Kollekolle 2003

- Day 3, Session 2
- Clustering, Graph II, Standards
- VJ Carey <stvjc@channing.harvard.edu>
Iyer517

- Iyer517 is a package providing information on a 1999 Science paper from the Eisen lab investigating the transcriptional response of human fibroblasts to exposure to serum.
- used cDNA arrays.
- they published a clustering of the longitudinally measured expression.
- made a portion of the data available as a web resource.
Iyer 517 eset

> show(Iyer517)

Expression Set (exprSet) with
517 genes
19 samples

phenoData object with 2 variables and 19 cases

varLabels

time.hrs: time, NA=Unsync
cycloheximide: cycloheximide present

• The IyerAnnotated data frame gives additional annotation for each gene, and the clustering label from the Iyer analysis

> data(IyerAnnotated)
> print(table(ic <- IyerAnnotated$Iclust))

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100</td>
<td>145</td>
<td>34</td>
<td>43</td>
<td>7</td>
<td>34</td>
<td>15</td>
<td>63</td>
<td>19</td>
<td>25</td>
<td>32</td>
</tr>
</tbody>
</table>
A few of Iyer’s clusters over time (raw)
A heatmap of a subset
A heatmap of a subset
disallow clustering by time
Issues in clustering

- a traditional exploratory technique (unsupervised learning)
- given \( p \) features on \( N \) objects, we typically aim to assign a number \( 1, \ldots, K \) to each object indicating its cluster membership
- for most algorithms \( K \) must be specified in advance
- the measure of distance between objects must be specified
- it may not be necessary to use all \( p \) features, or features may be combined or transformed
Heatmaps

• The common heatmaps are a juxtaposition of
  • a color rendering of relative magnitudes of element values
  • dendrograms derived from agglomerative hierarchical clustering by row and columns
• it is important to note that dendrograms
  • impose rather than reveal structure in the data (viz., the rearrangement of times in Iyer data)
  • do not dictate a partitioning until a cut is made
Hierarchical methods

- Hierarchical clustering methods produce a tree or dendrogram.
- They avoid specifying how many clusters are appropriate by providing a partition for each $K$ obtained from cutting the tree at some level.
- The tree can be built in two distinct ways
  - bottom–up: agglomerative clustering;
  - top–down: divisive clustering.
Agglomerative methods

- Start with $n$ mRNA sample (or $G$ gene) clusters.
- At each step, merge the two closest clusters using a measure of between-cluster dissimilarity which reflects the shape of the clusters.
- Between-cluster dissimilarity measures:
  - *Unweighted Pair Group Method with Arithmetic mean (UPGMA)*: average of pairwise dissimilarities;
  - *Single-link*: minimum of pairwise dissimilarities;
  - *Complete-link*: maximum of pairwise dissimilarities.
Divisive methods

- Start with only one cluster.
- At each step, split clusters into two parts.
- Advantages: Obtain the main structure of the data, i.e., focus on upper levels of dendrogram.
- Disadvantages: Computational difficulties when considering all possible divisions into two groups.
- Examples
  - Self–Organizing Tree Algorithm – SOTA (Dopazo & Carazo, 1997);
Partitioning methods

• Partition the data into a **prespecified** number $K$ of mutually exclusive and exhaustive groups.

• Iteratively reallocate the observations to clusters until some criterion is met, e.g. minimize within–cluster sums–of–squares.

• Examples:
  - $k$–means; fuzzy $k$–means;
  - Partitioning Around Medoids – PAM (Kaufman & Rousseeuw, 1990);
  - Self–Organizing Maps – SOM (Kohonen, 2001);
Partitioning around medoids

Partitioning around medoids or PAM of Kaufman and Rousseeuw (1990) is a partitioning methods which operates on a dissimilarity matrix, e.g. Euclidean distance matrix.

For a prespecified number of clusters $K$, the PAM procedure is based on the search for $K$ representative objects, or medoids, among the observations to be clustered.

