Statistical analysis of *in vitro* screening for inhibitors of viral infection

Normalization and target selection methods

Franck Touret, Magali Gilles, Karine Barral, Antoine Nougaire, Etienne Decroly, Xavier de Lamballerie,

2020-04-16

Contents

```
#### Parameters ####

## Significance threshold
alpha <- 0.05

## Highlight colors
markColor <- c(
  cellCtl = "grey",
  virusCtl = "red",
  treated = "blue",
  Arbidol = "black",
  Hydroxychloroquine = "orange"
)

## Highlight point shaapes
markPCh <- c(
  cellCtl = 5,
  virusCtl = 17,
  treated = 1,
  Arbidol = 19,
  Hydroxychloroquine = 13
)
```

Introduction

This document describes in detail the procedure used to select molecules having a potential inhibitory effect on the infection of cultured cells by covid-19.

Pre-publication in bioRxiv:

- In vitro screening of a FDA approved chemical library reveals potential inhibitors of SARS-CoV-2 replication. Franck Touret, Magali Gilles, Karine Barral, Antoine Nougairede, Etienne Decroly, Xavier de Lamballerie, Bruno Coutard. bioRxiv 2020.04.03.023846. DOI 10.1101/2020.04.03.023846

Data

Sources

Doc	URL
bioRxiv article	https://www.biorxiv.org/content/10.1101/2020.04.03.023846v1
Supplementary tables	https://www.biorxiv.org/content/biorxiv/early/2020/04/05/2020.04.03.023846/DC1/embed/media-1.xlsx?download=true

Supplementary data tables

```
#### Directories ####
message("Directories and files")

dir <- c(data = "../data",
         results = "../results",
         figures = "figures")
dir.create(dir["results"], showWarnings = FALSE, recursive = TRUE)

## Data file
supTableFile <- file.path(dir["data"], "suppl-table_Touret-2020.xlsx")
```

```
#### Load data from Excel workbook ####
message("Loading data from excel workbook.")
supTable <- read.xlsx(file = supTableFile, sheetIndex = 1)
#supTable <- read_xlsx(path = supTableFile, sheet = 1, col_names = TRUE)
# dim(supTable)
# View(supTable)
# names(supTable)

colNames <- colnames(supTable)
colNames[1] <- "ID"
colnames(supTable) <- colNames

## Suppress the last row (NA)
supTable <- supTable[!is.na(supTable$ID), ]
# dim(supTable)

## Assign row names for convenience
# View(supTable)


## Extract plate number
supTable$plateNumber <- as.numeric(substr(supTable[, 1], start = 1, stop = 2))
# table(supTable$plateNumber)
plateNumbers <- unique(supTable$plateNumber)

## Assign a color to each molecule according to its plate number
```

```
plateColor <- rainbow(n = length(plateNumbers))
names(plateColor) <- unique(supTable$plateNumber)

supTable$color <- plateColor[supTable$plateNumber]
message("\tLoaded main table with ", nrow(supTable), " rows ")
```

The supplementary table downloaded from bioRxiv contains 1520 molecules.

Viability measurements

Cell Titer Blue intensity (CTB)

The number of viable cells per well is measured by a colorimetric test. The primary measure is the **Cell Titer Blue intensity** (CTB).

```
#### Read the CTB values from the Excel workbook ####
message("Reading Cell Titer Blue (CTB) values from file ", supTableFile)
nbPlates <- 19
rowsPerPlate <- 8
columnsPerPlate <- 12

dataPerPlate <- list()

## Control 1: uninfected cells
cellControl <- data.frame(matrix(ncol = 8, nrow = nbPlates))
colnames(cellControl) <- LETTERS[1:rowsPerPlate]

## Control 2: untreated infected cells
virusControl <- data.frame(matrix(ncol = 6, nrow = nbPlates))
colnames(virusControl) <- LETTERS[3:rowsPerPlate]

## Prepare a table to store the raw data
inhibTable <- data.frame(matrix(ncol = 8, nrow = nbPlates*rowsPerPlate * columnsPerPlate))
colnames(inhibTable) <- c("ID",
                          "Plate",
                          "Row",
                          "Column",
                          "CTB",
                          "cellControl",
                          "virusControl",
                          "Chemical.name")

i <- 2 ## for quick test
for (i in 1:nbPlates) {
  message("\tLoading data from plate ", i)
  sheetName <- paste0("Plate", i)

  ## Raw measures
  # rawMeasures <- read.xlsx(file = supTableFile,
  #                          sheetName = sheetName,
  #                          rowIndex = 30:37,
  #                          colIndex = 2:13, header = FALSE)
  rawMeasures <- read_xlsx(path = supTableFile, col_names = FALSE,
                           sheet = sheetName,
```

```
                            range = "B30:M37", progress = FALSE)
  rawMeasures <- as.data.frame(rawMeasures)
  rownames(rawMeasures) <- LETTERS[1:nrow(rawMeasures)]
  colnames(rawMeasures) <- 1:ncol(rawMeasures)
  # dim(rawMeasures)
  # View(rawMeasures)

  ## Extract control values
  cellControl[i, ] <- as.vector(rawMeasures[,1])
  virusControl[i, ] <- as.vector(rawMeasures[3:8,12])
  platevc <- mean(unlist(virusControl[i, ]))
  platecc <- mean(unlist(cellControl[i, ]))

  ## Extract all values
  r <- 1
  for (r in 1:rowsPerPlate) {
    currentRowName <- LETTERS[r]
    currentValues <- unlist(rawMeasures[currentRowName,])
    id <- paste0(sprintf("%02d",i),
                 currentRowName,
                 sprintf("%02d",1:columnsPerPlate))

    ## Compute the start index for the data table
    startIndex <- (i - 1) * (rowsPerPlate * columnsPerPlate) + (r - 1) * columnsPerPlate + 1
    # message(cat("\t\tIDs\t",  startIndex, id))
    indices <- startIndex:(startIndex + columnsPerPlate - 1)
    # length(indices)
    inhibTable[indices, "ID"] <- id
    inhibTable[indices, "Plate"] <- i
    inhibTable[indices, "Row"] <- currentRowName
    inhibTable[indices, "Column"] <- 1:columnsPerPlate
    inhibTable[indices, "CTB"] <- currentValues
    inhibTable[indices, "virusControl"] <- platevc
    inhibTable[indices, "cellControl"] <- platecc
  }

  dataPerPlate[[i]] <- list()
  dataPerPlate[[i]][["rawMeasures"]] <- rawMeasures
}

# dim(inhibTable)
# names(inhibTable)
# View(inhibTable)
# View(dataPerPlate)
# View(dataPerPlate[[1]][["rawMeasures"]])
# table(inhibTable$Row, inhibTable$Column) ## Check that there are 19 entries for each plate position

## Use the plate well ID as rowname
rownames(inhibTable) <- inhibTable$ID

## Check consistency between IDs in supplementary Touret Table 1
##  and those created here
touretIDs <- unlist(supTable$ID)
```

```r
# length(touretIDs)
inhibIDs <- inhibTable$ID
# length(inhibIDs)

## Cell control: uninfected cells
cellControlIndices <- inhibTable$Column == 1
inhibTable[cellControlIndices, "Chemical.name"] <- "uninfected"

## Virus control: infected cells, no treatment
virusControlIndices <- (inhibTable$Column == 12) & (inhibTable$Row %in% LETTERS[3:8])
# table(virusControlIndices)
inhibTable[virusControlIndices, "Chemical.name"] <- "infected no treatment"

## Define the treatment type
wellType <- NA
wellType[cellControlIndices] <- "cellCtl"
wellType[virusControlIndices] <- "virusCtl"

## All the other ones are treated with a given molecule
wellType[!(virusControlIndices | cellControlIndices)] <- "treated"

inhibTable[, "wellType"] <- wellType


#### Retrieve fields from the bioRxiv supplementary Table 1 ####

for (field in c("Chemical.name",
                "Broad.Therapeutic.class",
                "Reported.therapeutic.effect",
                "Inhibition.Index")) {
  inhibTable[, field] <- NA
  inhibTable[inhibTable$ID %in% touretIDs, field] <-
    as.vector(supTable[, field])
}


# ## Retrieve the molecule names from Table 1 of the bioRxiv workbook
# inhibTable$Chemical.name <- NA
# inhibTable[inhibTable$ID %in% touretIDs, "Chemical.name"] <-
#   as.vector(supTable$Chemical.name)




kable(t(table(inhibTable$wellType)), caption = "Well types. cellCtl: no infection; virusCtl: infection w
```

Table 2: Well types. cellCtl: no infection; virusCtl: infection without treatment; treated: inhected and treated with one molecule

cellCtl	treated	virusCtl
152	1558	114

The raw data contains 19 plates with 8 rows (indiced A to H) and 12 columns (indiced from 1 to 12.)

The raw data consists of CTB measurements in cell cultures.

Distribution of raw CTB values

```
#### Distribution of raw measurements ####
classInterval <- 500
# xmin <- floor(min(inhibTable$CTB)/classInterval) * classInterval
xmin <- 0 ## Intently start the scale at0 to show the remnant CTB
xmax <- ceiling(max(inhibTable$CTB)/classInterval) * classInterval
breaks = seq(from = xmin, to = xmax, by = classInterval)
# range(inhibTable$CTB)

par(mfrow = c(3, 1))
par(mar = c(2,5,3,1))
hist(inhibTable[wellType == "cellCtl", "CTB"],
     main = "Uninfected (cell control)",
     xlab = "CTB",
     ylab = "Number of plate wells",
     las = 1,
     breaks = breaks, col = "palegreen", border = "palegreen")

hist(inhibTable[wellType == "virusCtl", "CTB"],
     main = "Infected, no treatment (virus control))",
     xlab = "CTB",
     ylab = "Number of plate wells",
     las = 1,
     breaks = breaks, col = "orange", border = "orange")

hist(inhibTable[wellType == "treated", "CTB"],
     main = "Treated cells",
     xlab = "CTB",
     ylab = "Number of plate wells",
     las = 1,
     breaks = breaks, col = "#AACCFF", border = "#AACCFF")
```

```
par(par.ori)
```

- ***Cell control***.
  - The top panel (green) shows the distribution of CTB measurements in controls cultures, where the cells were neither infected by the virus nor treated with a drug.
  - Each plate contains 8 wells with uninfected cells (total = 152).
- ***Virus control***.
  - The middle panel (orange) shows the distribution of CTB measurements in infected cells without drug treatment
  - The virus control was performed on 6 wells per plate, total = 114).
- ***Treated cells***.
  - The bottom distribution (pale blue) shows the CTB values for cells infected and treated with a given drug.
  - Note that each drug was tested on a single well (no replicates). Indeed, in order to face the COVID-19 emergency, the study attempted to test as fast as possible a wide range of molecules.

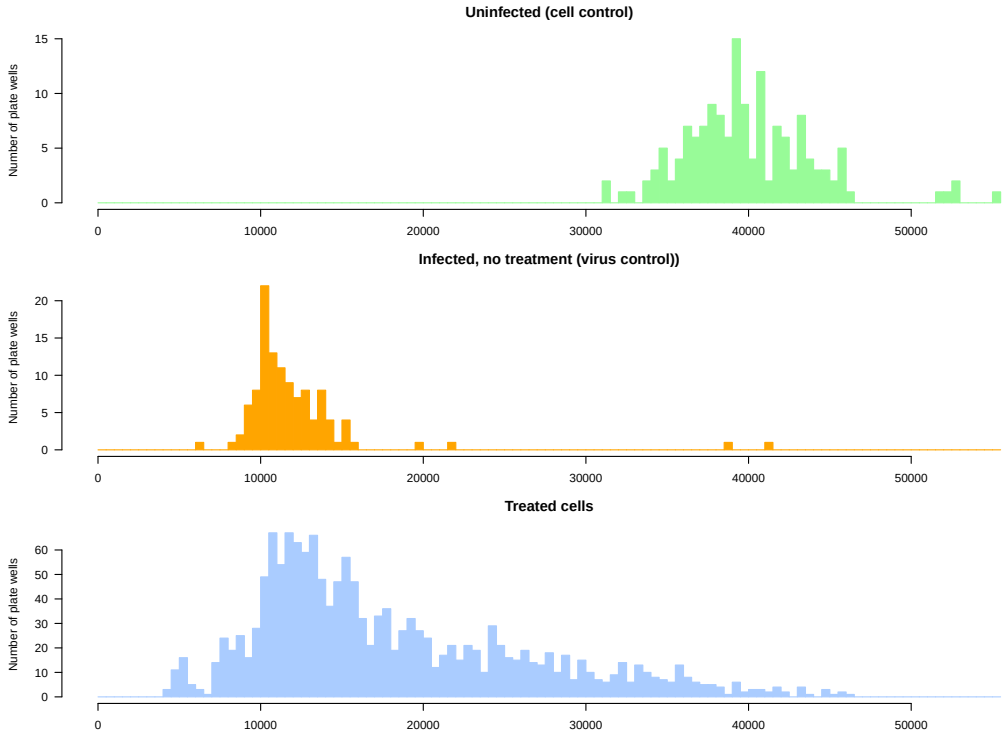

Figure 1: Distributions of the CTB measures. Top: uninfected cells. Middle: infected cells without treatment. Bottom: infected cells treated with a specific molecule.

This first screening was thus performed without replicates. This has to be taken into account for the normalization, which should be done with no estimation of the variance for the individual drugs.

– The distribution is strongly asymmetrical, and seems bi- or multi-modal. This distribution can be considered as a mixture between different distributions;

  * all the drugs that have no inhibitory effect (and are thus expected to have a CTB similar to the virus control);
  * various drugs that inhibit the action of the virus, each one with its specific level of inhibition. This probably corresponds to the widely dispersed values above the bulk of distribution (and above the virus control distribution)

```
#### Plate-wise colors ####

## Assign a color to each plate
## A trick: we alternate the colors of the rainbow in order
## to see the contrast between successive plates
platePalette <- rainbow(n = length(plateNumbers))
plateColor <- vector(length = nbPlates)
oddIndices <- seq(1, nbPlates, 2)
evenIndices <- seq(2, nbPlates, 2)
plateColor[oddIndices] <- platePalette[1:length(oddIndices)]
plateColor[evenIndices] <- platePalette[(length(oddIndices) + 1):nbPlates]
names(plateColor) <- 1:nbPlates

## Assign a color to each result according to its plate
inhibTable$color <- plateColor[inhibTable$Plate]
```

```
inhibTable$pch <- 1 # default point type for dot plots
inhibTable[inhibTable$wellType == "cellCtl", "pch"] <- markPCh["cellCtl"]
inhibTable[inhibTable$wellType == "virusCtl", "pch"] <- markPCh["virusCtl"]
# table(inhibTable$color) ## Check that each plate has 96 wells
# table(inhibTable$pch)
```

Arbidol treatment

A treatment with 10µM Arbidol – a broad-spectrum antiviral – was used as control, with duplicate test in 2 wells per plate.

```
#### Select arbidol duplicates as plate-wise milestones ####
arbidolWells <- (inhibTable$Column == 12) & (inhibTable$Row %in% c("A", "B"))
inhibTable[arbidolWells, "Chemical.name"] <- "Arbidol"
# inhibTable[arbidolWells, c("ID", "Chemical.name")]

#### Extract raw CTB measures per plate for arbidol ####
arbidoltV <- inhibTable[arbidolWells, c("Plate", "CTB")]
inhibTable[arbidolWells, "color"] <- markColor["Arbidol"]
inhibTable[arbidolWells, "pch"] <- markPCh["Arbidol"]


# table(inhibTable$color)
```

Hydroxychloroquine sulfate

We assin a specific label to Hydroxychloroquine sulfate, which has a specific interest since it is one of the molecules tested in an European clinical trial.

```
HOClSindex <- which(inhibTable$Chemical.name == "Hydroxychloroquine sulfate")
HOClSpch <- 13
inhibTable[HOClSindex, "color"] <- markColor["Hydroxychloroquine"]
inhibTable[HOClSindex, "pch"] <- HOClSpch
```

CTB boxplots

```
#### Boxplots per plate  ####

boxplot(CTB ~ Plate + wellType,
        main = "Cell Ttiter Blue (CTB) per plate",
        data = inhibTable,
        las = 1, col = plateColor,
        xlab = "CTB",
        cex.axis = 0.5, cex = 0.5,
        horizontal = TRUE
        )
abline(v = seq(from = 0, to = max(inhibTable$CTB), by = 2000), col = "#EEEEEE", lty = "dashed")
abline(v = seq(from = 0, to = max(inhibTable$CTB), by = 10000), col = "grey")

## Add points to denote the arbidol controls
stripchart(CTB ~ Plate, vertical = FALSE,
           data = inhibTable[arbidolWells, ],
           method = "jitter", add = TRUE, cex = 0.7,
           col = markColor["Arbidol"], pch = markPCh["Arbidol"])
```

```
legend("topright", legend = names(plateColor),
       title = "Plate", fill = plateColor, cex = 0.8)
```

The boxplot of the CTB measurements highlights a plate effect, for the treated cells (middle barplots) but also for the untreated virus control (top boxplots) and uninfected cell control (bottom boxplots).

- **Treated:**
  - The medians and interquartile ranges show strong variations between plates.
  - In particular, plate 1 (in red) has a the smallest median and a remarkably compact interquartile range. There are many statistical outliers (empty circles) in this plate, which might correspond to the molecules having a significant inhibitory effect.
  - In contrast, plates 11 to 15 show a high median and a wide dispersion of CTB measures, and there is not a single statistical outlier.
- **Virus control** (vc)
  - As expected, untreated infected cells generally gave a very small CTB, with small variations (very compact interquartile rectangles)
  - There is a problem for the virus control of plate 17, which shows a broad range of values, with a third quartile falling in the range of the uninfected cells. This suggests a problem with at least 2 of the 6 replicates (missed infection ?). However, the median of the virus controls for this plate falls in the same range as for the virus control of the other plates.
- **Cell control** (cc)
  - The cell control performs as expected in all the plates, with high CTB values.
  - Note however that the CTB of uninfected cells show inter-plate variations, with median values ranging from ~37,000 to ~53,000.

Importantly, there is a consistency between the inter-plate differences observed for virus control, treated cells and cell control, respectively. For example, plate 2 whose consistently higher value than the other plates for the three types of wells. This highlights the importances to perform a plate-wise standardization.

Plate-wise standardization

Plate-wise control points

Taking into account the above-reported results, we apply the following procedure to standardize the individual viability measures.

For each plate, we define two control values:

- $CTB_{cc,i}$: median CTB of the 8 **cell controls** (uninfected cells) of plate i
- $CTB_{vc,i}$: median CTB of the 6 **virus controls** (infected untreated cells) of plate i

These two values are deliberately estimated with the median measurement of the control replicate, in order to avoid the effect of outliers as denoted for the virus control of plate 17.

```
#### Compute plate-wise statistics ####
plateStat <- data.frame(
  plate = plateNumbers,

  ## Mean CTB
  CTBmean = as.vector(by(
    data = inhibTable$CTB,
```

