NLS_Report

Censoring criteria

```
db %>% filter(CENSOR != "NO") %>% distinct(CMPD, SITE, TIME, CENSOR) %>%
  pivot_wider(names_from=SITE, values_from=TIME, values_fn=function(x){
    paste(x, collapse=", ")
  })
```

```
## # A tibble: 10 x 6
##    CMPD     CENSOR INJ_PC  INJ_IP  `IMM + INJ_PC (Combination)` IMM
##    <fct>    <chr>  <chr>   <chr>   <chr>                        <chr>
##  1 S-CY3A   AGGREG 0.25    0.25    0.25                         <NA>
##  2 S-CY5.5A BLOQ   <NA>    <NA>    <NA>                         1, 3
##  3 S-CY5.5A AGGREG 0.25    0.25    0.25                         <NA>
##  4 S-CY5A   BLOQ   <NA>    <NA>    <NA>                         1, 3, 6
##  5 S-CY5A   UNEXPL 0.25, 1 0.25, 1 0.25, 1                      <NA>
##  6 FAMA     UNEXPL 0.25, 1 0.25, 1 0.25, 1                      <NA>
##  7 TAMRA    AGGREG 0.25    0.25    0.25                         <NA>
##  8 R6GA     BLOQ   <NA>    <NA>    <NA>                         1, 3, 6
##  9 R6GA     AGGREG 0.25    0.25    0.25                         <NA>
## 10 CY3A     AGGREG 0.25, 1 0.25, 1 0.25, 1                      <NA>
```

