Section 1: What we measure and why

Mammaprint Gene Signature

- Exploring genes used in the Mammaprint gene signature - assess risk of breast cancer
- Diagnostic signature using gene expression levels of 70 genes
- Information about the 70 gene signature used in the Mammaprint algorithm

```
library(genefu)
```

```
## Loading required package: survcomp
## Loading required package: survival
## Loading required package: prodlim
## Loading required package: mclust
## Package 'mclust' version 5.4.7
## Type 'citation("mclust")' for citing this R package in publications.
## Loading required package: limma
## Loading required package: biomaRt
## Loading required package: iC10
## Loading required package: pamr
## Loading required package: cluster
## Loading required package: impute
## Loading required package: iC10TrainingData
## Loading required package: AIMS
## Loading required package: e1071
## Loading required package: Biobase
## Loading required package: BiocGenerics
## Loading required package: parallel
##
## Attaching package: 'BiocGenerics'
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
## The following object is masked from 'package:limma':
##
##     plotMA
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
```

```
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min
```

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
data(sig.gene70)
dim(sig.gene70)
```

```
## [1] 70  9
```

```
head(sig.gene70)[,1:6]
```

```
##                         probe correlation average.good.prognosis.profile
## NM_003748           NM_003748   -0.420671                      0.12350000
## NM_003862           NM_003862   -0.410964                      0.05159091
## Contig32125_RC Contig32125_RC   -0.409054                      0.05409091
## U82987                 U82987   -0.407002                      0.06150000
## AB037863            AB037863    -0.402335                      0.06334091
## NM_020974           NM_020974   -0.399987                     -0.06231818
##                EntrezGene.ID NCBI.gene.symbol HUGO.gene.symbol
## NM_003748               8659          ALDH4A1          ALDH4A1
## NM_003862               8817            FGF18            FGF18
## Contig32125_RC            NA             <NA>             <NA>
## U82987                 27113             BBC3             BBC3
## AB037863                  NA             <NA>             <NA>
## NM_020974              57758           SCUBE2           SCUBE2
```

```
count_nan_gene_symbol <- sum(is.na(sig.gene70$NCBI.gene.symbol))
paste("Count of NaN NCBI gene symbols: ", count_nan_gene_symbol)
```

```
## [1] "Count of NaN NCBI gene symbols:  14"
```

```
subset_matching_desc <- sig.gene70[which(sig.gene70$Description == "cyclin E2"), ]
paste("NCBI gene matching the description cyclin E2: ", subset_matching_desc$NCBI.gene.symbol)
```

```
## [1] "NCBI gene matching the description cyclin E2:  CCNE2"
```

```
number_kinase_coding_genes <- length(grep("kinase", sig.gene70$Description))
paste("Number of kinase coding genes responsible for cell to cell communication: ", number_kinase_coding
```

```
## [1] "Number of kinase coding genes responsible for cell to cell communication:  4"
```

Assessment: Phenotypes

- COPDSexualDimorphism.data package - phenotypes (cols) individuals (rows)
- Data to assess incidence of COPD and emphysema by gender and smoking status
- The pkyrs variable in the expr.meta data.frame represents pack years smoked. Other variables include gender and diagmaj (disease status). These variables correspond to phenotypes.

