**Title:** Project RSI Translation from CCLE: Initial Comparisons
**Project:** RSI Translation from CCLE
**CICPT:** 007
**Date:** 07 May, 2020
**Investigators:** ,
**Organization:**

## Contents

# 1 Overview

RSI was originally developed on the Affymetrix platform and has been validated in a number of Affymetrix-based GeneChips. It has also been translated to other microarray platforms based on sequence similarity among probes. The translation to RNASeq is important, however many RNASeq platforms provide gene-level quantification of expression. This is different from the microarray-based approach where specific regions are identified to be reporters for the entire gene. Of course, the term gene is used loosely and what is more specifically considered are transcripts. In the case of a microarray, the probe may cover a region/exon that is common among most (if not all) transcripts for the gene. RNASeq, in contrast, may align all reads to a specific transcript representing the gene (depending on the specific gene model used).

With respect to the ten genes that make up the RSI signature, the Affymetrix probesets for many of these genes are reasonable although likely 3' biased. Although not part of this specific study, the nature and correlation structure of these genes (particularly on the Affy array) is under consideration. Here, suffice it to say that we have frequently observed many genes that correlate well across platforms, etc. Several genes are consistently problematic, including PAK2. The probeset for PAK2 that the U133A/Plus detects is beyond the last exon of the gene, therefore it is likely to not be an accurate representation of PAK2.

This experiment utilizes the CCLE cell line dataset that has been assayed on both the HG-U133Plus 2.0 array and RNASeq over approximately 1,000 cell lines. This offers us a unique opportunity to evaluate the gene expression of the ten RSI genes in a clean setup (no tissue contamination). We will examine the similarity in expression for these reporters, including looking at transcript-level expression. The goal is to determine if RNASeq can be used for RSI, and which specific gene/transcript elements to use for the most accurate translation.

# 2 Setup

```
suppressPackageStartupMessages({
library(affy)
library(readxl)
library(ggplot2)
library(ggpubr)
  library(knitr)
})
```

The data has been pre-processed (downloaded from CCLE web site) and organized. We have developed the translation table internally for mapping probe identities across platforms.

```
ccle_cel<-readRDS(file="../data.raw/ccle_cel.rds")
ccle_rpkm<-readRDS(file="../data.raw/ccle_rnaseq_genes_rpkm.rds")
ccle_rsem<-readRDS(file="../data.raw/ccle_rnaseq_genes_rsem.rds")
ccle_rsem_transcripts<-readRDS(file="../data.raw/ccle_rnaseq_transcripts_rsem.rds")

translation_table<-read_excel("../../../datasets/Translation.xlsx")
```

```
## New names:
```

```
## * `` -> ...11
```

For the RNASeq data, we take the log values so they are comparable to microarray (which have been RMA-normalized and are thus on the log scale). A small value (+1) is added to avoid negative expression values (floor).

```
exprs(ccle_rpkm)<-log2(exprs(ccle_rpkm)+1)
exprs(ccle_rsem)<-log2(exprs(ccle_rsem)+1)
exprs(ccle_rsem_transcripts)<-log2(exprs(ccle_rsem_transcripts)+1)
```

## 3 Initial RSI Comparison

We have performed this experiment in the past, namely to just apply the RSI algorithm to RNASeq data from the same genes. We do that here and compare the results.

```
source("../src/RNASeqPredictor.R")
source("../src/U133Predictor.R")
cel_rsi<-predict.u133.rsi.rank(exprs(ccle_cel))
rpkm_rsi<-predict.rnaseq.rsi.rank(exprs(ccle_rpkm))
rsem_rsi<-predict.rnaseq.rsi.rank(exprs(ccle_rsem))

write.table(file="../data.derived/rsi_results_ccle.txt", quote=F,sep="\t",col.names=FALSE,
            data.frame(Samples=cel_rsi$Patient.ID,
                       CEL=cel_rsi$RSI.Rank,
                       RPKM=rpkm_rsi$RSI.Rank,
                       RSEM=rsem_rsi$RSI.Rank
                       )
            )
```

We can visualize this several ways, the most obvious being a scatter plot.

```
df<-data.frame(
  sample=rpkm_rsi$Patient.ID,
  rpkm=rpkm_rsi$RSI.Rank,
  rpkm_error=abs(rpkm_rsi$RSI.Rank-cel_rsi$RSI.Rank),
  rsem=rsem_rsi$RSI.Rank,
  rsem_error=abs(rsem_rsi$RSI.Rank-cel_rsi$RSI.Rank),
  cel=cel_rsi$RSI.Rank
)

a<-ggplot(df, aes(x=cel, y=rpkm, fill=rpkm_error>0.1)) +
  geom_point(shape = 21,  col="black", size=2) +
  geom_abline(slope=1, intercept=0, col="black", linetype=2) +
  labs(caption=sprintf("R=%5.2f",cor(df$cel, df$rpkm))) +
  xlab("CEL based RSI") +
  ylab("RPKM based RSI") +
  scale_fill_manual(values=c("#2b83ba","#d7191c"), name="RSI error", labels=c("<0.1",">0.1")) +
  theme_bw()
```

```
b<-ggplot(df, aes(x=cel, y=rsem, fill=rsem_error>0.1)) +
  geom_point(shape = 21,  col="black", size=2) +
  geom_abline(slope=1, intercept=0, col="black", linetype=2) +
  xlab("CEL based RSI") +
  ylab("RSEM based RSI") +
  labs(caption=sprintf("R=%5.2f",cor(df$cel, df$rsem))) +
  scale_fill_manual(values=c("#2b83ba","#d7191c"), name="RSI error", labels=c("<0.1",">0.1")) +
  theme_bw()

ggarrange(a, b, ncol=2, nrow=1, labels="AUTO", common.legend=TRUE)
```

