
樋口 千洋

Rを使う理由

- さまざまなマイニング手法が実装されている
  - PCA、PLS、SVM、SOM、Random Forests、ICA …
- 操作が簡便で統一的である
  - オリジナルのマイニングアプリケーション等の煩雑さを吸収
  - オブジェクト指向により様々なデータ型のオブジェクトをも統一操作
- Bioconductorが提供されている
  - 様々なゲノム解析手法を網羅している
    - バイオインフォマティクス関係論文の実装例として公開
  - 多くの商用ソフトウェアがRバインディングを提供
- インターネットへのアクセスが容易である
  - RCurlパッケージ、SSOAPパッケージ
- オープンソースかつフリーソフトウェアであること
  - ライセンスのコンプライアンス上有利
- 最近まで気がつかなかったのですが …
  - コンプリーション機能(TABキーで候補一覧を表示)が便利

操作が簡便で統一的

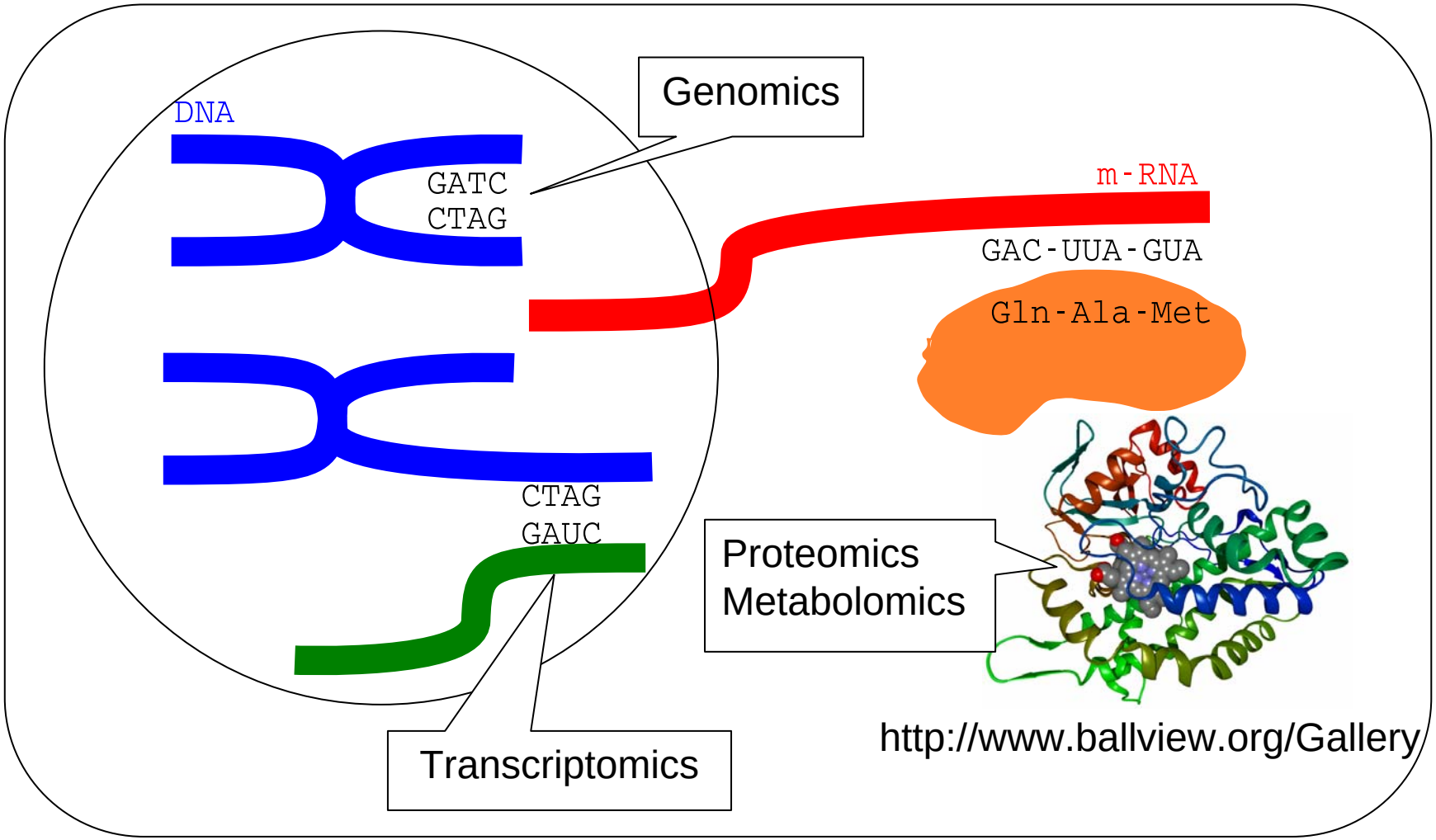
- SVM学習過程をlibSVMとR(e1071パッケージ)で比較



RとBioconductor

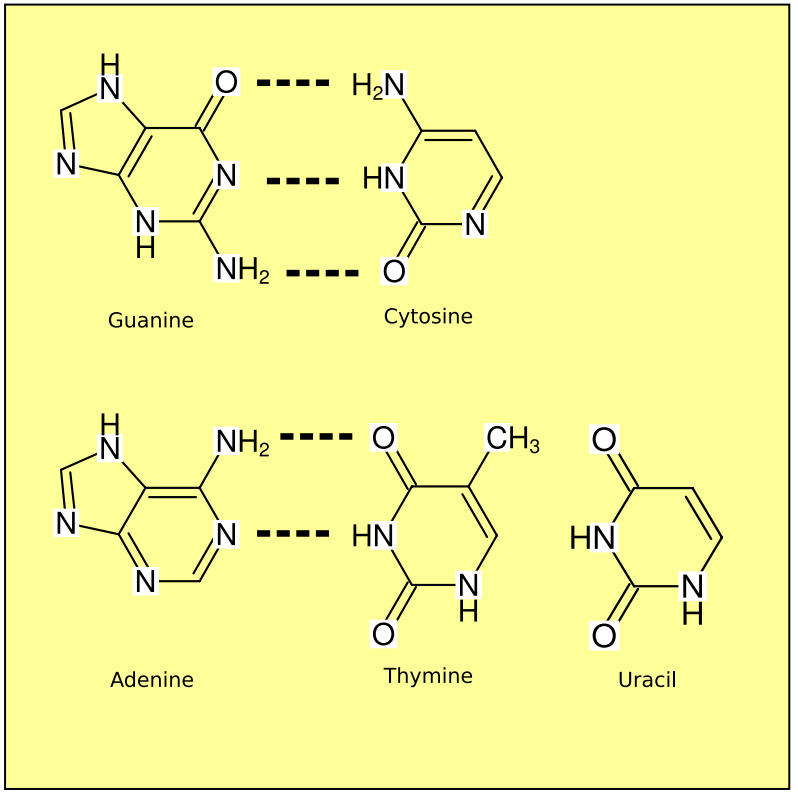
- Bioconductor
  - 遺伝子解析に特化した処理を担当
    - マイクロアレイ解析
    - プロテオーム解析(スペクトル処理)
    - グラフ処理
  - S4クラスオブジェクトが主
  - データマイニングは既存の(cran)パッケージを利用
- オミクス解析で(講演者が)よく使うRパッケージ
  - stats、som、amap、e1071、randomForest、ape、seqinr
  - 研究領域によってはベイズが必須
- Bioconductor「だけ」ではオミクス解析は不可

