Tutorial 2b: Digging Deeper with Drug Response Summarization

Introduction

- IC50 and AUC statistics are designed to summarize drug response curves into a single number. This summarization step facilitates downstream analyses.
- Apart from summarizing drug responses, IC50 and AUC values also provide measures of the effect of drugs on cell lines.
- For an overview about these statistics, have a look at the **Tutorial 1b** ("Exploring Replicability with the `summarizedPharmacoData` Dataset").
- A limitation, of these summary statistics is that they usually require making assumptions about the data.
    - As we will see in this tutorial, some of these assumption might not always hold. When going through this tutorial, try to think about the following question: **Can the inconsistencies between the different studies be attributed to the modelling assumptions?**

Setup Workspace

We start by loading the tidyverse family of packages and specifying a default plotting theme for our `ggplot` graphics.

```
library(tidyverse)
theme_set(theme_bw())
```

Load Summarized Dataset

We will be using both the raw and summarized pharmacological data in this tutorial.

```
pharmacoData <- readRDS(file.path("..", "data", "rawPharmacoData.rds"))
summarizedData <- readRDS(file.path("..", "data", "summarizedPharmacoData.rds"))
```

Original Summaries

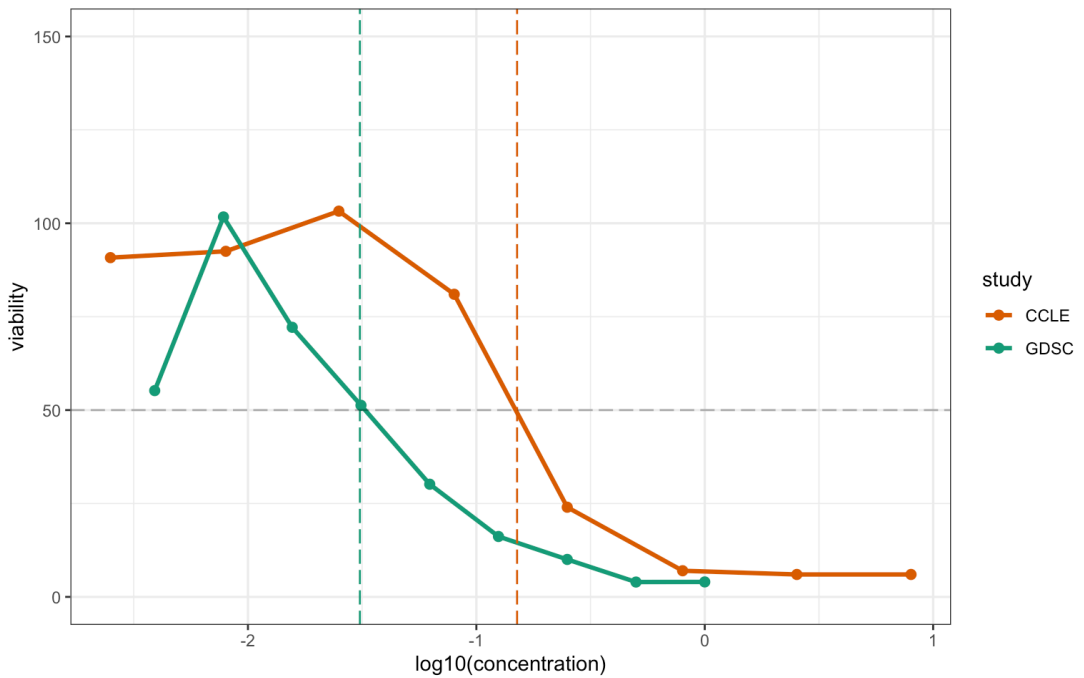
- Let's start by exploring the IC50 and the AUC statistics that were published in the original manuscripts.
- To do this, we'll define a function, `plotResponse`, that allows us to visualize the relation between drug response and drug concentration.
- By writing a function to do the plotting, we reduce the amount of copying and pasting of code in our analysis (which can often introduce unexpected errors!).
- It also allows us to define a consistent way of plotting that can be applied to different subsets of the data.