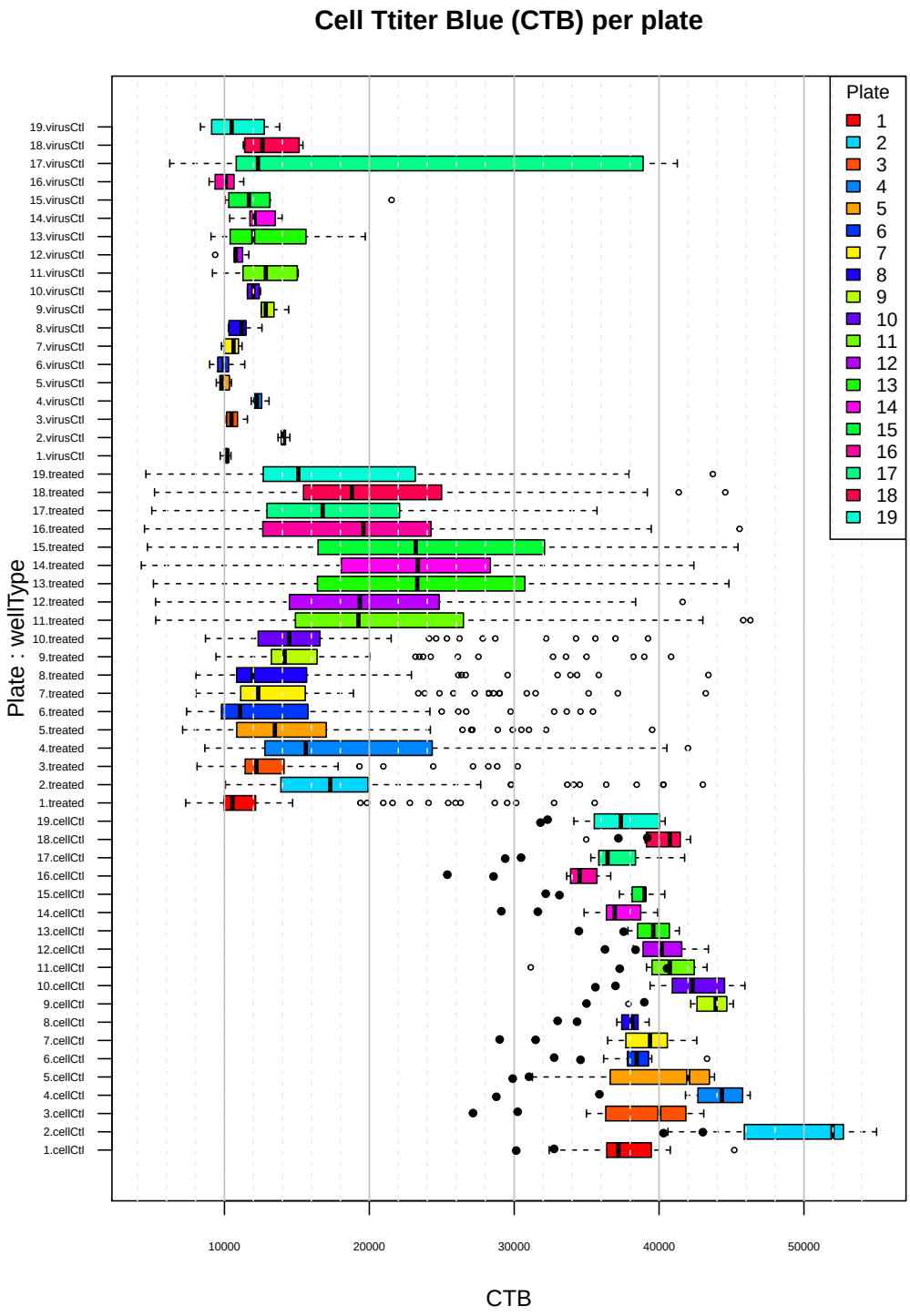

Figure 2: Distribution of CTB values per plate. Top virus control (untreated infected cells); middle: treated cells; bottom: cell control (uninfected). Black plain circles: arbidol control duplicates.

```
    INDICES = inhibTable$Plate,
    FUN = mean)),

  ## Standard deviation
  CTBsd = as.vector(by(
    data = inhibTable$CTB,
    INDICES = inhibTable$Plate,
    FUN = sd)),

  ## Median
  CTBmedian = as.vector(by(
    data = inhibTable$CTB,
    INDICES = inhibTable$Plate,
    FUN = median)),

  ## Minimum
  CTBmin = as.vector(by(
    data = inhibTable$CTB,
    INDICES = inhibTable$Plate,
    FUN = min)),

  ## maximum
 CTBmax = as.vector(by(
    data = inhibTable$CTB,
    INDICES = inhibTable$Plate,
    FUN = max)),

  # ## interquartile range
  # CTBiqr = as.vector(by(
  #   data = inhibTable$CTB,
  #   INDICES = inhibTable$Plate,
  #   FUN = IQR)),

  ## Plate-wise virus control
  CTBvc = as.vector(by(
    data = inhibTable[wellType == "virusCtl", "CTB"],
    INDICES = inhibTable[wellType == "virusCtl", "Plate"],
    FUN = median)),

  ## Plate-wise cell control
  CTBcc = as.vector(by(
    data = inhibTable[wellType == "cellCtl", "CTB"],
    INDICES = inhibTable[wellType == "cellCtl", "Plate"],
    FUN = median))

)
rownames(plateStat) <- plateStat$plate

## print the plate-wise stats in the report
kable(plateStat, caption = "Plate-wise statistics on the CTB")
```

Table 3: Plate-wise statistics on the CTB

plate	CTBmean	CTBsd	CTBmedian	CTBmin	CTBmax	CTBvc	CTBcc
1	14968.79	9179.079	10568.5	7326	45194	10202.0	37197.5
2	21151.54	11359.022	17339.0	10069	55014	14018.0	51971.0
3	15561.16	8274.558	12261.0	8110	43083	10482.0	40016.0
4	20376.58	10986.807	15742.0	8648	46289	12241.0	44342.0
5	16882.96	9640.466	13660.5	7114	43824	9876.5	42012.0
6	15654.72	9488.589	11291.5	7389	43312	9942.0	38458.5
7	17024.33	9779.790	12338.0	8040	43221	10626.0	39375.0
8	16583.02	9286.450	12010.5	8033	43402	11185.5	38161.0
9	18603.48	9702.154	14293.5	9418	45126	12852.5	43904.5
10	18183.64	9574.292	14639.0	8690	45927	11993.0	42310.5
11	22241.27	10113.348	19335.0	5255	46301	12856.0	40742.5
12	21527.03	9444.469	19471.5	5248	43404	10781.5	40187.5
13	24039.02	10818.207	23996.5	5097	44821	11987.0	39596.5
14	23585.09	9364.788	23567.0	4263	42400	12083.0	36942.0
15	24082.91	10388.209	23327.0	4685	45445	11686.0	38995.5
16	19891.76	8991.913	19738.0	4483	45555	10090.0	34521.5
17	19518.59	9053.786	17587.0	4983	41762	12314.5	36445.0
18	21942.55	9706.601	19075.5	5173	44573	12623.0	40740.5
19	19526.67	9930.452	15299.5	4578	43712	10505.0	37360.5

```
## Add columns with the control values according to the plate
inhibTable$CTBvc <- plateStat$CTBvc[inhibTable$Plate]
inhibTable$CTBcc <- plateStat$CTBcc[inhibTable$Plate]
# sort(table(inhibTable$CTBvc))
# sort(table(inhibTable$CTBcc))
```

Viability ratio and log2(ratio)

The **viability ratio** (R) associated to a given treatment with molecule m on a given plate i is defined as the ratio between

- the CTB measured on infected cells treated with this molecule ($CTB_{m,i}$) and
- the median CTB of 8 replicates of uninfected cells (denoted cc for *cell control*) on the same plate ($CTB_{cc,i}$).

$$R_{m,i} = \frac{CTB_{m,i}}{CTB_{cc,i}}$$

We further apply logarithmic transformation in order to normalise this ratio.

$$R_{m,i} = log2(R) = log2\left(\frac{CTB_{m,i}}{CTB_{cc,i}}\right)$$

```
#### Compute plate-relative viability ####
inhibTable$Vratio <- inhibTable$CTB / inhibTable$CTBcc
inhibTable$Vlog2R <- log2(inhibTable$Vratio)
# table(wellType)
# hist(inhibTable[wellType == "cellCtl", "Vratio"], breaks = 50)
# hist(inhibTable[wellType == "cellCtl", "Vlog2R"], breaks = 50)
```

```
## Plate-wise virus control viability ratio
plateStat$Rvc <- as.vector(by(
  data = inhibTable[wellType == "virusCtl", "Vratio"],
  INDICES = inhibTable[wellType == "virusCtl", "Plate"],
  FUN = median))

## Plate-wise cell control viability ratio
## Note: this is 1 by definition, we compute it for validation
plateStat$Rcc <- as.vector(by(
  data = inhibTable[wellType == "cellCtl", "Vratio"],
  INDICES = inhibTable[wellType == "cellCtl", "Plate"],
  FUN = median))


## Plate-wise virus control viability log2 ratio
plateStat$Lvc <- as.vector(by(
  data = inhibTable[wellType == "virusCtl", "Vlog2R"],
  INDICES = inhibTable[wellType == "virusCtl", "Plate"],
  FUN = median))

## Plate-wise cell control viability log2 ratio
plateStat$Lcc <- as.vector(by(
  data = inhibTable[wellType == "cellCtl", "Vlog2R"],
  INDICES = inhibTable[wellType == "cellCtl", "Plate"],
  FUN = median))
```

Viability ratio boxplots

```
par(mfrow = c(1,2))


#### Boxplots of viability ratios per plate ####

Rfloor <- floor(min(inhibTable$Vratio))
Rceiling <- max(inhibTable$Vratio) * 1.1

## Box plot per plate and well type
boxplot(Vratio ~ Plate + wellType,
        data = inhibTable,
        las = 1, col = plateColor,
        xlab = "R = CTB / CTBcc",
        ylim = c(min(inhibTable$Vratio), Rceiling),
        cex.axis = 0.5, cex = 0.5,
        horizontal = TRUE)
abline(v = 1, col = "#00BB00")
abline(v = seq(from = Rfloor, to = Rceiling, by = 0.05), col = "#EEEEEE", lty = "dashed")
abline(v = seq(from = Rfloor, to = Rceiling, by = 0.2), col = "grey")
legend("topright", legend = names(plateColor),
       title = "Plate", fill = plateColor, cex = 0.6)

## Add points to denote the arbidol controls
stripchart(Vratio ~ Plate, vertical = FALSE,
           data = inhibTable[arbidolWells, ],
```

```
           method = "jitter", add = TRUE, cex = 0.7,
           col = markColor["Arbidol"], pch = markPCh["Arbidol"])

legend("bottomleft",
       title = "Markers",
       legend = "Arbidol",
       col = markColor["Arbidol"],
       pch = markPCh["Arbidol"],
       cex = 0.8)


#### Boxplots of viability log(ratios) per plate ####

Lfloor <- floor(min(inhibTable$Vlog2R))
Lceiling <- ceiling(max(inhibTable$Vlog2R))

## Box plot per plate and well type
boxplot(Vlog2R ~ Plate + wellType,
        main = "Viability log2(ratio) distribution",
        data = inhibTable,
        las = 1, col = plateColor,
        xlab = "L = log2(CTB / CTBcc)",
        ylim = c(min(inhibTable$Vlog2R), Lceiling),
        cex.axis = 0.5, cex = 0.5,
        horizontal = TRUE)
abline(v = seq(from = Lfloor, to = Lceiling, by = 0.2), col = "#EEEEEE", lty = "dashed")
abline(v = seq(from = Lfloor, to = Lceiling, by = 1), col = "grey")
legend("topright", legend = names(plateColor),
       title = "Plate", fill = plateColor, cex = 0.6)

## Add points to denote the arbidol controls
stripchart(Vlog2R ~ Plate, vertical = FALSE,
           data = inhibTable[arbidolWells, ],
           method = "jitter", add = TRUE, cex = 0.7,
           col = markColor["Arbidol"], pch = markPCh["Arbidol"])

legend("bottomleft",
       title = "Markers",
       legend = "Arbidol",
       col = markColor["Arbidol"],
       pch = markPCh["Arbidol"],
       cex = 0.8)
```

```
par(par.ori)
```

The barplots of log2-transformed viability measures show another indication for the possible existence of a plate bias: in plates 11 to 19, several molecules are associated to a much smaller viability than in any of the untreated cells. This might reflect a cytotoxic effect of the drug that would enforce the viral infection, but there is a priori no reason to expect such effects to be concentrated on the last plates.

Two-points scaling: defining a plate-wise relative viability ($I_{m,i}$)

A **plate-wise relative viability** $I_{m,i}$ is defined to indicate the viability of cells treated with a given molecule m relative to the two controls f their plate (i):

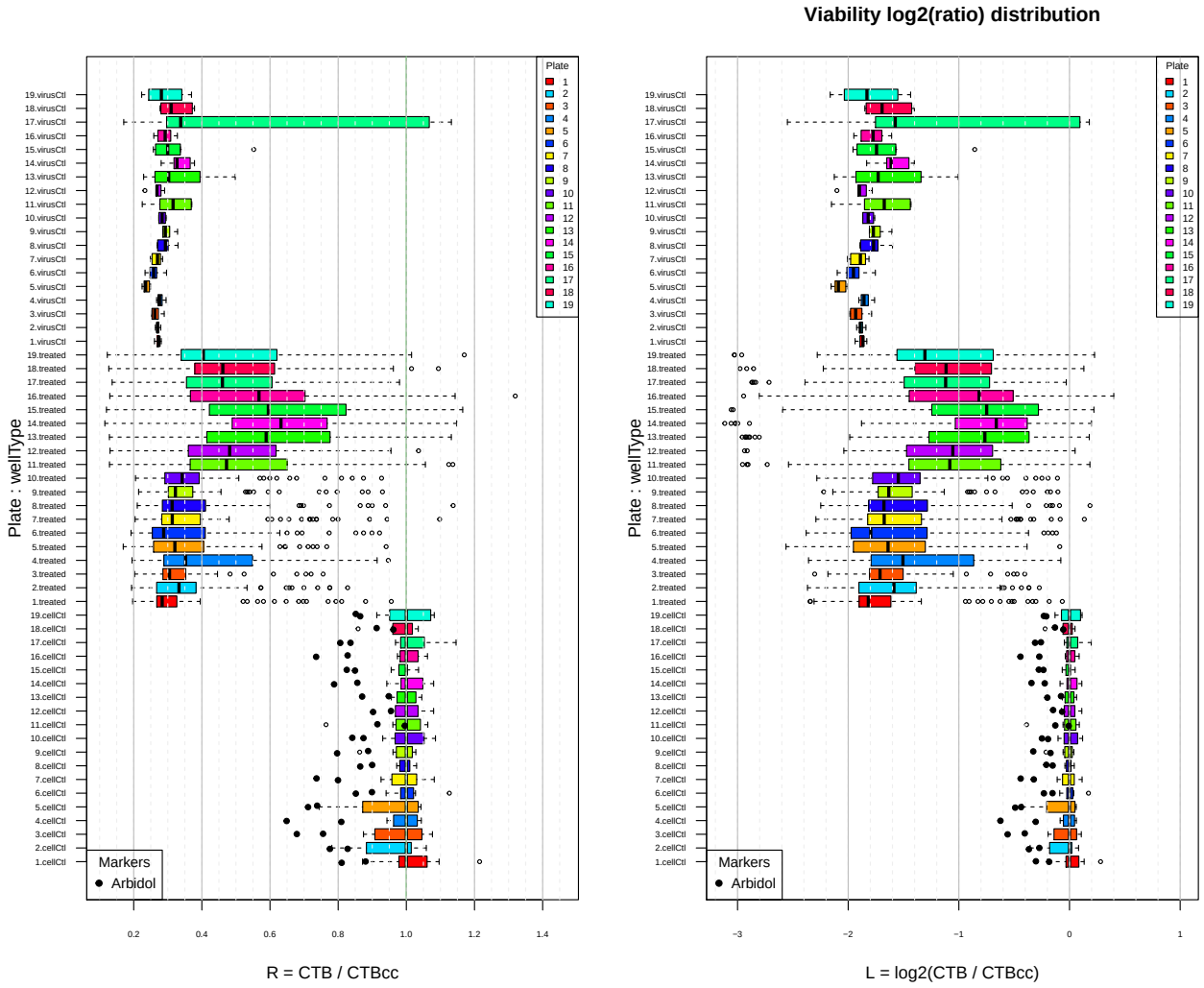

Figure 3: Distribution of the viability ratio (R) per plate. Top virus control (untreated infected cells); middle: treated cells; bottom: cell control (uninfected). Black plain circles: arbidol control duplicates.

- virus control (vc): infected untreated cells;
- cell control (cc): uninfected cells.

It is computed as follows.

$$I_{m,i} = 100 \cdot \frac{L_{m,i} - L_{vc}}{L_{cc} - L_{vc}}$$

where $L_{cc,i}$ and $L_{vc,i}$ respectively denote the median values of cell controls and virus controls for plate i.

The plate-wise relative viability $I_{m,i}$ is measured on a scale where 0 corresponds to the median of infected untreated cells (virus control), and 100 to the median of uninfected cells (cell control).

Note that $I_{m,i}$ values lower than 0 denote treatments with a lower viability than the untreated virus infection. This might result from various sources: experimental fluctuations, cytotoxic effect of the drug or plate bias.

$I_{m,i}$ can also take values higher than 100, denoting a highly efficient treatment.

```
#### Compute relative viability from log-ratios ####
inhibTable$Lrel <- 100 *
  (inhibTable$Vlog2R - plateStat$Lvc[inhibTable$Plate]) /
    (plateStat$Lcc[inhibTable$Plate] - plateStat$Lvc[inhibTable$Plate])

# hist(inhibTable$I, breaks = 100)
#
#### Compute relative viability from ratios ####
inhibTable$Vrel <- 100 *
  (inhibTable$Vratio - plateStat$Rvc[inhibTable$Plate]) /
    (plateStat$Rcc[inhibTable$Plate] - plateStat$Rvc[inhibTable$Plate])
```

Relative viability boxplots

```
#### Boxplots of relative viability per plate ####
VrFloor = floor(min(inhibTable$Vrel))
VrCeiling = max(inhibTable$Vrel) * 1.2

## Box plot per plate and well type
boxplot(Vrel ~ Plate + wellType,
        main = "Relative viability",
        data = inhibTable,
        las = 1, col = plateColor,
        xlab = "Vr = (R - Rvc) / (Rcc - Rvc)",
        ylim = c(VrFloor, VrCeiling),
        cex.axis = 0.5, cex = 0.5,
        horizontal = TRUE)
abline(v = seq(from = VrFloor, to = VrCeiling, by = 5), col = "#EEEEEE", lty = "dashed")
abline(v = seq(from = -100, to = 150, by = 25), col = "grey")
legend("topright", legend = names(plateColor),
       title = "Plate", fill = plateColor, cex = 0.6)

## Add points to denote the arbidol controls
stripchart(Vrel ~ Plate, vertical = FALSE,
           data = inhibTable[arbidolWells, ],
           method = "jitter", add = TRUE, cex = 0.7,
           col = markColor["Arbidol"], pch = markPCh["Arbidol"])
```

```
legend("bottomleft",
       title = "Markers",
       legend = "Arbidol",
       col = markColor["Arbidol"],
       pch = markPCh["Arbidol"],
       cex = 0.8)
```

```
par(par.ori)
```

The box plots show that the relative viability standardises the measures by positioning each treatment relative to two milestones of its own plate:

- the median virus control (0)
- the cell control (100)

The virus controls are well regrouped in the range of smaller v_r values, except for plate 17.

The cell controls occupy the high range (their first quartile is higher than 80) and are quite compactly grouped around 100.

However, we still observe a strong difference between plates 11-19 and the other plates:

- their median is mich higher than for the plates 1 to 10;
- they also show a much wider inter-quartile rectangle, denoting a wide dispersion of values on this plate;
- for plates 11 and 13-19, this wider dispersion is even visible for the virus controls (untreated infected cells), and it is thus unlikely that it results from the particular molecules sampled on this second half of the plates.

We thus need a way to perform a between-plates standardization of the variance.

Dot plots: relative viability

```
#### Dot plots of the relative viability ####
VrFloor <- floor(min(inhibTable$Vrel / 10)) * 10
VrCeiling <- max(inhibTable$Vrel) * 1.1

## Virus control plots
par(mfrow = c(4,1))
plot(inhibTable[wellType == "virusCtl", "Vrel"],
     main = "Virus control (infected, untreated)",
     xlab = "Replicates, sorted per plate",
     ylab = "Vr = (R - Rvc) / (Rcc - Rvc)",
     col = inhibTable[wellType == "virusCtl", "color"],
     pch = inhibTable[wellType == "virusCtl", "pch"],
     xlim = c(0, (nbPlates * 6 * 1.05)),
     ylim = c(VrFloor, VrCeiling),
     panel.first = c(abline(h = 0, col = "red", lwd = 2),
                     abline(h = 100, col = "#008800", lwd = 2),
                     abline(h = seq(VrFloor,VrCeiling, 10), col = "#DDDDDD"),
                     abline(v = (0:19) * 6, col = "#999999")
                     ),
     xaxt = "n",
     las = 1,
     cex = 0.5
     )
mtext(plateNumbers, at = (1:19) * 6 - 3, side = 1)
legend("topright",
```