セントラルドグマと
オミクス(-omics)



DNA (デオキシリボ核酸)

- G、A、T、Cの四種類の塩基で構成される
  - Guanine、Adenine、Thymine、Cytosine
- 二重らせん構造
  - 相補なTとA、CとGで結合
- 相補(complimentaly)
  - TとAおよびCとGが対
- 転写されてRNAを生成
  - 相補な塩基配列の生成



RNA (リボ核酸)

- G、A、U、Cの四種類の塩基で構成される
  - Guanine、Adenine、Uracil、Cytosine
- さまざまな形態
  - mRNA、tRNA、rRNA、shRNA、dsRNA
- エクソンとイントロン

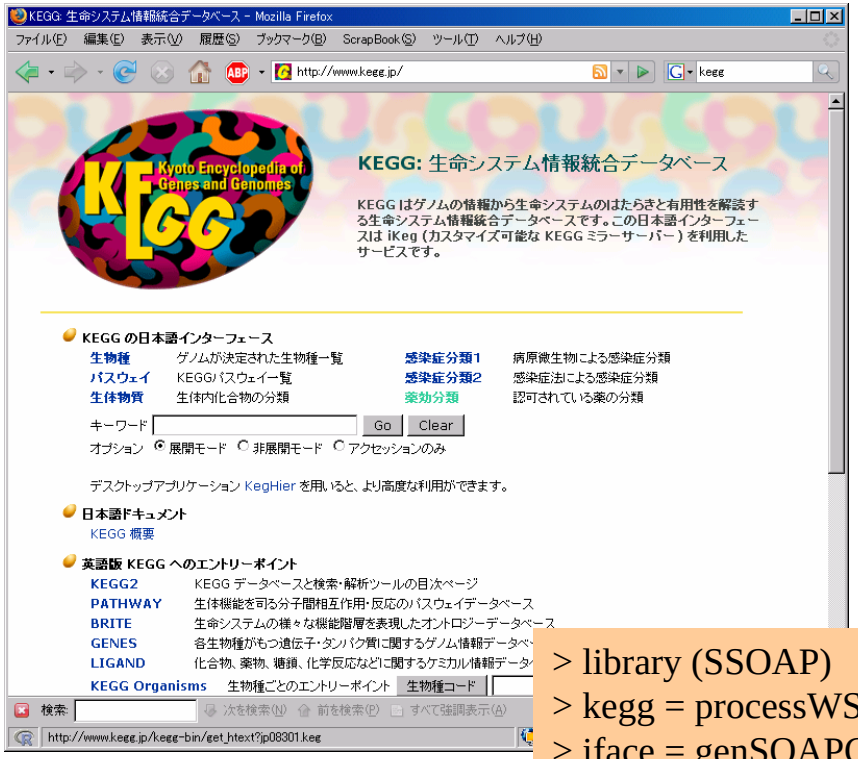
Bioconductor
http://www.bioconductor.org/

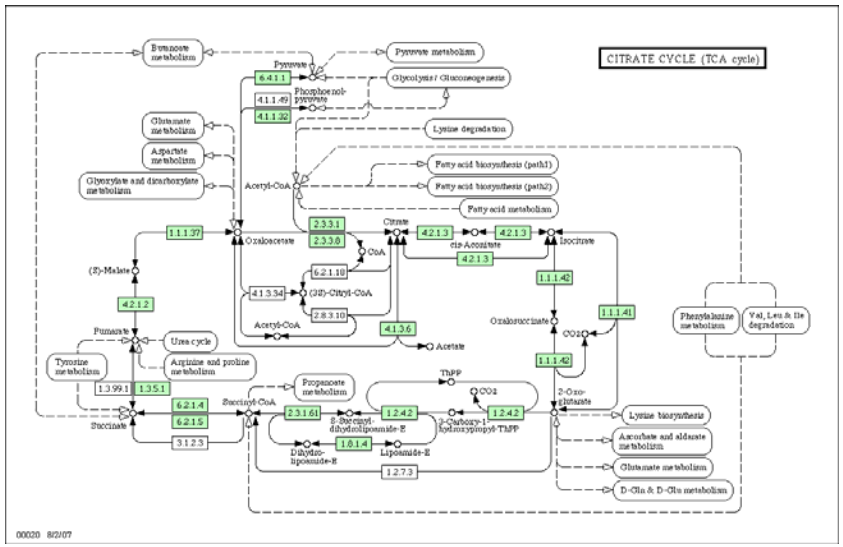
- 現在のバージョンは2.1
  - R 2.6.0ベース(CRAN、Omegahat Projectと共にリポジトリの一つ)
  - ソフトウェア/メタデータ/実験データの三つのカテゴリ
- 守備範囲
  - バイオ系データベース検索
  - マイクロアレイ解析
  - プロテオーム解析
  - グラフ構造による解析
- 注意点
  - バージョン変更による仕様変更を覚悟すること
    - 基本的にクラス構造の変更は対応するメソッドがカバーする
    - 関数そのものやパラメータ仕様の変更は注意するしかない
  - バージョンがあがった場合全パッケージをアップデート
    - 部分的なアップデートのために不整合を催す場合もある

バイオ系データベース検索

- さまざまなデータベースが存在
  - 配列データベース
  - 文献情報データベース
  - 遺伝子発現データベース
  - 代謝経路データベース
  - 相互作用データベース
  - 化合物データベース
- データベースの迅速アクセスが望まれる