The ploting code below will visualize the viability scores of a single cell line, `cellLineA`, for a single drug, `drugA`, as a function of the drug concentrations in each study. The vertical dotted lines display the IC50 value published from each study.

```
plotResponse <- function(drugA, cellLineA) {
    pharSub <- filter(pharmacoData, drug == drugA, cellLine == cellLineA)
    sumSub <- filter(summarizedData, drug == drugA, cellLine == cellLineA)
    ggplot(pharSub, aes(x = log10(concentration), y = viability, color = study)) +
        geom_point(size = 2.1) +
        geom_line(lwd = 1.1) +
        ylim(0, 150) +
        geom_vline(xintercept = log10(sumSub[,"ic50_CCLE"]),
                   color = "#d95f02", linetype = "longdash") +
        geom_vline(xintercept = log10(sumSub[,"ic50_GDSC"]),
                   color = "#1b9e77", linetype = "longdash") +
        geom_hline(yintercept = 50, col = "#00000050", linetype = "longdash") +
        scale_colour_manual(values = c("CCLE" = "#d95f02", "GDSC" = "#1b9e77")) +
        xlim(range(log10(c(pharSub$concentration, sumSub$ic50_CCLE, sumSub$ic50_GDSC))))
}
```

Let's start by exploring how the response curve for the drug 17−AAG behaves in the cell-line H4. Notice that this drug had consistent viability responses between the two studies.

```
1 plotResponse(drugA = "17-AAG", cellLineA = "H4")
```

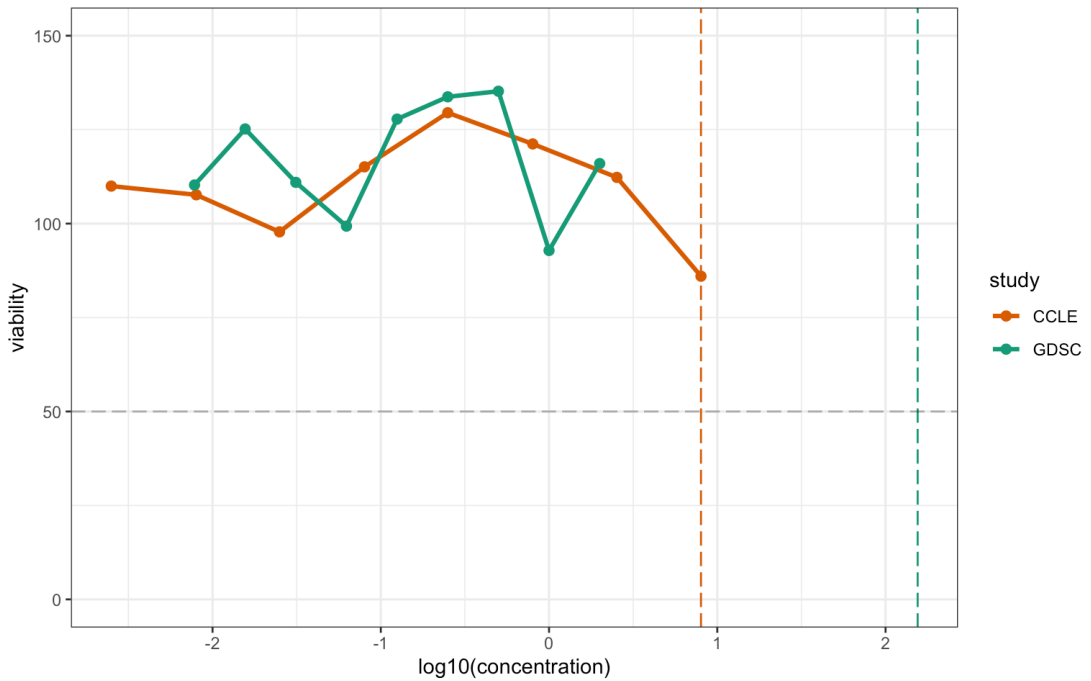


What observations can you draw from this curve? Are the response data holding the assumptions to estimate an IC50 value?

> Place your answer here

Let’s now select another drug and cell line combination.

```
1  plotResponse(drugA = "Nilotinib", cellLineA = "22RV1")
```



Are the reported IC50 values reflecting the actual behaviour of the response curves? How can IC50 values be estimated if there are no viabilities below 50% for the second example? How did the two different studies deal with these cases?

> Place your answer here

Recomputed Summaries

Hopefully it is now clear that there is more than one way to compute the IC50 and AUC statistics.

- Since different approaches were used to compute these values in the original CCLE and GDSC studies, agreement in the raw data may not be translating to agreement in the summarized values (as we saw above).
- This discordance in how the statistics were computed may be contributing to the lower replicability between the two studies.

To address this problem, we have recomputed the IC50 and AUC statistics using a common approach based on a modified logistic regression model.