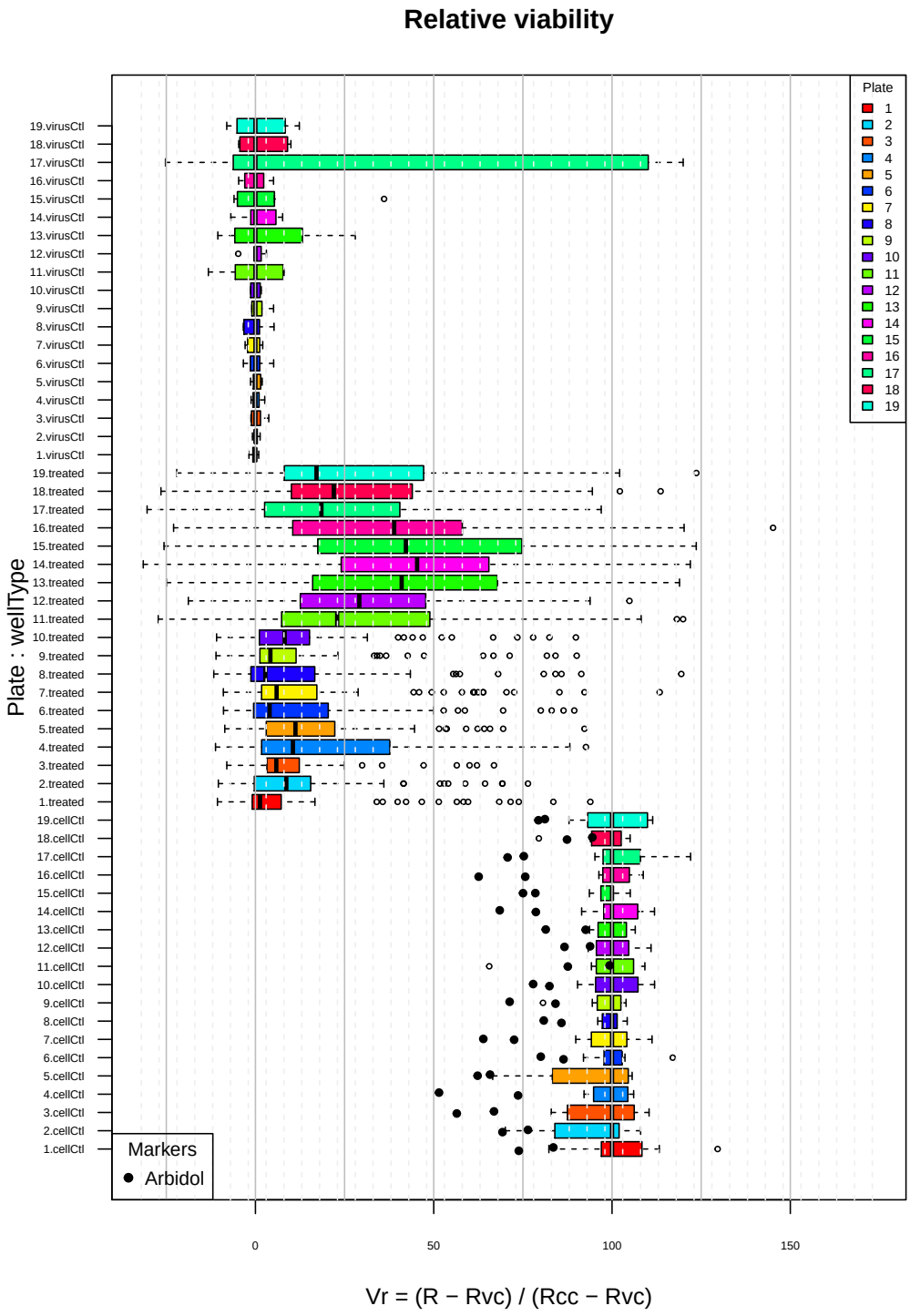

Figure 4: Distribution of relative viability (I) values per plate. Top virus control (untreated infected cells); middle: treated cells; bottom: cell control (uninfected). Black plain circles: arbidol control duplicates.

```
       legend = names(plateColor),
       col = plateColor, pch = 1,
       cex = 0.7)

## Treated cells
plot(inhibTable[wellType == "treated", "Vrel"],
     main = "Relative viability (Vr)",
     xlab = "Molecules",
     ylab = "relative viability",
     col = inhibTable[wellType == "treated", "color"],
     pch = inhibTable[wellType == "treated", "pch"],
     xlim = c(0, (nbPlates * 80 * 1.05)),
     ylim = c(VrFloor, VrCeiling),
     panel.first = c(abline(h = 0, col = "red", lwd = 2),
                     abline(h = 100, col = "#008800", lwd = 2),
                     abline(h = seq(VrFloor,VrCeiling, 10), col = "#DDDDDD"),
                     abline(v = (0:19) * 80, col = "#999999")
                     ),
     xaxt = "n",
     las = 1,
     cex = 0.5
     )
mtext(plateNumbers, at = (1:19) * 80 - 40, side = 1)
legend("topright",
       legend = names(plateColor),
       col = plateColor, pch = 1,
       cex = 0.6)

## Cell control
plot(inhibTable[wellType == "cellCtl", "Vrel"],
     main = "Cell control (uninfected)",
     xlab = "Replicates, sorted per plate",
     ylab = "relative viability",
     col = inhibTable[wellType == "cellCtl", "color"],
     pch = inhibTable[wellType == "cellCtl", "pch"],
     xlim = c(0, (nbPlates * 8 * 1.05)),
     ylim = c(VrFloor, VrCeiling),
     panel.first = c(abline(h = 0, col = "red", lwd = 2),
                     abline(h = 100, col = "#008800", lwd = 2),
                     abline(h = seq(VrFloor,VrCeiling, 10), col = "#DDDDDD"),
                     abline(v = (0:19) * 8, col = "#999999")
                     ),
     xaxt = "n",
     las = 1,
     cex = 0.5
     )
mtext(plateNumbers, at = (1:19) * 8 - 4, side = 1)
legend("topright",
       legend = names(plateColor),
       col = plateColor, pch = 1,
       cex = 0.7)
# names(supTable)
```

```
## Rank plot
VrRank <- order(inhibTable$Vrel, decreasing = TRUE)
plot(inhibTable[VrRank, "Vrel"],
     main = "Ranked relative viability values",
     xlab = "Molecules (ranked by relative viability)",
     ylab = "relative viability",
     col = inhibTable[VrRank, "color"],
     pch = inhibTable[VrRank, "pch"],
     cex = 0.5,
     panel.first = grid(),
     xlim = c(0, length(VrRank) * 1.05)
     )
abline(h = 0, col = "red", lwd = 2)
abline(h = 100, col = "#008800", lwd = 2)
legend("topright",
       legend = names(plateColor),
       col = plateColor, pch = 1,
       cex = 0.4)
```

```
par(mfrow = c(1,1))
```

IQR-based standardization

Plate-wise IQR-standardized viability

To take into account the between-plate differences in variance denoted above, we compute a z-scores from the original value.

- centering: substract an estimator of the plate-wise mean $\hat{\mu}$;
- scaling: divide by an estimator of the plate-wise standard deviation ($\hat{\sigma}_i$)

$$z_{c,i} = \frac{V_c - \hat{\mu}_i}{\hat{\sigma}_i}$$

where

- V_c is the viability for molecule c;
- $\hat{\mu_i}$ is the estimate for the mean viability of all the treated cells in plate i;
- $\hat{\sigma_i}$ is the estimate for the standard deviation of all the treated cells in plate i;

In classical statistics, the estimators of centrality and dispersion are derived from the sample mean and standard deviation, respectively:

- the population mean is used as maximum likelihood estimator of the population: $\hat{\mu} = \bar{x}$
- the population standard deviation (σ) is estimated with the sample standard deviation, corrected for the systematic bias: $\hat{\sigma} = \sqrt{n/(n-1)} \cdot s$)

However, we must be careful because each plate supposedly contain a mixture of inactive (no inhibitory effect) and active (inhibitory) molecules. The previous histograms and box plots show that these inhibitory molecules appear as statistiacal outliers (with very high viability values) and would thus strongly bias the estimation of the background variance (the variance due to fluctuations in absence of treatment).

One possibility would be to use the standard deviation of the virus control to this purpose, but this would leat to instable estimators, since they would be based on 6 points per plate. In addition, the boxplots show that the variance among treated cells is higher than the virus control, suggesting some generic effect of the treatments.

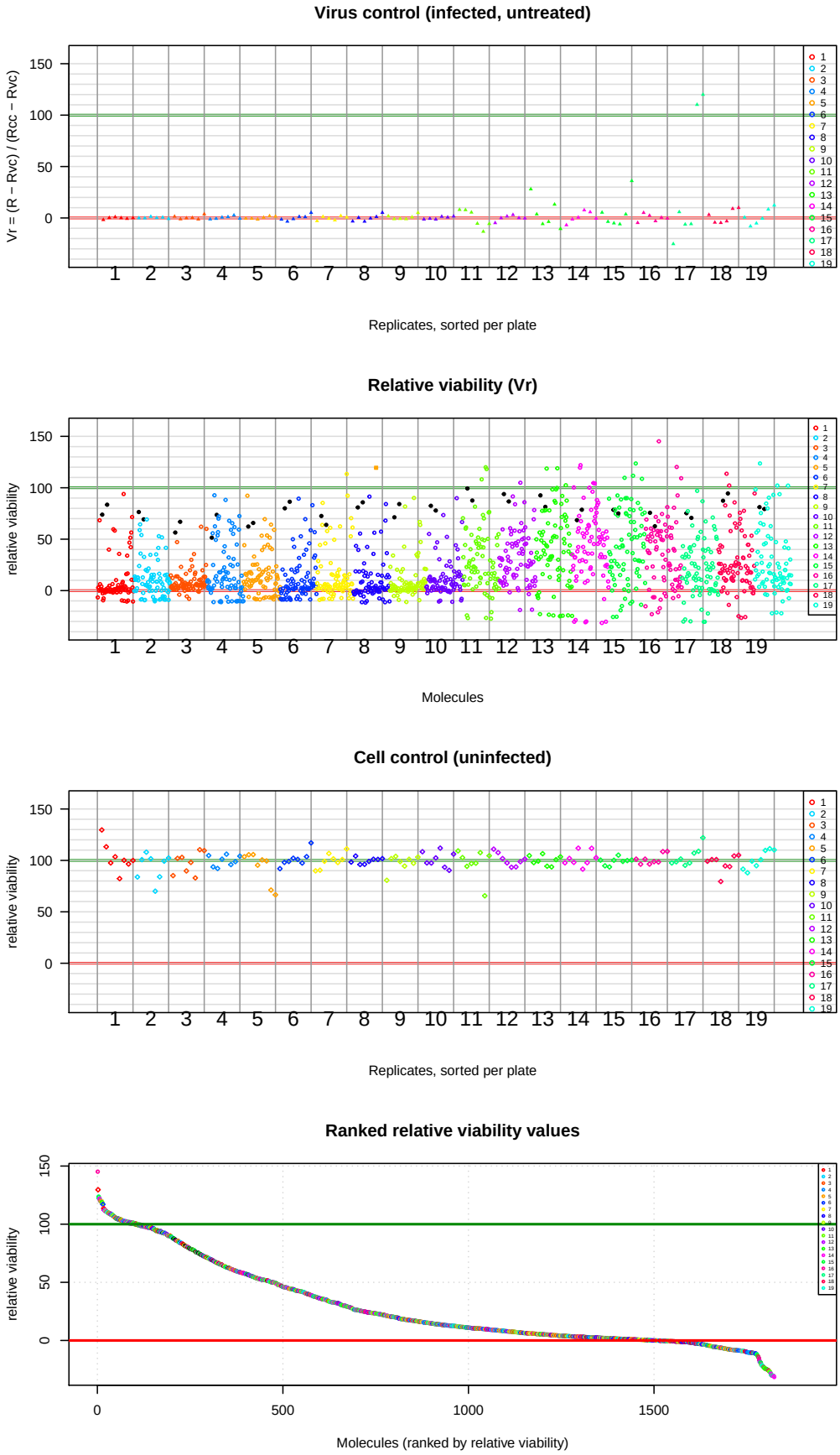

Figure 5: Values of the plate-wise relative viability for all the tested molecules. Molecules are colored according to the plate number. A: virus control (infected untreated); B: treated cells; C: cell control (untreated cells); D: ranked values (all types). Plain triangles: virus control (untreated infected cells). Black plain circles: arbidol control duplicates per plate. Orange square: Hydroxychloroquine sulfate.

Another strategy is to consider that the variance (and standard deviation) can be estimated from the bulk of treated cell viability measures themselves, and to use **robust estimators** of the central tendency (i.e. the plate-wise median) and dispersion (i.e. the plate-wise interquartile range).

This approach relies on the assumption that, *in each plate*, the number of active molecule (statistical outliers). Since teach plate contains tests of 80 molecules, there are 19 molecules above the third quartile ($Q3$). However, it has to be noted that the plates were manufactured with some grouping of molecules of the same structural family. It might thus happen that some plates contain more than 19 molecules having an inhibitory effect. Such a situation would result in a loss of sensitivity, since the presence of active molecules in the inter-quartile range would lead to over-estimate the dispersion.

An alternative is to estimate the dispersion based on the range between the first quartile ($Q1$) and the median ($\tilde{x}$) of each plate.

In summary, we compute robust estimators, in order to avoid the effect of outliers (in this case, the suspected outliers are the molecules having an actual inhibitory effect). To this purpose, we use:

- the median plate-wise relative viability for all the molecules ($\tilde{I}_i$) to estimate the mean
- the plate-wise standardized inter-quantile range; (IQR_i) standardized by the normal IQR to estimate the standard deviation.

```
#### Compute plate-wise statistics ####
statPerPlate <- data.frame(
  Plate = plateNumbers,
  TrMean = as.vector(by(
    data = inhibTable[wellType == "treated", "Vrel"],
    INDICES = inhibTable[wellType == "treated", "Plate"],
    FUN = mean)),
  TrSD = as.vector(by(
    data = inhibTable[wellType == "treated", "Vrel"],
    INDICES = inhibTable[wellType == "treated", "Plate"],
    FUN = sd)),
  TrMedian = as.vector(by(
    data = inhibTable[wellType == "treated", "Vrel"],
    INDICES = inhibTable[wellType == "treated", "Plate"],
    FUN = median)),
  vcMedian = as.vector(by(
    data = inhibTable[wellType == "virusCtl", "Vrel"],
    INDICES = inhibTable[wellType == "virusCtl", "Plate"],
    FUN = median)),
  ccMedian = as.vector(by(
    data = inhibTable[wellType == "cellCtl", "Vrel"],
    INDICES = inhibTable[wellType == "cellCtl", "Plate"],
    FUN = median)),
  arbidolMean = as.vector(by(
    data = inhibTable[arbidolWells, "Vrel"],
    INDICES = inhibTable[arbidolWells, "Plate"],
    FUN = mean)),
  TrMin = as.vector(by(
    data = inhibTable[wellType == "treated", "Vrel"],
    INDICES = inhibTable[wellType == "treated", "Plate"],
    FUN = min)),
  TrMax = as.vector(by(
    data = inhibTable[wellType == "treated", "Vrel"],
    INDICES = inhibTable[wellType == "treated", "Plate"],
    FUN = max)),
```

```
  TrQ1 = as.vector(by(
    data = inhibTable[wellType == "treated", "Vrel"],
    INDICES = inhibTable[wellType == "treated", "Plate"],
    FUN = quantile, probs = 0.25)),
  TrQ3 = as.vector(by(
    data = inhibTable[wellType == "treated", "Vrel"],
    INDICES = inhibTable[wellType == "treated", "Plate"],
    FUN = quantile, probs = 0.75)),
  TrIQR = as.vector(by(    data = inhibTable[wellType == "treated", "Vrel"],
    INDICES = inhibTable[wellType == "treated", "Plate"],
    FUN = IQR))
)
rownames(statPerPlate) <- statPerPlate$Plate
```

We define a plate-wise scaling factor from the interquantile range, standardized by the inter-quartile range of a Gaussian distribution.

$$S_i = \frac{Q3_N - Q1_N}{Q3_i - Q3_i} = \frac{1.349}{Q3_i - Q3_i}$$

Where $Q1$ and $Q3$ denote the first and third quartile, N the Normal distribution, and i is the plate number.

The relative viabilities of each plate are then multiplied by the corresponding scaling factor to obtain plate-wise standardized values (z), which will have the same inter-quartile range as the normal distribution.

$$z_{c,i} = \frac{I_{m,i} - \hat{\mu}_i}{\hat{\sigma}_i} = (I_{m,i} - \tilde{I}_i)\frac{Q3_N - Q1_N}{Q3_i - Q3_i} = (v_c - \tilde{v_i}) \cdot S_i$$

where

- $Q1_N$ and $Q3_N$ are the first and third quartiles of the normal distribution,
- $Q1_i$ and $Q3_i$ are the first and third quartiles of the relative viability for all the molecules tested on plate i,

The table below indicates the plate-wise statistics and scaling factor.

```
#### Scaling factor per plate ####

## Compute scaling factor based on the standardized inter-quartile range.
statPerPlate$scaling <-
  (qnorm(p = 0.75) - qnorm(p = 0.25)) /
  (statPerPlate$TrQ3 - statPerPlate$TrQ1)

kable(statPerPlate, caption = "Plate-wise statistics of treated cells. Column prefixes: Tr = treated ce
```

Table 4: Plate-wise statistics of treated cells. Column prefixes: Tr = treated cells; cc = cell control (uninfected cells); vc = virus control (infected, untreated cells).

Plate	TrMean	TrSD	TrMedian	vcMedian	ccMedian	arbidolMean	TrMin	TrMax	TrQ1	
1	10.65724	23.44074	1.287252	0	100	78.69645	-10.653628	93.92306	-0.8612547	
2	12.84856	20.48569	8.656760	0	100	72.84800	-10.404975	76.42084	-0.2766580	15
3	10.57859	14.39757	5.852577	0	100	61.68145	-8.031421	66.89917	3.2724995	1
4	19.93263	27.39853	10.524594	0	100	62.56503	-11.192798	92.68559	1.7600698	3
5	16.38903	21.51424	11.215011	0	100	64.04599	-8.596412	92.26401	3.1468314	2

Plate	TrMean	TrSD	TrMedian	vcMedian	ccMedian	arbidolMean	TrMin	TrMax	TrQ1	
6	13.55398	23.87465	3.974892	0	100	83.20621	-8.952712	89.41139	-0.4278225	2
7	16.34907	26.20411	5.937598	0	100	68.22324	-8.995095	113.37786	1.8070194	1
8	13.70064	26.29018	2.776594	0	100	83.34600	-11.686530	119.42874	-1.2455747	1
9	12.09549	21.35717	4.191356	0	100	77.72124	-11.060479	90.10209	1.3179505	1
10	14.05262	21.26600	8.196586	0	100	80.16822	-10.894698	89.86229	1.2896842	1
11	29.94993	31.95786	22.921485	0	100	93.46817	-27.256916	119.93258	8.0397325	4
12	32.97116	26.23733	29.140311	0	100	90.25879	-18.817588	104.88506	12.6887370	4
13	41.03991	36.73360	41.005813	0	100	87.00809	-24.955178	118.92283	16.6355783	6
14	44.18947	34.61745	45.337705	0	100	73.57697	-31.457420	121.95583	24.0607024	6
15	43.10173	35.62465	42.192277	0	100	76.77182	-25.635768	123.61632	17.6907303	7
16	37.09870	32.99190	38.908786	0	100	69.13411	-22.949880	145.16096	10.5232998	5
17	22.42951	28.21706	18.507698	0	100	72.99269	-30.382711	96.97478	2.5921552	4
18	29.19213	29.89163	21.979194	0	100	90.92914	-26.495954	113.63030	10.0969147	4
19	29.41383	32.44893	17.123122	0	100	80.26475	-22.069967	123.65065	8.2692558	4

```
#### Compute plate-wise IQR-standardized viability ####

## Centering: substract the median
## Scaling: divide by IQR
## Standardize: multiply by IQR of the normal distribution
plate <- as.vector(inhibTable$Plate)
inhibTable$z <- (inhibTable$Vrel
      - statPerPlate[plate, "TrMedian"]) * statPerPlate[plate, "scaling"]
# IQR(inhibTable$z)
# IQR(rnorm(n = 1000000))
# as.vector(by(data = inhibTable$z[wellType == "treated"], INDICES = inhibTable$Plate[wellType == "trea
# as.vector(by(data = inhibTable$z[wellType == "treated"], INDICES = inhibTable$Plate[wellType == "trea
```

```
### Descriptive statistics on the IQR-standardized viability ####
zstat <- data.frame(
  mean = mean(inhibTable$z[wellType == "treated"]),
  sd = sd(inhibTable$z[wellType == "treated"]),
  IQR = IQR(inhibTable$z[wellType == "treated"]),
  var = var(inhibTable$z[wellType == "treated"]),
  min = min(inhibTable$z[wellType == "treated"]),
  Q1 = as.vector(quantile(inhibTable$z[wellType == "treated"], probs = 0.25)),
  median = median(inhibTable$z[wellType == "treated"]),
  Q3 = as.vector(quantile(inhibTable$z[wellType == "treated"], probs = 0.75)),
  max = max(inhibTable$z[wellType == "treated"])
)

kable(t(zstat),
      col.names = "Stat",
      caption = "Statistics of the plate-wise IQR-standardized viability")
```

Table 5: Statistics of the plate-wise IQR-standardized viability

	Stat
mean	0.4566598
sd	1.8532899
IQR	1.3083010
var	3.4346835

	Stat
min	-2.5268796
Q1	-0.4810615
median	0.0000000
Q3	0.8272395
max	15.8639101

Histograms of inter-quartile standardized viability

The histogram of plate-wise IQR-standardized viability shows a clear improvement: the median is much closer to the mode than with the raw or log-transformed II values.

```
#### Histograms of IQR-standardized viability ####
histBreaks = seq(from = floor(min(inhibTable$z)),
                 to = ceiling(max(inhibTable$z)), by = 0.1)

par(mfrow = c(3,1))

## Virus controls
hist(inhibTable[wellType == "virusCtl", "z"],
     main = "Virus control - IQR standardized viability",
     breaks = histBreaks,
     xlab = "Plate-wise z-score",
     col = "orange", border = "orange")
abline(v = 0)
abline(v = c(-1, 1), lty = "dotted")

## Treated cells
hist(inhibTable[wellType == "treated", "z"],
     main = "Treated cells - IQR standardized viability",
     breaks = histBreaks,
     xlab = "Plate-wise z-score",
     col = "grey", border = "grey")
abline(v = 0)
abline(v = c(-1, 1), lty = "dotted")
abline(v = mean(inhibTable[wellType == "treated", "z"]), col = "blue")
abline(v = median(inhibTable[wellType == "treated", "z"]), col = "darkgreen")
legend("topright", legend = c("mean", "median"),
       col = c("blue", "darkgreen"),
       lwd = 2)

## Cell controls
hist(inhibTable[wellType == "cellCtl", "z"],
     main = "Cell control (untreated) - IQR standardized viability",
     breaks = histBreaks,
     xlab = "Plate-wise z-score",
     col = "palegreen", border = "palegreen")
abline(v = 0)
abline(v = c(-1, 1), lty = "dotted")
```