SOAPでKEGG APIにアクセス





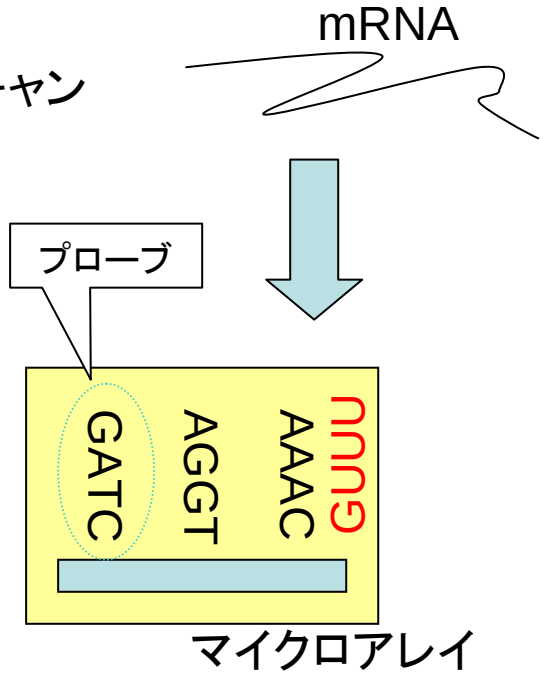
path:hsa00020 ヒトのTCA回路

ヒトTCA回路中の全化合物リストを出力

```
> library (SSOAP)
> kegg = processWSDL("http://soap.genome.jp/KEGG.wsdl")
> iface = genSOAPClientInterface(def = kegg, nameSpaces = "1.2")
> iface@functions$get_compounds_by_pathway("path:hsa00020")
 [1] "cpd:C00010" "cpd:C00011" "cpd:C00022" "cpd:C00024" "cpd:C00026"
 [6] "cpd:C00033" "cpd:C00036" "cpd:C00042" "cpd:C00068" "cpd:C00074"
[11] "cpd:C00091" "cpd:C00122" "cpd:C00149" "cpd:C00158" "cpd:C00311"
[16] "cpd:C00417" "cpd:C05379" "cpd:C05381" "cpd:C15972" "cpd:C15973"
[21] "cpd:C16254"
```

マイクロアレイ解析

- マイクロアレイ
  - 塩基の相補性を利用してmRNAをトラップ
    - RNAを蛍光標識しておいて蛍光イメージをスキャン
      - 蛍光強度からトラップできたRNAを検出
    - プローブ
      - RNAと相補にデザインされた塩基配列
  - (現在では)Affymetrix社のGeneChipが主流
- affyパッケージ
  - GeneChipを扱うためのパッケージ
    - Bioconductorのデフォルトで導入される
  - 基本クラス
    - AffyBatch
      - 蛍光データファイル(CELファイル)を一括格納
    - ExpressionSet(以前はexprSet; Biobaseパッケージから継承)
      - AffyBatchクラスオブジェクトを対象に統計解析した結果を格納



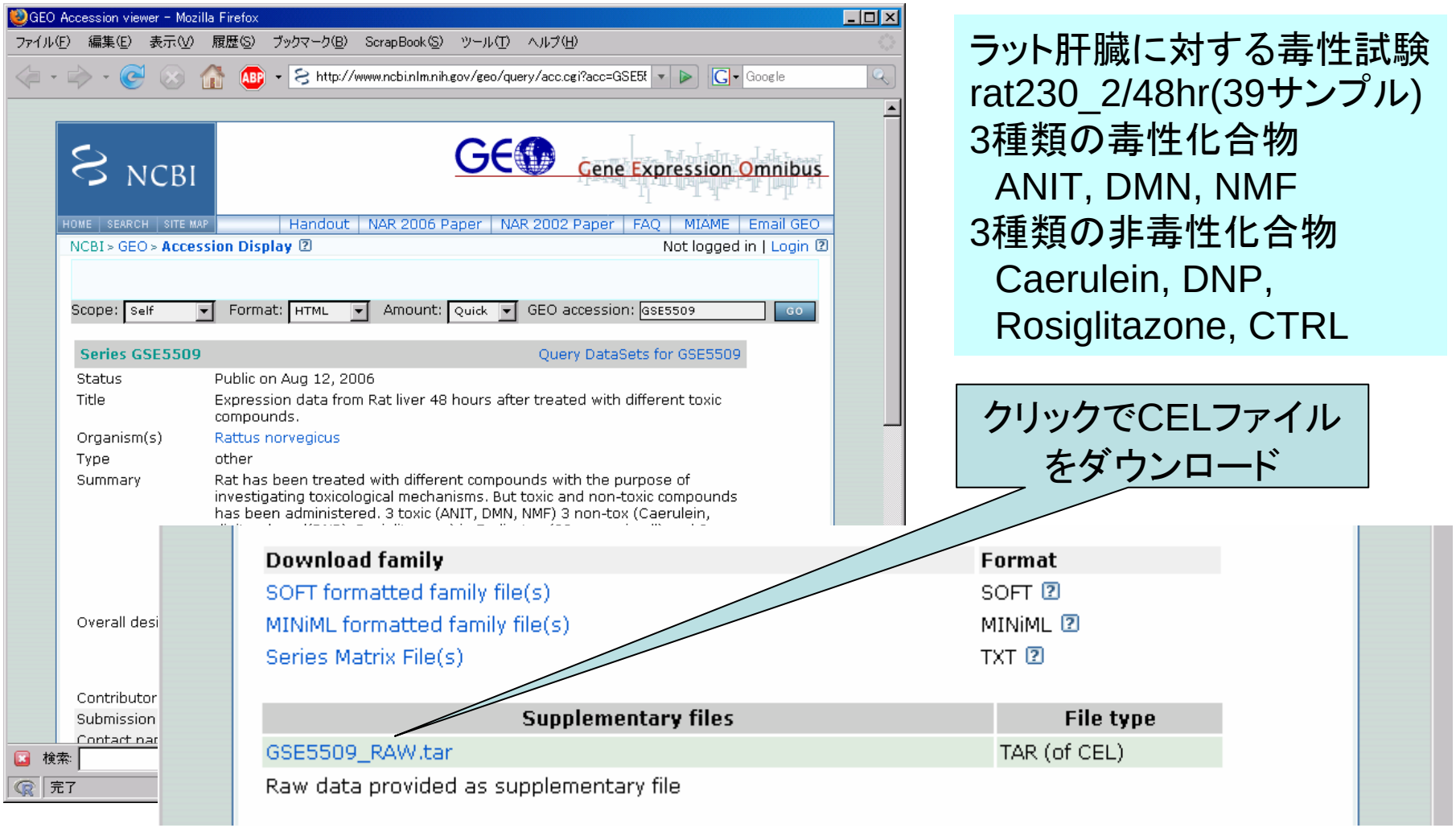
プローブの解析

- GeneChipの蛍光データから発現解析
- Affymetrix社チップに対する統計処理手法
  - MAS5
  - plier
  - dChip
  - RMA
  - GCRMA
  - DFW

AffyBatchクラス

- CELファイルを格納するオブジェクト
  - GeneChipの蛍光強度情報を格納したファイル
- ReadAffy関数で生成
  - 引数なし　　カレントディレクトリ下の全て
    - gz形式でも可能
  - ディレクトリ指定　　celfile.path = /some/where
  - ファイル名指定　　filenames = list (“”, “”…)
- サンプルクラスオブジェクト
  - > data(affybatch.example)　仮想のGeneChip　x3
  - > library(affydata)
  - > data(Dilution)　hgu95av2　x4