```
library(COPDSexualDimorphism.data)
data(lgrc.expr.meta)
head(expr.meta)
```

```
##     tissueid     sample_name  newid   GENDER age       cigever pkyrs
## 1 LT001098RU LT001098RU_COPD 161745 2-Female  46 2-Ever (>100)    35
## 2 LT001796RU LT001796RU_CTRL 212671   1-Male  48 2-Ever (>100)    19
## 3 LT005419RU LT005419RU_COPD 291396   1-Male  70 2-Ever (>100)    43
## 4 LT007392RU LT007392RU_COPD 169067   1-Male  46 2-Ever (>100)    45
## 5 LT009615LU LT009615LU_CTRL  49801 2-Female  49 2-Ever (>100)    45
## 6 LT010491LL LT010491LL_COPD 180409   1-Male  78 2-Ever (>100)    51
##            diagmaj   gender
## 1 2-COPD/Emphysema 2-Female
## 2        3-Control   1-Male
## 3 2-COPD/Emphysema   1-Male
## 4 2-COPD/Emphysema   1-Male
## 5        3-Control 2-Female
## 6 2-COPD/Emphysema   1-Male
```

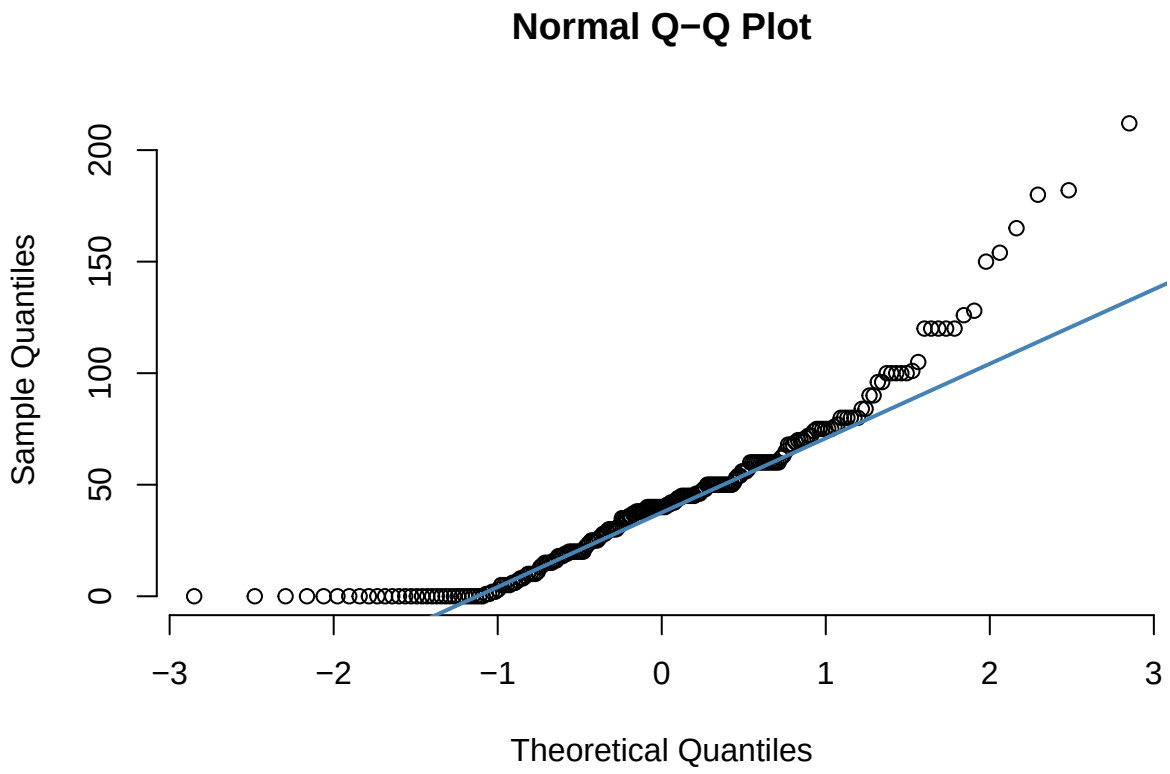
```
table(expr.meta$GENDER)
```