Model fit

```
cmpdDb <- lapply(unique(db$CMPD), function(myCMPD) {
  db %>%
  mutate(Dose = ifelse(SITE == "IMM", 0, 2E-6*0.25/1000), #2mg/kg (=2ug/g) at 0.25mg bodyweight
         M = ifelse(HAS_IMM, 10/1000/1000*MW, 0)) %>% #10uM
  filter(CMPD==myCMPD)# %>%
  #mutate(PRED = fun(TIME, Dose=Dose, M=M, FLUOR=10E9, Ke=0.01, Q=1E-9))
})

formula <- RFU ~ FLUOR*(Dose*exp(-Ke*TIME)+M*Q/Ke *(1-exp(-Ke*TIME)))
start <- c(Q = 1E-9, Ke=0.01, FLUOR=10E9)
```

```
m2 <- lapply(seq_along(cmpdDb), function(i) {
  cat("===== COMPOUND ",cmpdDb[[i]]$CMPD[1], "\n")
  fitDb <- filter(cmpdDb[[i]], CENSOR == "NO")
  #fitDb <- cmpdDb[[i]]
  m <- nls(formula,
      data=fitDb,
      start=start,
      # lower=start*0,
      # upper=start*100,
      # algorithm = "port",
      control=nls.control(maxiter=1000*100, minFactor=1E-24))
  nlstools::overview(m)
  m
})
```

```
## ===== COMPOUND  1
##
## ------
## Formula: RFU ~ FLUOR * (Dose * exp(-Ke * TIME) + M * Q/Ke * (1 - exp(-Ke
*
##     TIME)))
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## Q     2.318e-10  1.075e-10   2.156   0.0323 *
## Ke    1.208e-02  1.842e-03   6.556 4.79e-10 ***
## FLUOR 8.293e+09  2.472e+08  33.551  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.041 on 196 degrees of freedom
##
## Number of iterations to convergence: 5
## Achieved convergence tolerance: 9.729e-07
##
## ------
## Residual sum of squares: 213
##
```

```
## ------
## t-based confidence interval:
##               2.5%        97.5%
## Q     1.975150e-11 4.438884e-10
## Ke    8.443300e-03 1.570865e-02
## FLUOR 7.805764e+09 8.780731e+09
##
## ------
## Correlation matrix:
##                   Q        Ke        FLUOR
## Q      1.000000000 0.3765945 -0.004486099
## Ke     0.376594468 1.0000000  0.561522269
## FLUOR -0.004486099 0.5615223  1.000000000
##
## ===== COMPOUND  2
##
## ------
## Formula: RFU ~ FLUOR * (Dose * exp(-Ke * TIME) + M * Q/Ke * (1 - exp(-Ke
*
##     TIME)))
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## Q     3.426e-10  7.470e-11   4.585 8.57e-06 ***
## Ke    2.291e-02  2.249e-03  10.186  < 2e-16 ***
## FLUOR 2.632e+09  6.954e+07  37.854  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2763 on 176 degrees of freedom
##
## Number of iterations to convergence: 6
## Achieved convergence tolerance: 6.075e-06
##
## ------
## Residual sum of squares: 13.4
##
```

```
## ------
## t-based confidence interval:
##               2.5%        97.5%
## Q     1.951194e-10 4.899830e-10
## Ke    1.846802e-02 2.734361e-02
## FLUOR 2.495141e+09 2.769625e+09
##
## ------
## Correlation matrix:
##                  Q        Ke       FLUOR
## Q     1.000000000 0.4476229 0.005580052
## Ke    0.447622906 1.0000000 0.527399997
## FLUOR 0.005580052 0.5274000 1.000000000
##
## ===== COMPOUND  3
##
## ------
## Formula: RFU ~ FLUOR * (Dose * exp(-Ke * TIME) + M * Q/Ke * (1 - exp(-Ke
*
##     TIME)))
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## Q     1.608e-09  1.520e-10  10.580  < 2e-16 ***
## Ke    1.375e-02  1.927e-03   7.137  5.3e-11 ***
## FLUOR 1.088e+09  3.764e+07  28.918  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1251 on 135 degrees of freedom
##
## Number of iterations to convergence: 5
## Achieved convergence tolerance: 9.937e-07
##
## ------
## Residual sum of squares: 2.11
##
```

```
## ------
## t-based confidence interval:
##               2.5%        97.5%
## Q     1.307518e-09 1.908732e-09
## Ke    9.939665e-03 1.755985e-02
## FLUOR 1.013955e+09 1.162826e+09
##
## ------
## Correlation matrix:
##                Q        Ke      FLUOR
## Q      1.0000000 0.4368652 -0.1191826
## Ke     0.4368652 1.0000000  0.6131015
## FLUOR -0.1191826 0.6131015  1.0000000
##
## ===== COMPOUND  4
##
## ------
## Formula: RFU ~ FLUOR * (Dose * exp(-Ke * TIME) + M * Q/Ke * (1 - exp(-Ke
*
##     TIME)))
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## Q     1.708e-09  2.259e-10   7.560 3.01e-12 ***
## Ke    1.207e-02  2.240e-03   5.391 2.50e-07 ***
## FLUOR 4.352e+09  1.892e+08  23.006  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6056 on 159 degrees of freedom
##
## Number of iterations to convergence: 5
## Achieved convergence tolerance: 8.29e-06
##
## ------
## Residual sum of squares: 58.3
##
```

```
## ------
## t-based confidence interval:
##               2.5%        97.5%
## Q     1.261944e-09 2.154404e-09
## Ke    7.649905e-03 1.649689e-02
## FLUOR 3.978084e+09 4.725239e+09
##
## ------
## Correlation matrix:
##                Q        Ke      FLUOR
## Q      1.0000000 0.3709325 -0.1193837
## Ke     0.3709325 1.0000000  0.6398915
## FLUOR -0.1193837 0.6398915  1.0000000
##
## ===== COMPOUND  5
##
## ------
## Formula: RFU ~ FLUOR * (Dose * exp(-Ke * TIME) + M * Q/Ke * (1 - exp(-Ke
*
##     TIME)))
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## Q     1.453e-09  2.525e-10   5.754 3.41e-08 ***
## Ke    5.194e-02  4.969e-03  10.452  < 2e-16 ***
## FLUOR 1.970e+10  5.773e+08  34.131  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.891 on 192 degrees of freedom
##
## Number of iterations to convergence: 6
## Achieved convergence tolerance: 5.478e-06
##
## ------
## Residual sum of squares: 686
##
```

```
## ------
## t-based confidence interval:
##               2.5%        97.5%
## Q     9.545567e-10 1.950435e-09
## Ke    4.213601e-02 6.173743e-02
## FLUOR 1.856534e+10 2.084269e+10
##
## ------
## Correlation matrix:
##               Q        Ke     FLUOR
## Q     1.0000000 0.5363328 0.1057257
## Ke    0.5363328 1.0000000 0.5779940
## FLUOR 0.1057257 0.5779940 1.0000000
##
## ===== COMPOUND  6
##
## ------
## Formula: RFU ~ FLUOR * (Dose * exp(-Ke * TIME) + M * Q/Ke * (1 - exp(-Ke
*
##     TIME)))
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## Q     3.226e-09  2.945e-10   10.96  < 2e-16 ***
## Ke    1.077e-02  2.373e-03    4.54 1.08e-05 ***
## FLUOR 1.388e+09  5.574e+07   24.90  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2421 on 165 degrees of freedom
##
## Number of iterations to convergence: 4
## Achieved convergence tolerance: 2.091e-06
##
## ------
## Residual sum of squares: 9.67
##
```

```
## ------
## t-based confidence interval:
##               2.5%        97.5%
## Q     2.644912e-09 3.807841e-09
## Ke    6.088411e-03 1.546020e-02
## FLUOR 1.277797e+09 1.497898e+09
##
## ------
## Correlation matrix:
##                Q        Ke      FLUOR
## Q      1.0000000 0.5211084 -0.1753926
## Ke     0.5211084 1.0000000  0.5317858
## FLUOR -0.1753926 0.5317858  1.0000000
##
## ===== COMPOUND  7
##
## ------
## Formula: RFU ~ FLUOR * (Dose * exp(-Ke * TIME) + M * Q/Ke * (1 - exp(-Ke
*
##     TIME)))
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## Q     8.365e-09  9.226e-10   9.067 4.37e-16 ***
## Ke    4.981e-02  7.564e-03   6.586 6.28e-10 ***
## FLUOR 5.444e+09  3.497e+08  15.565  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9914 on 159 degrees of freedom
##
## Number of iterations to convergence: 55
## Achieved convergence tolerance: 9.637e-06
##
## ------
## Residual sum of squares: 156
##
```

```
## ------
## t-based confidence interval:
##                2.5%         97.5%
## Q      6.542843e-09 1.018712e-08
## Ke     3.487355e-02 6.475101e-02
## FLUOR 4.753014e+09 6.134517e+09
##
## ------
## Correlation matrix:
##                  Q         Ke       FLUOR
## Q       1.0000000 0.7178605 -0.2070472
## Ke      0.7178605 1.0000000  0.4475268
## FLUOR -0.2070472 0.4475268  1.0000000
```