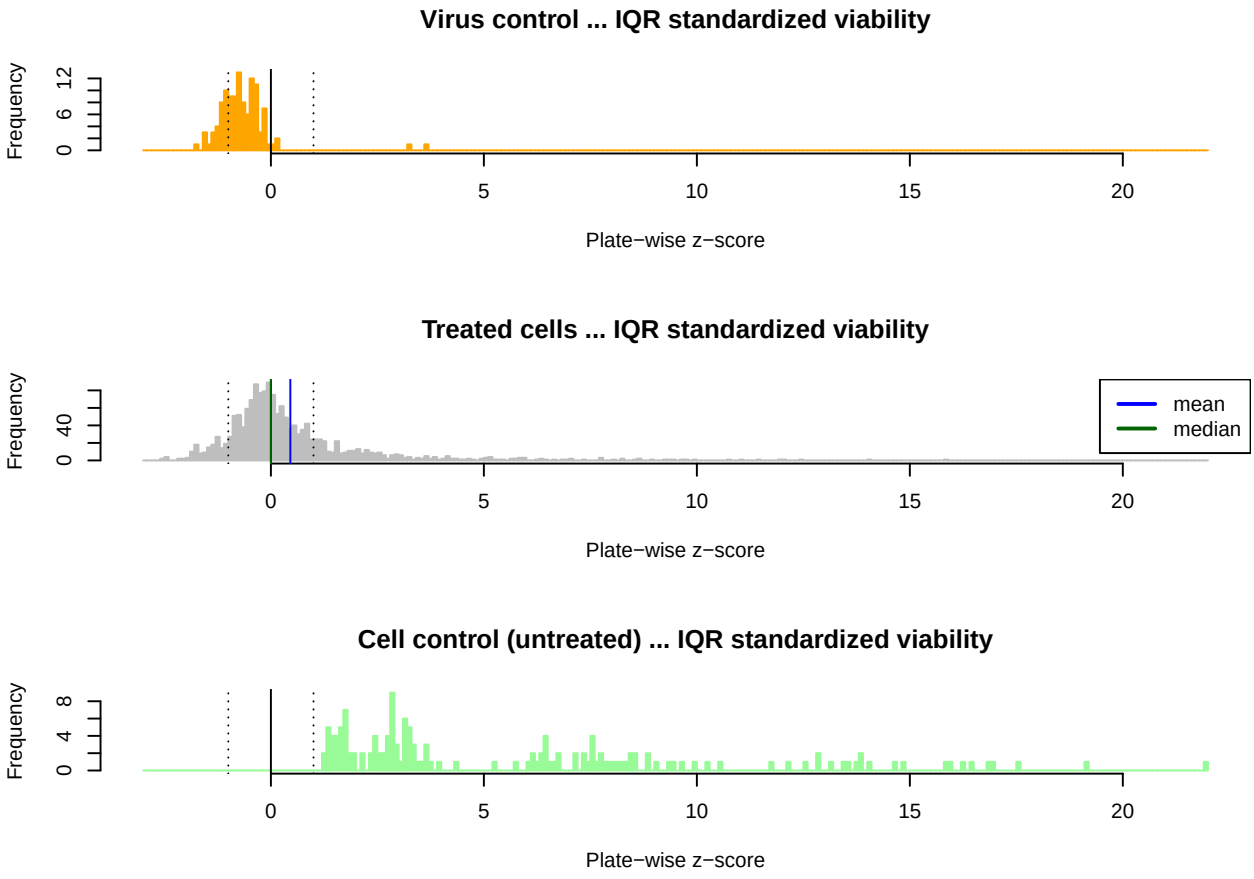

Figure 6: Distribution of the plate-wise IQR-standardized viability (z-scores) Top: virus control (infected, untreated); middle: treated; bottom: cell control (untreated)

```
par(par.ori)
```

Boxplots: inter-quartile standardized viability

```
#### Boxplots of IQR-standardized viability per plate ####

zfloor <- floor(min(inhibTable$z))
zceiling <- max(inhibTable$z) * 1.1

## Box plot per plate and well type
boxplot(z ~ Plate + wellType,
        main = "Inter-quartile standardized viability",
        data = inhibTable,
        las = 1, col = plateColor,
        xlab = "z",
        ylim = c(zfloor, zceiling),
        cex.axis = 0.5,
        cex = 0.5,
        horizontal = TRUE)
abline(v = seq(from = zfloor, to = zceiling, by = 1), col = "#EEEEEE")
abline(v = seq(from = zfloor, to = zceiling, by = 5), col = "grey")
abline(v = 0)
abline(v = c(-1, 1), lty = "dotted")
legend("topright", legend = names(plateColor),
       title = "Plate", fill = plateColor, cex = 0.6)

## Add points to denote the arbidol controls
stripchart(z ~ Plate, vertical = FALSE,
           data = inhibTable[arbidolWells, ],
           method = "jitter", add = TRUE, cex = 0.7,
           col = markColor["Arbidol"], pch = markPCh["Arbidol"])
legend("bottomleft",
       title = "Markers",
       legend = "Arbidol",
       col = markColor["Arbidol"],
       pch = markPCh["Arbidol"],
       cex = 0.8)
```

```
par(par.ori)
```

The above boxplots show that the inter-quartile standardization efficiently corrects for the over-dispersion of the plates 11 to 19. However we may expect a lot of sensitivity for these plates. There are however still some weaknesses.

- The virus control show good properties: in absence of treatment, infected cells have sligtly negative values, except for 2 outliers in plate 17.
- The cell control box plots show wide variation in their medians and dispersion:
  - Uninfected cells (cell control) have much lower values in some plates (plates 4 and 11-19) than in other ones. This is not expected, since these cells should by definition have the same inhibition values.
  - Even for the plates where the cell controls have a high relative viability after IQR-based standardization, there are strong between-plates variations.

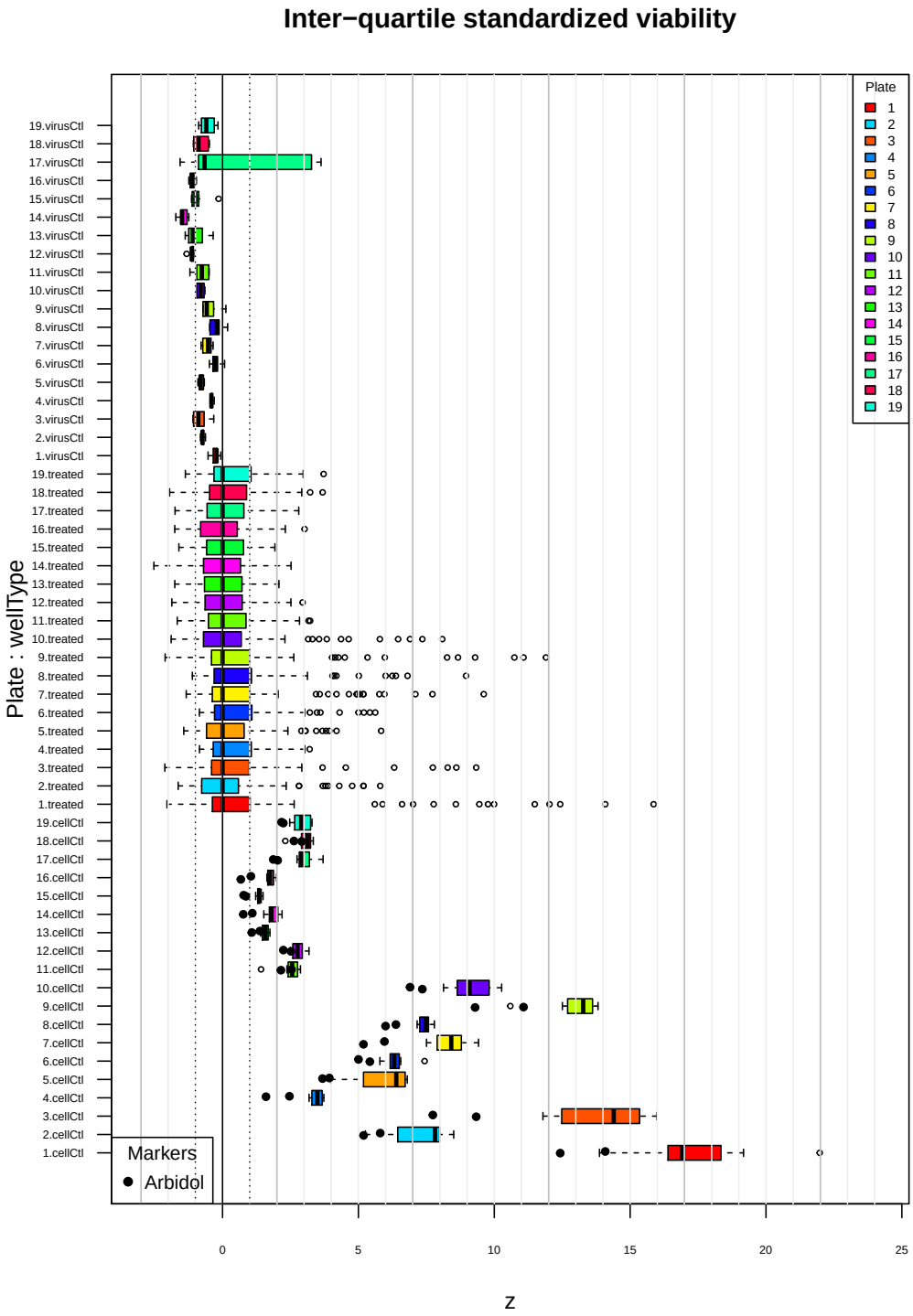

Figure 7: Distribution of inter-quartile standardized viability (z) values per plate. Top virus control (untreated infected cells); middle: treated cells; bottom: cell control (uninfected). Black plain circles: arbidol control duplicates.

This standardization seems thus efficient to correct the apparent over-dispersion of plates 11 to 19, and thereby reduce the rate of likely false positives, but the wide between-plate variability of the untreated cells suggest that the resulting z-scores should not be interpreted as indicators of inhibition.

Dot plots: inter-quartile standardized viability

```
zceiling <- max(inhibTable$z) * 1.1
zRange <- c(zfloor, zceiling)

## Virus control
par(mfrow = c(4,1))
plot(inhibTable[wellType == "virusCtl", "z"],
     panel.first = grid(),
     main = "Virus control (infected, untreated)",
     xlab = "Replicates, sorted per plate",
     ylab = "z",
     ylim = zRange,
     xlim = c(0, (19*6*1.1)),
     col = inhibTable[wellType == "virusCtl", "color"],
     pch = inhibTable[wellType == "virusCtl", "pch"],
     cex = 0.5, las = 1
     )
abline(h = seq(from = zfloor, to = zceiling, by = 1), col = "#EEEEEE")
abline(h = seq(from = zfloor, to = zceiling, by = 5), col = "grey")
abline(h = 0)
abline(h = c(-1, 1), lty = "dotted")
legend("topright",
       legend = names(plateColor),
       col = plateColor, pch = 1,
       cex = 0.7)

## Treated cells
plot(inhibTable[wellType == "treated", "z"],
     panel.first = grid(),
     main = "IQR-standardized viability (z)",
     xlab = "Molecules",
     ylab = "z",
     ylim = zRange,
     col = inhibTable[wellType == "treated", "color"],
     pch = inhibTable[wellType == "treated", "pch"],
     cex = 0.5, las = 1,
     xlim = c(0, length(inhibTable[wellType == "treated", "z"])*1.1)
     )
abline(h = seq(from = zfloor, to = zceiling, by = 1), col = "#EEEEEE")
abline(h = seq(from = zfloor, to = zceiling, by = 5), col = "grey")
abline(h = 0)
abline(h = c(-1, 1), lty = "dotted")
legend("topright",
       legend = names(plateColor),
       col = plateColor, pch = 1,
       cex = 0.6)
legend("bottomleft",
       title = "Markers",
       legend = names(markColor),
```

```
       col = markColor,
       pch = markPCh,
       cex = 0.6)

## Cell control
plot(inhibTable[wellType == "cellCtl", "z"],
     panel.first = grid(),
     main = "Cell control (uninfected)",
     xlab = "Replicates, sorted per plate",
     ylab = "z",
     ylim = zRange,
     col = inhibTable[wellType == "cellCtl", "color"],
     pch = inhibTable[wellType == "cellCtl", "pch"],
     cex = 0.5, las = 1
     )
abline(h = seq(from = zfloor, to = zceiling, by = 1), col = "#EEEEEE")
abline(h = seq(from = zfloor, to = zceiling, by = 5), col = "grey")
abline(h = 0)
abline(h = c(-1, 1), lty = "dotted")
legend("topright",
       legend = names(plateColor),
       col = plateColor, pch = 1,
       cex = 0.7)
# names(supTable)

## Rank plot
zrank <- order(inhibTable$z, decreasing = TRUE)
plot(inhibTable[zrank, "z"],
     main = "Ranked IQR-standardized viability values",
     xlab = "Molecules (ranked by z index)",
     ylab = "z",
     col = inhibTable[zrank, "color"],
     pch = inhibTable[zrank, "pch"],
     cex = 0.5, las = 1,
     panel.first = grid(),
     xlim = c(0, length(zrank) * 1.05)
     )
abline(h = seq(from = zfloor, to = zceiling, by = 1), col = "#EEEEEE")
abline(h = seq(from = zfloor, to = zceiling, by = 5), col = "grey")
abline(h = 0)
abline(h = c(-1, 1), lty = "dotted")
legend("topright",
       legend = names(plateColor),
       col = plateColor, pch = 1,
       cex = 0.4)
```

```
par(mfrow = c(1,1))
```

The plot of IQR-standardized viability values (top panel) clearly shows that the plate-wise normalization suppressed the background bias. However it denotes a new problem: the cell controls show striking differences depending on the plates. Noticeably,they show very high values in plates 1 and 3, and very low values in plates 11 to 19, as well as in plate 4.

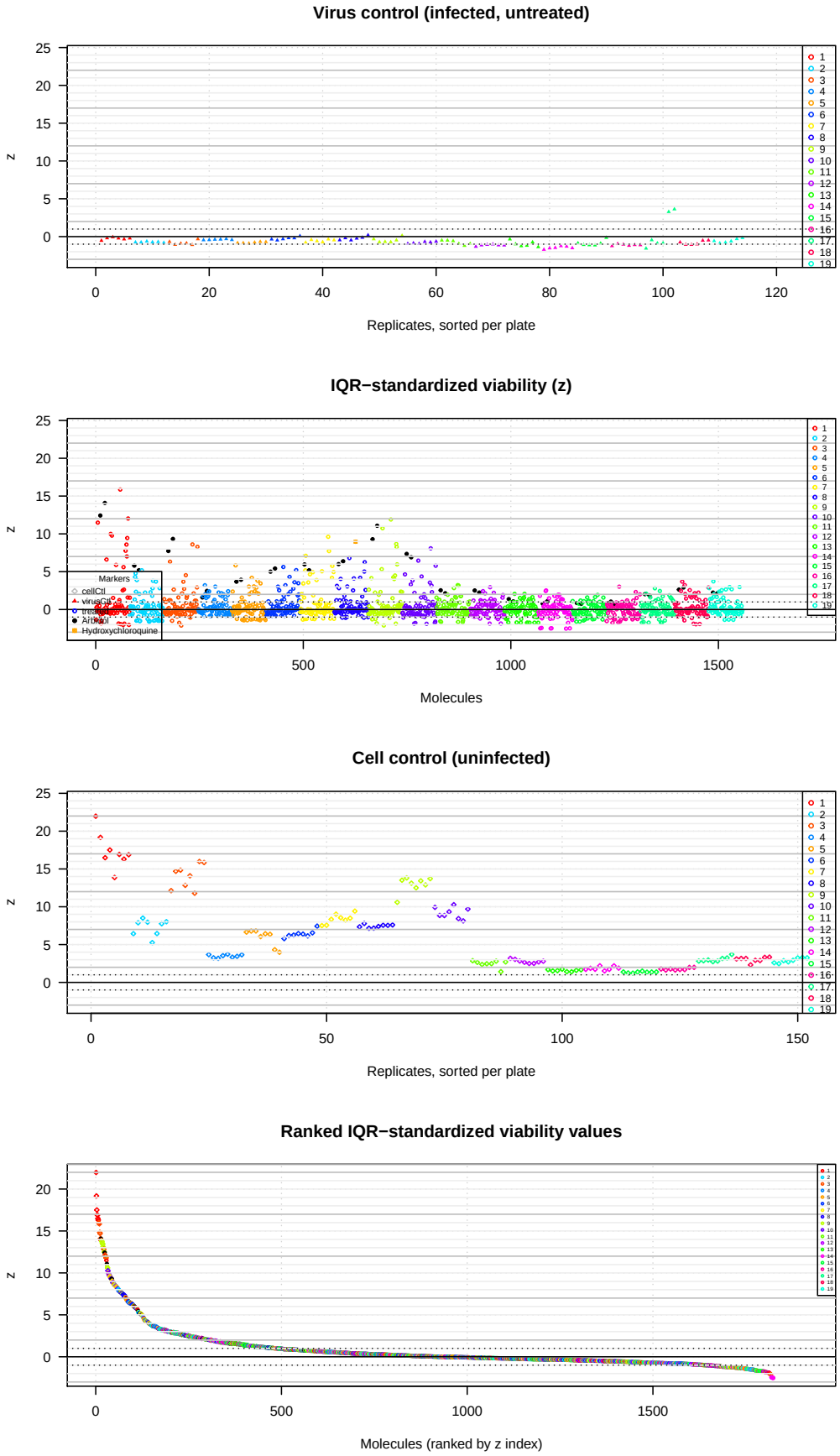

Figure 8: Values of the plate-wise IQR-standardized relative viability (z) for all the tested molecules. Molecules are colored according to the plate number. A: virus control (infected untreated); B: treated cells; C: cell control (untreated cells); D: ranked values (all types). Plain triangles: virus control (untreated infected cells). Black plain circles: arbidol control duplicates per plate. Orange square: Hydroxychloroquine sulfate.

P-value computation

We compute the p-value as the upper tail of the normal distribution (rigth-side test) in order to detect significantly high values of the plate-wise IQR-standardized viability.

```
#### Compute P-value from the IQR-standardized viability ####
inhibTable$p.value <- pnorm(inhibTable$z, mean = 0, sd = 1, lower.tail = FALSE)
inhibTable$log10Pval <- log10(inhibTable$p.value)
inhibTable$e.value <- inhibTable$p.value * sum(wellType == "treated")
inhibTable$FDR <- NA
inhibTable[wellType == "treated", "FDR"] <-
  p.adjust(inhibTable[wellType == "treated", "p.value"], method = "fdr")
inhibTable$log10FDR <- log10(inhibTable$FDR)
inhibTable$sig <- -inhibTable$log10FDR
```

P-value histogram

```
hist(inhibTable[wellType == "treated", "p.value"],
     breaks = 20,
     col = "grey",
     main = "P-value histogram after plate-wise normalization",
     xlab = "Nominal P-value (unadjusted)",
     ylab = "Frequency")
```

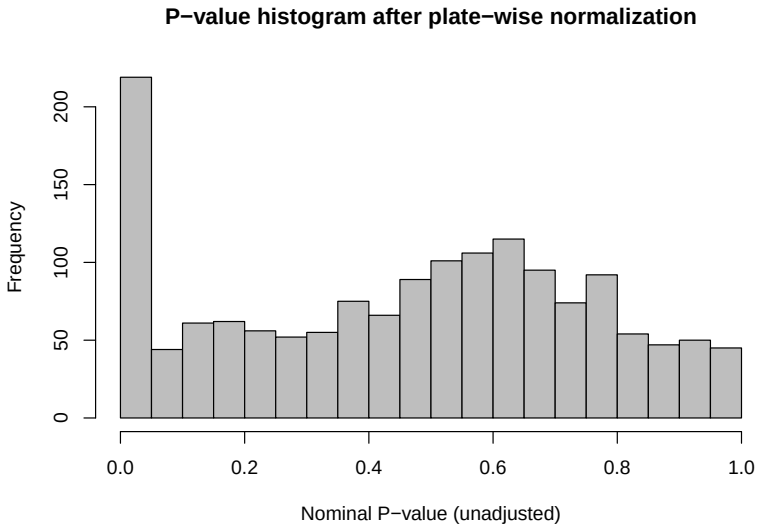

Figure 9: Histogram of the nominal (unadjusted) p-values derived from the plate-wise IQR-standardized relative viability.

```
## Estimate the proportion of tests under H0 and H1
## following the method proposed by Storey-Tibshirani (2003)
# lambda <- 0.40
# table(inhibTable[wellType == "treated", "p.value"] > lambda)
# m0 <- sum(inhibTable[wellType == "treated", "p.value"] > lambda) / (1 - lambda)
# m1 <- sum(wellType == "treated") - m0
# print(m0)
# print(m1)
#
```

Significance plot

```
#### Manhattan plot ####
sigFloor <- 0
sigCeiling <- ceiling(max(inhibTable$sig, na.rm = TRUE))

plot(x = inhibTable[wellType == "treated", "sig"],
     col = inhibTable[wellType == "treated", "color"],
     pch = inhibTable[wellType == "treated", "pch"],
     main = "Significance plot",
     xlab = "Molecules sorted per plate",
     ylab = "Significance = -log10(FDR)",
     xlim = c(0, sum(wellType == "treated") * 1.1),
     las = 1,
     xaxt = "n",
     cex = 0.5)
abline(h = 0)
abline(h = seq(0, sigCeiling, 1), lty = "dotted", col = "grey")
abline(h = -log10(alpha), col = "red")
abline(v = (0:19) * 82, col = "grey")
mtext(plateNumbers, at = (1:19) * 82 - 41, side = 1)

## Legends
legend("bottomright",
       legend = names(plateColor),
       col = plateColor, pch = 1,
       cex = 0.6)
legend("topright", legend = names(markColor),
       title = "Markers",
       col = markColor,
       pch = markPCh,
       cex = 0.6)
```

Volcano plot

```
#### Volcano plot ####

plot(x = inhibTable[wellType == "treated", "Vrel"],
     y = inhibTable[wellType == "treated", "sig"],
     col = inhibTable[wellType == "treated", "color"],
     pch = inhibTable[wellType == "treated", "pch"],
     main = "Volcano plot",
     xlab = "Relative viability",
     ylab = "Significance = -log10(FDR)",
     xlim = c(VrFloor, VrCeiling),
     panel.first = c(
       abline(h = seq(sigFloor, sigCeiling, 1), col = "#DDDDDD"),
       abline(v = seq(VrFloor, VrCeiling, 10), col = "#DDDDDD"),
       abline(v = seq(VrFloor, VrCeiling, 50), col = "#BBBBBB")
     ),
     las = 1,
     cex = 0.7)
abline(h = 0)
abline(h = -log10(alpha), col = "red")
```