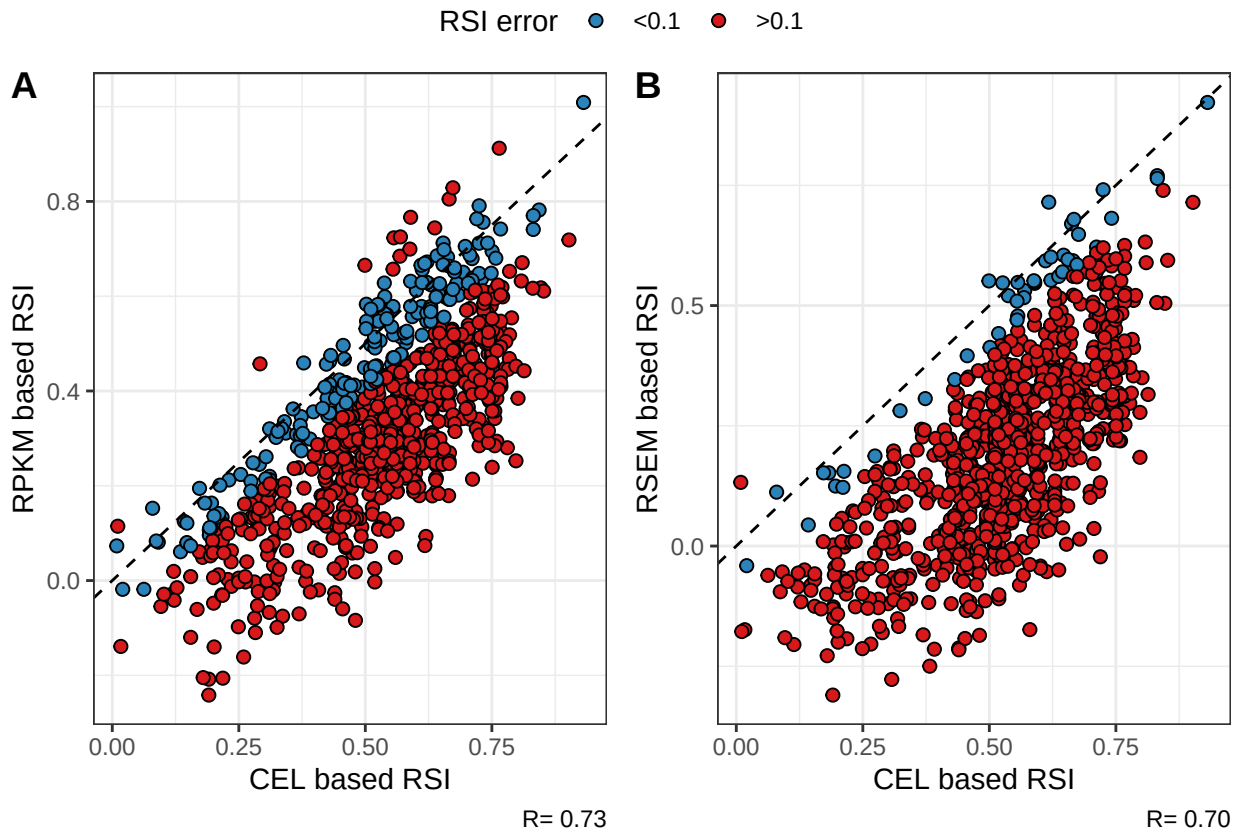



Figure 1: RSI values in CCLE computed from CEL files, RPKM and RSEM data. A) RSI in CEL files vs RPKM expression. B) RSI in CEL files vs RSEM expression.

The conclusion from this comparison is that RSI is not too different (correlation-wise), but is clearly shifted (RNASeq RSI is generally lower) and likely not within a tolerable error level.

## 4 Gene Comparisons

One reason that RSI can be different between platforms is that the gene expression of the ten individual genes can differ. The reasons for this are extensive, however we can empirically determine the degree of similarity of gene expression for these genes across CCLE.

### 4.1 Gene identification

We extract the ten genes from each dataset. For the Affymetrix platform, we know exactly which one to use. For RNASeq, we map to the Ensembl Gene identifier then consider whatever version is available in the RNASeq data. Note: The gene order for RSEM and RPKM differ. The resulting data frames have the genes in the same order, based on the translation table.

```
ccle_cel_tengenes<-ccle_cel[translation_table$`HG-U133 Plus 2.0 Probeset`,]
f<-1:dim(ccle_rpkm)[1]
names(f)<-sub("\\..+$","",featureNames(ccle_rpkm))
ccle_rpkm_tengenes<-ccle_rpkm[f[translation_table$`Ensembl Gene`],]

f<-1:dim(ccle_rsem)[1]
names(f)<-sub("\\..+$","",featureNames(ccle_rsem))
ccle_rsem_tengenes<-ccle_rsem[f[translation_table$`Ensembl Gene`],]
```

### 4.2 Gene-level comparisons

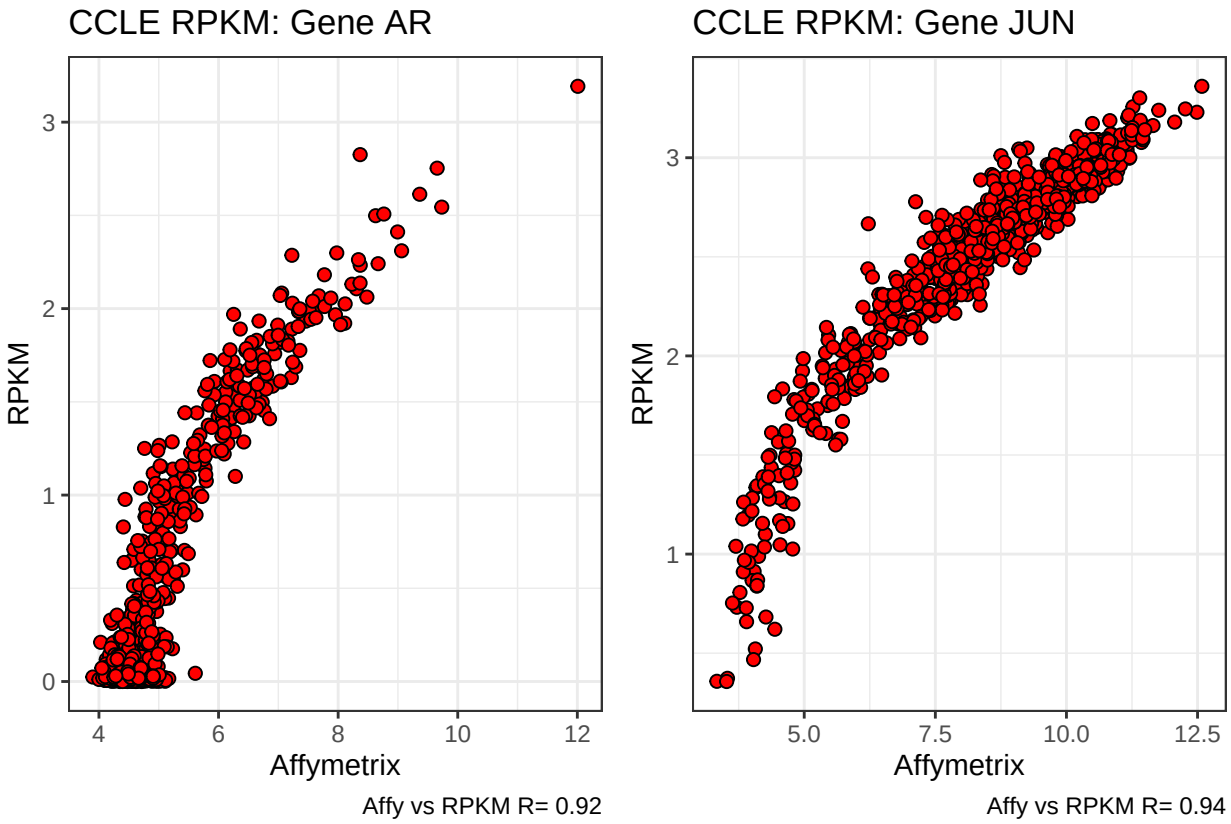
Plot the gene expression between the Affymetrix GeneChip (gold standard) and the two variations of RNASeq quantification.

```
rpkm_comparison_graphs<-list()
rsem_comparison_graphs<-list()

for (gene_index in 1:10) {
  gene_symbol<-translation_table$`Gene Symbol`[gene_index]
  df<-data.frame(
    affy=exprs(ccle_cel_tengenes)[gene_index,],
    rpkm=log2(exprs(ccle_rpkm_tengenes)[gene_index,]+1),
    rsem=exprs(ccle_rsem_tengenes)[gene_index,],
    site=ccle_rpkm_tengenes$Site_Primary
  )

  g<-ggplot(df, aes(x=affy, y=rpkm)) +
    geom_point(size=2,shape=21, fill="red") +
    theme_bw() +
    ggtitle(sprintf("CCLE RPKM: Gene %s",gene_symbol)) +
    xlab("Affymetrix") +
    ylab("RPKM") +
    labs(caption=sprintf("Affy vs RPKM R=%5.2f\n",cor(df$affy, df$rpkm)))

  rpkm_comparison_graphs[[gene_symbol]]<-g



  g<-ggplot(df, aes(x=affy, y=rsem)) +
    geom_point(size=2,shape=21, fill="red") +
    theme_bw() +
```