After finding a set of $K$ medoids, $K$ clusters are constructed by assigning each observation to the nearest medoid.
Partitioning around medoids

The goal is to find $K$ medoids, $\mathbf{M}^* = (m_1^*, \ldots, m_K^*)$, which minimize the sum of the dissimilarities of the observations to their closest medoid, that is,

$$\mathbf{M}^* = \underset{\mathbf{M}^*}{\text{argmin}} \sum_{i} \min_{k} d(x_i, m_k).$$

Tends to be more robust than $k$–means.
Silhouette plots

PAM provides a graphical display, the silhouette plot, which can be used to: (i) select the number of clusters and (ii) assess how well individual observations are clustered. The silhouette width of observation $i$ is defined as

$$
\text{sil}_i = \frac{(b_i - a_i)}{\max(a_i, b_i)},
$$

where $a_i$ denotes the average dissimilarity between $i$ and all other observations in the cluster to which $i$ belongs, and $b_i$ denotes the minimum average dissimilarity of $i$ to objects in other clusters. Intuitively, objects with large silhouette width $\text{sil}_i$ are well–clustered, those with small $\text{sil}_i$ tend to lie between clusters.
K-means

- classical iterative approach
- given $K$, establish clusters by requiring that elements of group $g$ are closer to the mean value of group $g$ than to the mean of any other group
- result can depend on initial assortment
### Comparing two clusterings

```r
> set.seed(1234)
> km1 <- kmeans(log(ie), 11)
> print(table(IC, km1$clust))

<table>
<thead>
<tr>
<th>IC</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>28</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>51</td>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>61</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>17</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>31</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>0</td>
<td>0</td>
<td>21</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>7</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>0</td>
<td>27</td>
<td>1</td>
<td>1</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>26</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>7</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>14</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
```

Comments

• k-means seems to discover additional structure in the group that Iyer et al called ‘A’
• let’s check visually
Iyer A vs K-means

Iyer A

KM 1

KM 7

KM 8
PAM, k=11

clusplot(pam(x = dLIE, k = 11))

These two components explain 76.7% of the point variability.
## Comparison: PAM/Iyer

### pam

<table>
<thead>
<tr>
<th>Iyer</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>40</td>
<td>2</td>
<td>25</td>
<td>33</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>9</td>
<td>43</td>
<td>0</td>
<td>93</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>8</td>
<td>1</td>
<td>23</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>13</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>19</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>H</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>19</td>
<td>16</td>
<td>4</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>9</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>15</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>N</td>
<td>3</td>
<td>12</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
## Comparison: PAM/K-means

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>km</strong></td>
<td><strong>pam</strong></td>
<td><strong>pam</strong></td>
<td><strong>pam</strong></td>
<td><strong>pam</strong></td>
<td><strong>pam</strong></td>
<td><strong>pam</strong></td>
<td><strong>pam</strong></td>
<td><strong>pam</strong></td>
<td><strong>pam</strong></td>
<td><strong>pam</strong></td>
<td><strong>pam</strong></td>
</tr>
<tr>
<td>1</td>
<td>34</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>32</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>19</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>66</td>
<td>0</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>19</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>1</td>
<td>43</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>19</td>
<td>0</td>
<td>0</td>
<td>84</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>0</td>
<td>23</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>22</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Silhouette plot

Silhouette plot of pam(x = LIE, k = 11)

n = 517

11 clusters $C_j$

j : $n_j | \text{ave}_{i \in C_j} s_i$

1 : 60 | 0.083
2 : 73 | 0.15
3 : 55 | 0.18
4 : 132 | 0.23
5 : 23 | 0.16
6 : 36 | 0.054
7 : 19 | 0.38
8 : 41 | 0.14
9 : 44 | 0.17
10 : 6 | 0.26
11 : 28 | 0.18

Average silhouette width : 0.17
Summary of clustering

- wide diversity of algorithms
- diagnostics and measures of clustering strength are open topics
- primarily hypothesis-generating
Graph structures II