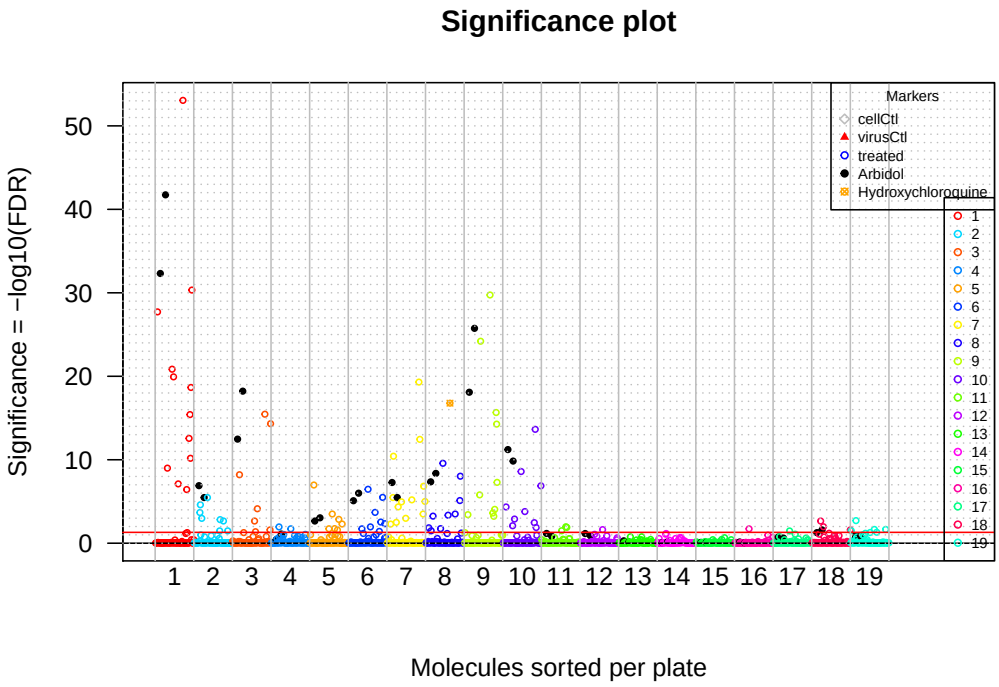

Figure 10: Volcano plot.

```
## Mark arbidol
points(x = inhibTable[arbidolWells, "Vrel"],
     y = inhibTable[arbidolWells, "sig"],
     col = inhibTable[arbidolWells, "color"],
     pch = inhibTable[arbidolWells, "pch"], cex = 0.7)
## Mark hydroxychloroquine
points(x = inhibTable[HOClSindex, "Vrel"],
     y = inhibTable[HOClSindex, "sig"],
     col = inhibTable[HOClSindex, "color"],
     pch = inhibTable[HOClSindex, "pch"], cex = 0.7)

## Legends
legend("topright",
       legend = names(plateColor),
       col = plateColor, pch = 1,
       cex = 0.7)
legend("topleft", legend = names(markColor),
       title = "Markers",
       col = markColor,
       pch = markPCh,
       cex = 0.7)
```

Significance of Arbidol after IQR-based standardisation

```
#### Relative viabilities for the arbidol contron ####
# names(inhibTable)
kable(inhibTable[arbidolWells, c("Plate", "CTB", "Vratio", "Vrel", "z", "FDR", "sig")], caption = "rela
```

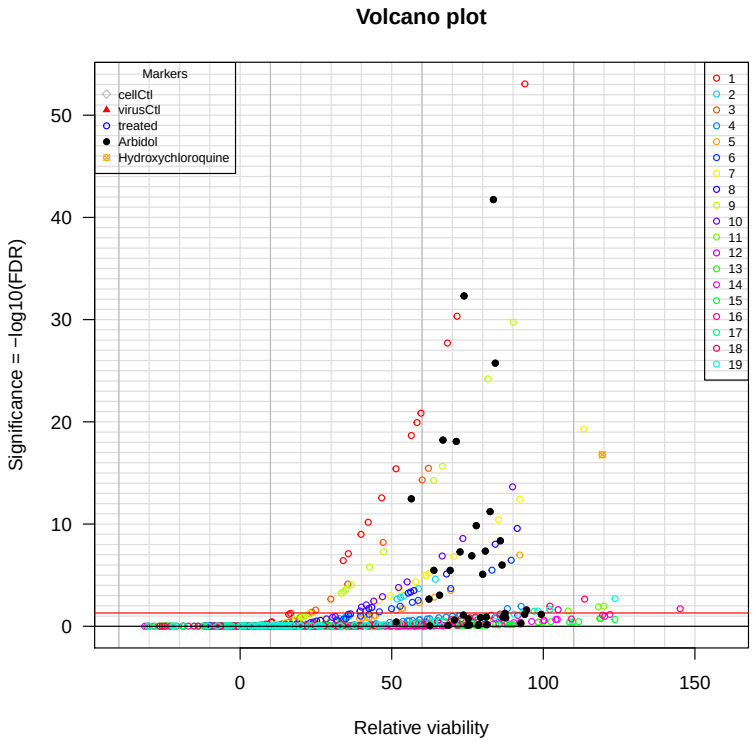

Figure 11: Volcano plot.

Table 6: relative viabilities for the arbidol controls (2 replicates per plate)

	Plate	CTB	Vratio	Vrel	z	FDR	sig
01A12	1	30140	0.8102695	73.85675	12.4275490	0.0000000	32.3178798
01B12	1	32753	0.8805162	83.53614	14.0851476	0.0000000	41.7386174
02A12	2	43022	0.8278078	76.42084	5.8018695	0.0000001	6.8937539
02B12	2	40310	0.7756249	69.27516	5.1900660	0.0000034	5.4705606
03A12	3	27158	0.6786785	56.46374	7.7389641	0.0000000	12.4691925
03B12	3	30240	0.7556977	66.89917	9.3346482	0.0000000	18.2166799
04A12	4	28768	0.6487754	51.48438	1.5981130	0.3821808	0.4177311
04B12	4	35882	0.8092102	73.64568	2.4627726	0.0801482	1.0961060
05A12	5	29898	0.7116538	62.30337	3.6787294	0.0022000	2.6575853
05B12	5	31018	0.7383129	65.78861	3.9296920	0.0009076	3.0420868
06A12	6	32760	0.8518273	80.01683	5.0026843	0.0000082	5.0885199
06B12	6	34579	0.8991250	86.39560	5.4223335	0.0000010	5.9921058
07A12	7	31481	0.7995175	72.54165	5.9595684	0.0000001	7.2737070
07B12	7	28998	0.7364571	63.90483	5.1867667	0.0000034	5.4705606
08A12	8	32999	0.8647310	80.86412	5.9986444	0.0000000	7.3537397
08B12	8	34338	0.8998192	85.82788	6.3799583	0.0000000	8.3653050
09A12	9	34990	0.7969570	71.29170	9.2955227	0.0000000	18.0887986
09B12	9	38983	0.8879044	84.15078	11.0769118	0.0000000	25.7430361
10A12	10	36991	0.8742747	82.45403	7.3529630	0.0000000	11.2189130
10B12	10	35605	0.8415169	77.88241	6.9002820	0.0000000	9.8404583
11A12	11	40557	0.9954470	99.33480	2.5390102	0.0676204	1.1699219
11B12	11	37285	0.9151378	87.60153	2.1491448	0.1483991	0.8285687
12A12	12	38378	0.9549736	93.84649	2.5169971	0.0693250	1.1591100

	Plate	CTB	Vratio	Vrel	z	FDR	sig
12B12	12	36268	0.9024697	86.67109	2.2378820	0.1253598	0.9018416
13A12	13	37554	0.9484172	92.60218	1.3741071	0.5161851	0.2871946
13B12	13	34465	0.8704052	81.41401	1.0761451	0.6992700	0.1553551
14A12	14	29114	0.7881003	68.51040	0.7624782	0.8390417	0.0762164
14B12	14	31633	0.8562882	78.64355	1.0959011	0.6863293	0.1634674
15A12	15	33124	0.8494313	78.50016	0.8600007	0.8021167	0.0957624
15B12	15	32180	0.8252234	75.04348	0.7781248	0.8334067	0.0791430
16A12	16	28576	0.8277740	75.66461	1.0445889	0.7143961	0.1460609
16B12	16	25385	0.7353388	62.60361	0.6733992	0.8804517	0.0552944
17A12	17	30476	0.8362190	75.26367	2.0282082	0.1821089	0.7396688
17B12	17	29380	0.8061462	70.72170	1.8658983	0.2441480	0.6123467
18A12	18	37191	0.9128754	87.37619	2.6284873	0.0543182	1.2650548
18B12	18	39189	0.9619175	94.48208	2.9140930	0.0255995	1.5917678
19A12	19	32305	0.8646833	81.17518	2.2361983	0.1253598	0.9018416
19B12	19	31816	0.8515946	79.35432	2.1726282	0.1415895	0.8489690

Comparison between viability scores

CTB versus viability

```
#### Dot plots to compare viability scores ####


# names(inhibTable)
# par(mfrow = c(2,2))

#### Dot plot: viability ratio versus relative viability ####
plot(inhibTable[, c("CTB", "Vrel")],
     main = "CTB vs relative viability",
     xlab = "Cell Titer Blue intensity (CTB)",
     ylab = "Relative viability (I)",
     col = inhibTable[, "color"],
     pch = inhibTable[, "pch"],
#     xlim = c(0, max(inhibTable$R)*1.1),
     cex = 0.5,
     las = 1)

## Mark cell controls
points(inhibTable[wellType == "cellCtl", c("CTB", "Vrel")],
       col = markColor["cellCtl"], pch = markPCh["cellCtl"], cex = 0.5)
## Mark virus controls
points(inhibTable[wellType == "virusCtl", c("CTB", "Vrel")],
       col = markColor["virusCtl"], pch = markPCh["virusCtl"], cex = 0.5)
## Mark arbidol controls
points(inhibTable[arbidolWells, c("CTB", "Vrel")],
       col = markColor["Arbidol"], pch = markPCh["Arbidol"], cex = 0.5)
## Mark Hydroxychloroquine
points(inhibTable[HOClSindex, c("CTB", "Vrel")],
       col = markColor["Hydroxychloroquine"], pch = markPCh["Hydroxychloroquine"], cex = 0.5)

## Grid + specific values for the selected metrics
```

```
abline(h = seq(from = -100, to = 150, by = 10), col = "#DDDDDD")
abline(h = 0, col = "red")
abline(h = 100, col = "#00BB00")
abline(v = seq(from = 0, to = 1.4, by = 0.1), col = "#DDDDDD")
abline(v = 1)

## Legends
legend("topright", legend = names(plateColor),
       title = "Plate", fill = plateColor, cex = 0.6)
legend("bottomright", legend = names(markColor),
       title = "Markers",
       col = markColor,
       pch = markPCh,
       cex = 0.6)
```

Relative versus IQR-standardised viability

```
#### Dot plot: IQR-standardized versus relative viability ####
plot(inhibTable[, c("Vrel", "z")],
     main = "Relative viability vs IQR-standardized viability",
     xlab = "Relative viability (I)",
     ylab = "IQR-standardized viability (z-score)",
     col = inhibTable[, "color"],
     pch = inhibTable[, "pch"],
     cex = 0.5,
     xlim = c(VrFloor, VrCeiling),
     panel.first = grid(),
     las = 1)

## Mark cell controls
points(inhibTable[wellType == "cellCtl", c("Vrel", "z")],
       col = markColor["cellCtl"], pch = markPCh["cellCtl"], cex = 0.5)
## Mark virus controls
points(inhibTable[wellType == "virusCtl", c("Vrel", "z")],
       col = markColor["virusCtl"], pch = markPCh["virusCtl"], cex = 0.5)
## Mark arbidol controls
points(inhibTable[arbidolWells, c("Vrel", "z")],
       col = markColor["Arbidol"], pch = markPCh["Arbidol"], cex = 0.5)
## Mark Hydroxychloroquine
points(inhibTable[HOClSindex, c("Vrel", "z")],
       col = markColor["Hydroxychloroquine"], pch = markPCh["Hydroxychloroquine"], cex = 0.5)

## Mark milestones for the selected metrics
abline(v = seq(-100, 150, 10), col = "#EEEEEE")
abline(v = seq(-100, 150, 50), col = "#BBBBBB")
abline(v = 0, col = "red")
abline(v = 100, col = "#00BB00")
abline(h = seq(-5, 10, 1), col = "#EEEEEE")
abline(h = seq(-5, 10, 5), col = "#BBBBBB")
abline(h = 0)
abline(h = c(-1, 1), col = "#BBBBBB")

## Legends
```

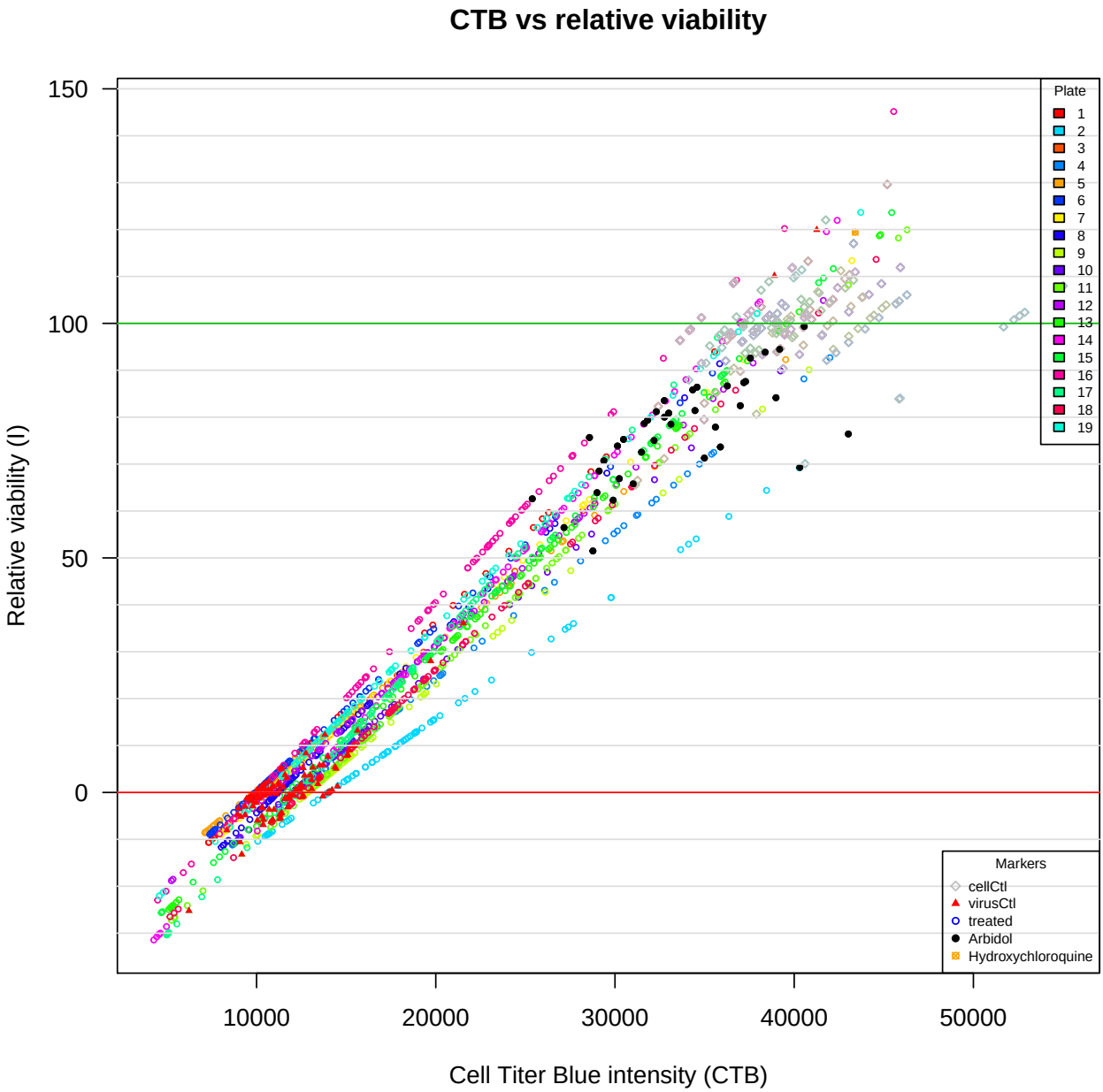

Figure 12: Comparison between viability scores. CTB versus relative viability (V_r). Diamonds: cell control (uninfected cells). Plain triangles: virus control (untreated infected cells). Black plain circles: arbidol control duplicates per plate. Orange square: Hydroxychloroquine sulfate.

```
legend("topright", legend = names(plateColor),
       title = "Plate", fill = plateColor, cex = 0.7)
legend("topleft", legend = names(markColor),
       title = "Markers",
       col = markColor,
       pch = markPCh,
       cex = 0.7)
```

FDR versus relative viability

```
#### Dot plot: FDR versus relative viability ####
plot(x = inhibTable[, "Vrel"],
     y = -inhibTable[, "log10FDR"],
     main = "relative viability vs FDR",
     xlab = "relative viability",
     ylab = "-log10(FDR)",
     col = inhibTable[, "color"],
     pch = inhibTable[, "pch"],
     cex = 0.7,
     xlim = c(VrFloor, VrCeiling),
     panel.first = grid(),
     las = 1)
## Mark arbidol controls
points(x = inhibTable[arbidolWells, "Vrel"],
       y = -inhibTable[arbidolWells, "log10FDR"],
       col = markColor["Arbidol"],
       pch = markPCh["Arbidol"], cex = 0.7)
## Mark Hydroxychloroquine
points(x = inhibTable[HOClSindex, "Vrel"],
       y = -inhibTable[HOClSindex, "log10FDR"],
       col = markColor["Hydroxychloroquine"],
       pch = markPCh["Hydroxychloroquine"], cex = 0.7)

## Grid + specific values for the selected metrics
abline(v = 0, col = "red")
abline(v = 100, col = "#00BB00")
abline(h = 0)
abline(h = -log10(alpha), col = "blue", lwd = 2)

## Legends
legend("topright", legend = names(plateColor),
       title = "Plate", fill = plateColor, cex = 0.7)
legend("topleft", legend = names(markColor),
       title = "Markers",
       col = markColor,
       pch = markPCh,
       cex = 0.7)
```

Relative viability versus inhibition index

. We compare hereafter the values of the relative variability with the inhibition index defined in the bioRxiv publication (DOI 10.1101/2020.04.03.023846).

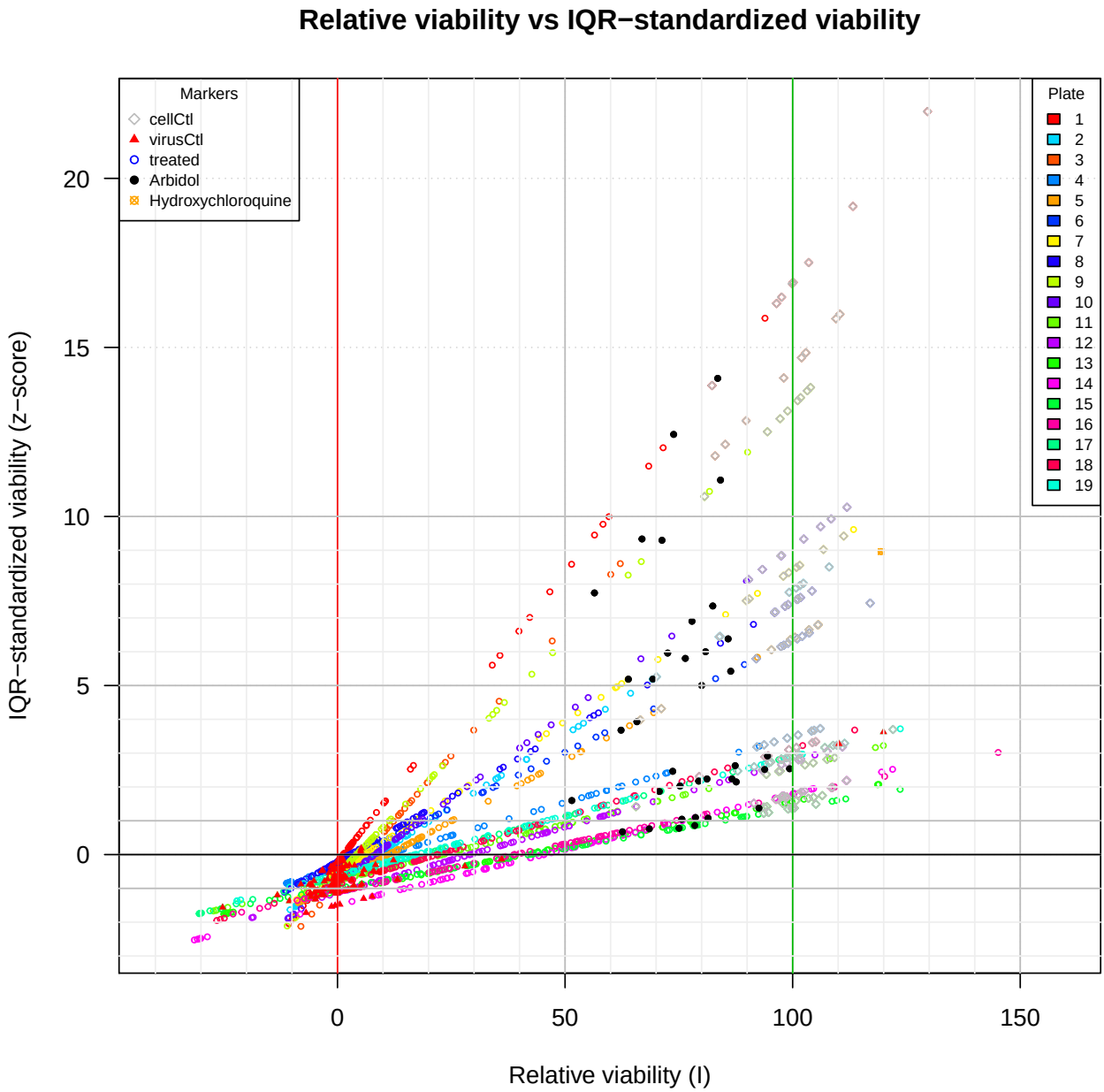

Figure 13: Comparison between viability scores. Relative viability versus IQR-standardized viability (z-score). Diamonds: cell control (uninfected cells). Plain triangles: virus control (untreated infected cells). Black plain circles: arbidol control duplicates per plate. Orange square: Hydroxychloroquine sulfate.