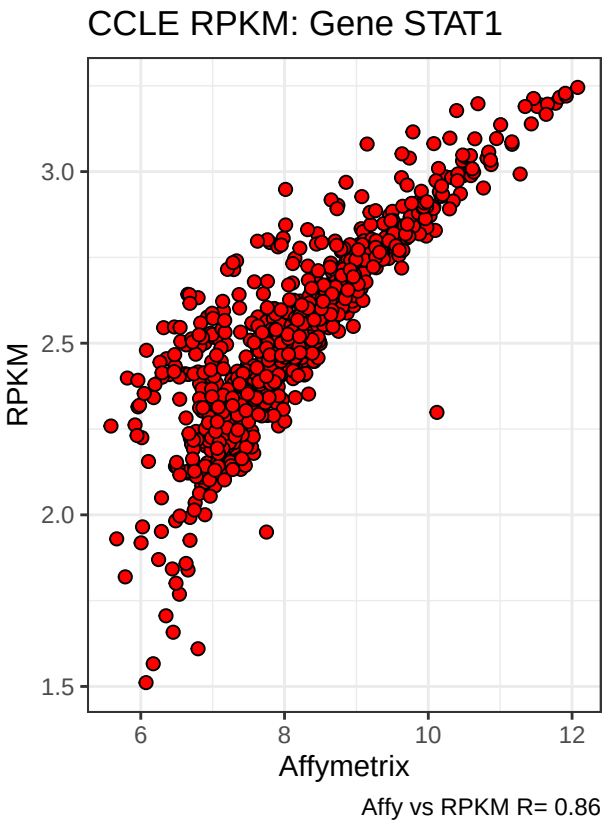
```
    ggtitle(sprintf("CCLE RSEM: Gene %s",gene_symbol)) +
    xlab("Affymetrix") +
    ylab("RSEM") +
    labs(caption=sprintf("Affy vs RSEM R=%5.2f\n", cor(df$affy, df$rsem)))

  rsem_comparison_graphs[[gene_symbol]]<-g

}
```

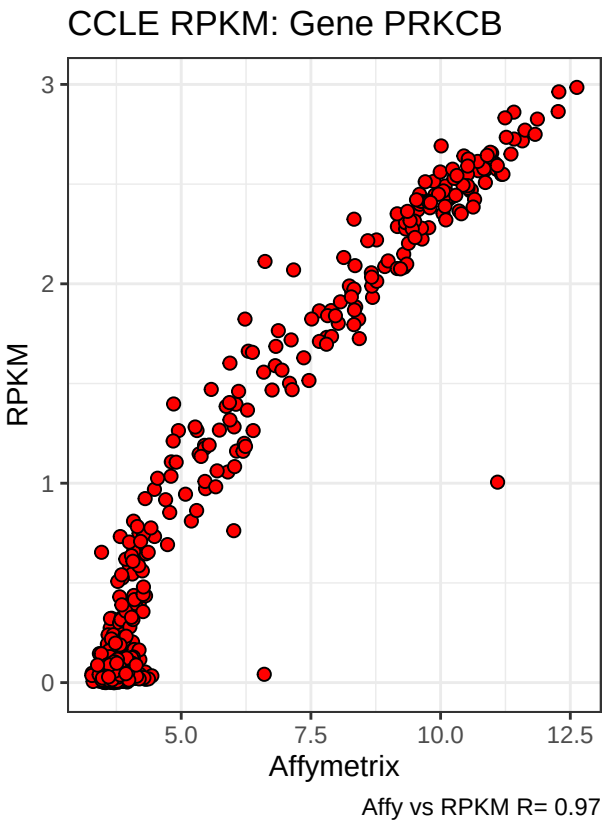
```
ggarrange(plotlist=rpkm_comparison_graphs, ncol=2, common.legend=TRUE)
```

```
## $`1`
```

