Gene Expression in Cancer Cells

Daniel Kaplan
CVC Workshop, July 2014

Gene Expression and Gene Expression Studies

For those unfamiliar with gene expression, we include the following description:

> Gene expression is the process by which information from a gene is used in the synthesis of a functional gene product. These products are often proteins, but in non-protein coding genes such as ribosomal RNA (rRNA), transfer RNA (tRNA) or small nuclear RNA (snRNA) genes, the product is a functional RNA... Several steps in the gene expression process may be modulated, including the transcription, RNA splicing, translation, and post-translational modification of a protein. Gene regulation gives the cell control over structure and function, and is the basis for cellular differentiation, morphogenesis and the versatility and adaptability of any organism... The timing, location, and amount of gene expression can have a profound effect on the functions (actions) of the gene in a cell or in a multicellular organism. In genetics, gene expression is the most fundamental level at which the genotype gives rise to the phenotype. The genetic code stored in DNA is "interpreted" by gene expression, and the properties of the expression give rise to the organism's phenotype. Such phenotypes are often expressed by the synthesis of proteins that control the organism's shape, or that act as enzymes catalysing specific metabolic pathways characterising the organism.
>
> Source: http://en.wikipedia.org/wiki/Gene_expression

> Developed in the 1990s, DNA microarrays have revolutionized the way in which gene expression is now analyzed by allowing the RNA products of thousands of genes to be monitored at once. By examining the expression of so many genes simultaneously, we can now begin to identify and study the gene expression patterns that underlie cellular physiology: we can see which genes are switched on (or off) as cells grow, divide, or respond to hormones or to toxins.
>
> DNA microarrays are little more than glass microscope slides studded with a large number of DNA fragments, each containing a nucleotide sequence that serves as a probe for a specific gene. The most dense arrays may contain tens of thousands of these fragments in an area smaller than a postage stamp, allowing thousands of hybridization reactions to be performed in parallel (Figure 8-62). Some microarrays are generated from large DNA fragments that have been generated by PCR and then spotted onto the slides by a robot. Others contain short oligonucleotides that are synthesized on the surface of the glass wafer with techniques similar to those that are used to etch circuits onto computer chips. In either case, the exact sequence—and position—of every probe on the chip is known. Thus any nucleotide fragment that hybridizes to a probe on the array can be identified as the product of a specific gene simply by detecting the position to which it is bound.
>
> Source: www.ncbi.nlm.nih.gov/books/NBK26818/

NCI 60

In the 1980s, the National Cancer Institute developed a set of 60 cancer cell lines, called NCI60. The original purpose was for screening potential anti-cancer drugs. A recent overview in *Nature* describes some of the ongoing work with these cell lines.

Here you will examine gene expression in these cell lines using data described in Staunton *et al.*. More than 41,000 probes were used for each of the 60 cell lines.

For convenience, the data are provided by the `DCIdevel` package.

```
# load required packages
require(DCFdevel)
require(mosaic)
# this will load two data frames: nci60expression and nci60celline
data(nci60)
```

The `nci60` data tables are lightly re-organized from the form provided by Staunton *et al.*.

Two separate data frames are provided: `nci60expression` and `nci60cellLine`. The expression data is quite big — 41,078 probes by 60 cell lines — so there's little point in looking at it directly:

```
dim(nci60expression)
```

```
[1] 41078    61
```

Each of the 2,454,680 numbers in the expression data is on a log scale.

To help visualize how the process of data processing, here is a small subset of 15 cases and 6 cell lines from the data. These are presented in the same format as the original. The name of the cell line starts with two or three letters indicating the tissue type: `BR` is breast, `CNS` is brain (central nervous system), `CO` is colon.

```
set.seed(300)
row.names(nci60expression) <- make.names(nci60expression$Probe, unique=TRUE)
Small <- nci60expression %>%
  select(c(1,2,4,7,8,14,15)) %>%       # only these columns
  sample(size=15, orig.ids=FALSE)    # 15 random rows
```