- Logistic regression is a common approach to modeling viability response curves and both the CCLE and GDSC studies used (different) variants of this model to compute the original summary statistics.
- These values are stored in a separate RDS file, `modelSummarizedPharmacoData.rds`.
- Details on how the regression model was fit and code to regenerate the RDS file are provided in the Supplementary Tutorial, **Supplement: Regression**.

```
1 mySummarizedData <- readRDS(file.path("..", "data", "modelSummarizedPharmacoData.rds"))
```

Let’s start by comparing the agreement across studies for both the original and recomputed statistics. To do this, we start by merging the two datasets.

```
allSummarizedData <- inner_join(summarizedData, mySummarizedData,
                                by = c("drug", "cellLine"),
                                suffix = c("_original", "_updated"))
head(allSummarizedData)
```

```
  cellLine      drug ic50_CCLE_original auc_CCLE_original ic50_GDSC_original
1    22RV1 Nilotinib           8.000000         0.0000000          155.269917
2     5637 Nilotinib           7.475355         0.0072625          219.934550
3    639-V Nilotinib           8.000000         0.0710125           92.177125
4      697 Nilotinib           1.910434         0.1573375            3.063552
5    769-P Nilotinib           8.000000         0.0000000           19.633514
6    786-0 Nilotinib           8.000000         0.0750125          137.066882
  auc_GDSC_original ic50_CCLE_updated auc_CCLE_updated ic50_GDSC_updated
1          0.003935          8.000000       0.01368318         2.0000000
2          0.003616                NA               NA         0.0078125
3          0.007622          8.000000       0.08307900         2.0000000
4          0.069265          2.821824       0.17328185         2.0000000
5          0.028758          7.910106       0.05703088         2.0000000
6          0.005482          8.000000       0.07074420         0.0078125
  auc_GDSC_updated
1      0.006718582
2      0.001520709
3      0.020623143
4      0.084926370
5      0.048912825
6      0.016638998
```

We can also use some tidyverse magic to reorganize this data.

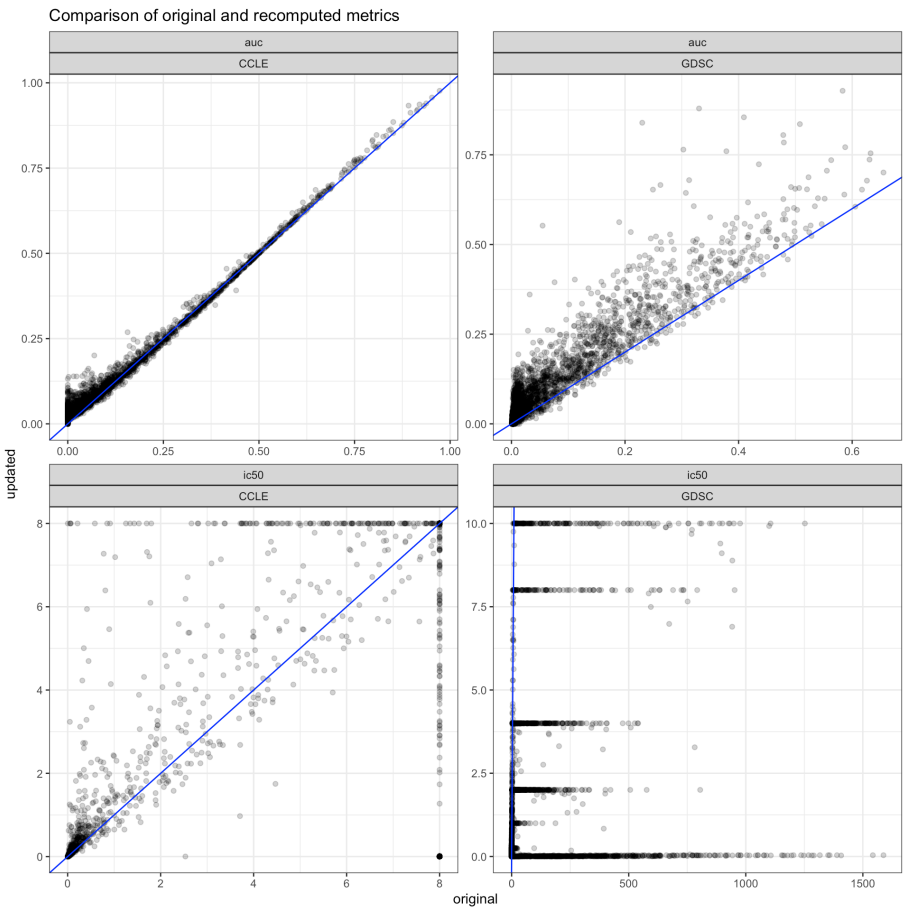
```
longSummarizedData <-
    allSummarizedData %>%
    tidyr::gather(metric, value, -cellLine, -drug) %>%
    tidyr::separate(metric, c("metric", "study", "calc"), sep = "_")
head(longSummarizedData)
```

```
  cellLine      drug metric study     calc    value
1    22RV1 Nilotinib   ic50  CCLE original 8.000000
2     5637 Nilotinib   ic50  CCLE original 7.475355
3    639-V Nilotinib   ic50  CCLE original 8.000000
4      697 Nilotinib   ic50  CCLE original 1.910434
5    769-P Nilotinib   ic50  CCLE original 8.000000
6    786-0 Nilotinib   ic50  CCLE original 8.000000
```