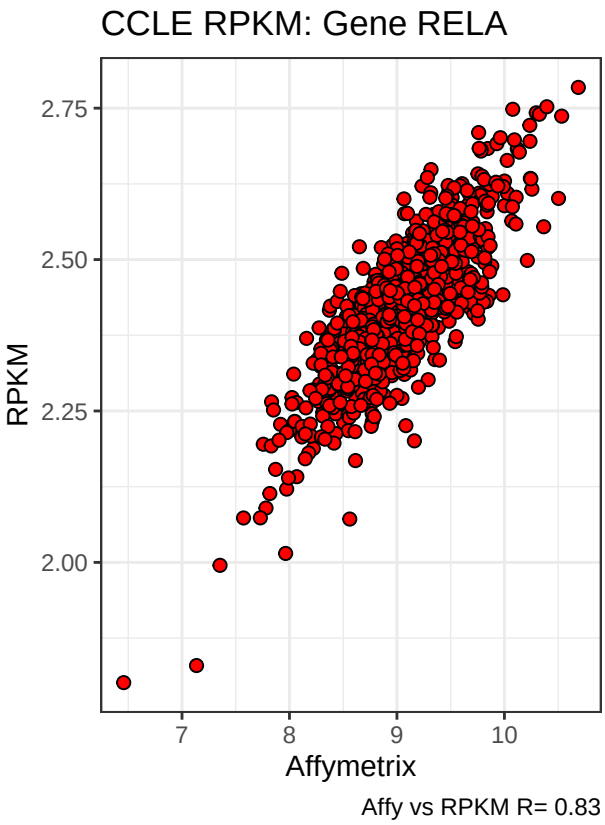


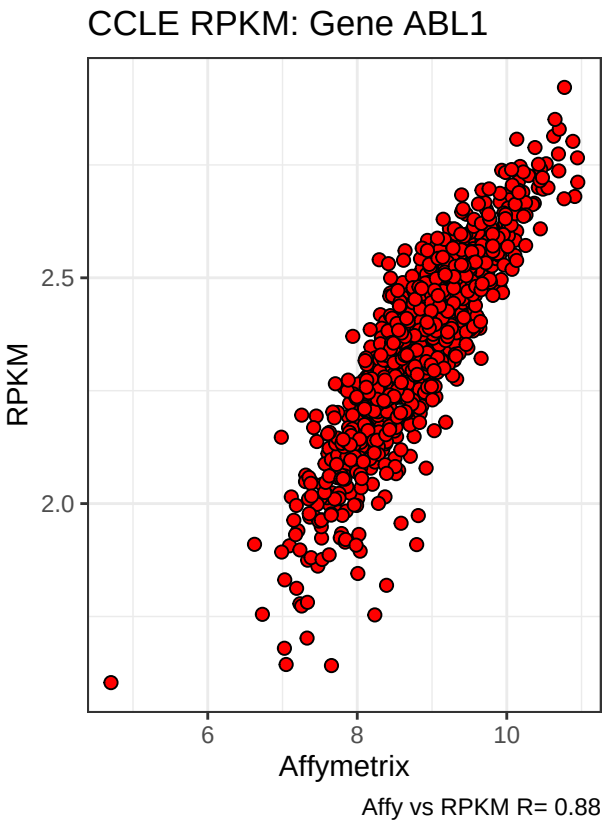

```
##
## $`2`
```


CCLE RPKM: Gene STAT1
RPKM
3.0
2.5
2.0
1.5
6
8
10
12
Affymetrix
Affy vs RPKM R= 0.86

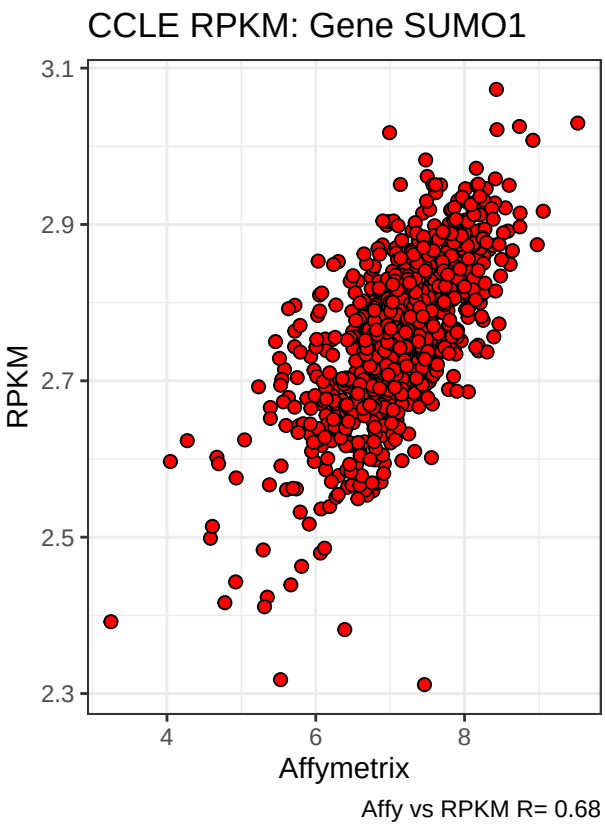

CCLE RPKM: Gene PRKCB
RPKM
3
2
1
0
5.0
7.5
10.0
12.5
Affymetrix
Affy vs RPKM R= 0.97

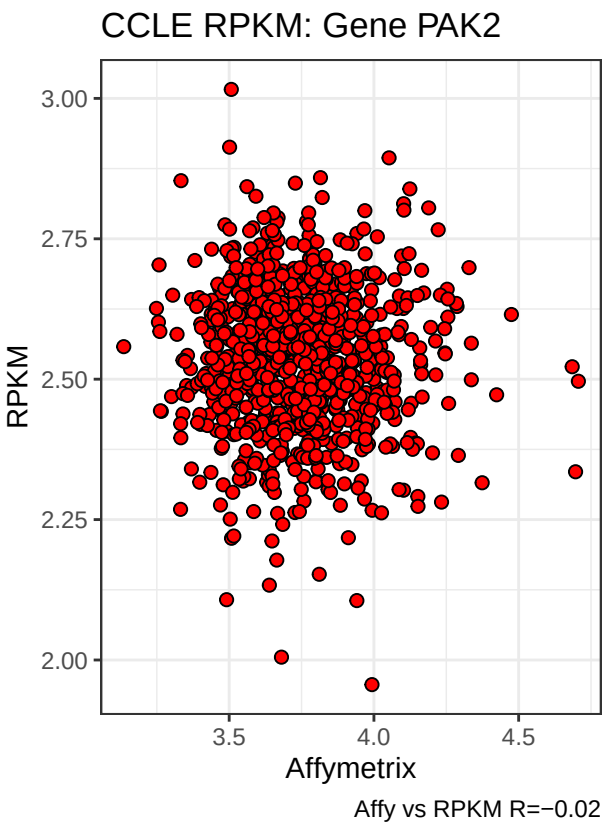
```
##
## $`3`
```


CCLE RPKM: Gene RELA
RPKM
2.75
2.50
2.25
2.00
7
8
9
10
Affymetrix
Affy vs RPKM R= 0.83