GEOでGSE5509を検索



ラット肝臓に対する毒性試験
rat230_2/48hr(39サンプル)
3種類の毒性化合物
　ANIT, DMN, NMF
3種類の非毒性化合物
　Caerulein, DNP,
　Rosiglitazone, CTRL

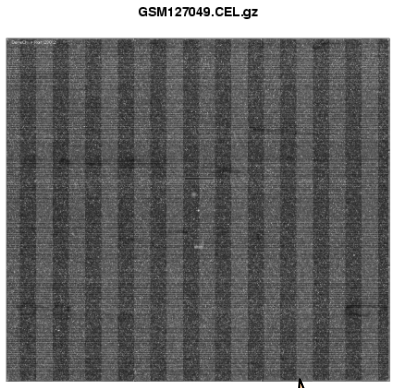
クリックでCELファイル
をダウンロード

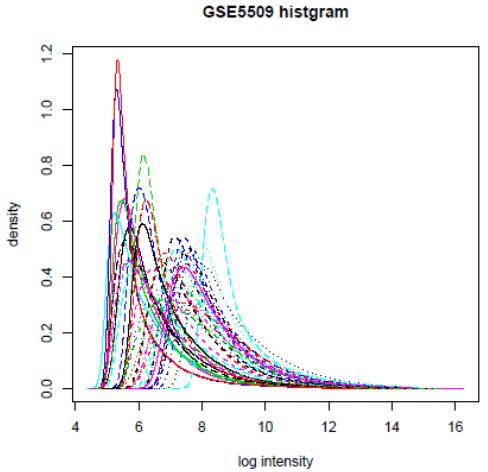
GEOからCELファイルを取得

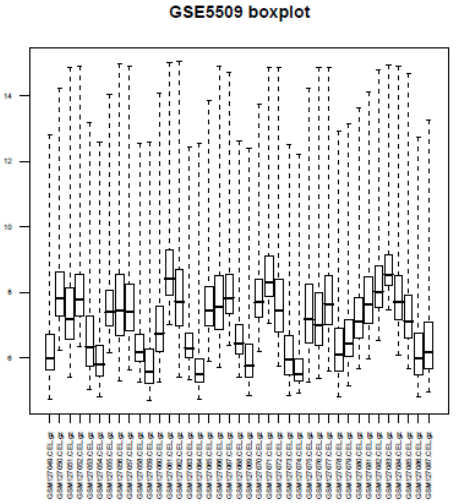
```
$ mkdir geo
$ cd geo
lftp ftp.ncbi.nlm.nih.gov:/pub/geo/DATA/supplementary/series/GSE5509>
-r--r--r-- 1 ftp anonymous 169943040 Aug 14 2006 GSE5509_RAW.tar
-r--r--r-- 1 ftp anonymous 2198 Jun 1 09:34 filelist.txt
lftp (途中省略)/series/GSE5509> get GSE5509_RAW.tar
lftp (途中省略)/series/GSE5509> quit
$ mkdir CELfiles
$ cd CELfiles
$ tar xf ../GSE5509_RAW.tar
$ ls
GSM127049.CEL.gz GSM127059.CEL.gz GSM127069.CEL.gz GSM127079.CEL.gz
GSM127050.CEL.gz GSM127060.CEL.gz GSM127070.CEL.gz GSM127080.CEL.gz
GSM127051.CEL.gz GSM127061.CEL.gz GSM127071.CEL.gz GSM127081.CEL.gz
GSM127052.CEL.gz GSM127062.CEL.gz GSM127072.CEL.gz GSM127082.CEL.gz
GSM127053.CEL.gz GSM127063.CEL.gz GSM127073.CEL.gz GSM127083.CEL.gz
GSM127054.CEL.gz GSM127064.CEL.gz GSM127074.CEL.gz GSM127084.CEL.gz
GSM127055.CEL.gz GSM127065.CEL.gz GSM127075.CEL.gz GSM127085.CEL.gz
GSM127056.CEL.gz GSM127066.CEL.gz GSM127076.CEL.gz GSM127086.CEL.gz
GSM127057.CEL.gz GSM127067.CEL.gz GSM127077.CEL.gz
```

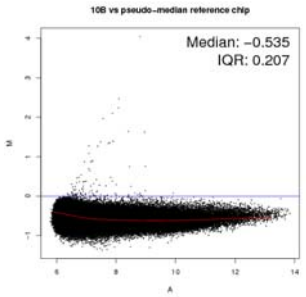
CELファイルから AffyBatchクラスを作成

```
$ cd geo/CELfiles
$ sudo R
> source("http://bioconductor.org/biocLite.R")
> biocLite()
> biocLite(c("gcrma", "plier"))
> library(affy)
> library(gcrma)
> library(plier)
> GSE5509 <- ReadAffy()
> class(GSE5509)
[1] "AffyBatch"
attr(,"package")
[1] "affy"
> hist(GSE5509, main="GSE5509 histgram")
> boxplot(GSE5509, las=2, cex.axis=0.5,
+  main="GSE5509 boxplot")
> image(GSE5509)
> MAplot(GSE5509)
```