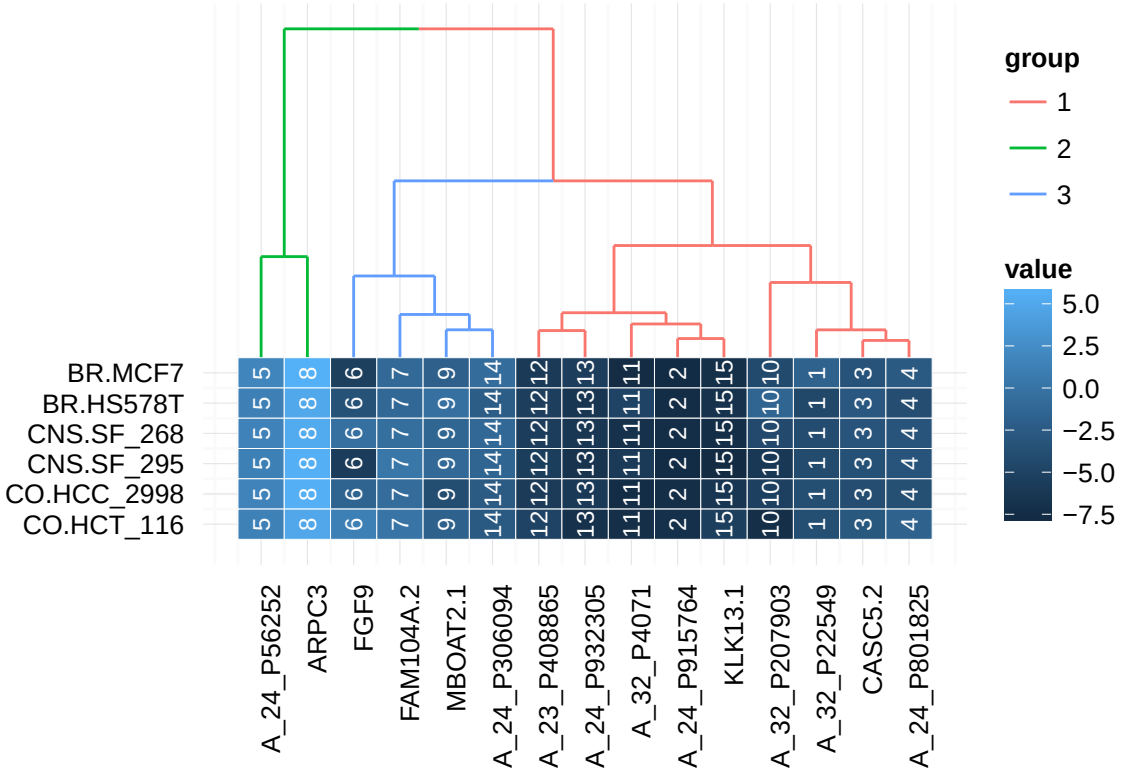
```
print(Small, row.names=FALSE)
```

```
       Probe BR.MCF7 BR.HS578T CNS.SF_268 CNS.SF_295 CO.HCC_2998 CO.HCT_116
  A_32_P22549   -1.65     -4.51      -3.98      -4.10       -3.77      -3.77
 A_24_P915764   -7.45     -7.30      -7.25      -7.41       -7.20      -6.97
       CASC5   -3.22     -2.84      -3.12      -3.47       -3.91      -2.80
 A_24_P801825   -2.57     -3.89      -3.16      -3.11       -3.72      -1.66
  A_24_P56252    1.70      1.21       1.39       1.14        1.47       1.28
        FGF9   -5.28     -3.37      -0.33      -5.64       -1.56       1.15
     FAM104A   -1.11     -1.13      -0.53       0.29       -0.77       0.05
       ARPC3    5.63      4.85       5.11       5.51        5.51       4.80
      MBOAT2   -1.92     -0.35      -1.36      -1.11       -3.85      -1.88
 A_32_P207903   -4.21     -1.51      -4.19      -6.13       -5.23      -7.11
   A_32_P4071   -7.40     -4.91      -6.93      -5.90       -7.36      -7.11
 A_23_P408865   -5.96     -5.44      -5.56      -5.48       -5.82      -4.89
 A_24_P932305   -5.62     -6.40      -6.22      -6.70       -6.02      -6.89
 A_24_P306094   -0.75     -2.25      -1.13      -1.42       -2.53      -2.59
       KLK13   -7.04     -6.78      -7.25      -7.63       -6.38      -5.40
```

Construct this small dataset for developing your computer statements.

Once you have the `small` dataframe, it's relatively easy to display how the data indicate the cell lines are related.

```
Small2 <- Small %>% select(-Probe)
SmallDend <- Small2 %>% dist() %>% hclust()
mplot(SmallDend, data=Small2, k=3, heatmap=0.5, labels=TRUE)
```



But this method wouldn't scale nicely to all 41,000 probes. You'll have to do some work to get the data in form where you can construct the dendrograms.