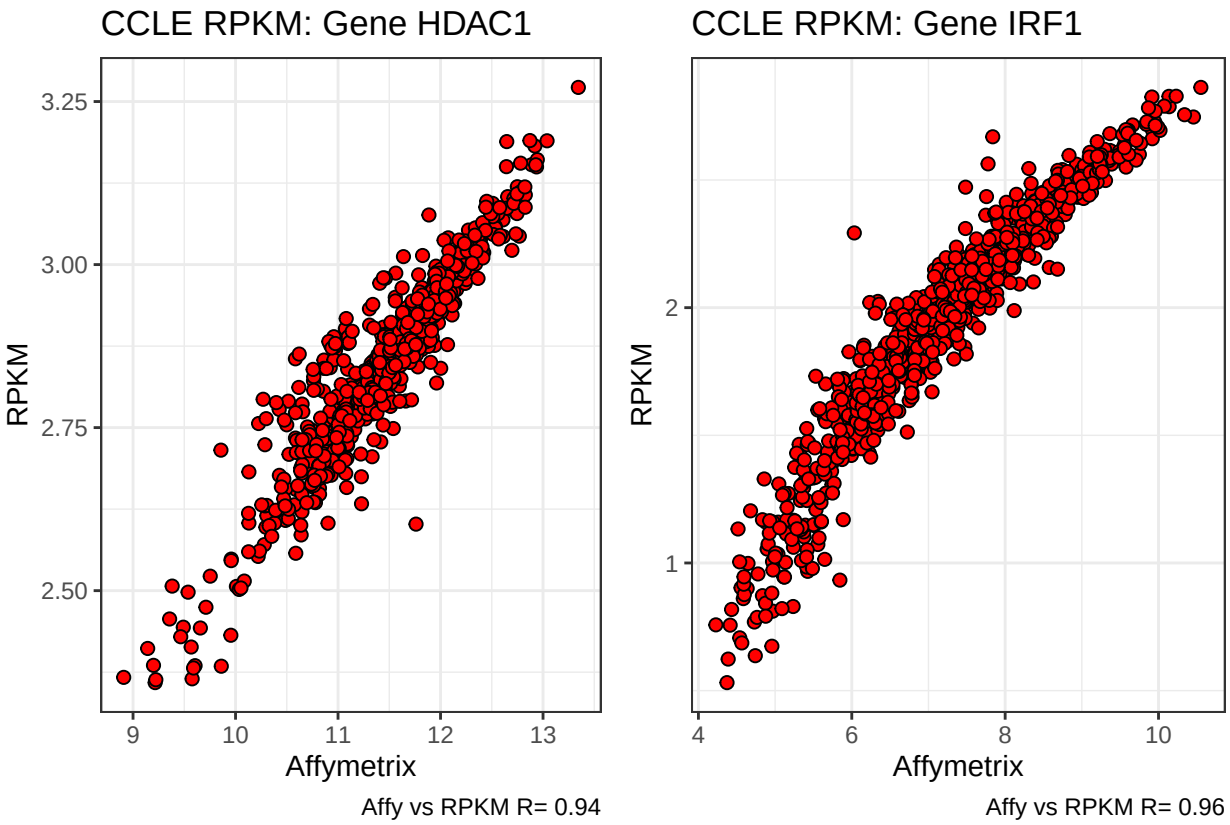

CCLE RPKM: Gene ABL1
RPKM
2.5
2.0
6
8
10
Affymetrix
Affy vs RPKM R= 0.88

```
##
## $`4`
```


CCLE RPKM: Gene SUMO1
RPKM
Affymetrix
3.1
2.9
2.7
2.5
2.3
4
6
8
Affy vs RPKM R= 0.68


CCLE RPKM: Gene PAK2
RPKM
Affymetrix
3.00
2.75
2.50
2.25
2.00
3.5
4.0
4.5
Affy vs RPKM R=−0.02

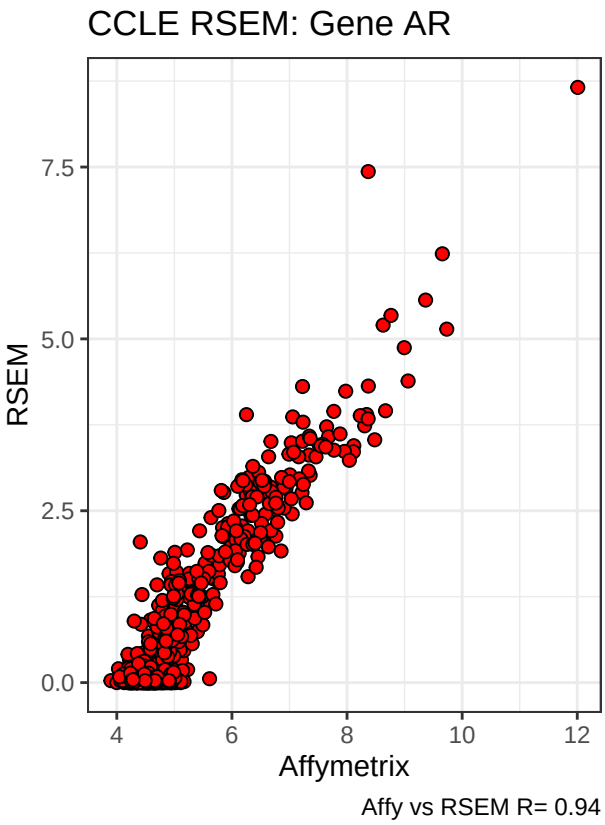
##
## $`5`

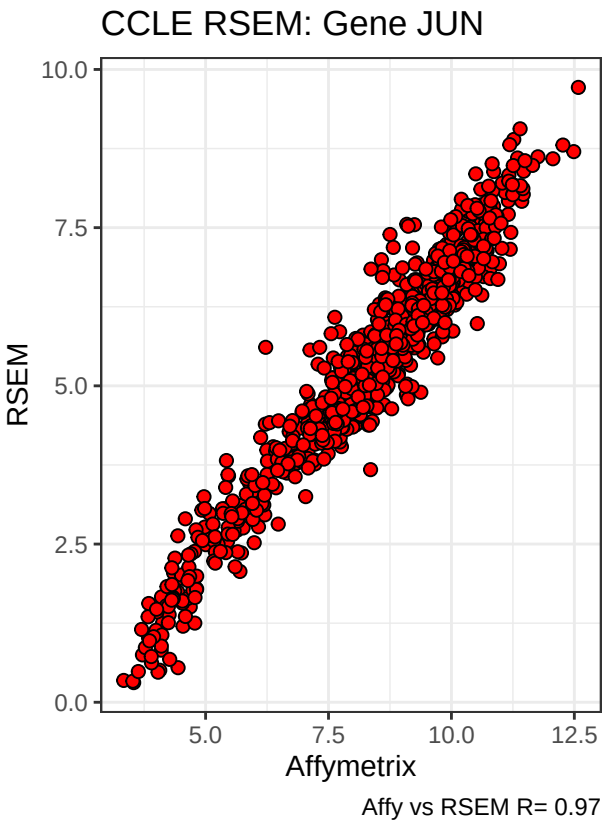



```
##
## attr(,"class")
## [1] "list"      "ggarrange"
```

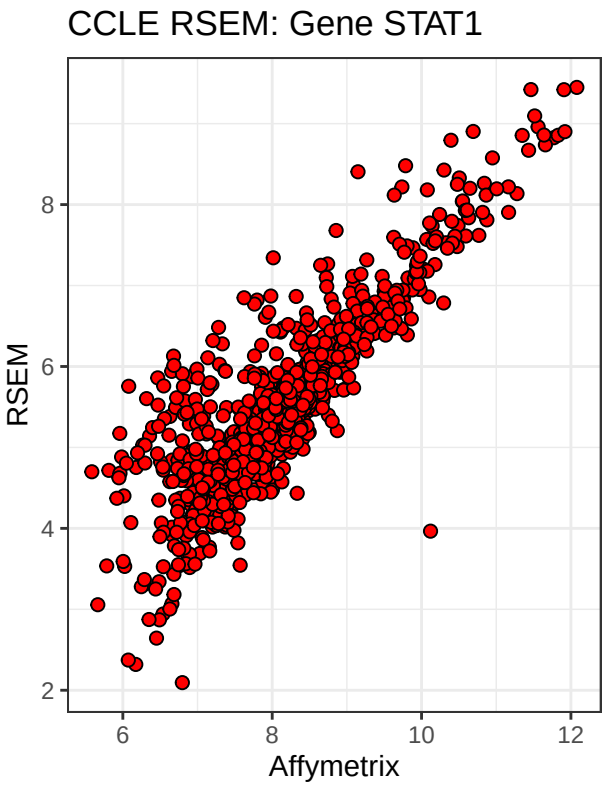
```
ggarrange(plotlist=rsem_comparison_graphs, ncol=2, common.legend=TRUE)
```