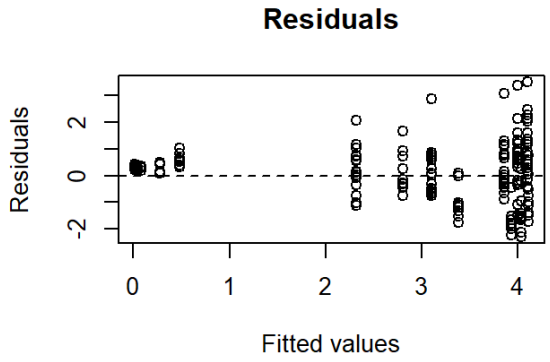
Coefficients

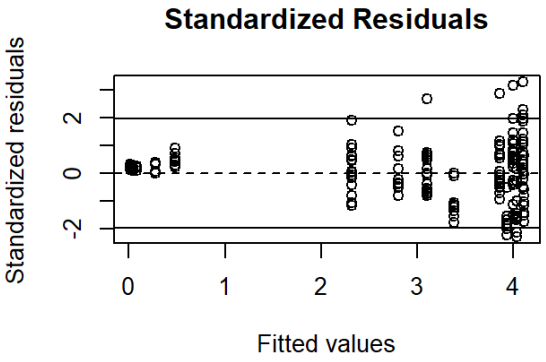
```
lapply(m2, coef) %>%
  bind_rows %>%
  mutate(T12 = log(2)/Ke, CMPDName = as.character(unique(db$CMPD))) %>%
  pander::pander()
```

