The CTC saga: C1 comparison

Stelios Sfakianakis

Date: 2014-09-21

Datasets

C1 is the comparison between normal and cancer tissue. We will merge an number of datasets to create a big one and then perform common differential expression algorithms.

Katerina has selected the following datasets:

- GSE22820:"Gene expression profiles were generated from 176 primary breast cancer patients and 12 normal breast samples", PMID: 21356353 (*Agilent Whole Human Genome Microarray 4x44K G4112F*). The GEO dataset contains **176 primary breast cancer** and **10 normal patients**.
- GSE19783:"mRNA profiling from 115 breast cancer samples was performed", PMID: 21364938 (*Agilent Whole Human Genome Microarray 4x44K G4112F*) Here two samples represent metastatic tumors and were removed so we finally keep **113** breast cancer samples.
- GSE31364:"Seventy-two primary breast cancer tumor have been analyzed against a breast cancer reference pool.", PMID: 22384245 (*Agendia_human_DiscoverPrint_v1* **custom platform**) All samples are kept.
- GSE9574: "Gene expression abnormalities in histologically normal breast epithelium of breast cancer patients", PMID: 18058819 (*Affymetrix Human Genome U133A Array*). We selected **14 normal samples** that have been identified as "disease-state: normal (reduction mammoplasty)"
- GSE18672: "Mammographic density and genetics - A study of breast biopsies in relation to mammoraphic density", PMID: 20799965 (Agilent Whole Human Genome Oligo Microarray G4112A). Here we used all the **79 women with no malignancy** (healthy women) and **64 newly diagnosed breast cancer patients**.

Filtering Genes per Dataset

A "non specific" filtering is performed in each data set:

```
> nsFilter(gse, require.entrez=TRUE,remove.dupEntrez=TRUE,
>               var.func=IQR, var.cutoff=0.5, var.filter=FALSE)
```

where `gse` is the dataset. This means that probes/transcripts that do not map in Entrez genes identifiers are removed and in the case that multiple probes map to the same identifier we remove all but one that has the largest IQR (i.e. the largest variance).

In some datasets where there were missing values I did imputation with the 10 "nearest neighbors".

Now loading the data:

Merging

I am using the inSilicoMerging package of Bioconductor that features some of the most well known cross platform normalization methods (COMBAT is Empirical Bayes, DWD stands for Distance Weighted Discrimination, and GENENORM means gene “z-score” normalization) for the merger of these datasets:

```
> require(inSilicoMerging)
> mgse_COMBAT <- merge(datasets, method="COMBAT");
```

The distribution of the different classes in the initial datasets is shown in the next table:

	GSE18672	GSE19783	GSE22820	GSE31364	GSE9574
Cancer	64	113	176	72	14
Control	79	2	10	0	15

For each dataset we keep specific samples in the final (merged) dataset.

```
> gse22820_samples = sampleNames(gse22820)
> gse31364_samples = sampleNames(gse31364)
> gse18672_samples = sampleNames(gse18672)
>
> ## Two samples are metastatic -- remove those!!
> idx = gse19783$characteristics_ch1.1 == 'disease state: Breast Primary Tumor'
> ## and keep the rest which are all tumor
> gse19783_samples = sampleNames(gse19783[,idx])
>
> # Keep only the normal
> idx = gse9574$characteristics_ch1 == 'disease-state: normal (reduction mammoplasty)'
> gse9574_samples = sampleNames(gse9574[,idx])
>
>
> samples = c(gse22820_samples, gse19783_samples, gse31364_samples, gse18672_samples, gse9574_samples)
>
> c1.data = mgse_COMBAT[,samples]
```

After the selection of samples their distribution is as follows:

	GSE18672	GSE19783	GSE22820	GSE31364	GSE9574
Cancer	64	113	176	72	0
Control	79	0	10	0	15

We can compare the output (how the different data sets are mixed together) by computing a Multidimensional Scaling (MDS) plot for the 3 methods used:

```
> plotMDS(c1.data, "Study", "Disease", main="Distribution of samples in the merged datasets")
```

The plot is shown in Figure 1.

After the merging and selection of samples what is the proportion of normal versus tumor cases?

```
> dim(c1.data);
```