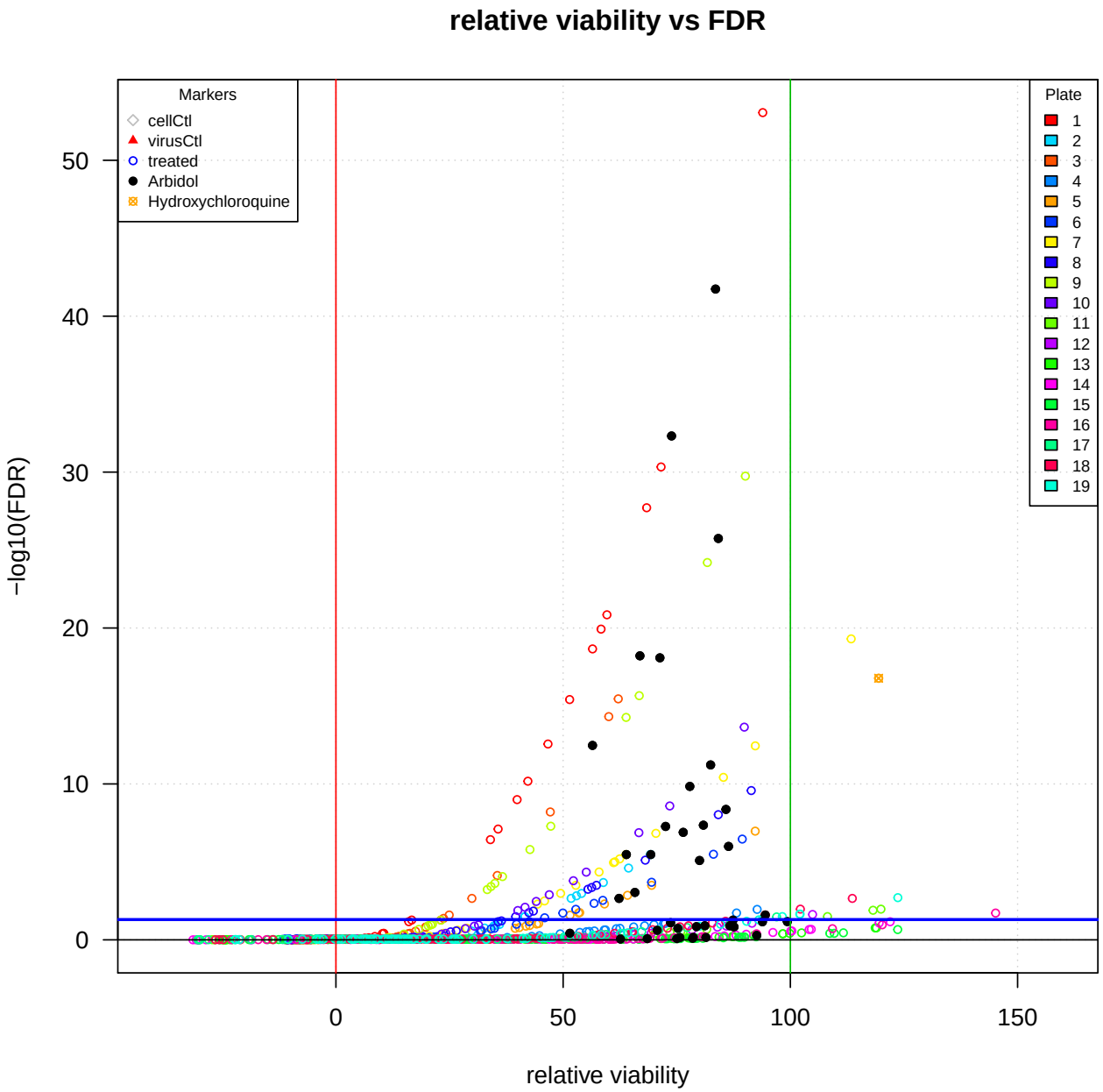

Figure 14: Comparison between viability scores. Diamonds: cell control (uninfected cells). Plain triangles: virus control (untreated infected cells). Black plain circles: arbidol control duplicates per plate. Orange square: Hydroxychloroquine sulfate.

```
iiFloor <- floor(min(inhibTable$Inhibition.Index, na.rm = TRUE))
iiCeiling <- ceiling(max(inhibTable$Inhibition.Index, na.rm = TRUE))

#### Dot plot: Relative viability versus inhibition index ####
plot(x = inhibTable[, "Inhibition.Index"],
     y = inhibTable[, "Vrel"],
     main = "Relative viability vs inhibition index",
     xlab = "Inhibition index",
     ylab = "Vr = (R - Lvc) / (Lcc - Lvc)",
     col = inhibTable[, "color"],
     pch = inhibTable[, "pch"],
     cex = 0.5,
     xlim = c(iiFloor, iiCeiling),
     panel.first = c(
       abline(v = seq(-0.5, 2, 0.5), col = "#BBBBBB"),
       abline(v = 1, col = "blue", lwd = 2),
       abline(h = seq(iiFloor, iiCeiling, 0.2), col = "#DDDDDD"),
       abline(h = 1)),
     las = 1)

## Mark Hydroxychloroquine
points(x = inhibTable[HOClSindex, "Inhibition.Index"],
       y = inhibTable[HOClSindex, "Vrel"],
       col = markColor["Hydroxychloroquine"],
       pch = markPCh["Hydroxychloroquine"], cex = 0.7)

## Legends
legend("topright", legend = names(plateColor),
       title = "Plate", fill = plateColor, cex = 0.7)
legend("topleft",
       title = "Markers",
       legend = "Hydroxychloroquine",
       col = markColor["Hydroxychloroquine"],
       pch = markPCh["Hydroxychloroquine"],
       cex = 0.7)
```

```
par(par.ori)
```

Hit molecules selected by the different criteria

```
#### Select candidate molecules accordint to different criteria ####

## False Discovery Rate computed from the IQR-standardized viabilities
inhibTable$selected.FDR <- as.numeric(inhibTable$FDR < alpha)
# `table(inhibTable$selected.FDR)
# sum(inhibTable$selected.FDR, na.rm = TRUE)

## Previous inhibition index above 1 (Arbidol)
inhibTable$selected.ii <- as.numeric(inhibTable$Inhibition.Index >= 1)
# table(inhibTable$selected.ii)
# sum(inhibTable$selected.ii, na.rm = TRUE)

## Relative viability higher than the mean of arbidol values on the same plate
```

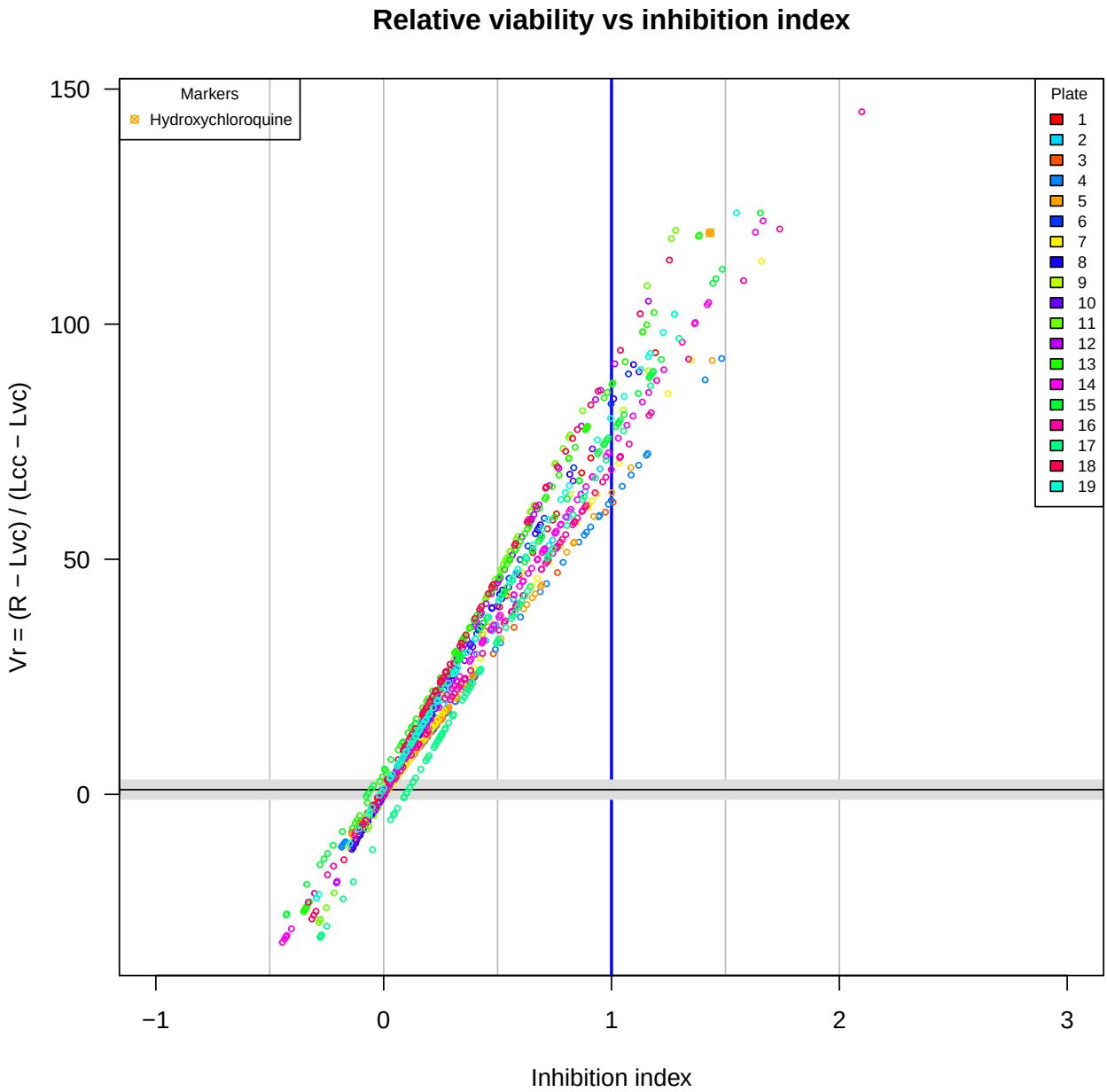

Figure 15: Comparison between viability scores. Diamonds: cell control (uninfected cells). Plain triangles: virus control (untreated infected cells). Black plain circles: arbidol control duplicates per plate. Orange square: Hydroxychloroquine sulfate.

```
inhibTable[wellType == "treated", "selected.arbidolMean"] <- as.numeric(
  inhibTable[wellType == "treated", "Vrel"] >
    statPerPlate[inhibTable[wellType == "treated", "Plate"], "arbidolMean"])

# table(inhibTable[arbidolWells, "selected.arbidolMean"] )
# table(inhibTable[, "selected.arbidolMean"] )
# table(inhibTable[, c("selected.ii", "selected.arbidolMean")] )

# diff <- !is.na(inhibTable$selected.ii) & (inhibTable$selected.ii != inhibTable$selected.arbidolMean)
# View(inhibTable[diff, ])


kable(table(inhibTable[, c("selected.FDR", "selected.arbidolMean")]),
      caption = "Contingency table of the molecules selected by different criteria. Columns: inhibition
```

Table 7: Contingency table of the molecules selected by different criteria. Columns: inhibition index >= 1. Rows: Vrel > arbidol mean.

	0	1
0	1372	67
1	77	42

```
## Compute some combinations between criteria
inhibTable$selected.FDRandArbidol <-
  as.numeric(inhibTable$selected.FDR & inhibTable$selected.arbidolMean)
inhibTable$selected.FDRorArbidol <-
  as.numeric(inhibTable$selected.FDR | inhibTable$selected.arbidolMean)
inhibTable$selected.FDRonly <-
  as.numeric(inhibTable$selected.FDR & !inhibTable$selected.arbidolMean)
inhibTable$selected.ArbidolOnly <-
  as.numeric(!inhibTable$selected.FDR & inhibTable$selected.arbidolMean)



par(par.ori)
```

Selected hits

```
#### Select significant normalized II values ####

kable(t(table(inhibTable$FDR < alpha)), caption = paste("Number of tests declared positive with FDR <",
```

Table 8: Number of tests declared positive with FDR < 0.05

FALSE	TRUE
1439	119

```
# table(inhibTable$FDR < alpha)
selected <- subset(inhibTable, inhibTable$FDR < alpha)
# names(selected)

## Sort by decreasing adjusted p-value
selected <- selected[order(selected$FDR, decreasing = FALSE), ]
# kable(names(selected), row.names=TRUE)
names(selected) <- sub(pattern = "selected.",
                       replacement = "+",
                       x = names(selected))

selectedFields <- c("ID",
                    # "CTB",
                    # "cellControl",
                    # "virusControl",
                    "Chemical.name",
                    # "Broad.Therapeutic.class",
                    "Reported.therapeutic.effect",
                    "Inhibition.Index",
                    "Vratio",
                    "Vrel",
                    "z",
                    # "p.value",
                    # "FDR",
                    "sig",
                    "+FDR",
                    "+ii",
                    "+arbidolMean",
                    "+FDRandArbidol",
                    "+FDRorArbidol",
                    "+FDRonly",
                    "+ArbidolOnly")
# kable(selectedFields)
# View(selected[ , selectedFields])

## Print selected molecules
kable(selected[ , selectedFields],
      row.names = FALSE,
      digits = 4,
      caption = "Candidate moecules sorted by significance after plate-wise normalization. ")
```

Table
norm

ID	Chemical.name	Reported.therapeutic.effect
01F08	Benoxinate hydrochloride	Local anesthetic
01B12	Arbidol	NA
01A12	Arbidol	NA
01H07	Dibucaine	Local anesthetic
09F04	Promazine hydrochloride	Antipsychotic
01A06	Atracurium besylate	Curarizing
09B12	Arbidol	NA
09D04	Opipramol dihydrochloride	Antidepressant 'Antipsychotic

ID	Chemical.name	Reported.therapeutic.effect
01D05	Triamterene	Antihypertensive 'Diuretic
01D08	Pyrimethamine	Antimalarial 'Antiprotozoal 'Antineoplastic
07G07	Pregnenolone	Anabolic 'Anti-inflammatory
01H05	Amitryptiline hydrochloride	Antidepressant
03B12	Arbidol	NA
09A12	Arbidol	NA
08E11	Hydroxychloroquine sulfate	Antimalarial
09G07	Chlorcyclizine hydrochloride	Antiemetic 'Antihistaminic 'Sedative
03G08	Clemizole hydrochloride	Antibacterial 'Antifungal 'Antihistaminic 'Antipruritic
01H03	Imipramine hydrochloride	Antidepressant
03H10	Orphenadrine hydrochloride	Antihistaminic 'Antiparkinsonian
09G08	Diphenylpyraline hydrochloride	Antihistaminic 'Antipruritic 'Sedative
10G08	Merbromin disodium salt	Antibacterial 'Antineoplastic
01G11	Tolnaftate	Antifungal 'Antifungal
03A12	Arbidol	NA
07G09	Chloroquine diphosphate	Anti-inflammatory 'Antimalarial 'Antiprotozoal
10A12	Arbidol	NA
07B04	Omeprazole	Antiulcer
01H04	Sulindac	Analgesic 'Anti-inflammatory 'Antipyretic
10B12	Arbidol	NA
08D06	Exemestane	Antineoplastic
01C05	Norethynodrel	Contraceptive
10D08	Chlorotrianisene	Antineoplastic
08B12	Arbidol	NA
03B05	Alverine citrate salt	Antispastic
08H03	Dipivefrin hydrochloride	Antiglaucoma
08A12	Arbidol	NA
09G09	Benzethonium chloride	Antibacterial 'Antiseptic 'Antineoplastic
07A12	Arbidol	NA
01E08	Ticlopidine hydrochloride	Anticoagulant 'Antiplatelet
05A10	Tacrine hydrochloride	CNS Stimulant
02A12	Arbidol	NA
10H10	Pridinol methanesulfonate salt	Antiparkinsonian
07H07	Mirtazapine	Antidepressant
06D11	Epiandrosterone	Anabolic
01G06	Diphenhydramine hydrochloride	Antiemetic 'Antihistaminic 'Antitussive 'Sedative
06B12	Arbidol	NA
09D02	Dydrogesterone	Progestogen
06H02	Vatalanib	Antineoplastic
02B12	Arbidol	NA
02C08	Pioglitazone	NA
07B12	Arbidol	NA
07B03	Nitrofural	Antibacterial 'Antidotes
07F02	Pirenperone	Anxiolytic
08H02	Alendronate sodium	Antiosteoporetic
06A12	Arbidol	NA
07H10	Tazarotene	Antipsoriatic 'Antiacneic
07C10	Bromperidol	Antipsychotic
02B04	Azacyclonol	Antipsychotic
07C03	Biperiden hydrochloride	Antiparkinsonian 'Antineoplastic
10A08	Liranaftate	Antifungal
03F02	Piroxicam	Analgesic 'Anticoagulant 'Anti-inflammatory 'Antiplatelet 'Antip

ID	Chemical.name	Reported.therapeutic.effect
09G04	Famprofazone	Analgesic 'Antipyretic
10E06	Ethoxyquin	Antifungal 'Antineoplastic
06F06	Mebhydroline 1,5-naphtalenedisulfonate	Antihistaminic
02B03	Zimelidine dihydrochloride monohydrate	Antidepressant
09F10	Hexestrol	Antineoplastic
05E07	Ambroxol hydrochloride	Expectorant 'Mucolytic
07H06	Nifuroxazide	Antibacterial 'Antineoplastic
08G02	Mizolastine	Antihistaminic
09A09	Nilvadipine	Antianginal 'Antihypertensive
08E07	Rimantadine hydrochloride	Antiviral
08B06	Tenatoprazole	Antiulcer
09G02	Trihexyphenidyl-D,L hydrochloride	Antiparkinsonian
05B12	Arbidol	NA
07D09	Budesonide	Anti-inflammatory 'Antineoplastic
02B07	Guanabenz acetate	Antihypertensive
10C04	Medrysone	Anti-inflammatory
05F11	Clebopride maleate	Antiemetic 'Antispastic
02F05	Diltiazem hydrochloride	Antianginal 'Antiarrhythmic 'Antihypertensive 'Antiplatelet 'Diu
19B02	Budralazine	Vasodilator 'Antihypertensive
02G02	Verapamil hydrochloride	Antihypertensive 'Antineoplastic
03E06	Tolfenamic acid	Analgesic 'Anti-inflammatory
05A12	Arbidol	NA
18B09	Eperisone HCl	Muscle relaxant 'Antispastic
06G08	Rifapentine	Antibacterial
07B10	Propafenone hydrochloride	Antiarrhythmic
10G06	Dicumarol	Anticoagulant
06H07	Primaquine diphosphate	Antimalarial
05G08	Carbetapentane citrate	Antispastic 'Antitussive 'Local anesthetic
07A09	Dosulepin hydrochloride	Antidepressant 'CNS Stimulant
10B11	Amprenavir	Antiviral
11E11	Equilin	NA
18C04	Methandrostenolone	Anabolic
06E06	Cyclobenzaprine hydrochloride	Muscle relaxant
04B07	Isotretinoin	Keratolytic 'Antineoplastic
11F03	Nylidrin	Vasodilator
10G10	Drofenine hydrochloride	Antispastic
08A08	Itraconazole	Antifungal 'Antifungal
05F02	Ketotifen fumarate	Antihistaminic
08C11	Etretinate	Antipsoriatic
05D10	Dextromethorphan hydrobromide monohydrate	Antitussive
04D11	Quinidine hydrochloride monohydrate	Antiarrhythmic 'Antimalarial
06C08	Dibenzepine hydrochloride	Antidepressant
16C10	Ethoxzolamide	Antiglaucoma 'Antiulcer 'Diuretic 'Antiepileptic
19E10	Dilevalol	Antihypertensive
19H04	Acetyl spiramycin	Antibacterial
12E07	Idazoxan hydrochloride	Antiparkinsonian 'Antipsychotic
03H09	Lomefloxacin hydrochloride	Antibacterial
18B12	Arbidol	NA
18H11	Artenimol	Antimalarial 'Antipaludic
05F08	Nafronyl oxalate	Anti-ischemic 'Antispastic 'Vasodilator
11E02	Olanzapine	Antipsychotic
19A03	Eletriptan	Antimigraine

ID	Chemical.name	Reported.therapeutic.effect
08B02	Tetracaïne hydrochloride	NA
02G11	Dihydroergotamine tartrate	Antimigraine
02F04	Hydroxyzine dihydrochloride	Antiemetic 'Antihistaminic 'Antipruritic 'Anxiolytic 'Sedative
17D04	Oxiglutatione	Antidote
06G05	Quetiapine hemifumarate	Antipsychotic
03E08	Tibolone	NA
19F05	Vonoprazan	Antiulcer

Venn diagram

```
#### Draw a Venn diagram of the selected molecules ####

## Venn diagram
vennTable <- na.omit(inhibTable[, c("selected.FDR", "selected.arbidolMean")])
vennDiagram(object = vennTable,
            names = c(paste("FDR <", alpha),
                      paste("I >=", 1)),
            circle.col = c("#00BB00", "blue"), mar = c(0,0,0,0)
            )
```

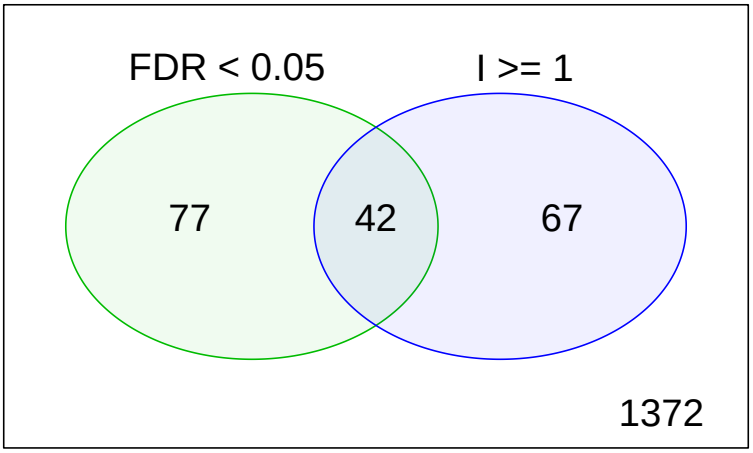

Figure 16: Venn diagram of the molecules selected by different criteria.