Narrow vs. Wide Data Formats

- The expression data is provided in *wide* format. What would it look like in *narrow* format.
- Create the *narrow* format data, using `Expression` as the name of that variable in the result and `CellLine` as the variable name for that information. `melt()` let's you specify the `value.name` and the `variable.name` as character strings. You'll have to figure out which is which.

```
Narrow <- reshape2::melt(Small, value.name='Expression', variable.name='CellLine')
```

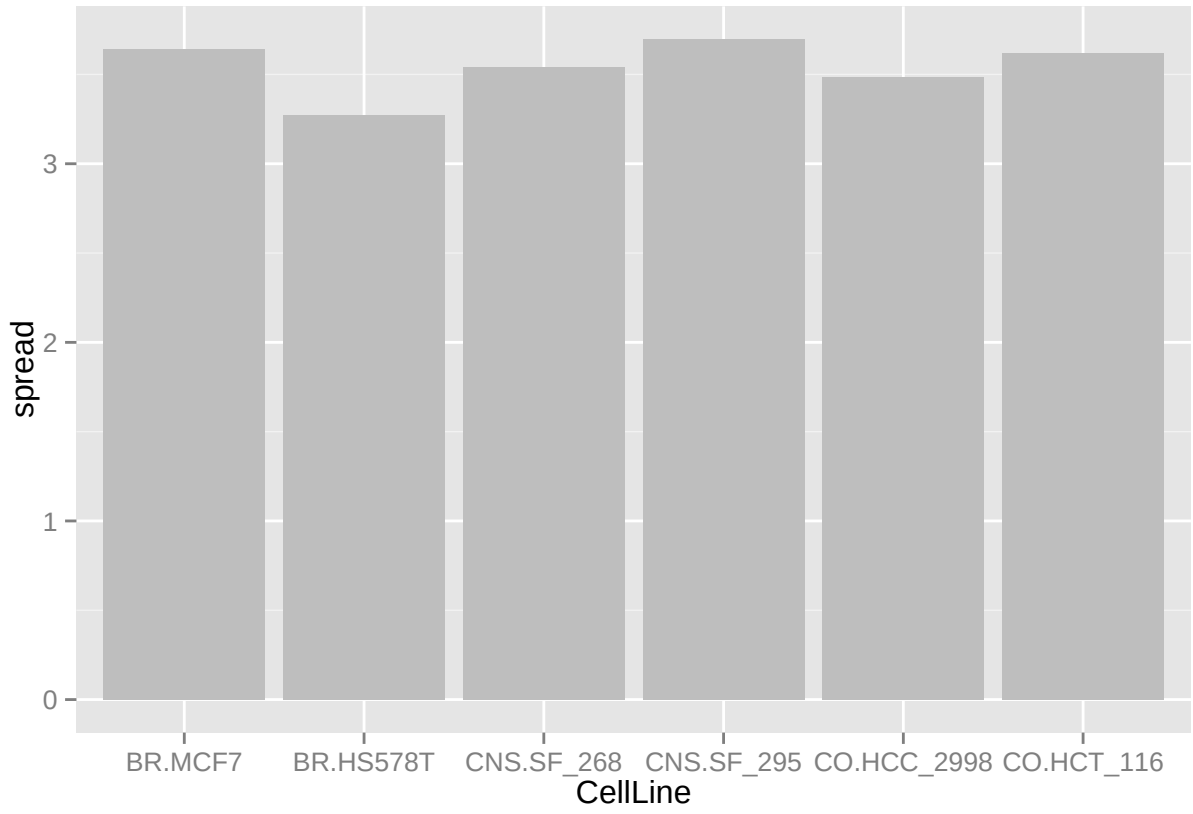
```
Using Probe as id variables
```

- If the expression does not vary much for each cell line across probes, that cell line might be faulty. One way to measure the variation is with the standard deviation (`sd()`) Make a chart to compare the variation in expression across probes in each of the cell lines. Explain why or why not your chart form would be effective for the full data.

```
NarrowByCellLine <-
Narrow %>%
  group_by( CellLine ) %>%
```

```
  summarise( spread=sd(Expression), n=n() )

ggplot(data=NarrowByCellLine, aes(x=CellLine, y=spread)) +
  geom_bar(fill='grey', stat='identity')
```