```
##
##   1-Male 2-Female
##      119      110
```

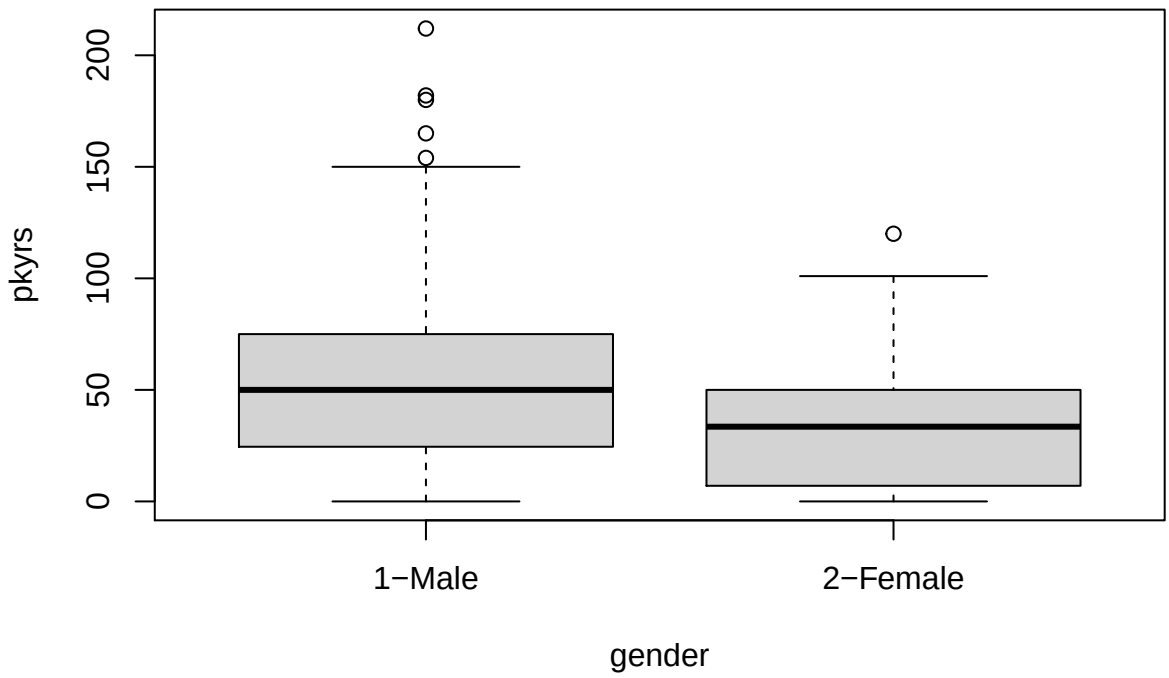
```
summary(expr.meta$pkyrs)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   15.00   40.00   44.17   60.00  212.00
```

```
qqnorm(expr.meta$pkyrs, pch=1, frame=FALSE)
qqline(expr.meta$pkyrs, col = "steelblue", lwd = 2)
```




```
boxplot(pkyrs~gender, data=expr.meta)
```



Assessment: Chromosomes and SNPs

- GWAS (Genome-wide association studies)
- Comparing individuals with disease vs. controls using SNP chips or DNA sequencing.
- SNPs with association are investigated for disruption of gene regulation or function

- Bioconductor *gwascat* package

```
library(gwascat)
```

```
## gwascat loaded.  Use makeCurrentGwascat() to extract current image.

##  from EBI.  The data folder of this package has some legacy extracts.
```

```
data(ebicat_2020_04_30)
ebicat_2020_04_30
```

```
## gwasloc instance with 50000 records and 38 attributes per record.
## Extracted:  2020-04-30 23:24:51
## metadata()$badpos includes records for which no unique locus was given.
## Genome:  GRCh38
## Excerpt:
## GRanges object with 5 ranges and 3 metadata columns:
##       seqnames    ranges strand |   DISEASE/TRAIT        SNPS   P-VALUE
##          <Rle> <IRanges>  <Rle> |     <character> <character> <numeric>
##   [1]       10  58153390      * | Crohn's disease   rs1819658     9e-17
##   [2]        1 206766559      * | Crohn's disease   rs3024505     2e-14
##   [3]       13  42478744      * | Crohn's disease   rs2062305     5e-10
##   [4]       19   1124836      * | Crohn's disease    rs740495     8e-12
##   [5]       12  40398498      * | Crohn's disease  rs11564258     6e-21
##   -------
##   seqinfo: 24 sequences from GRCh38 genome
```