Hits per plate

```
#### Compute the number of candidates per plate depending on the criterion ####

## Count the number of candidates per plate for the different criteria
criteria <- c("ii", "arbidolMean", "FDR", "FDRandArbidol", "FDRorArbidol", "FDRonly", "ArbidolOnly")

# table(inhibTable$selected.ii, inhibTable$selected.arbidolMean)

candidatesPerPlate <- data.frame(matrix(
  nrow = nbPlates,
  ncol = length(criteria), 0))
row.names(candidatesPerPlate) <- 1:nbPlates
names(candidatesPerPlate) <- criteria
```

```
for (criterion in criteria) {
  candidates <- as.data.frame.table(
    table(
    subset(x = inhibTable,
           subset = inhibTable[paste0("selected.", criterion)] == 1,
           select = "Plate")))
  names(candidates) <- c("Plate", "n")
  candidatesPerPlate[as.vector(candidates$Plate), criterion] <- candidates$n
}
# apply(candidatesPerPlate, 2, sum)

ccpp <- candidatesPerPlate
ccpp["Total", ] <- apply(ccpp, 2, sum)
kable(ccpp, row.names = TRUE, ccaption = "Candidates per plate depending on the selection criteria")
```

	ii	arbidolMean	FDR	FDRandArbidol	FDRorArbidol	FDRonly	ArbidolOnly
1	1	2	14	2	14	12	0
2	0	1	10	1	10	9	0
3	1	2	9	2	9	7	0
4	7	8	2	2	8	0	6
5	3	4	9	4	9	5	0
6	1	2	10	2	10	8	0
7	4	5	15	5	15	10	0
8	3	4	12	4	12	8	0
9	2	3	12	3	12	9	0
10	1	2	11	2	11	9	0
11	3	4	3	3	4	0	1
12	2	3	1	1	3	0	2
13	9	10	0	0	10	0	10
14	14	15	0	0	15	0	15
15	16	17	0	0	17	0	17
16	9	10	1	1	10	0	9
17	3	4	1	1	4	0	3
18	3	4	4	4	4	0	0
19	8	9	5	5	9	0	4
Total	90	109	119	42	186	77	67

```
maxc <-  max(candidatesPerPlate)
plot(candidatesPerPlate[, c("FDR", "ii")],
     main = "Candidates per plate",
     xlab = paste("FDR < ", alpha),
     ylab = paste("Inhibition index >", 1),
     xlim = c(0, maxc * 1.1),
     las = 1, pch = 20,
     panel.first =
       c(abline(h = seq(0, maxc, by = 1), col = "#DDDDDD"),
         abline(h = seq(0, maxc, by = 5), col = "#BBBBBB"),
         abline(v = seq(0, maxc, by = 1), col = "#EEEEEE"),
         abline(v = seq(0, maxc, by = 5), col = "#BBBBBB")),
     col = plateColor[rownames(candidatesPerPlate)])
legend("topright", legend = names(plateColor),
       title = "Plate", col = plateColor, pch = 20, cex = 0.8)
```

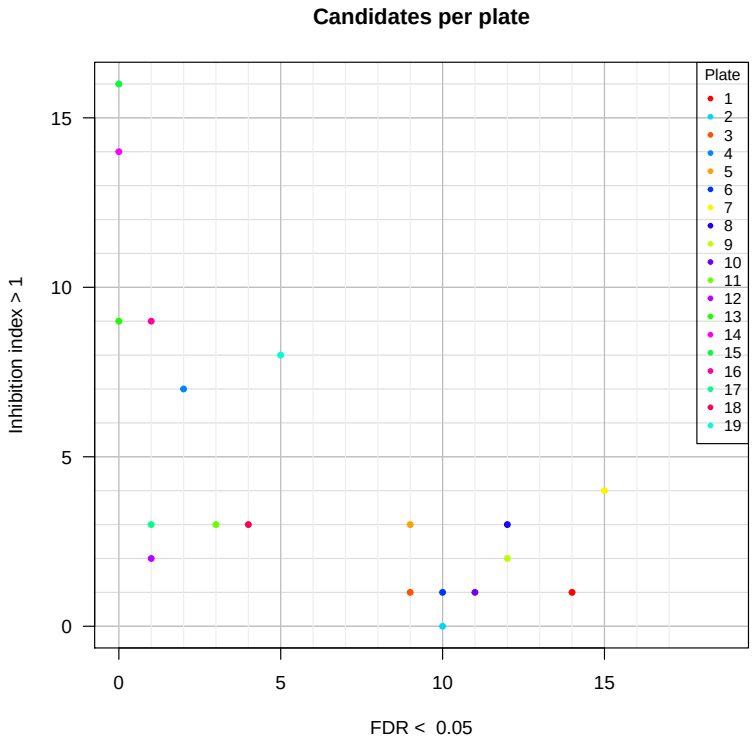

Figure 17: Number of candidate molecules per plate depending on the method.

Result files

```
#### Export result tables ####

## Define output file names
outFiles <- list(
  "All results (tsv)" =  file.path(dir["results"], "result_table_all-molecules.tsv"),
  "All results (xlsx)" =  file.path(dir["results"], "result_table_all-molecules.xlsx")
)


write.table(x = inhibTable, file = outFiles$`All results (tsv)`,
            sep = "\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

write.xlsx2(x = inhibTable, file = outFiles$`All results (xlsx)`,
            row.names = FALSE, col.names = TRUE)

# system(paste("open", dir["results"]))

## Prepare a data frame with the relative links to output files
fileLinks <- data.frame(
  name = names(outFiles),
  path = unlist(outFiles),
  basename = basename(unlist(outFiles))
)
```

```
fileLinks$link <- paste0("<a href='", fileLinks$path, "'>", fileLinks$basename, "</a>")

kable(fileLinks[, c("name", "link")], row.names = FALSE, caption = "Links to the result tables. ")
```

Table 11: Links to the result tables.

name	link
All results (tsv)	result_table_all-molecules.tsv
All results (xlsx)	result_table_all-molecules.xlsx

Analysis of Touret's original Inhibition Index (II)

Inhibition index

The inhibition index is derived from the raw viability measurement in the following way.

Descriptive stats

```
ii <- supTable$Inhibition.Index
iiStat <- list(
  mean = mean(ii),
  sd = sd(ii),
  var = var(ii),
  min = min(ii),
  Q1 = as.vector(quantile(ii, probs = 0.25)),
  median = median(ii),
  Q3 = as.vector(quantile(ii, probs = 0.75)),
  max = max(ii)
)

kable(t(as.data.frame.list(iiStat)), col.names = "Stat")
```

	Stat
mean	0.2865268
sd	0.3758726
var	0.1412802
min	-0.4444311
Q1	0.0279599
median	0.1665643
Q3	0.4917926
max	2.0988878

Distribution

The distribution of inhibition index values is trongly asymmetrical. The mode is much lower than the mean and the median (robust estimator of central tendency). A normal fit will tuhs give a poor estimate of the p-values.

```r
hist(supTable$Inhibition.Index, breaks = 100, col = "grey", border = "grey")
abline(v = iiStat$mean, col = "blue")
abline(v = iiStat$median, col = "darkgreen")

legend("topright", legend = c("mean", "median"),
       col = c("blue", "darkgreen"),
       lwd = 2)
```

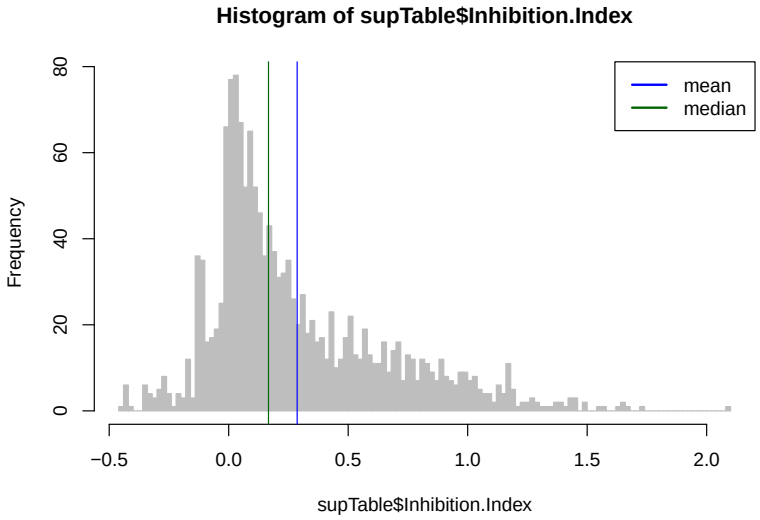

Figure 18: Distribution of the inhibition index

Normalization

Log transform

A classical method for normalization is to take the log of the values. We first had to shift the data in order for all of them to take positive values.

```r
# fit.gamma <- fitdist(data = ii - iiStat$min + 1, distr = "gamma", method = "mge")
# summary(fit.gamma)
#
# plot(fit.gamma)

#### Compute a normalized distribution from inhibition indices ####
iiPositive <- ii - iiStat$min + 1 ## shift the distrib to achieve non-negative numbers
logII <- log(iiPositive)

logIIStat <- list(
  mean = mean(logII),
  sd = sd(logII),
  var = var(logII),
  min = min(logII),
  Q1 = as.vector(quantile(logII, probs = 0.25)),
  median = median(logII),
  Q3 = as.vector(quantile(logII, probs = 0.75)),
  max = max(logII)
)
```

```
kable(t(as.data.frame.list(logIIStat)), col.names = "Stat", caption = "Parameters of the log-normalized
```

Table 13: Parameters of the log-normalized inhibition index distribution

	Stat
mean	0.5272798
sd	0.2027703
var	0.0411158
min	0.0000000
Q1	0.3868877
median	0.4768523
Q3	0.6607395
max	1.2650638

However, even after log transformation the distribution remains highly asymmetrical, with a mode much smaller than the median and mean.

```
#### Histogram of log-normalized values ####
hist(logII, breaks = 100, col = "grey", border = "grey")
abline(v = mean(logII), col = "blue")
abline(v = median(logII), col = "darkgreen")

legend("topright", legend = c("mean", "median"),
       col = c("blue", "darkgreen"),
       lwd = 2)
```

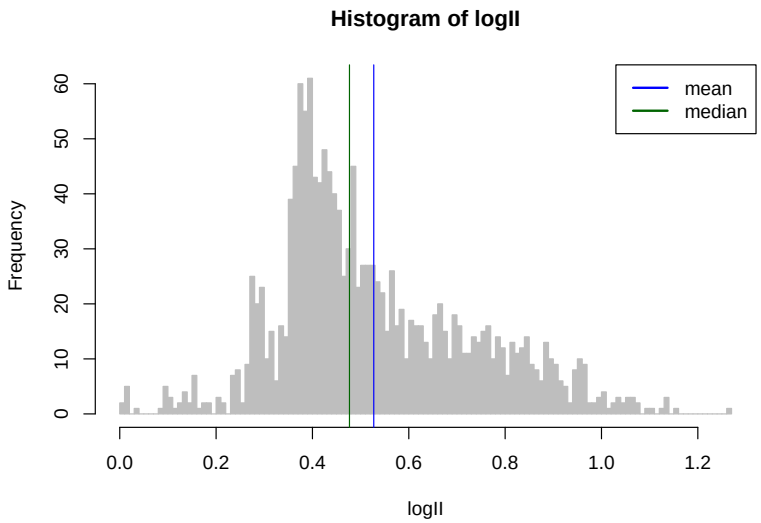

Figure 19: Distribution of the inhibition index

Evidence of a plate bias

Ranked values

We plot the inhibition index values ordered by plate and position number (top) or ranked by decreasing value (bottom). In both cases, the color denotes the plate number.

```
par(mfrow = c(2,1))
plot(ii,
     panel.first = grid(),
     main = "Inhibition index values",
     xlab = "Molecules (ranked by inhibition index)",
     ylab = "Inhibition index",
     col = supTable$color,
     cex = 0.5,
     xlim = c(0, length(ii)*1.05)
     )
legend("topright",
       legend = names(plateColor),
       col = plateColor, pch = 1,
       cex = 0.7)

sortedTable <- supTable[order(supTable$Inhibition.Index, decreasing = TRUE), ]
plot(sortedTable$Inhibition.Index,
     panel.first = grid(),
     main = "Ranked inhibition index values",
     xlab = "Molecules (ranked by inhibition index)",
     ylab = "Inhibition index",
     col = sortedTable$color,
     cex = 0.5,
     xlim = c(0, length(ii)*1.05)
     )
legend("topright",
       legend = names(plateColor),
       col = plateColor, pch = 1,
       cex = 0.7)
```

```
par(mfrow = c(1, 1))
```

The molecule-wised colored plots of inhibition index suggest a plate-wise effect.

Plate-wise normalization

We perform a plate-wise normalization using robust estimators, in order to avoid the effect of outliers (in this case, the suspected outliers are the molecules having an actual inhibitory effect).

To this purpose, we use: - plate-wise median to estimate the mean - plate-wise standardized inter-quantile range (IQR) to estimate the standard deviation

```
#### Compute plate-wise statistics ####
plateStat <- data.frame(
  plate = plateNumbers,
  mean = as.vector(by(
    data = supTable$Inhibition.Index,
    INDICES = supTable$plateNumber,
    FUN = mean)),
  sd = as.vector(by(
    data = supTable$Inhibition.Index,
    INDICES = supTable$plateNumber,
    FUN = sd)),
  median = as.vector(by(
    data = supTable$Inhibition.Index,
```

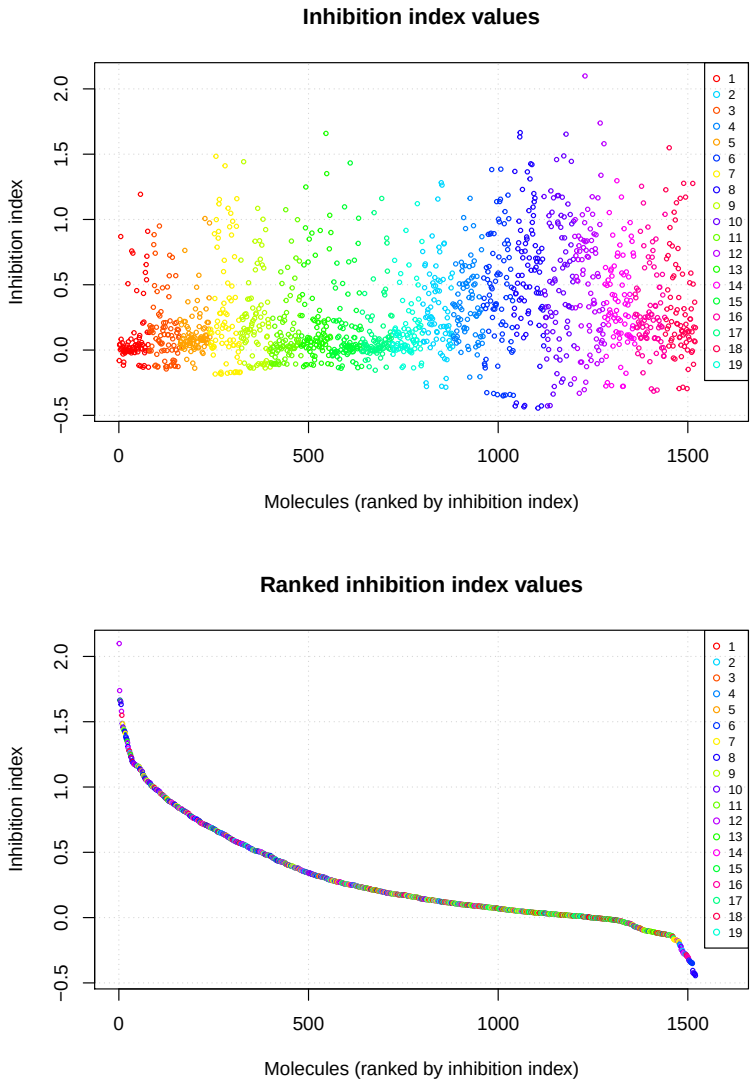

Figure 20: Values of the inhibition index for all the tested molecules. Molecules are colored according to the plate number.

```
    INDICES = supTable$plateNumber,
    FUN = median)),
  min = as.vector(by(
    data = supTable$Inhibition.Index,
    INDICES = supTable$plateNumber,
    FUN = min)),
  max = as.vector(by(
    data = supTable$Inhibition.Index,
    INDICES = supTable$plateNumber,
    FUN = max)),
  IQR = as.vector(by(
    data = supTable$Inhibition.Index,
    INDICES = supTable$plateNumber,
    FUN = IQR))
)
rownames(plateStat) <- plateStat$plate

kable(plateStat, caption = "Plate-wise statistics oninhibition index")
```

Table 14: Plate-wise statistics oninhibition index

plate	mean	sd	median	min	max	IQR
1	0.1165661	0.2665169	0.0184570	-0.1318422	1.1928831	0.0914910
2	0.1541782	0.2523084	0.1133156	-0.1450055	0.9498831	0.2156783
3	0.1443563	0.1960778	0.0863385	-0.1387721	1.0073009	0.1349143
4	0.2982710	0.4307195	0.1601301	-0.1844422	1.4836918	0.4739120
5	0.2345274	0.3194481	0.1628049	-0.1383338	1.4421869	0.2814504
6	0.1393632	0.2580634	0.0412022	-0.1109600	1.0748024	0.2271337
7	0.2241196	0.3673457	0.0873107	-0.1267828	1.6589036	0.2073556
8	0.1439492	0.2894590	0.0331407	-0.1396088	1.4326962	0.1976995
9	0.1253725	0.2449408	0.0420476	-0.1543792	1.1609812	0.1107754
10	0.1541551	0.2335481	0.0948601	-0.1365916	1.1209955	0.1617424
11	0.3071834	0.3265338	0.2478965	-0.2846758	1.2816167	0.4140013
12	0.3499997	0.2757037	0.3159377	-0.2074233	1.1619058	0.3761552
13	0.4314013	0.4397504	0.4146992	-0.3511397	1.3851376	0.5813343
14	0.5857602	0.4775020	0.6038263	-0.4444311	1.6653052	0.5380693
15	0.5187760	0.4972562	0.4842358	-0.4279441	1.6531873	0.8015369
16	0.5253860	0.4768292	0.5615571	-0.3309767	2.0988878	0.6858842
17	0.3590275	0.3386716	0.3050932	-0.2784932	1.2965981	0.4539432
18	0.2912052	0.3201378	0.2263674	-0.3152611	1.2542724	0.3426682
19	0.3404116	0.4026624	0.1982078	-0.2950106	1.5490336	0.4724498

```
## Centering: substract the median
## Scaling: divide by IQR
## Standardize: multiply by IQR of the normal distribution
normII <- (supTable$Inhibition.Index - plateStat[supTable$plateNumber, "median"]) / plateStat[supTable$
# IQR(normII)
# IQR(rnorm(n = 1000000))

normIQR <- qnorm(p = 0.75) - qnorm(p = 0.25)
normII <- normII * normIQR
# sd(normII)
```

```
# IQR(normII)

supTable$normInhibIndex <- normII
```

```
### Descriptive statistics on the normalized Inhibition Index ####
normIIStat <- list(
  mean = mean(normII),
  sd = sd(normII),
  IQR = IQR(normII),
  var = var(normII),
  min = min(normII),
  Q1 = as.vector(quantile(normII, probs = 0.25)),
  median = median(normII),
  Q3 = as.vector(quantile(normII, probs = 0.75)),
  max = max(normII)
)

kable(t(as.data.frame.list(normIIStat)), col.names = "Stat", caption = "Statistics of the plate-wise no
```

Table 15: Statistics of the plate-wise normalized inhibition index

	Stat
mean	0.4144692
sd	1.8129741
IQR	1.3183994
var	3.2868752
min	-2.6280587
Q1	-0.5094508
median	0.0000000
Q3	0.8089486
max	17.3161990