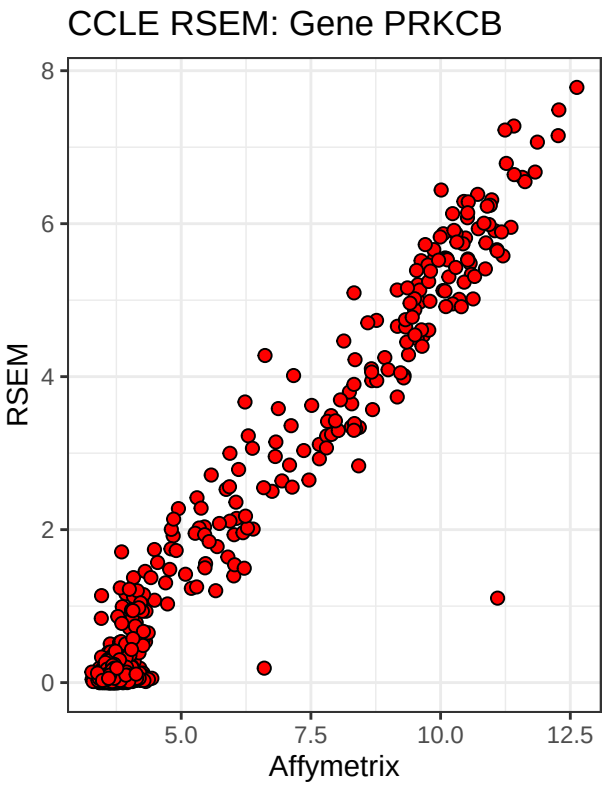
```
## $`1`
```


CCLE RSEM: Gene AR
RSEM
Affymetrix
0.0
2.5
5.0
7.5
4
6
8
10
12
Affy vs RSEM R= 0.94

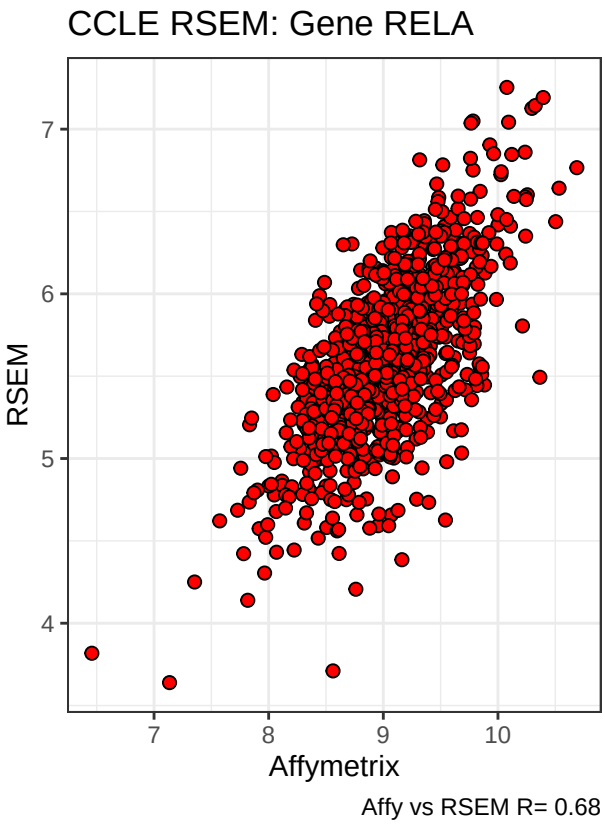

CCLE RSEM: Gene JUN
RSEM
Affymetrix
0.0
2.5
5.0
7.5
10.0
5.0
7.5
10.0
12.5
Affy vs RSEM R= 0.97

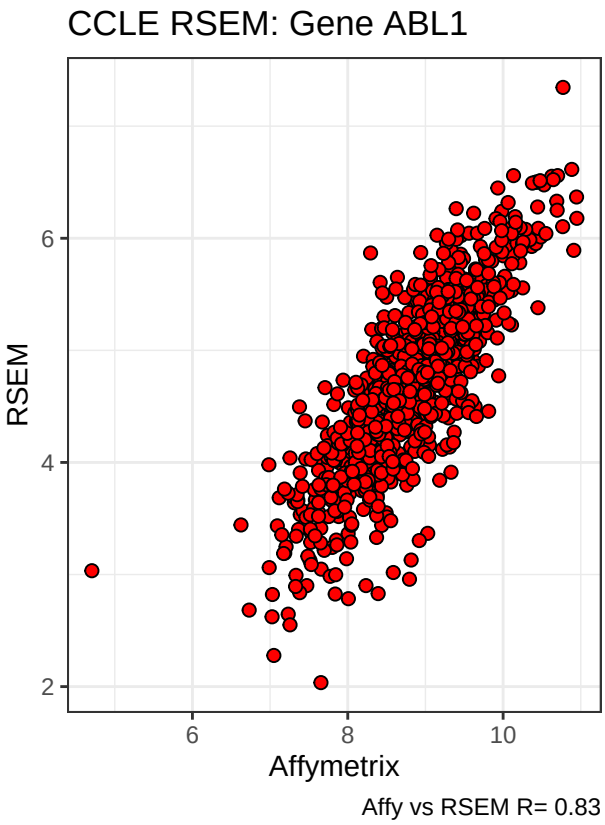
##
## $`2`


CCLE RSEM: Gene STAT1
RSEM
Affymetrix
2
4
6
8
6
8
10
12
Affy vs RSEM R= 0.86

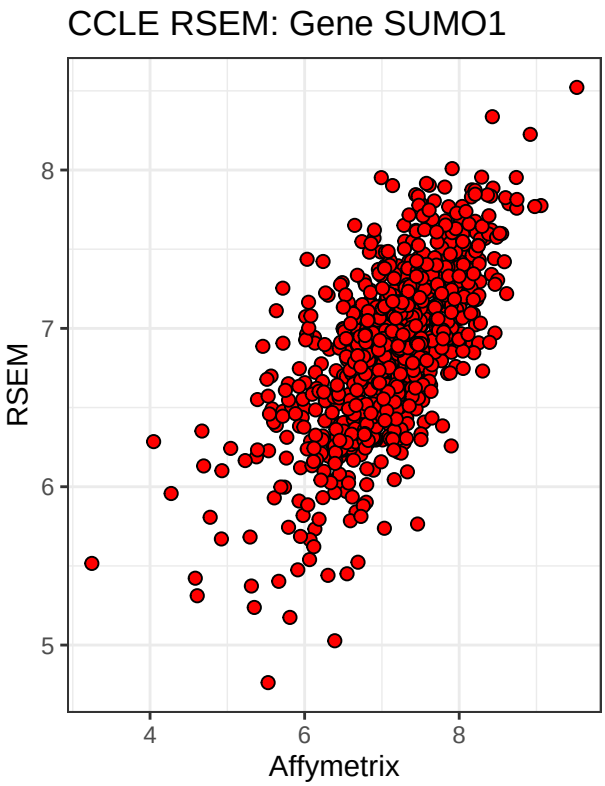

CCLE RSEM: Gene PRKCB
RSEM
Affymetrix
0
2
4
6
8
5.0
7.5
10.0
12.5
Affy vs RSEM R= 0.98