- in the discussion of ontologies, we saw a role for a DAG structure represented as a sparse matrix
- the graph package provides for a variety of native representations of network structures in R

```r
> getSlots("graph")

  edgemode
  "character"

> getClass("graphNEL")

Slots:

  Name: nodes edgeL edgemode
  Class: vector list character

  Extends: "graph"
```
pathway
manual construction in R

```r
> myNodes <- c("a", "b", "c", "d", "e")
> myEdgeL <- list(a = list(edges = c(2, 4)), b = list(edges = 1),
+                  c = list(edges = c(3, 4, 5)), d = list(edges = c(2, 3)),
+                  e = list(edges = character(0)))
> g <- new("graphNEL", nodes = myNodes, edgeL = myEdgeL, edgemode = "directed")
> show(g)

A graph with directed edges
Number of Nodes = 5
Number of Edges = 8

On linux you can use Rgraphviz to render g on an R graphics device. On windows we export to GXL, then convert to dot.

> cat(saveXML(toGXL(g)), file = "myg.gxl")
```
the gxl embodiment

```xml
<?xml version="1.0"?>
<gxl>
    <graph id="graphNEL" edgemode="directed">
        <node id="a"/>
        <node id="b"/>
        <node id="c"/>
        <node id="d"/>
        <node id="e"/>
        <edge id="e1" from="a" to="b"/>
        <edge id="e2" from="a" to="d"/>
        <edge id="e3" from="b" to="a"/>
        <edge id="e4" from="c" to="c"/>
        <edge id="e5" from="c" to="d"/>
        <edge id="e6" from="c" to="e"/>
        <edge id="e7" from="d" to="b"/>
        <edge id="e8" from="d" to="c"/>
    </graph>
</gxl>
```
dot’s ps rendering

- `gxl2dot myg.gxl | dot -Tps > myg.ps`
the dot embodiment of pathway

digraph G {
"glu-6-p" -> "fruc-6-p";
"fruc-6-p" -> "fruc-1,6-p";
"fruc-1,6-p" -> "glycerone-p";
"fruc-1,6-p" -> "glyceraldehyde-3p";
"glyceraldehyde-3p" -> pyruvate;
pyruvate -> "acetyl-CoA";
"acetyl-CoA" -> citrate;
citrate -> "[aconitate]";
"[aconitate]" -> isocitrate;
isocitrate -> "[oxalo-succinate]";
isocitrate -> succinate;
...
icocitrate -> glyoxylate;
}

RBGL

- the Boost Graph Library is a collection of graph theoretic structures and algorithms deployed in a ‘portable STL’ C++ system Boost
- we have partially implemented the library as a set of high-level functions on graph library objects
- tsort, sp.between, mstree.kruskal, connectedComp, edgeConnectivity are examples
edge connectivity of network

```r
> gly <- fromGXL(file("glytca.gxl"))
> ugly <- ugraph(gly)
> show(ugly)

A graph with undirected edges
Number of Nodes = 19
Number of Edges = 20

> print(ec <- edgeConnectivity(ugly))

$connectivity
[1] 1

$minDisconSet
$minDisconSet[[1]]
[1] "glu-6-p"  "fruc-6-p"
```
\textbf{disconnection}

\begin{verbatim}
> sg <- removeEdge("acetyl-CoA", "citrate", gly)
> print(cc <- connectedComp(sg))

"1"
[1] "glu-6-p"       "fruc-6-p"       "fruc-1,6-p"
[4] "glycerone-p"   "glyceraldehyde-3p" "pyruvate"
[7] "acetyl-CoA"

"2"
[1] "citrate"       "[aconitate]"   "isocitrate"
[4] "[oxalo-succinate]" "succinate"    "oxaloacetate"
[7] "2-oxo-glutarate" "fumarate"     "succinyl-CoA"
[10] "malate"       "glyoxylate"  "icocitrate"

> show(subGraph(cc[[2]], gly))

A graph with undirected edges
Number of Nodes = 12
Number of Edges = 6.5
\end{verbatim}
Summary

- graph infrastructure is in early stages but growing
- we plan to interoperate with LEDA (proprietary library)
- more boost/STL should be exposed
- good applications proceeding in proteomics
Standards: MIAME

- Biobase defines the class, widgets can prompt for the data

```r
> getClass("MIAME")
```

Slots:

<table>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>character</td>
</tr>
<tr>
<td>lab</td>
<td>character</td>
</tr>
<tr>
<td>contact</td>
<td>character</td>
</tr>
<tr>
<td>title</td>
<td>character</td>
</tr>
<tr>
<td>abstract</td>
<td>character</td>
</tr>
<tr>
<td>url</td>
<td>character</td>
</tr>
<tr>
<td>samples</td>
<td>list</td>
</tr>
<tr>
<td>hybridizations</td>
<td>list</td>
</tr>
<tr>
<td>normControls</td>
<td>list</td>
</tr>
<tr>
<td>preprocessing</td>
<td>list</td>
</tr>
<tr>
<td>other</td>
<td>list</td>
</tr>
</tbody>
</table>