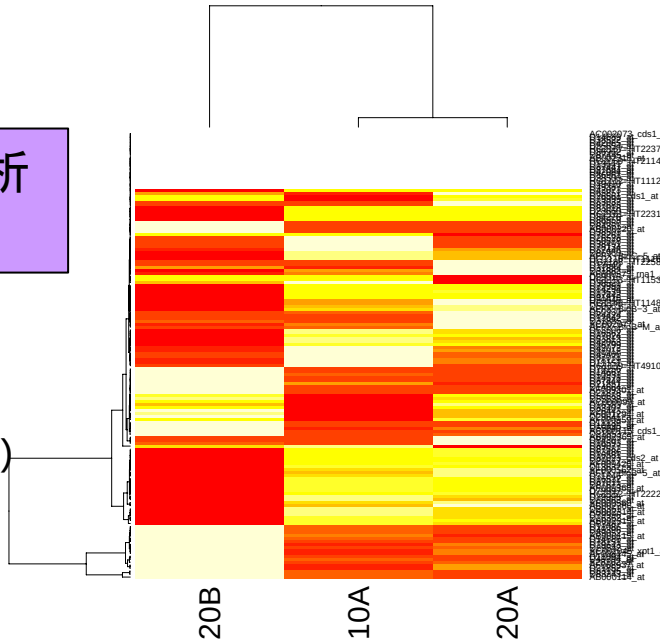


CELファイルの情報からGeneChipの蛍光イメージを再現

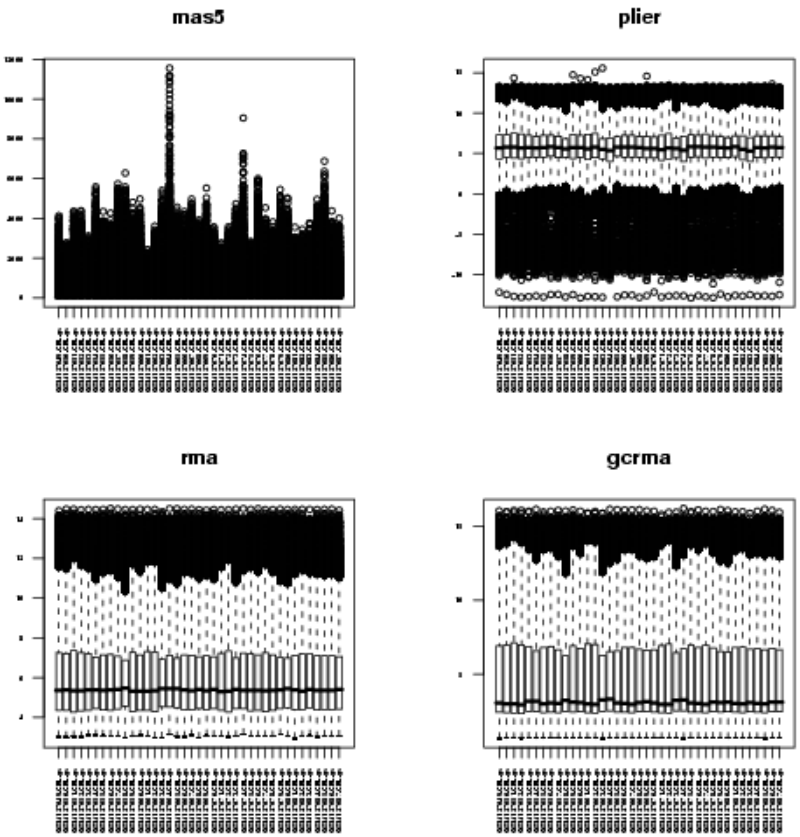
プローブレベルの発現解析処理 (MAS5、plier、RMA、dChip、GCRMA)

```
> eset.mas5 <- mas5(GSE5509, sc=50)
> class(eset.mas5)
[1] "ExpressionSet"
attr(,"package")
[1] "Biobase"
> expr.mas5 <- exprs(eset.mas5)
> dim(expr.mas5)
[1] 31099 39
> eset.plier <- justPlier(GSE5509, normalize=T)
> expr.plier <- exprs(eset.plier)
> eset.rma <- rma(GSE5509)
> expr.rma <- exprs(eset.rma)
> ai <- compute.affinities(cdfName(GSE5509))
> eset.gcrma <- gcrma(GSE5509, affinity.info=ai, type="affinities")
> expr.gcrma <- exprs(eset.gcrma)
> eset.dChip <- expresso(GSE5509, normalize.method="invariantset",
+ bg.correct=FALSE, pmcorrect.method="pmonly", summary.method="liwong")
> expr.dChip <- exprs(eset.dChip)
> par(mfrow=c(4,4))
> boxplot(as.data.frame(expr.rma), las=2, cex.axis=0.5, main="")
```