```
sort(table(ebicat_2020_04_30$CHR_ID), decreasing=TRUE)
```

```
##
##    1    2    6    3   11    5    4    7   12    8   17   10    9   16   19   15
## 4294 4290 4085 3202 2995 2908 2587 2530 2447 2307 2281 2138 2010 1972 1965 1746
##   14   20   18   13   22   21    X
## 1341 1270 1154 1090  790  401  197
```

Microarray Technology 1: How Hybridization Works

- Two technologies: microarray and NGS
- Both counting DNA or RNA molecules
- Both use a trick which allows us to take double-stranded DNA and convert to single-stranded
- Both require thousands - millions of molecules for us to be able to measure anything
- If a few cells only, they must be amplified

Microarray Technology

1. Denaturation (single-stranded)
2. Hybridization - when you have a single strand in solution and it finds complimentary DNA, it will hybridize to form 2 stranded DNA. This can be exploited to count molecules
3. Can create probes / troughs for different sequences. Put on location on piece of solid for the molecules we want to be able to count. Probes have compliments to the DNA that we want to count.

How microarray technology works

- Piece of solid where we put probes - 1x1 cm piece of silicone that gets divided into thousands to millions of cells (difference squares)

- squares correspond to probes which represent molecules we are trying to count
- 25bP long probes in example
- second step: label a sample with fluorescent tags and put on array. hope that right molecules hybridize to right probes

Two-color microarrays

- Hybridize two samples onto one array - two different labels that scanner can recognize
- Advantages: cost savings
- Sample 1: color 1, Sample 2: color 2. Let hybridize and get both hybridized to same probes, but scanner can distinguish two types of labels.
- Two numbers per probe – converted into RGB color combining red and green

Applications of microarray technology

- 3 different applications

1. **Measuring gene expression** - gene chip array.

- For every gene, we know the sequence and take 11 sequences for individual transcripts and hybridize.
- On this array, probes are towards 3' end of transcripts b/c RNA tends to degrade more on one side (5' end).
- 11 probes scattered around array to avoid confounding location with gene for each transcript.
- Label the RNA, put it on the array. Will see lots of hybridization if there are many copies of that transcript.
- High intensity = highly expressed gene. For each gene, select n probes and put them on the array and analyze the data.

2. **Genotyping SNP** - different alleles 2 of same or 1 of each.

- I.e., AA, AG, GG.
- If we want to know which of the three possibilities, we can do this for SNPs.
- Use probes to hybridize to piece of sequence which has A, G for example.
- Genotype millions of SNPs at a time. Arrays popular for GWAS studies to understand which alleles are associated with genes of interest.

3. **Detection of transcription factor binding sites** - genome is more than just sequence, measuring the chemical processes taking place around the genome, i.e. where specific protein is bound.

- Transcription factor = proteins that start gene expression. * Have DNA, want to know where specific protein is bound. Start by fragmenting DNA, some pieces have protein and others do not.
- Divide by presence of protein vs. not - hybridize the part with protein with tiling array and if lights up, the location is where the protein was bound.
- Intensities are not that reliable, must be controlled by hybridizing the total DNA for comparison.

Labeling

Need indirect ways to count molecules. Labeling adds a chemical to each molecule, use optical scanner to identify the different intensities based on # labels and quantify.

Design attribute of different technologies: synthetically sequenced, or cloned. Densities of probes put on the solid is also variable across different technologies. Also # samples on each array differs. Major manufacturers:

1. Affymetrix (high density, one color)
2. Agilent (circles on grid, one or two color)
3. Illumina (high density, one or two color)
    - Uses beads instead of in-situ sequencing

Brief introduction of NGS

- Early 21st century Human Genome sequenced
- Took DNA from several humans, pooled together and sequenced base x base the entire thing (1st generation)
- Back then, millions of clones 1k bP long. Different labs would sequence each clone, and then put together using computational methods. Cost billions of dollars
- NGS is high throughput - billions of bP in a week for thousands of dollars.
- Many copies of DNA to obtain measurement.

1. Fragment DNA using mol bio –> feed into NGS sequencer
2. Read sequences for all of the fragments to get the reads (bases)

Illumina flow cells - 8 lanes (1 per sample)

- For each lane, we get 160mill short reads (50-70bP long)
- Starts with DNA sample with fragments. Add adapters to each one to allow fragments to attach to pieces of solid. Once attached, amplify each one so that we have millions of copies (clusters) - adapter + fragment + copies.
- Add labeled nucleotides. Different from microArray - not having two molecules join.
- First base of the sequence attached to compliment, and starts forming double stranded. Once the first nucleotide attaches, take a picture and read intensity of labels. Keep doing this until we get through almost the entire fragment.
- Images from sequencing machine represent clusters - first base of sequence represented by cluster.
- Next step we get another image corresponding to bases on the second location and assemble the whole molecular sequence.

Applications of NGS

- Similarly to microArray technology, we can measure gene expression, genotype, location of transcription factor binding sites
- First application: sequencing the genome
- Resequencing: do not sequence the whole genome with a new subject of same species, only areas of interest. SNP discovery, genotyping, variant discovery and quantification
- Measuring methalation
- Used be 1000s genomes project, human epigenome project

Going from series of reads to measurements * First step in analyzing the NGS sequences: finding where the reads came from. All reads get mapped to the genome : matches to the reference genome * First application: Variant detection * Finding new SNPs. Take sample, sequence, align to genome, go to any specific location and ask if it is a SNP by analyzing whether there are G's and A's (heterozygous) * There are sequencing errors * Deletion: alleles are missing a base

- RNA seq - NGS to quantify gene expression
- We have RNA - two samples to see if they are different.
- RNA –> DNA, sequence DNA and map / align to reference genome
- Compare gene expression in the two samples
- ChipSeq - finding transcription factor binding sites
- DNA –> separate fragments bound to specific proteins and sequence those sections
- Location of genome bound to many reads –> means that it was bound to that protein
- Peak Detectors to identify locations of protein binding sites

Analyzing gene expression microarray dataset