Weeding Out

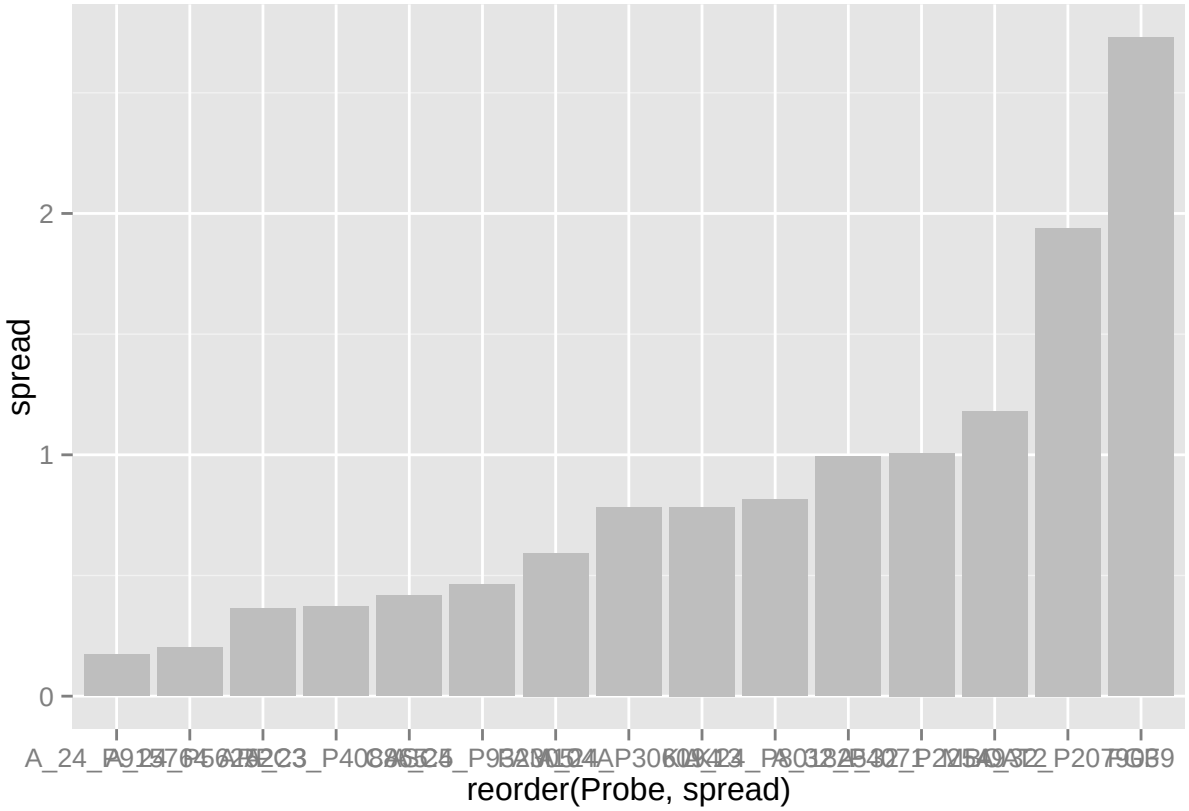
The variation of expression of a probe across cell lines can be used as an indicator of the relationship between the expression of that gene and cancer types. But there are many probes that hardly vary across the cell lines. For statistical reasons, to increase the "power" of a study, it's sensible to delete these probes from the data. The expression shown by most of the probes do not vary across cell lines.

- Find the standard deviation of the expression for each probe across cell lines.
  - Decide whether you want to do this from the data in long or wide format. Either is possible.

```
Narrow %>%
  group_by( Probe ) %>%
  summarize( spread=sd(Expression)) -> NarrowByProbe
```