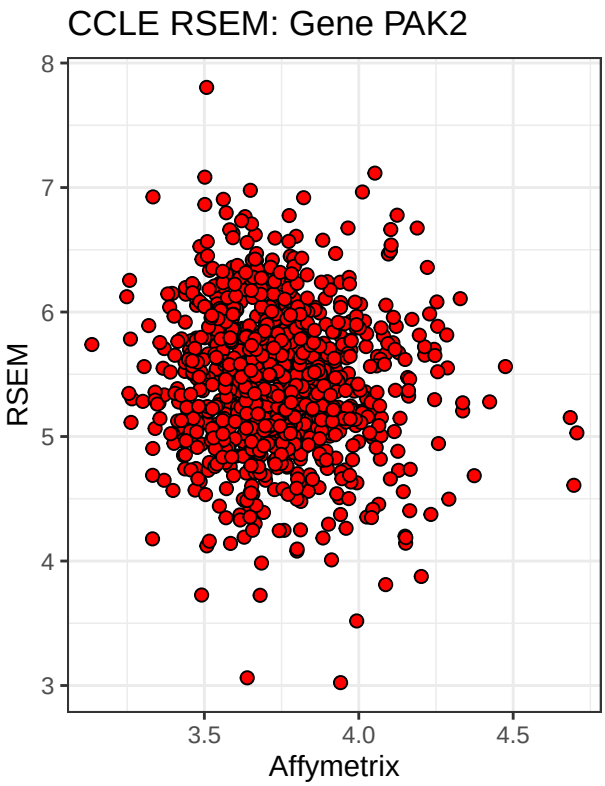
```
##
## $`3`
```


CCLE RSEM: Gene RELA
RSEM
Affymetrix
7
6
5
4
7
8
9
10
Affy vs RSEM R= 0.68

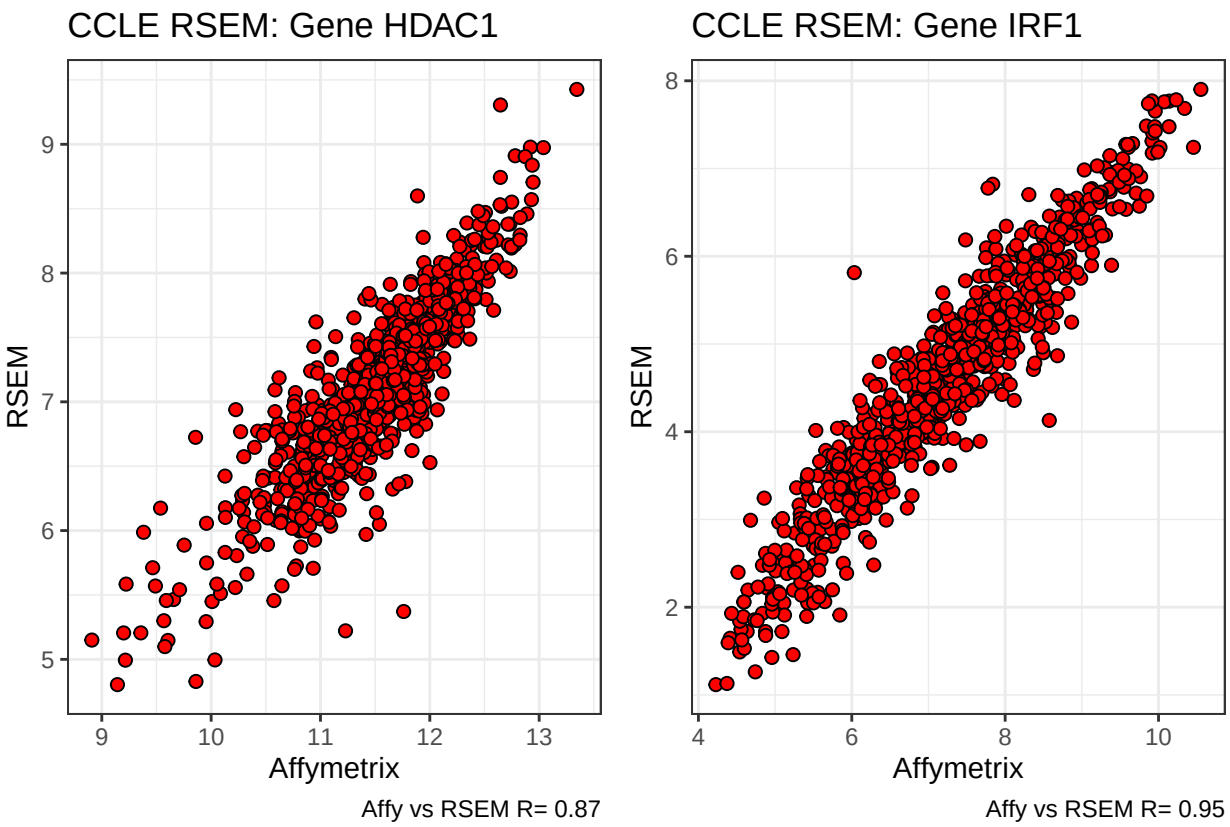

CCLE RSEM: Gene ABL1
RSEM
Affymetrix
6
4
2
6
8
10
Affy vs RSEM R= 0.83

```
##
## $`4`
```


CCLE RSEM: Gene SUMO1
RSEM
Affymetrix
Affy vs RSEM R= 0.67


CCLE RSEM: Gene PAK2
RSEM
Affymetrix
Affy vs RSEM R=−0.08

```
##
## $`5`
```




```
##
## attr(,"class")
## [1] "list"      "ggarrange"
```

### 4.3 Gene-level correlation to RSI

Instead of looking at the constituent parts of RSI across platforms, we can ask a more direction question: "What genes correlate with RSI?". That is, can we find a single gene that individually has good correlation with RSI? Perhaps these could be used by themselves to simply estimate RSI.

```
suppressWarnings(rsi_cors_rpkm<-apply(exprs(ccle_rpkm),1,function(y){cor(y,ccle_cel$RSI)}))
suppressWarnings(rsi_cors_rsem<-apply(exprs(ccle_rsem),1,function(y){cor(y,ccle_cel$RSI)}))
```

If we consider an R value of 0.5 as the minimally acceptable correlation threshold, what is correlated to RSI?

```
# This is ABL
u<-rsi_cors_rpkm[which(rsi_cors_rpkm>0.5)]
# This is IRF1
d<-rsi_cors_rpkm[which(rsi_cors_rpkm< -0.5)]
df<-data.frame(R=c(u,d), Gene=c("ABL","IRF1","PSMB10"))
kable(df, caption="RPKM genes most correlated with RSI (R=0.5 threshold).")
```

Table 1: RPKM genes most correlated with RSI (R=0.5 threshold).