```
library(tissuesGeneExpression)
data(tissuesGeneExpression)
paste("log scale intensities for microarray probes: ")
```

```
## [1] "log scale intensities for microarray probes: "
```

```
head(e[,1:5])
```

```
##           GSM11805.CEL.gz GSM11814.CEL.gz GSM11823.CEL.gz GSM11830.CEL.gz
## 1007_s_at       10.191267       10.509167       10.272027       10.252952
## 1053_at          6.040463        6.696075        6.144663        6.575153
## 117_at           7.447409        7.775354        7.696235        8.478135
## 121_at          12.025042       12.007817       11.633279       11.075286
## 1255_g_at        5.269269        5.180389        5.301714        5.372235
## 1294_at          8.535176        8.587241        8.277414        8.603650
##           GSM12067.CEL.gz
## 1007_s_at       10.157605
## 1053_at          6.606701
## 117_at           8.116336
## 121_at          10.832528
## 1255_g_at        5.334905
## 1294_at          8.303227
```

```
paste("tissue types of each sample: ")
```

```
## [1] "tissue types of each sample: "
```

```
table(tissue)
```

```
## tissue
##  cerebellum       colon endometrium hippocampus      kidney       liver
##          38          34          15          31          39          26
##    placenta
##           6
```

```
paste("overall mean expression of 209169_at", mean(e["209169_at",]))
```

```
## [1] "overall mean expression of 209169_at 7.26365039170786"
```

```
paste("mean expression by tissue: ")
```

```
## [1] "mean expression by tissue: "
```

```
sort(by(e["209169_at",], tissue, mean))
```

```
## tissue
##       colon    placenta       liver      kidney endometrium  cerebellum
##    5.076710    5.186711    5.249247    5.325370    5.585281   10.149866
## hippocampus
##   11.466372
```