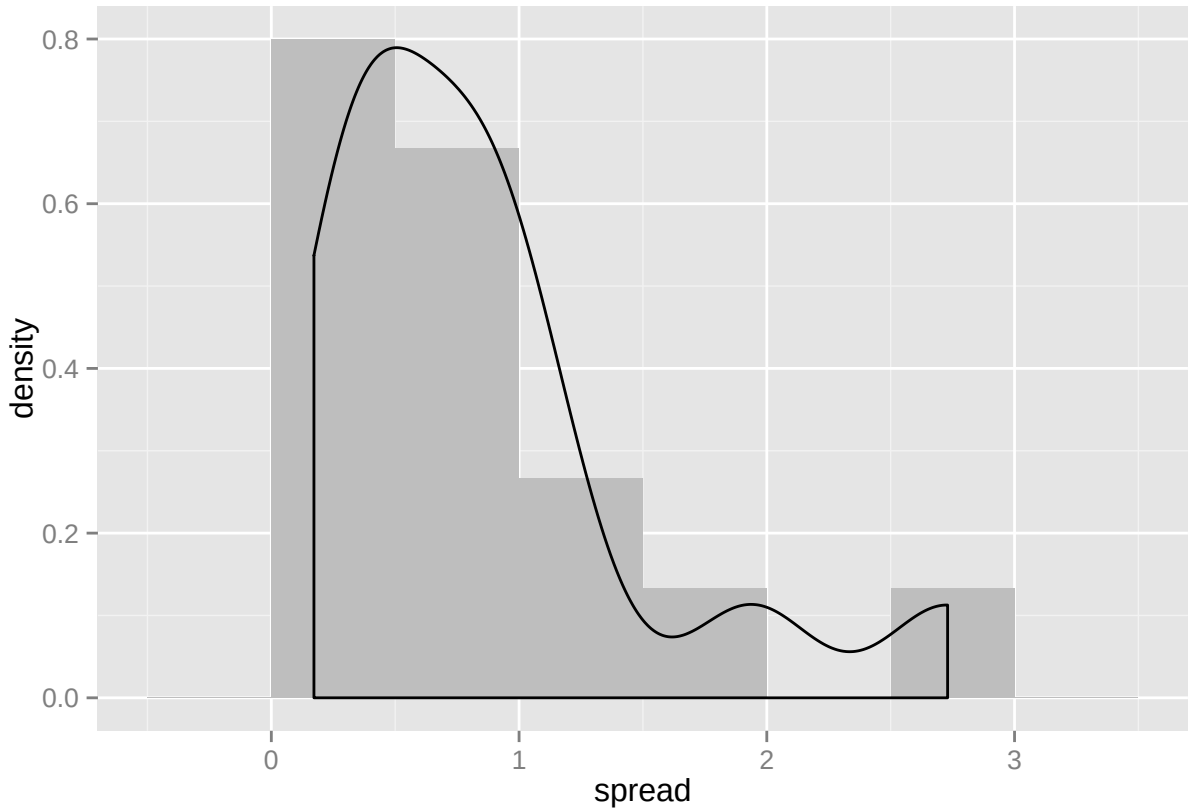
- Make a bar chart of `small` that shows how the expression varies across cell lines for each probe.
  - Explain why or why not your chart form would be effective for the full data.
  - What do you think would be the best ordering of probes for the purpose of spotting how many probes have high variation?

```
ggplot(data=NarrowByProbe, aes(x=reorder(Probe, spread), y=spread)) +
  geom_bar(fill='gray', stat='identity')
```



- Make a histogram and/or density plot showing how probes differ in terms of their variation across cell lines. *Explain why or why not this cart would be effective for the full data.

```
ggplot(data=NarrowByProbe, aes(x=spread)) +
geom_histogram(fill='gray', binwidth=0.5, aes(y= ..density..)) +
geom_density(aes(y=..density..))
```



- Make a dataset containing the expression data for the five probes with the highest standard deviation.

```
Highest <-
  NarrowByProbe %>%
  filter(rank(desc(spread)) <= 5)
```