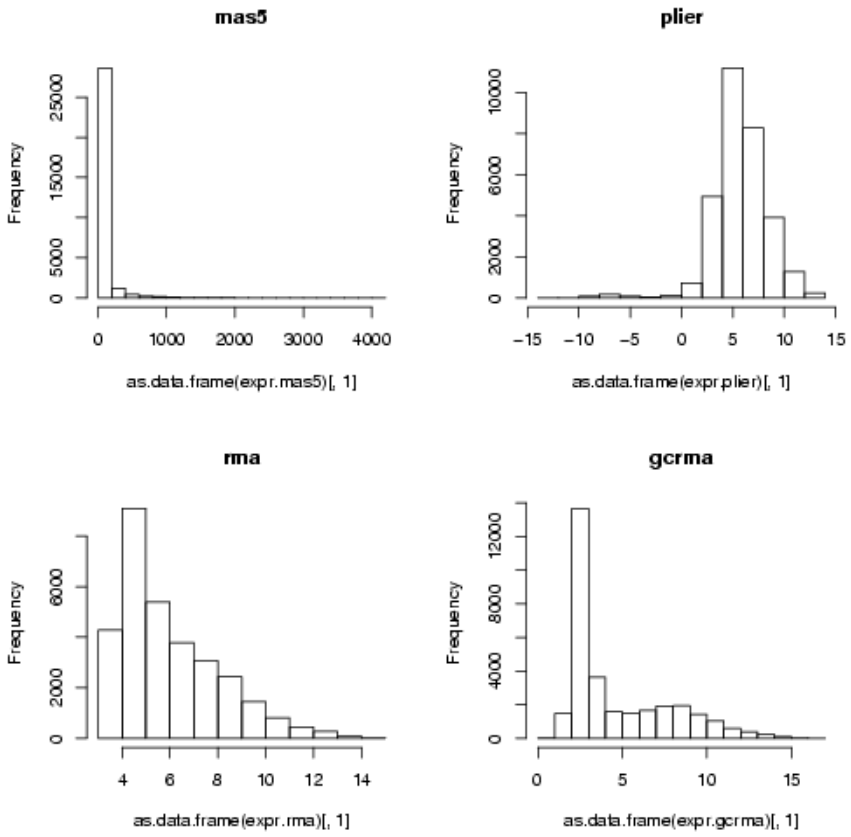
この結果からさらに統計解析
およびマイニングを実施



プローブ処理方式による結果の比較



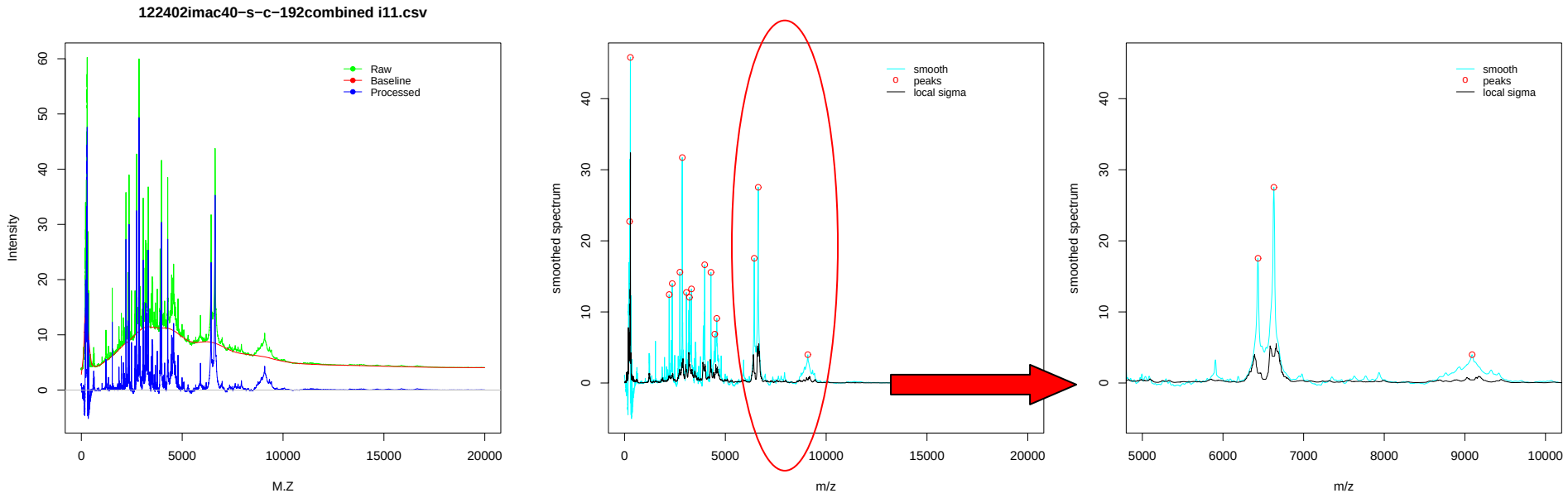
39チップ分のboxplotを横並びに配置



左1チップのhistgramを表示

プロテオーム解析

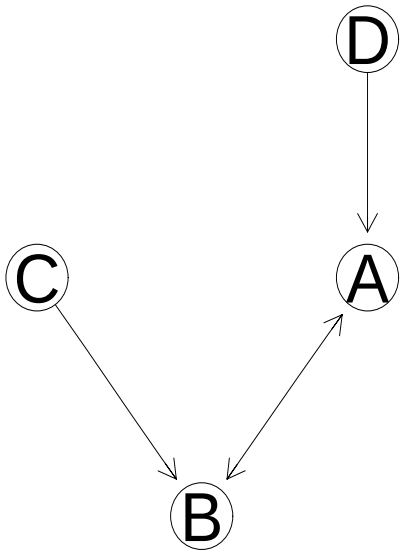
- MS波形の処理
  - SELDI-TOF-MSでのノイズ処理やピーク検出
  - LC-MSディファレンシャル解析
    - アラインメント機能の提供
    - netCDFが必要



グラフ構造による解析

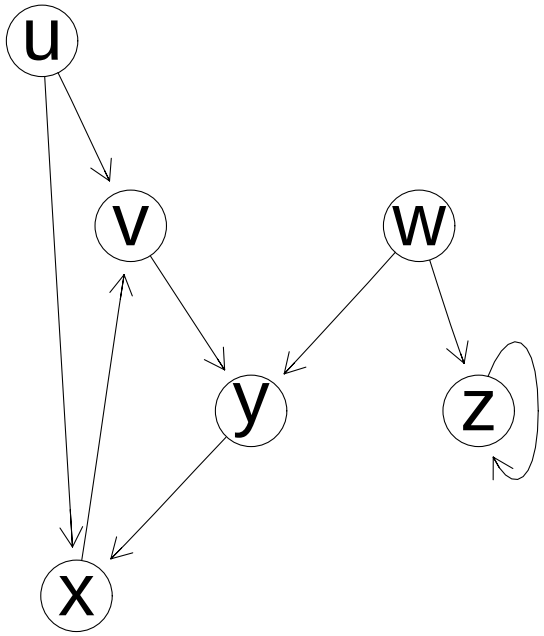
- グラフ構造の有用性
  - データ管理はリレーショナルデータベースが主流
    - 遺伝子やタンパクの発現プロファイル
    - 文献や遺伝子配列データベース
  - ツリー構造(グラフ)データも少なくない
    - 代謝ネットワーク
    - 相互作用ネットワーク
- Rが提供するグラフ構造処理パッケージ
  - graph　　基本的なクラスとメソッドを提供
  - RBGL　　グラフ探索アルゴリズムを提供
  - Rgraphviz　　可視化ツールGraphvizを利用

グラフの作成とオブジェクトの内容




```
> library(Rgraphviz)
> V <- LETTERS[1:4]
> edL2 <- vector("list", length=4)
> names(edL2) <- V
> for(i in 1:4)
+   edL2[[i]] <- list(edges=c(2,1,2,1)[i],
+  weights=sqrt(i))
> gR2 <- new("graphNEL", nodes=V, edgeL=edL2,
+  edgemode="directed")
> plot(gR2)
> class(gR2)
[1] "graphNEL"
attr(,"package")
[1] "graph"
> slotNames(gR2)
[1] "nodes"    "edgeL"    "edgemode" "edgeData"
"nodeData"
> gR2@nodes
[1] "A" "B" "C" "D"
> unlist(gR2@edgeL)
A.edges B.edges C.edges D.edges
      2       1       2       1
> gR2@edgemode
[1] "directed"
```

XMLでグラフを記述

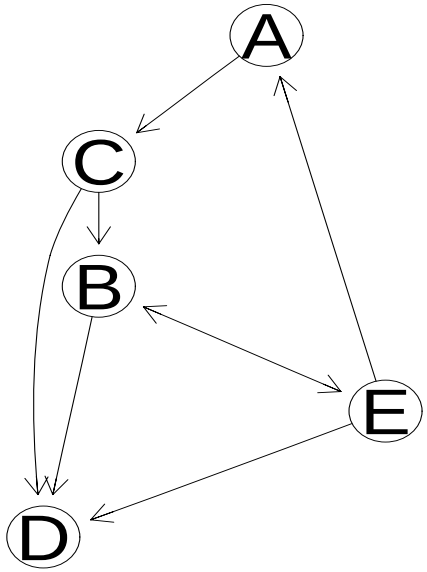



```
> library(Rgraphviz)
> df <- fromGXL(file(system.file
+         (“XML/dfsex.gxl”
+          package=“RBGL”)))
> plot(df)
```

```
<gxl>
 <graph id="G" edgemode="directed">
  <node id="u"/>
  <node id="v"/>
  <node id="w"/>
  <node id="x"/>
  <node id="y"/>
  <node id="z"/>
  <edge id="e1" from="u" to="v"/>
  <edge id="e2" from="u" to="x"/>
  <edge id="e3" from="v" to="y"/>
  <edge id="e4" from="w" to="y"/>
  <edge id="e5" from="w" to="z"/>
  <edge id="e6" from="x" to="v"/>
  <edge id="e7" from="y" to="x"/>
  <edge id="e8" from="z" to="z"/>
 </graph>
</gxl>
```

RBGL(R I/F Boost Graph Library)

- グラフ探索アルゴリズム集
  - BGL由来
    - Depth First Search
    - Breadth first Search
    - Shortest paths
    - Minimum spanning tree 他
  - BGLを利用して実装
    - Min-Cut
    - highlyConnSG
  - RBGLと無関係のアルゴリズム
    - maxClique
    - separates
    - kCores
    - kCliques