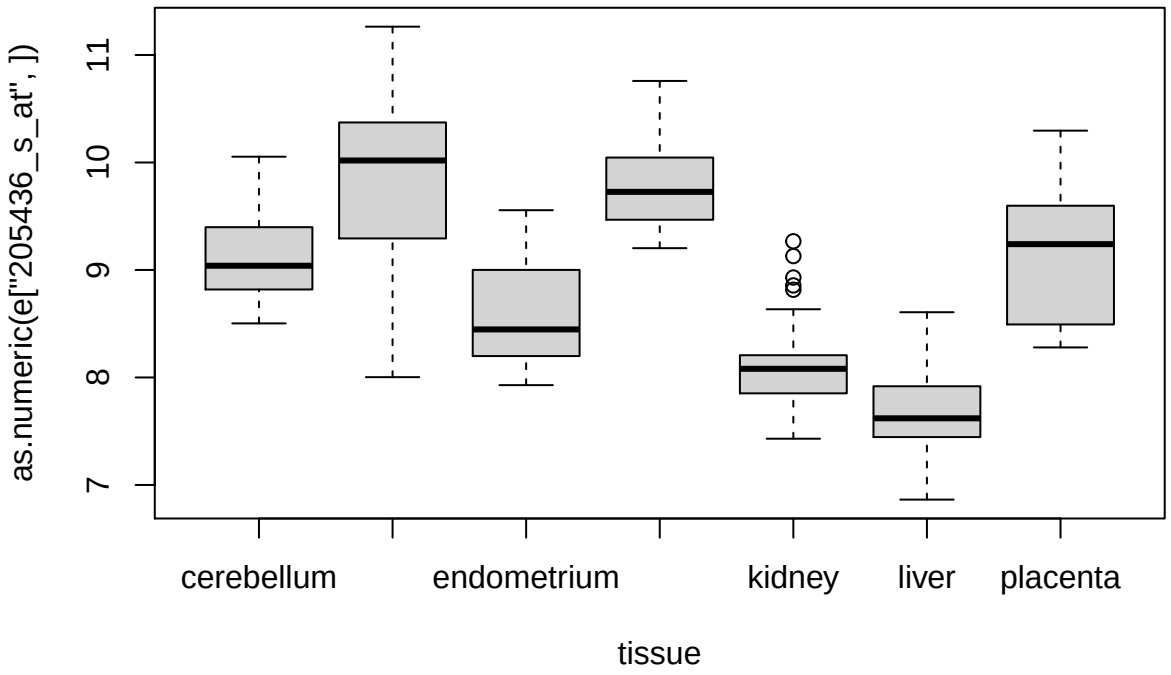
Gene Associated with probe ID

```
library(hgu133a.db)
```

```
## Loading required package: AnnotationDbi
```

```
## Loading required package: stats4
```

```
## Loading required package: IRanges
## Loading required package: S4Vectors
##
## Attaching package: 'S4Vectors'
## The following object is masked from 'package:base':
##
##     expand.grid
## Loading required package: org.Hs.eg.db
##
##
symbol = mapIds(hgu133a.db, keys=rownames(e), column="SYMBOL", keytype="PROBEID")
## 'select()' returned 1:many mapping between keys and columns
paste("gene associated with probe ID 209169_at", symbol["209169_at"])
## [1] "gene associated with probe ID 209169_at GPM6B"
num_features <- sum(symbol == "H2AX", na.rm=TRUE)
paste("number of features measuring expression of H2AX: ", num_features)
## [1] "number of features measuring expression of H2AX:  4"
paste("associated probes: ")
## [1] "associated probes: "
symbol[grep("H2AX", symbol)]
## 205436_s_at 212524_x_at 212525_s_at 213344_s_at
##      "H2AX"      "H2AX"      "H2AX"      "H2AX"
paste("comparing distributions across tissues: ")
## [1] "comparing distributions across tissues: "
boxplot(as.numeric(e["205436_s_at",])~tissue)
```




```
paste("finding gene specific to placenta: ")
```

```
## [1] "finding gene specific to placenta: "
```

```
IDs = c("201884_at", "209169_at", "206269_at", "207437_at", "219832_s_at", "212827_at")
sort(rowMeans(e[IDs, which(tissue == "placenta")]))
```

```
##   207437_at   209169_at   212827_at   201884_at 219832_s_at   206269_at
##    4.410814    5.186711    6.481558    6.637451    7.319096   11.372918
```