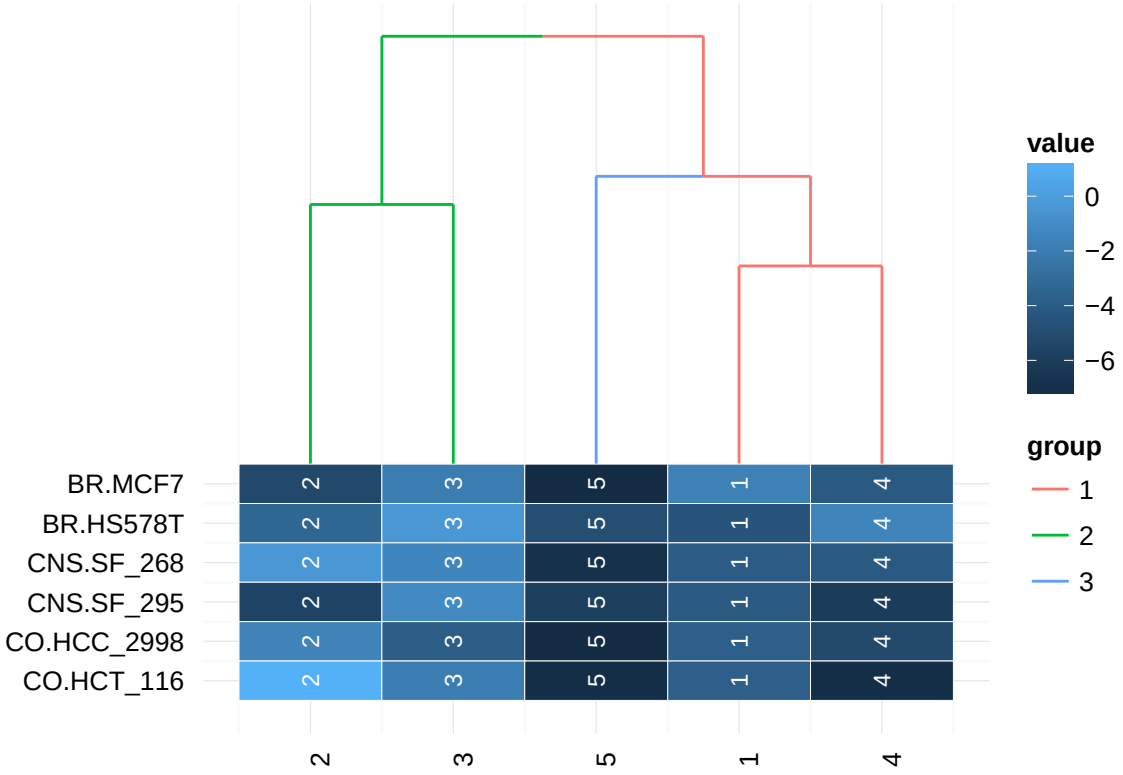
- Join the highest varying probes with the `small` dataset.
  - Experiment with the different sorts of join to find one that keeps just the highest varying probes in `small`.
  - When you have a working result, use `select()` to get rid of the `spread` variable. It's done its work and is no longer needed.

```
Keepers <- Small %>% inner_join(Highest) %>% select(-spread)
```

```
Joining by: "Probe"
```

The dendrogram/heat-map will look like this:

```
KeepersTmp <- Keepers %>% select(-Probe)
KeepersDend <- KeepersTmp %>% dist() %>% hclust()
mplot(KeepersDend, data=KeepersTmp, k=3, heatmap=0.5, labels=TRUE)
```