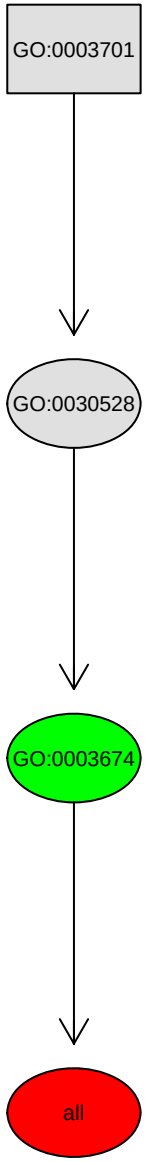



```
> library(RBGL)
> km <- fromGXL(file
+  (system.file("XML/kmstEx.gxl",
+  package="RBGL")))
> separates("B", "A", "E", km)
[1] TRUE
> separates("B", "A", "C", km)
[1] FALSE
```

Gene Ontologyとグラフ処理

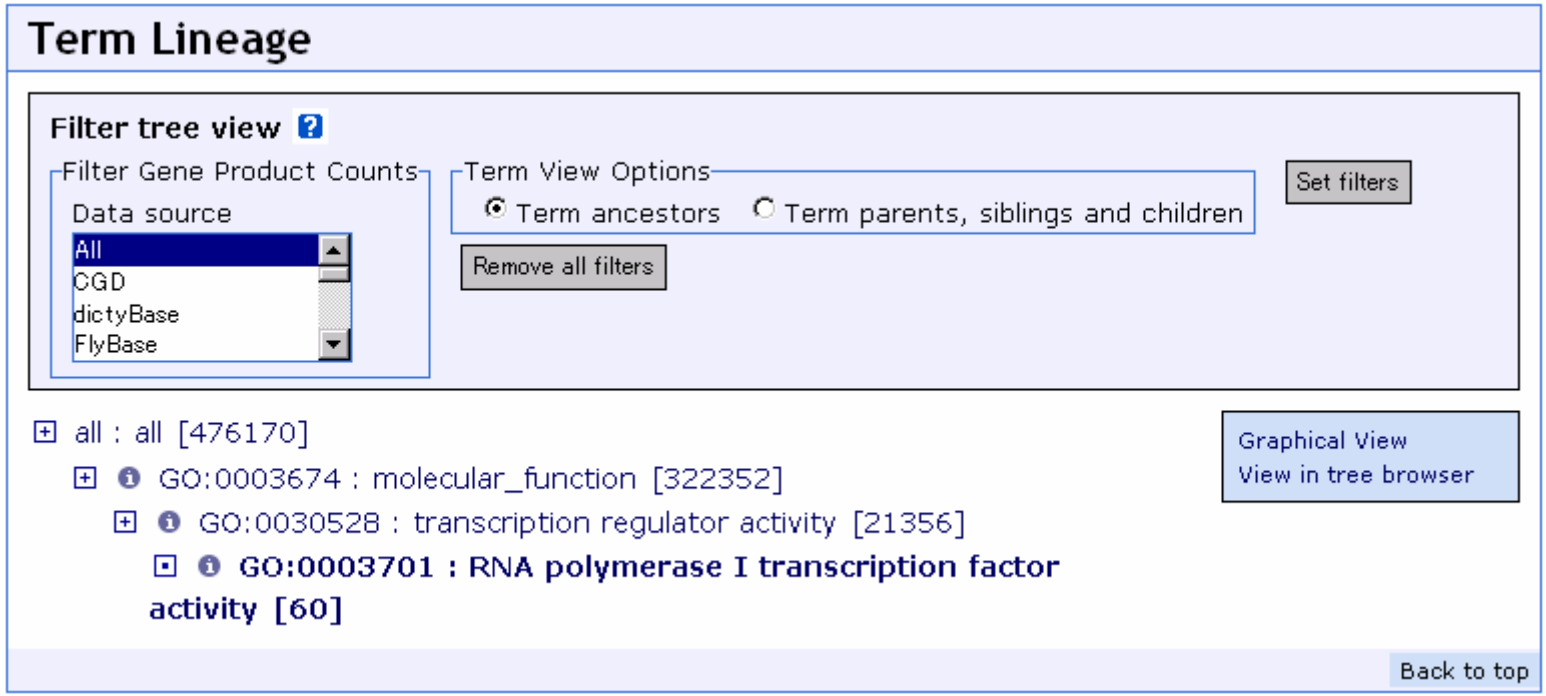
- Gene Ontology
  - 遺伝子の概念辞書
    - Molecular Function/Biological Process/Cellular component
  - セマンティックWeb
- GOパッケージ
  - 環境で与えられる
- GOstatsパッケージ
  - GOパッケージのデータからグラフを作成

```
> as.list(GOMFCHILDREN)$“GO:0003674” # (Molecular Functionの下の概念の一覧)
         isa          isa          isa          isa          isa          isa
"GO:0000332" "GO:0003774" "GO:0003824" "GO:0004871" "GO:0005198" "GO:0005215"
         isa          isa          isa          isa          isa          isa
"GO:0005488" "GO:0015457" "GO:0016209" "GO:0030188" "GO:0030234" "GO:0030528"
         isa          isa          isa          isa          isa          isa
"GO:0030533" "GO:0031386" "GO:0031992" "GO:0042056" "GO:0045182" "GO:0045499"
         isa
"GO:0045735"
```

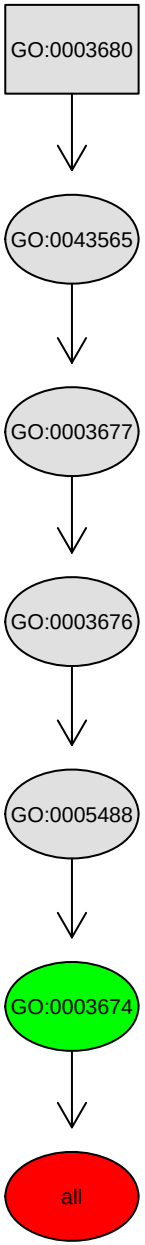


GO:0003701から上流を作成

```
> library(GOstats)
> library(Rgraphviz)
> g1 <- oneGOgraph("GO:0003701", GOMFPARENTS)
> plot(g1)
```