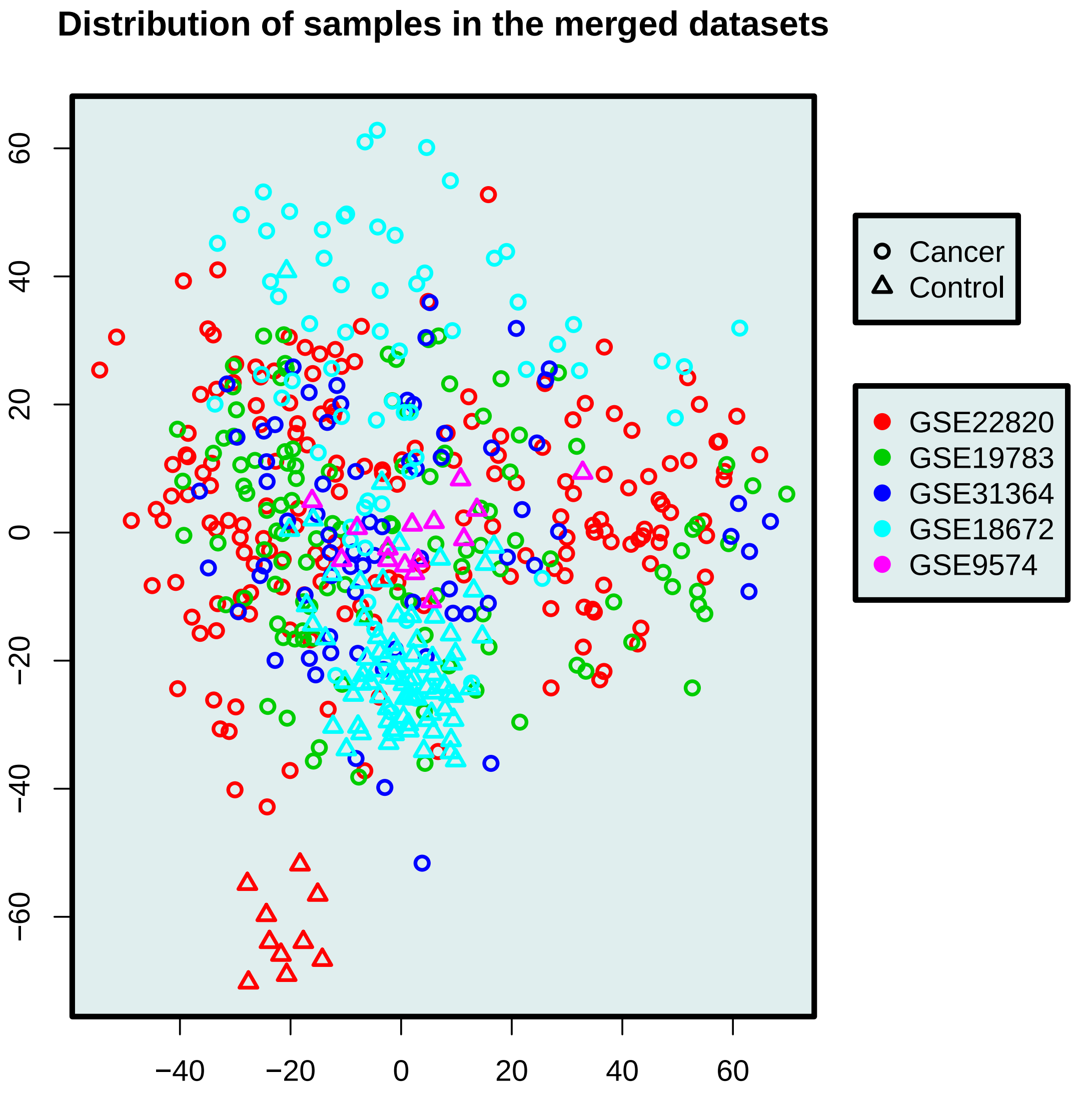

Figure 1: 2D (MDS) plot of the merged dataset

```
Features  Samples
   11928      529
```

```
> table(c1.data$Disease);
```

```

 Cancer Control
    425     104
```

Let's cache the dataset for subsequent analyses:

```
> saveRDS(c1.data, file=file.path("intermediate", "c1.data.rds"))
```

Gene selection

I am using Significance Analysis of Microarrays (SAM) for finding differential expressed genes. We try to select the differentially expressed genes so that the estimated False Discovery Rate (FDR) to be bounded by `0.05` (i.e. to expect 1% falsely assumed differentially expressed genes):

```
> cl = ifelse(c1.data$Disease == 'Control', 0, 1)
> sam.out <- sam(exprs(c1.data), cl, B=500, rand=0xDEAD);
> dd <- findDelta(sam.out, fdr=0.05, verbose=FALSE)[2,1]
```

```
The threshold seems to be at
  Delta Called     FDR
5  1.22   5731 0.05002
6  1.22   5730 0.04984
```

```
> sam.sum <- summary(sam.out, delta=dd)
> ee=exprs(c1.data)[sam.sum@row.sig.genes,]
> num.diff.exp = dim(ee)[1]
```

So by choosing 1.2198 as our "delta" we find 5730 differential expressed genes:

```
> plot(sam.out, dd)
```

Let's find out which of the genes are over-expressed in cancer samples:

```
> w <- which(sam.sum@mat.sig$d.value > 0)
> num.genes.over <- length(w)
> num.genes.over
```

```
[1] 3725
```

```
> siggenes.all <- list.siggenes(sam.out, dd)
> siggenes.over <- list.siggenes(sam.out, dd)[w]
```

So there are 3725 genes over-expressed in cancer state. They can be found in the `c1_siggenes.txt` file.