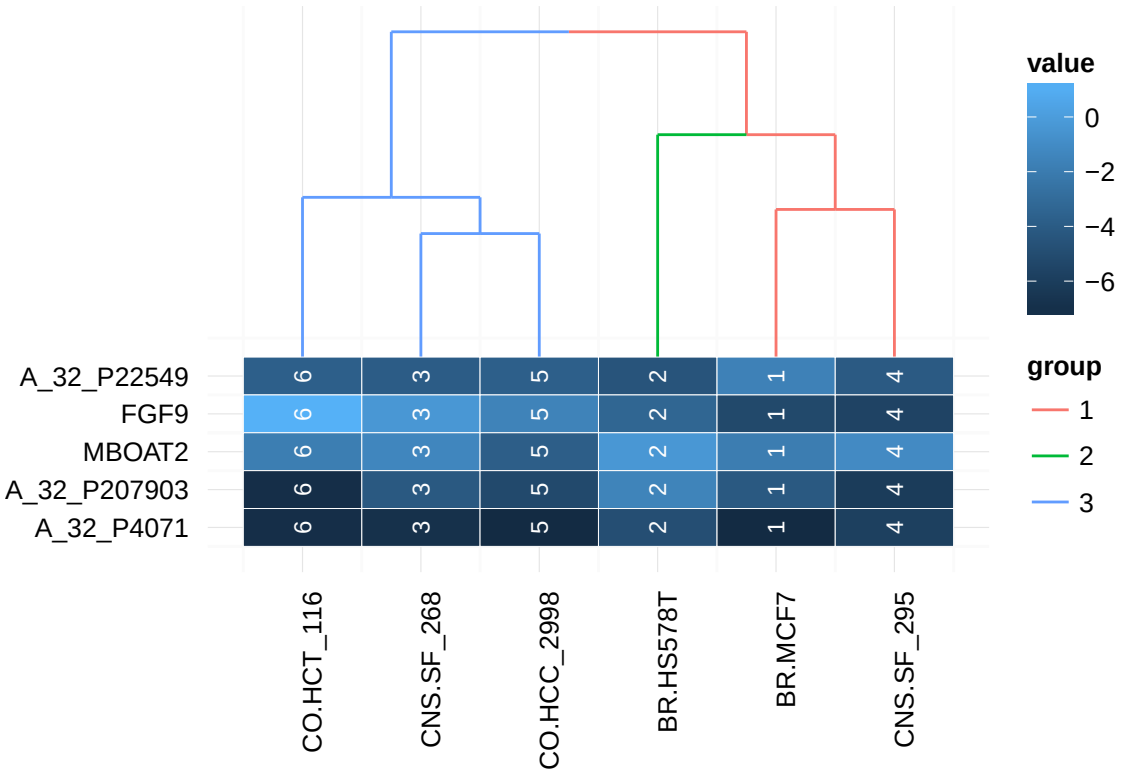


You might prefer to create a dendrogram of the cell types rather than the probes. This is a matter of taking the transpose of the data. Here's how:

```
KeepersTranspose <- Keepers %>% select(-Probe) %>% t(.)
colnames(KeepersTranspose) <- Keepers$Probe
KeepersTransposed <- as.data.frame(KeepersTranspose)
```

And the corresponding dendrogram:

```
Dend <- hclust(dist(KeepersTransposed))
mplot(Dend, data=KeepersTransposed, k=3, heatmap=0.5, labels=TRUE)
```



This dendrogram doesn't show a good classification of the tissue types, but there isn't much data going into it. More data might produce a better classification.

On Your Own

When you have the above working ...

For the whole data set — picking out just 100 or so probes with the greatest variation — make dendrograms of

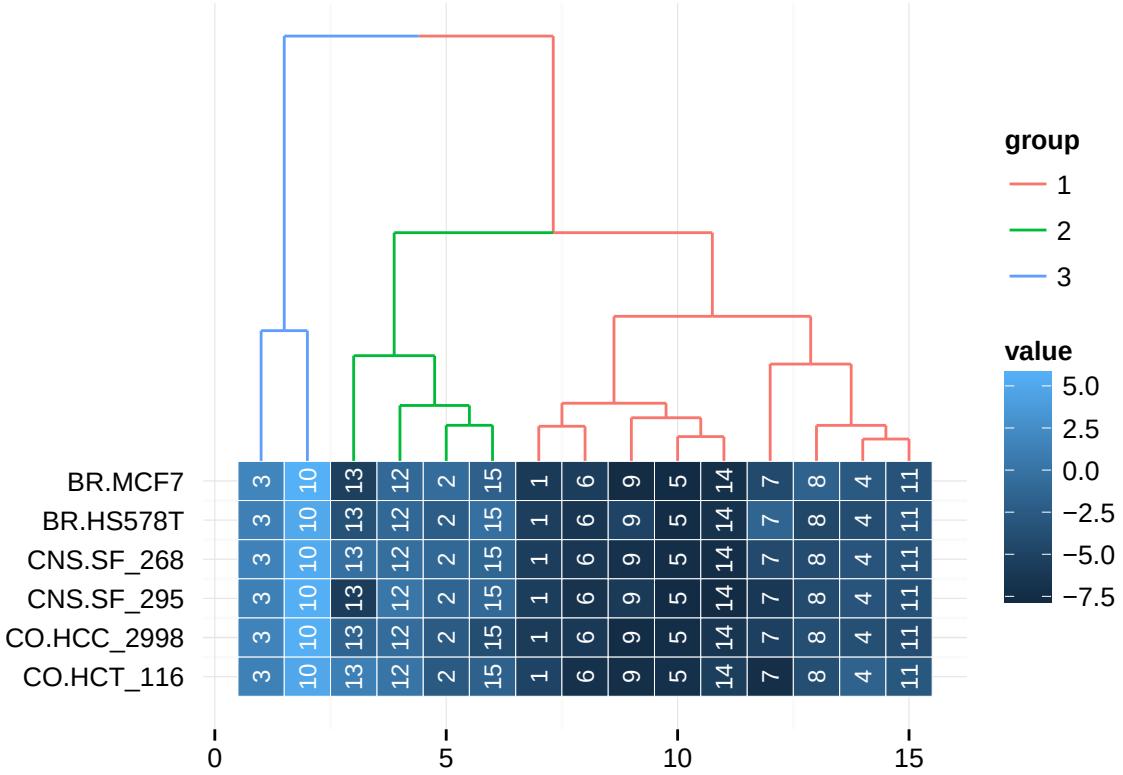
- the cell lines using probe expression
- the probes using cell lines

```
Highest <-
  NarrowByProbe %>%
  filter(rank(desc(spread)) <= 200)
Keepers <- Small %>% inner_join(Highest) %>% select(-spread)
```

```
Joining by: "Probe"
```