| | R | Gene |
|---|---|---|
| ENSG00000097007.13 | 0.5723796 | ABL |
| ENSG00000125347.9 | -0.5944863 | IRF1 |
| ENSG00000205220.7 | -0.5225072 | PSMB10 |

```
# Nothing is correlated 0.5
u<-rsi_cors_rsem[which(rsi_cors_rsem>0.5)]
# This is IRF1
d<-rsi_cors_rsem[which(rsi_cors_rsem< -0.5)]
df<-data.frame(R=d, Gene=c("IRF1"))
kable(df,caption="RSEM genes most correlated with RSI (R=0.5 threshold).")
```

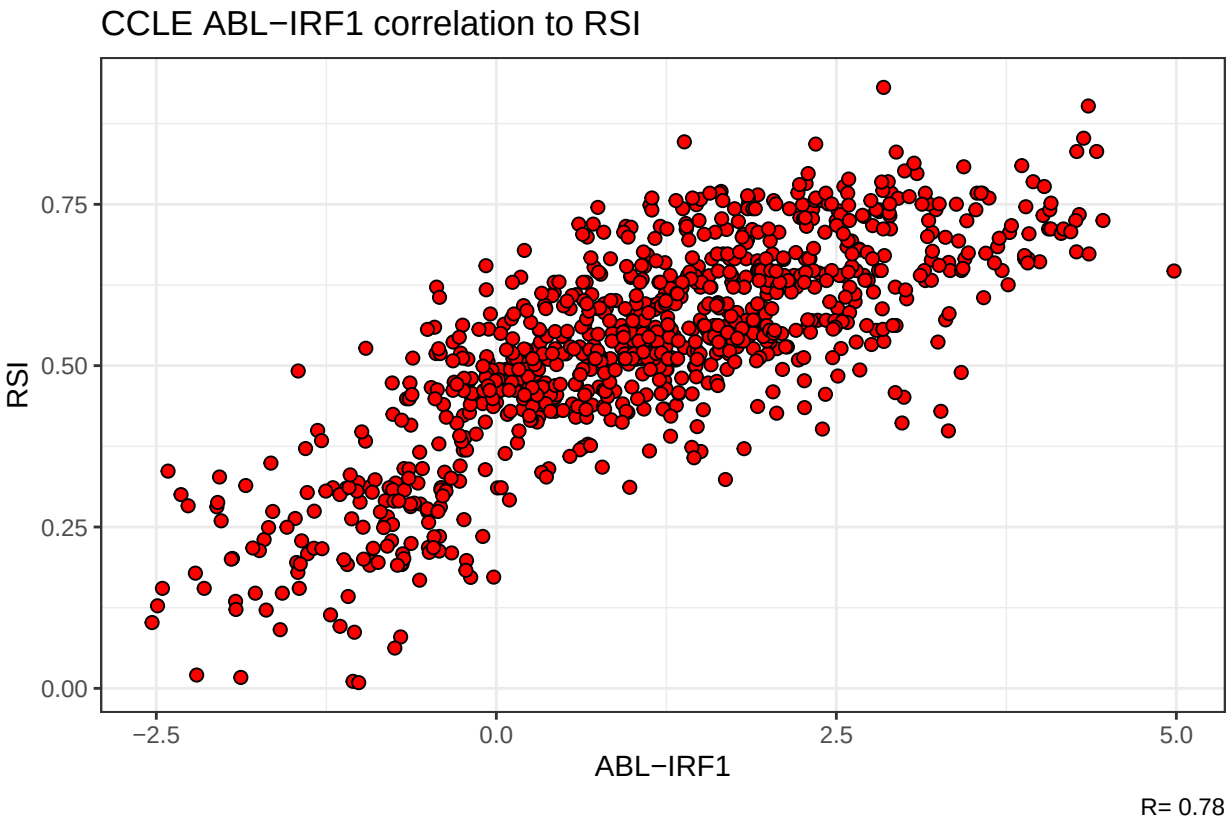
Table 2: RSEM genes most correlated with RSI (R=0.5 threshold).

| | R | Gene |
|---|---|---|
| ENSG00000125347.9 | -0.5431636 | IRF1 |

It is interesting that ABL and IRF1 have higher correlations with RSI. IRF1 is the most negative coefficient (-0.04) while ABL has the highest positive coefficient (0.107). Therefore, we already know they control much of the RSI result.

As a final curiosity, how well does the ratio of ABL and IRF1 correlate with RSI? More precisely, the difference in the log2 gene expression (which is the ratio in unlogged space).

```
df<-data.frame(
  IRF1=exprs(ccle_rpkm)["ENSG00000125347.9",],
  ABL=exprs(ccle_rpkm)["ENSG00000097007.13",],
  ratio=exprs(ccle_rpkm)["ENSG00000097007.13",]-exprs(ccle_rpkm)["ENSG00000125347.9",],
  RSI=ccle_cel$RSI
)
ggplot(df, aes(x=ratio, y=RSI)) +
    geom_point(size=2,shape=21, fill="red") +
    theme_bw() +
    ggtitle(sprintf("CCLE ABL-IRF1 correlation to RSI")) +
    xlab("ABL-IRF1") +
    ylab("RSI") +
    labs(caption=sprintf("R=%5.2f\n", cor(df$ratio, df$RSI)))
```




This is an interesting result, in that the correlation (~0.78) is pretty decent although the scales are clearly farther off. Also of note, this correlation is higher than just using RSI out-of-the-box in the RNASeq platform.

### 4.4 Gene Level Summary

Some conclusions - 8/10 genes correlate well between affy and RNASeq - There is a difference between RSEM and RPKM, for at least one gene. This should be investigated to determine which approach is preferred. - Gene expression ranges (and relationship to each other) will likely be the challenge in the translation. - Ratio of two genes (IRF1 and ABL) do surprisingly well and they translate well, so a new model is reasonable to consider. - PAK2 and SUMO1 have worse correlation than the other genes and are essentially unusable in RNASeq at this point. We can compare these at the transcript level to see if the relationship improves.

## 5 Transcript Comparisons

Similar to the analysis performed at the gene level, we can also compare RSI and indvidual expression at the transcript level (for RNA). We just repeat the prior analysis with transcripts instead of genes.

```
pak2_rsem_transcript_correlation<-apply(exprs(ccle_rsem_transcripts),1,function(y){cor(y, exprs
sumo1_rsem_transcript_correlation<-apply(exprs(ccle_rsem_transcripts),1,function(y){cor(y,exprs

hist(pak2_rsem_transcript_correlation)
```

```
summary(pak2_rsem_transcript_correlation)

hist(sumo1_rsem_transcript_correlation)
summary(sumo1_rsem_transcript_correlation)
```

PAK2 does not have any transcript expression correlated to the Affymetrix probeset. SUMO1 does not have any transcript level expression that improves upon the gene level (RSEM) expression.

```
rsi_cors_rsem_transcripts<-apply(exprs(ccle_rsem_transcripts),1,function(y){cor(y,ccle_cel$RSI
which(rsi_cors_rsem_transcripts>0.5) # ABL1

which(rsi_cors_rsem_transcripts < -0.5) # IRF1, PSMB?
```