The histogram of plate-wise normalized values shows a clear improvement : the median is much closer to the mode than with the raw or log-transformed II values.

```
hist(normII, breaks = 100, col = "grey", border = "grey")
abline(v = mean(normII), col = "blue")
abline(v = median(normII), col = "darkgreen")

legend("topright", legend = c("mean", "median"),
       col = c("blue", "darkgreen"),
       lwd = 2)
```

Normalized II plots

The plot of normalized II values (top panel) clearly shows that the plate-wise normalization suppressed the background bias.

```
par(mfrow = c(2,1))
plot(normII,
     panel.first = grid(),
     main = "Inhibition index values",
     xlab = "Molecules (ranked by inhibition index)",
```

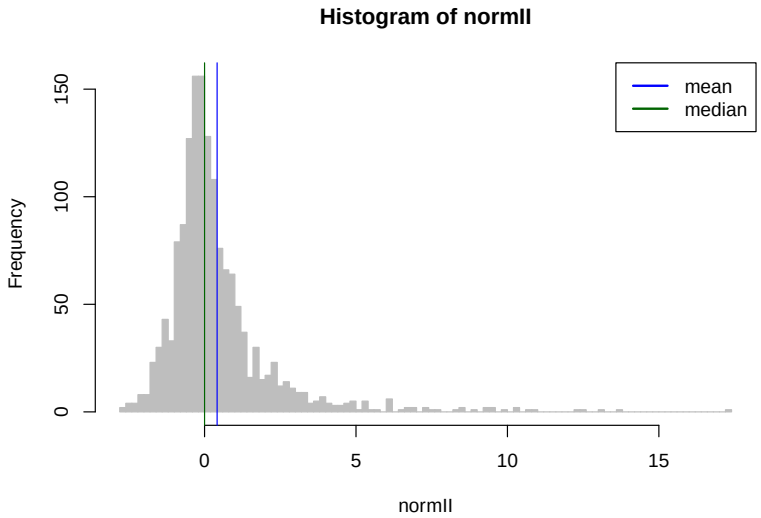

Figure 21: Distribution of the plate-wise normalized inhibition index

```
     ylab = "Inhibition index",
     col = supTable$color,
     cex = 0.5,
     xlim = c(0, length(normII)*1.05)
     )
legend("topright",
       legend = names(plateColor),
       col = plateColor, pch = 1,
       cex = 0.7)

# names(supTable)
normIIrank <- order(supTable$normInhibIndex, decreasing = TRUE)
plot(supTable[normIIrank, "normInhibIndex"],
     panel.first = grid(),
     main = "Ranked inhibition index values",
     xlab = "Molecules (ranked by inhibition index)",
     ylab = "Inhibition index",
     col = supTable[normIIrank, "color"],
     cex = 0.5,
     xlim = c(0, length(normII)*1.05)
     )
legend("topright",
       legend = names(plateColor),
       col = plateColor, pch = 1,
       cex = 0.4)
```

```
par(mfrow = c(1,1))
```

P-value computation

We compute the p-value as the upper tail of the normal distribution (rigth-side test) in order to detect significantly high values of the plate-wise normalized index.

```
#### Compute P-value for the inhibition index ####
supTable$p.value <- pnorm(normII, mean = 0, sd = 1, lower.tail = FALSE)
```

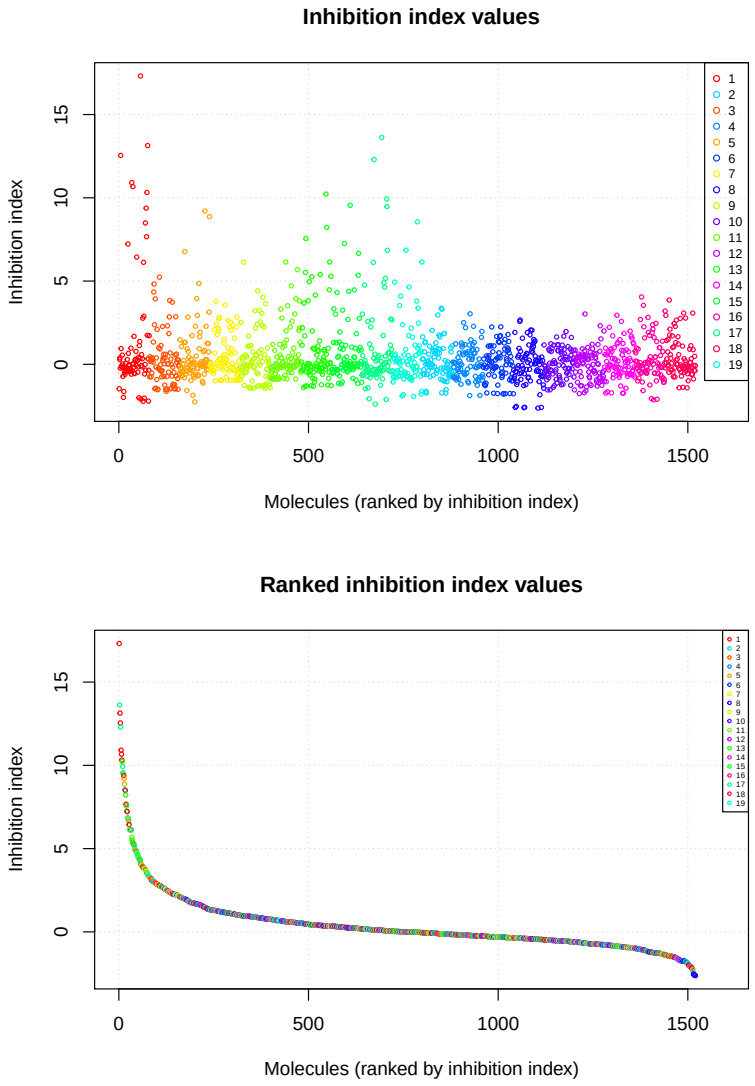

Figure 22: Values of the plate-wise normalized inhibition index for all the tested molecules. Molecules are colored according to the plate number.

```
supTable$log10Pval <- log10(supTable$p.value)
supTable$e.value <- supTable$p.value * length(normII)
supTable$FDR <- p.adjust(supTable$p.value, method = "fdr")
supTable$log10FDR <- log10(supTable$FDR)
```

P-value histogram

```
hist(supTable$p.value, breaks = 20,
     col = "grey",
     main = "P-value histogram after plate-wise normalization",
     xlab = "Nominal P-value (unadjusted)",
     ylab = "Frequency")
```

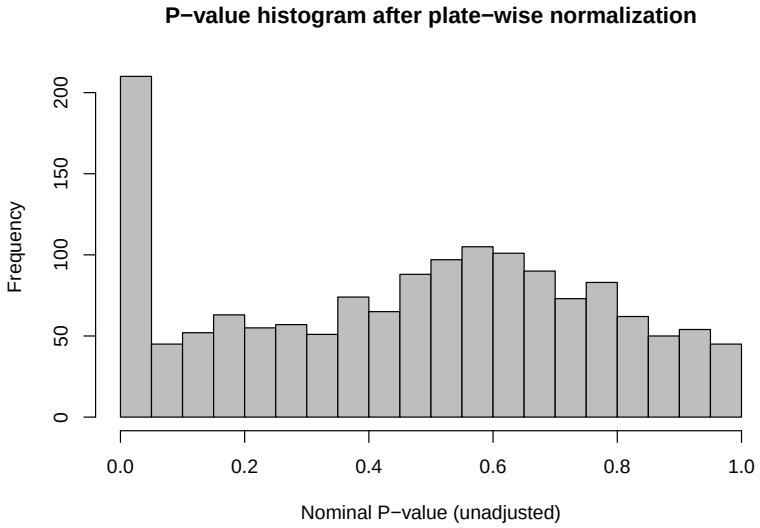

Figure 23: Histogram of the nominal (unadjusted) p-values derived from the plate-wise normalized inhibition index.

Volcano plot

```
#### Volcano plot ####

plot(x = supTable$normInhibIndex,
     y = -supTable$log10FDR,
     col = supTable$color,
     main = "Volcano plot",
     xlab = "Normalized inhibition index",
     ylab = "Significance = -log10(FDR)")
grid()
```

Selection of candidate molecules

```
#### Select significant normalized II values ####
alpha <- 0.05
# table(supTable$FDR < alpha)
selected <- subset(supTable, supTable$FDR < alpha)
```

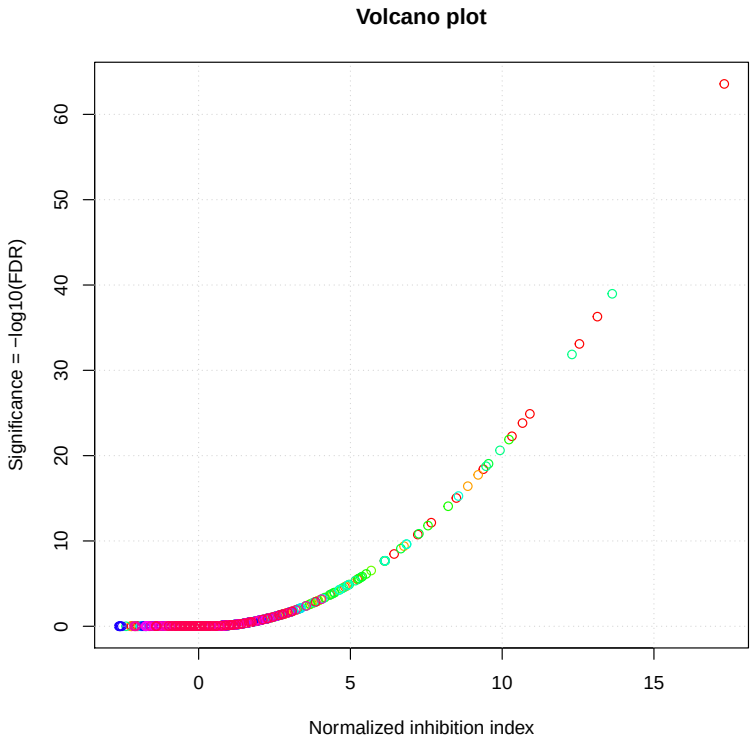

Figure 24: Volcano plot.

```
## Sort by decreasing adjusted p-value
selected <- selected[order(selected$FDR, decreasing = FALSE), ]
# kable(names(selected), row.names=TRUE)

## Print selected molecules
kable(selected[ , c(1:3, 5:7, 10, 12, 15)],
      row.names = FALSE,
      digits = 4,
      caption = "Candidate molecules sorted by significance after plate-wise normalization. ")
```

Table 16: Cand
normalization.

ID	Prestw.number	Chemical.name	Broad.Therapeutic.class
01F08	Prestw-57	Benoxinate hydrochloride	Neuromuscular
09F04	Prestw-693	Promazine hydrochloride	Central Nervous System
01H07	Prestw-76	Dibucaine	Neuromuscular
01A06	Prestw-5	Atracurium besylate	Neuromuscular
09D04	Prestw-1456	Opipramol dihydrochloride	Central Nervous System
01D05	Prestw-34	Triamterene	Metabolism
01D08	Prestw-37	Pyrimethamine	Infectiology
01H05	Prestw-74	Amitryptiline hydrochloride	Central Nervous System
07G07	Prestw-546	Pregnenolone	Endocrinology
09G07	Prestw-706	Chlorcyclizine hydrochloride	Allergology 'Central Nervous System
08E11	Prestw-1284	Hydroxychloroquine sulfate	Metabolism
09G08	Prestw-707	Diphenylpyraline hydrochloride	Allergology 'Central Nervous System
01H03	Prestw-72	Imipramine hydrochloride	Central Nervous System

ID	Prestw.number	Chemical.name	Broad.Therapeutic.class
03G08	Prestw-227	Clemizole hydrochloride	Allergology 'Dermatology 'Infectiology
03H10	Prestw-239	Orphenadrine hydrochloride	Allergology 'Central Nervous System
10G08	Prestw-787	Merbromin disodium salt	Infectiology
01G11	Prestw-70	Tolnaftate	Infectiology
07G09	Prestw-548	Chloroquine diphosphate	Metabolism
01H04	Prestw-73	Sulindac	Central Nervous System
07B04	Prestw-493	Omeprazole	Gastroenterology
08D06	Prestw-1410	Exemestane	Endocrinology
01C05	Prestw-24	Norethynodrel	Endocrinology
09G09	Prestw-708	Benzethonium chloride	Infectiology
10D08	Prestw-757	Chlorotrianisene	Endocrinology
03B05	Prestw-174	Alverine citrate salt	Neuromuscular
08H03	Prestw-632	Dipivefrin hydrochloride	Ophthalmology
01E08	Prestw-47	Ticlopidine hydrochloride	Hematology
06D11	Prestw-440	Epiandrosterone	Endocrinology
07H07	Prestw-1144	Mirtazapine	Central Nervous System
10H10	Prestw-799	Pridinol methanesulfonate salt	Central Nervous System
05A10	Prestw-329	Tacrine hydrochloride	Central Nervous System
01G06	Prestw-65	Diphenhydramine hydrochloride	Allergology 'Central Nervous System
09D02	Prestw-671	Dydrogesterone	Endocrinology
06H02	Prestw-1358	Vatalanib	Oncology
07B03	Prestw-492	Nitrofural	Infectiology
07F02	Prestw-531	Pirenperone	Central Nervous System
08H02	Prestw-1210	Alendronate sodium	Metabolism
07H10	Prestw-1817	Tazarotene	Dermatology
07C10	Prestw-509	Bromperidol	Central Nervous System
02C08	Prestw-1314	Pioglitazone	Endocrinology
09G04	Prestw-703	Famprofazone	Central Nervous System 'Metabolism
07C03	Prestw-502	Biperiden hydrochloride	Central Nervous System
10A08	Prestw-1140	Liranaftate	Infectiology
09F10	Prestw-699	Hexestrol	Endocrinology
03F02	Prestw-211	Piroxicam	Central Nervous System 'Hematology 'Metabol
02B04	Prestw-93	Azacyclonol	Central Nervous System
09A09	Prestw-1154	Nilvadipine	Cardiovascular
06F06	Prestw-455	Mebhydroline 1,5-naphtalenedisulfonate	Allergology
10E06	Prestw-765	Ethoxyquin	Metabolism
09G02	Prestw-701	Trihexyphenidyl-D,L hydrochloride	Central Nervous System
07H06	Prestw-555	Nifuroxazide	Infectiology 'Metabolism
08G02	Prestw-1506	Mizolastine	Allergology
05E07	Prestw-366	Ambroxol hydrochloride	Respiratory
08E07	Prestw-1331	Rimantadine hydrochloride	Infectiology
02B03	Prestw-92	Zimelidine dihydrochloride monohydrate	Central Nervous System
08B06	Prestw-1351	Tenatoprazole	Metabolism
07D09	Prestw-518	Budesonide	Endocrinology
10C04	Prestw-743	Medrysone	Metabolism
18B09	Prestw-1951	Eperisone HCl	Neuromuscular
05F11	Prestw-380	Clebopride maleate	Central Nervous System
03E06	Prestw-205	Tolfenamic acid	Central Nervous System 'Metabolism
06G08	Prestw-1157	Rifapentine	Infectiology
02B07	Prestw-96	Guanabenz acetate	Central Nervous System
19B02	Prestw-1996	Budralazine	Cardiovascular
02F05	Prestw-134	Diltiazem hydrochloride	Cardiovascular 'Hematology 'Metabolism

ID	Prestw.number	Chemical.name	Broad.Therapeutic.class
07B10	Prestw-499	Propafenone hydrochloride	Cardiovascular
06H07	Prestw-476	Primaquine diphosphate	Infectiology
10G06	Prestw-785	Dicumarol	Hematology
04B07	Prestw-256	Isotretinoin	Dermatology
02G02	Prestw-141	Verapamil hydrochloride	Cardiovascular
07A09	Prestw-488	Dosulepin hydrochloride	Central Nervous System
05G08	Prestw-387	Carbetapentane citrate	Central Nervous System 'Neuromuscular
04D11	Prestw-280	Quinidine hydrochloride monohydrate	Cardiovascular 'Infectiology
18C04	Prestw-1961	Methandrostenolone	Endocrinology
10B11	Prestw-1820	Amprenavir	Infectiology
06E06	Prestw-445	Cyclobenzaprine hydrochloride	Neuromuscular
10G10	Prestw-789	Drofenine hydrochloride	Neuromuscular
11E11	Prestw-850	Equilin	Endocrinology
08A08	Prestw-1139	Itraconazole	Infectiology 'Metabolism
06C08	Prestw-1393	Dibenzepine hydrochloride	Central Nervous System
11F03	Prestw-1454	Nylidrin	Cardiovascular
08C11	Prestw-1409	Etretinate	Dermatology
05F02	Prestw-371	Ketotifen fumarate	Allergology
05D10	Prestw-359	Dextromethorphan hydrobromide monohydrate	Central Nervous System
18H11	Prestw-2043	Artenimol	Infectiology
03H09	Prestw-238	Lomefloxacin hydrochloride	Infectiology
19E10	Prestw-2052	Dilevalol	Cardiovascular
19H04	Prestw-1940	Acetyl spiramycin	Infectiology
05F08	Prestw-377	Nafronyl oxalate	Cardiovascular 'Neuromuscular
12E07	Prestw-926	Idazoxan hydrochloride	Central Nervous System
06G05	Prestw-1323	Quetiapine hemifumarate	Central Nervous System
09A03	Prestw-1270	Gefitinib	Oncology
16C10	Prestw-1710	Ethoxzolamide	Ophthalmology 'Gastroenterology 'Central Ner
08B02	Prestw-571	Tetracaïne hydrochloride	Neuromuscular
11E02	Prestw-1455	Olanzapine	Central Nervous System
17D04	Prestw-1857	Oxiglutatione	Ophthalmology
19A03	Prestw-2045	Eletriptan	Central Nervous System
03E08	Prestw-1181	Tibolone	Endocrinology
01G07	Prestw-66	Minaprine dihydrochloride	Central Nervous System
02G11	Prestw-150	Dihydroergotamine tartrate	Central Nervous System
02F04	Prestw-133	Hydroxyzine dihydrochloride	Allergology 'Central Nervous System
04H02	Prestw-311	Ifenprodil tartrate	Cardiovascular
03C03	Prestw-182	Levamisole hydrochloride	Immunology 'Infectiology
04C06	Prestw-265	Dimenhydrinate	Allergology 'Central Nervous System
18D06	Prestw-2008	Azaribine	Oncology 'Dermatology
06C11	Prestw-430	Cisapride	Gastroenterology
19F05	Prestw-2019	Vonoprazan	Gastroenterology
01G04	Prestw-63	Nifedipine	Cardiovascular
08H05	Prestw-1463	Tomoxetine hydrochloride	Central Nervous System
19D11	Prestw-2067	Cyclofenil	Endocrinology
04C05	Prestw-264	Dyclonine hydrochloride	Neuromuscular
09H08	Prestw-717	Finasteride	Endocrinology
08E05	Prestw-1252	Butenafine hydrochloride	Infectiology 'Metabolism
18B06	Prestw-1945	Exifone	Central Nervous System

Conclusions

- The analysis strongly suggests a batch effect: the distribution of viability measures show strong inter-plate differences. In particular, the viability measures of plates 11 to 19 are spread over the whole range from the virus control to the cell control.
- To select candidate molecules, the fact to set a plate-wise threshold based on the mean viability of the arbidol duplicates might lead to ignore highly interesting candidates.
- I propose here a plate-wise standardisation based on the median (for centering) and inter-quartile range (to standardise the dispersion).
- This enables to compute a p-value (expected rate of false positive among all the tested molecules). This nominal p-value has to be corrected for multiple testing, in order to estimate the False Discovery Rate (FDR,i.e. the expected rate of false positives among the molecules declared positive). With a threshold of 0.05 on the FDR, 114 molecules are declared significant and could be considered as candidate for further characterization.

Libraries and versions

For the sake of reproducibility, we list hereafter the R libraries used to generate this report, as well as their versions.

```
sessionInfo()
```

```
R version 3.6.1 (2019-07-05)
Platform: x86_64-apple-darwin15.6.0 (64-bit)
Running under: macOS Mojave 10.14.6

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] grid      stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
 [1] vioplot_0.3.4       zoo_1.8-7           sm_2.2-5.6          fitdistrplus_1.0-14 npsurv_0.4-0

loaded via a namespace (and not attached):
 [1] Rcpp_1.0.4           highr_0.8            pillar_1.4.3         cellranger_1.1.0     compiler_3.6.1
[15] rlang_0.4.5          Matrix_1.2-18        cli_2.0.2            yaml_2.2.1           xfun_0.12
[29] lambda.r_1.2.4       magrittr_1.5         htmltools_0.4.0      ellipsis_0.3.0       splines_3.6.1
```

To do

- log2 transformatio before standardizing (and comparison of the targets)
- Add the formula of the inhibition index

- distribution de valeurs plaque par plaque –> vérifier si certaines plaques ont l'air d'avoir plus de 20 cibles (percentile75)
- $$II =$$

  : valeurs relative à la moyenne de l'arbidol. Comme l'arbidol change d'une plaque à l'autre, pas évident qu'on la conserve. Plutôt baser sur la viabilité des deux contrôles.

$$v'_{m,i} = log2(V_{m,i})$$

- Afficher les numéros des plaques sur les dot plots, plutôt que le nombre de molécules
- Vérifier si les plaques où il y a plein de molécules comportent des
- Hybrid approach
  - for each molecule, we compute a statistical significance
  - intersection: high-confidence molecules, i.e. staistically significant relative to their plate + biologically relevant with respect to the arbidol threshold
- replace candidates by hits
- Export
  - full result table
  - statistically significant
  - above arbidol
  - high-confidence hits (intersection between the two lists)