The dendrogram/heat-map will look like this:

```
KeepersTmp <- Keepers %>% select(-Probe)
KeepersDend <- KeepersTmp %>% dist() %>% hclust()
mplot(KeepersDend, data=KeepersTmp, k=3, heatmap=0.5, labels=FALSE)
```

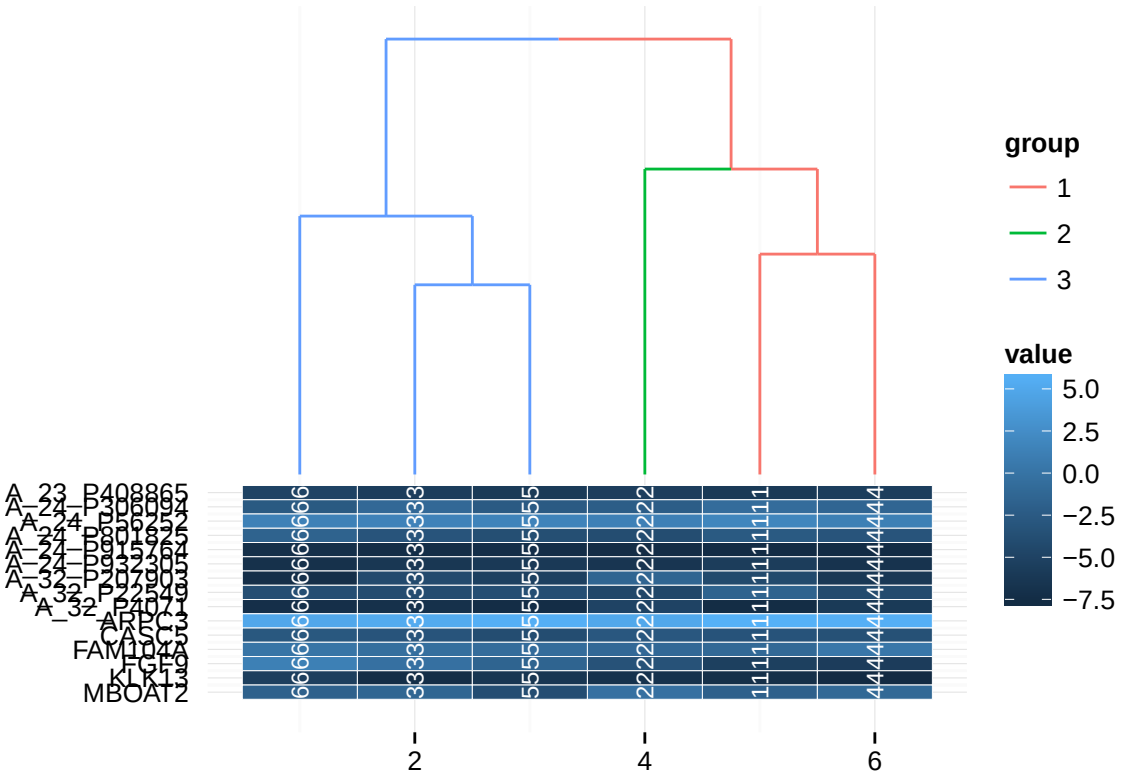


You might prefer to create a dendrogram of the cell types rather than the probes. This is a matter of taking the transpose of the data. Here's how:

```
KeepersTranspose <- Keepers %>% select(-Probe) %>% t(.)
colnames(KeepersTranspose) <- Keepers$Probe
KeepersTransposed <- as.data.frame(KeepersTranspose)
```

And the corresponding dendrogram:

```
Dend <- hclust(dist(KeepersTransposed))
mplot(Dend, data=KeepersTransposed, k=3, heatmap=0.5, labels=FALSE)
```


group
1
2
3
value
5.0
2.5
0.0
−2.5
−5.0
−7.5
A_23_P408865
A_24_P306094
A_24_P56252
A_24_P801825
A_24_P915764
A_24_P932305
A_32_P207903
A_32_P22549
A_32_P4071
ARPC3
CASC5
FAM104A
FGF9
KLK13
MBOAT2
2
4
6