Extends: "characterORMIAME"
## Standards: MAGE-OM

- a collection of concept packages defining metadata about a microarray experiment:

<table>
<thead>
<tr>
<th>Concept Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuditAndSecurity</td>
<td></td>
</tr>
<tr>
<td>BioMaterial</td>
<td>Array</td>
</tr>
<tr>
<td>Biosample</td>
<td>Manufact.LIMS</td>
</tr>
<tr>
<td>Characteristics</td>
<td>ArrayGroup</td>
</tr>
<tr>
<td>MaterialType</td>
<td>FeatureDefect</td>
</tr>
<tr>
<td>QCStatistics</td>
<td>defectType</td>
</tr>
<tr>
<td>Treatments</td>
<td>feature</td>
</tr>
<tr>
<td>Type</td>
<td>positionDelta</td>
</tr>
<tr>
<td>BioSource...</td>
<td>ZoneDefect...</td>
</tr>
<tr>
<td>BioAssay</td>
<td>ArrayDesign</td>
</tr>
<tr>
<td>BioAssayData</td>
<td>BioSequence</td>
</tr>
<tr>
<td>BioAssayDataTransformation</td>
<td>DesignElement</td>
</tr>
<tr>
<td>Experiment</td>
<td>QuantitationType</td>
</tr>
<tr>
<td>Measurement</td>
<td></td>
</tr>
<tr>
<td>Protocol</td>
<td>[Each with extensive substructure]</td>
</tr>
</tbody>
</table>
MAGE-ML

- a consortium of investigators, manufacturers and designers established a DTD for a markup of microarray metadata and data conforming to MAGE-OM

- see [www.mged.org](http://www.mged.org)

- The NCI Director’s Challenge includes a queryable repository of array results in MAGE-ML [gedp.nch.nih.gov/dc/index.jsp](http://gedp.nch.nih.gov/dc/index.jsp)
Example of MAGEML

```xml
<MAGE-ML identifier="Directors Challenge">
  <AuditAndSecurity_package>
    <Contact_assnlist>
      <Person identifier="198" URI="" address="A510A MSRB1, B0656,,Mi,48109-0656" phone="734-763-0917" email="rork@umich.edu" fax="">
      </Person>
    </Contact_assnlist>
  </AuditAndSecurity_package>
  <Protocol_package />
  <BioMaterial_package>
    <BioMaterial_assnlist>
      <BioSource identifier="3004">
        <Characteristics_assnlist>
          <OntologyEntry category="disease_stage" value="4">
            </OntologyEntry>
          <OntologyEntry category="disease_grade" value="2">
            </OntologyEntry>
          <OntologyEntry category="sex" value="F">
            </OntologyEntry>
          <OntologyEntry category="tissue" value="Ovary">
            </OntologyEntry>
          <OntologyEntry category="disease_state" value="cancer">
            </OntologyEntry>
        </Characteristics_assnlist>
      </BioSource>
    </BioMaterial_assnlist>
  </BioMaterial_package>
</MAGE-ML>
```
Drilling down

- need XML tools to excavate info from

```xml
<Characteristics_assnlist>
<OntologyEntry category="disease_stage" value="4">
</OntologyEntry>
<OntologyEntry category="disease_grade" value="2">
</OntologyEntry>
<OntologyEntry category="sex" value="F">
</OntologyEntry>
<OntologyEntry category="tissue" value="Ovary">
</OntologyEntry>
<OntologyEntry category="disease_state" value="cancer">
</OntologyEntry>
<OntologyEntry category="age" value=""/>
</OntologyEntry>
</Characteristics_assnlist>
```
Want something like

```
$"Characteristics_assnlist"
disease_stage  disease_grade  sex  tissue  disease_state
   "4"         "2"      "F"  "Ovary"     "cancer"
   ""           ""       ""    ""          ""
```

Quiz: Where would this go?

We can also drill into probe-level quantitation data and return annotated lists of results ... using xmlEventParse
MAGEML package

- Steffen Durinck and Joke Allemeersch of KU Leuven have contributed a package that reads MAGEML embodiments of cDNA array data and builds marray* class instances.
- The vignette is suitable for the final lab, for those who complete the other work.
- This package will be extended to:
  - read MAGEML embodiments of affy data and return exprSet objects
  - write MAGEML documents describing exprSets