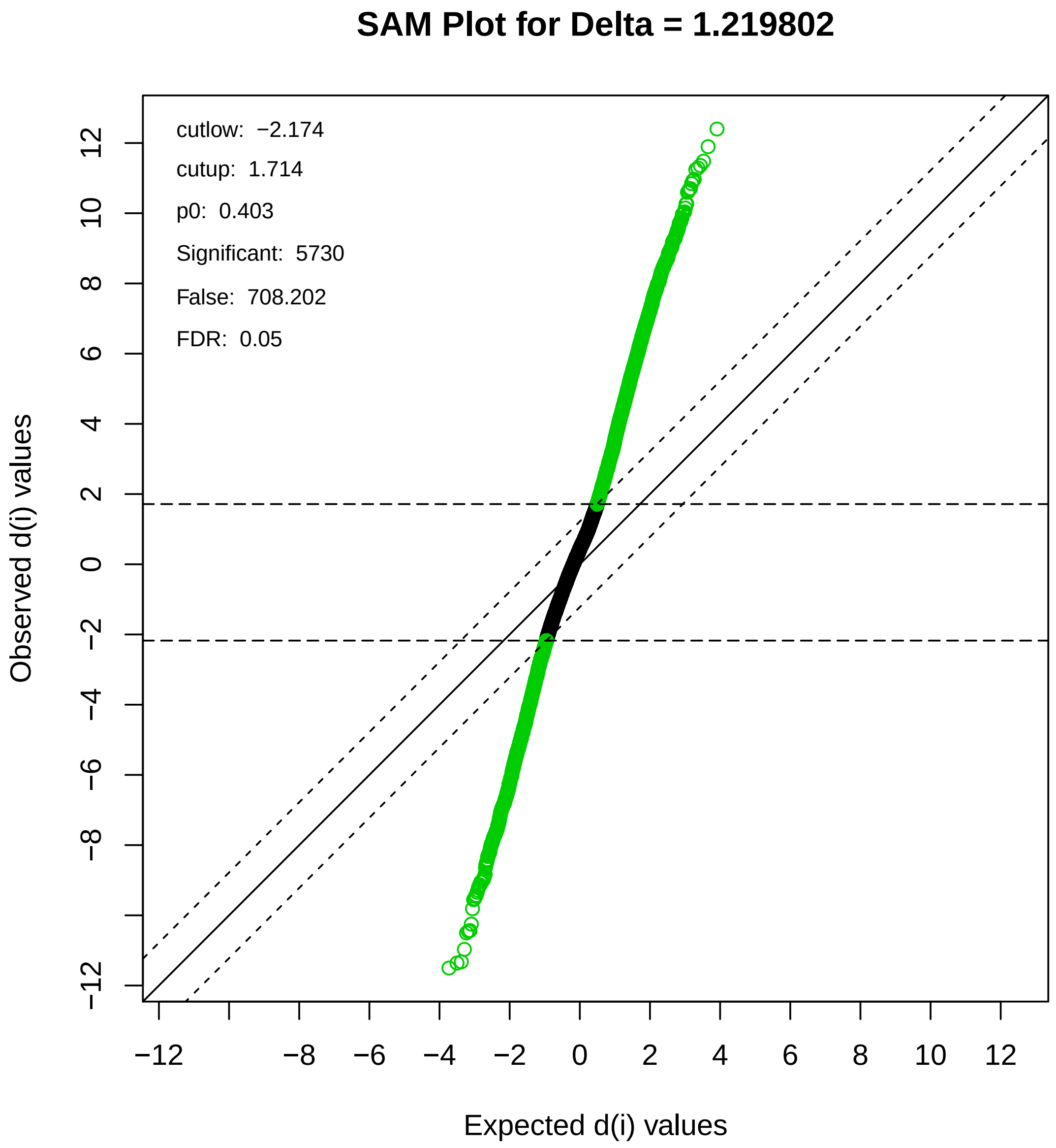

Figure 2: SAM plot

```
> sessionInfo()

R version 3.1.1 (2014-07-10)
Platform: x86_64-unknown-linux-gnu (64-bit)

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] splines   grDevices datasets  parallel  stats     graphics  utils
[8] methods   base

other attached packages:
 [1] xtable_1.7-3          ggplot2_1.0.0         plyr_1.8.1
 [4] inSilicoMerging_1.8.6 DWD_0.11              Matrix_1.1-4
 [7] siggenes_1.38.0       multtest_2.20.0       knitr_1.6
[10] Biobase_2.24.0        BiocGenerics_0.10.0   magrittr_1.1.0

loaded via a namespace (and not attached):
 [1] codetools_0.2-9  colorspace_1.2-4 digest_0.6.4     evaluate_0.5.5
 [5] formatR_0.10     grid_3.1.1       gtable_0.1.2     lattice_0.20-29
 [9] MASS_7.3-34      munsell_0.4.2    proto_0.3-10     Rcpp_0.11.2
[13] reshape2_1.4     scales_0.2.4     stats4_3.1.1     stringr_0.6.2
[17] survival_2.37-7  tools_3.1.1
```