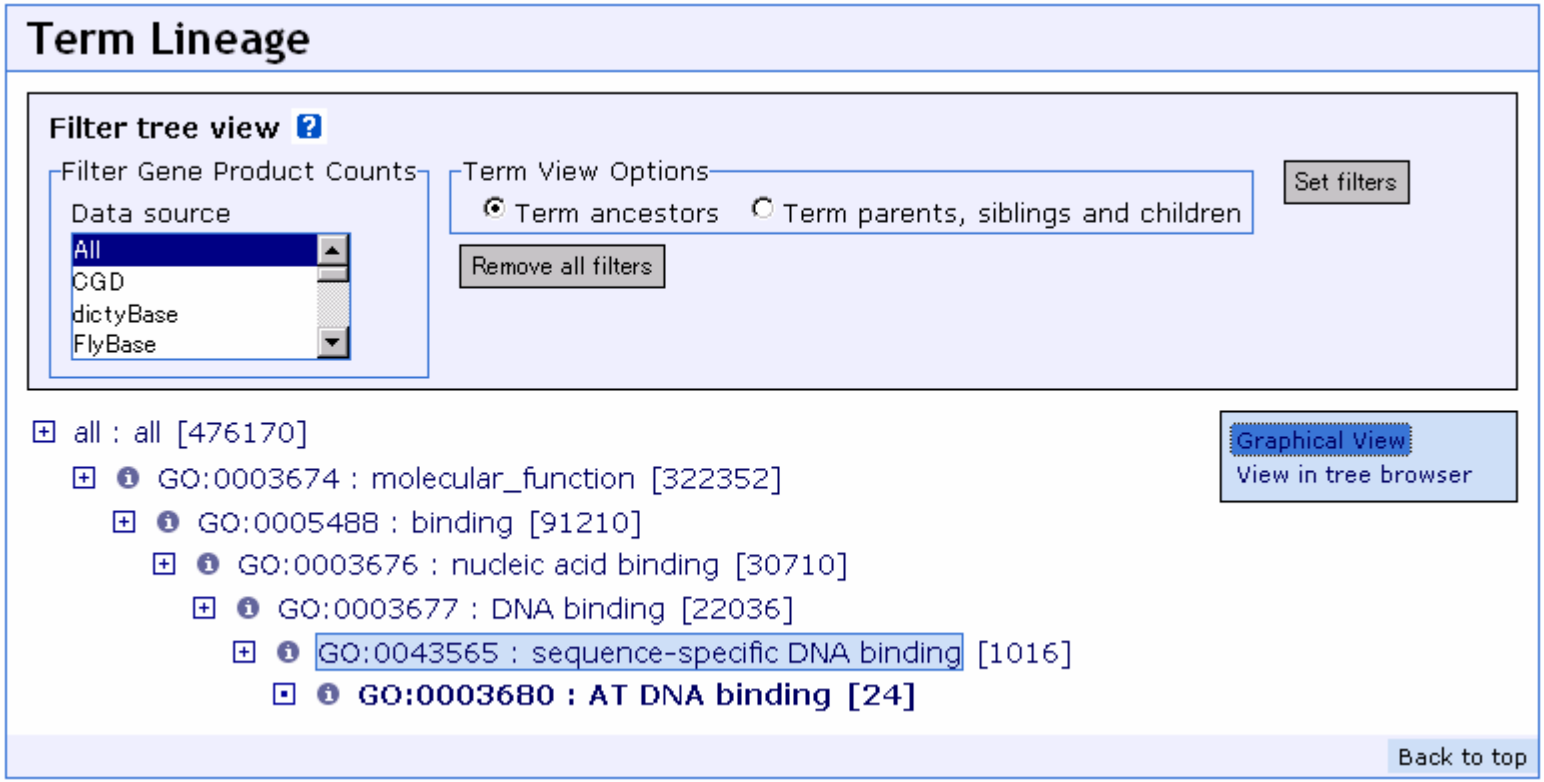


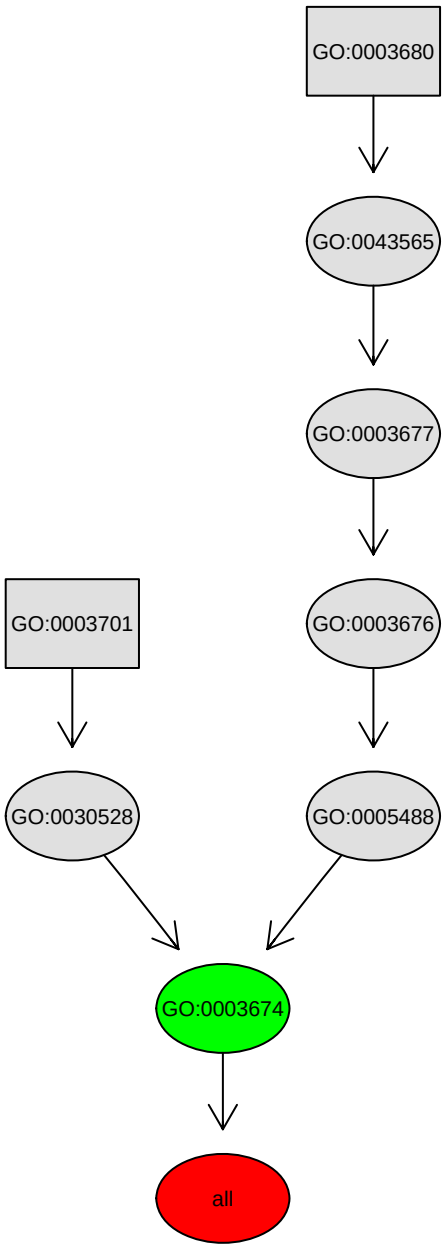
Amigo http://amigo.geneontology.org/cgi-bin/amigo/go.cgi



GO:0003680から上流を作成

```
> g2 <- oneGOgraph("GO:0003680", GOMFPARENTS)
> plot(g2)
```





2つのグラフを結合

```
> g3 <- join(g1, g2)
> plot(g3)
```

- **all : all [476170]**
  - GO:0008150 : biological_process [317190]
  - GO:0005575 : cellular_component [321578]
  - **GO:0003674 : molecular_function [322352]**
    - GO:0016209 : antioxidant activity [1162]
    - GO:0015457 : auxiliary transport protein activity [334]
    - GO:0005488 : binding [91210]
    - GO:0003824 : catalytic activity [105674]
    - GO:0030188 : chaperone regulator activity [144]
    - GO:0042056 : chemoattractant activity [26]
    - GO:0045499 : chemorepellent activity [16]
    - GO:0030234 : enzyme regulator activity [4866]
    - GO:0016530 : metallochaperone activity [82]
    - GO:0060089 : molecular transducer activity [16616]
    - GO:0003774 : motor activity [1144]
    - GO:0045735 : nutrient reservoir activity [102]
    - GO:0031386 : protein tag [36]
    - GO:0005198 : structural molecule activity [9152]
    - GO:0000332 : template for synthesis of G-rich strand of telomere DNA activity [8]
    - GO:0030528 : transcription regulator activity [21356]
    - GO:0045182 : translation regulator activity [1856]

Bioconductor参考書

謝辞

- ゲノム科学研究所ゲノミクス研究部
- ゲノム科学研究所構造生物研究部