Notice now that each row corresponds to a unique cell line, drug, study, metric value for either the original or recomputed approach.

Using this data, we take a look at how the original and recomputed values compare.

```
longSummarizedData %>%
    tidyr::spread(calc, value) %>%
    ggplot(aes(x = original, y = updated)) +
    geom_point(alpha = 1/5) +
    geom_abline(color = 'blue') +
    facet_wrap(~ metric + study, scales = "free") +
    ggtitle("Comparison of original and recomputed metrics")
```

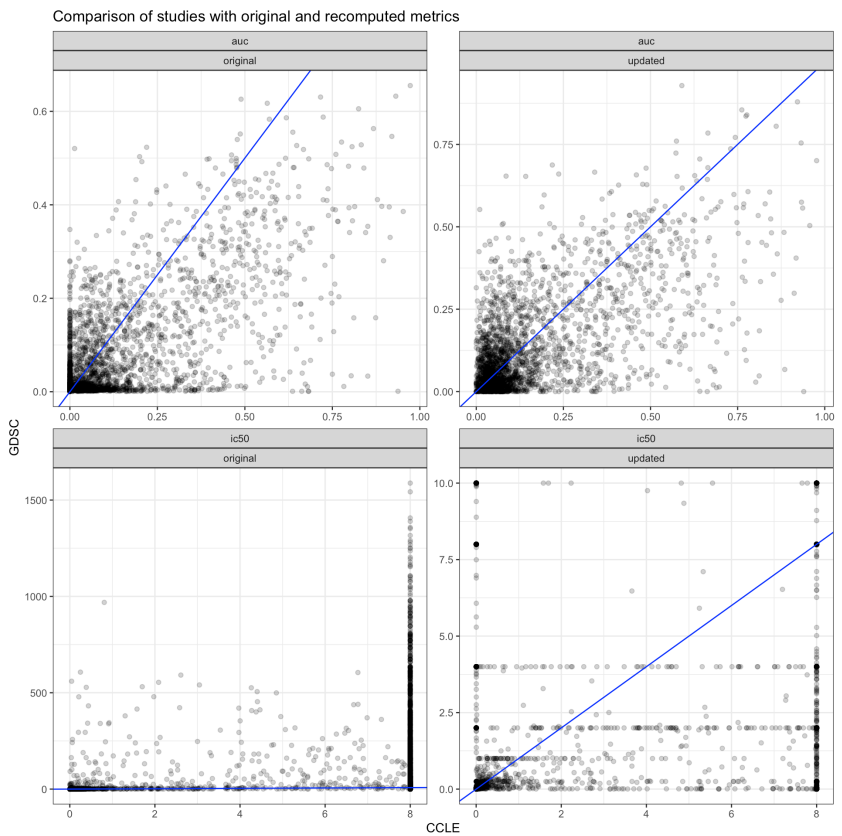


If the values were identical, they would lie across the blue diagonal line. Do the original and recomputed values agree? If not, are there any clear differences between the two? Recall that in an earlier analysis we also looked at IC50 values after log-transformation. How does the plot look after applying a log-transformation to these data?

> Place your answer here

We can also take a look at how the agreement of metrics between studies with both the original and recomputed values.

```
longSummarizedData %>%
    tidyr::spread(study, value) %>%
    ggplot(aes(x = CCLE, y = GDSC)) +
    geom_point(alpha = 1/5) +
    geom_abline(color = 'blue') +
    facet_wrap(~ metric + calc, scales = "free") +
    ggtitle("Comparison of studies with original and recomputed metrics")
```



Consider also computing correlation measures between the two studies with the original and recomputed values. Do the recomputed values improve or change the results of the analysis?

> Place your answer here