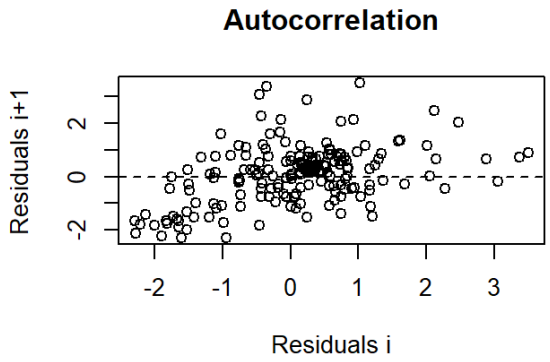
Q	Ke	FLUOR	T12	CMPDName
2.318e-10	0.01208	8.293e+09	57.4	S-CY3A
3.426e-10	0.02291	2.632e+09	30.26	S-CY5.5A
1.608e-09	0.01375	1.088e+09	50.41	S-CY5A
1.708e-09	0.01207	4.352e+09	57.41	FAMA
1.452e-09	0.05194	1.97e+10	13.35	TAMRA
3.226e-09	0.01077	1.388e+09	64.33	R6GA
8.365e-09	0.04981	5.444e+09	13.92	CY3A

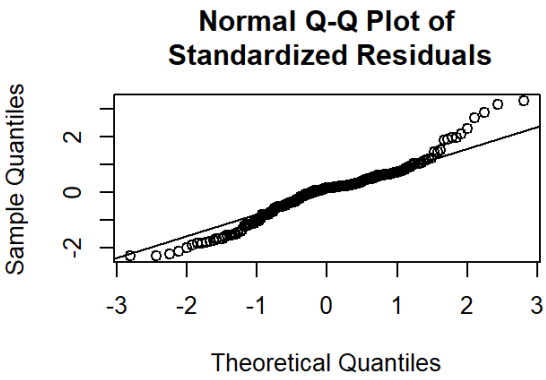
Figures

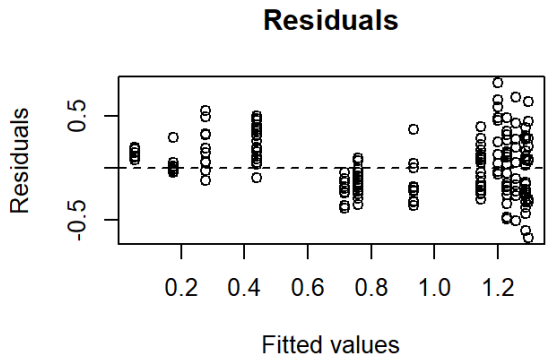
```
for(i in seq_along(cmpdDb)) {
  #cmpdDb[[i]]$IRES <- (cmpdDb[[i]]$IPRED - cmpdDb[[i]]$RFU)
  j <- cmpdDb[[i]]$CENSOR == "NO"
  res <- nlstools::nlsResiduals(m2[[i]])

  cmpdDb[[i]]$IRES <- cmpdDb[[i]]$RFU - predict(m2[[i]], newdata=cmpdDb[[i]
])

  cmpdDb[[i]]$IWRES <- NA
  cmpdDb[[i]]$IWRES[j] <- res$resi2[,2]
  cmpdDb[[i]]$ARES <- NA
  cmpdDb[[i]]$ARES[j] <- res$resi3[,2]
  cmpdDb[[i]]$MODEL <- list(m2[[i]])
  plot(res)
}
```

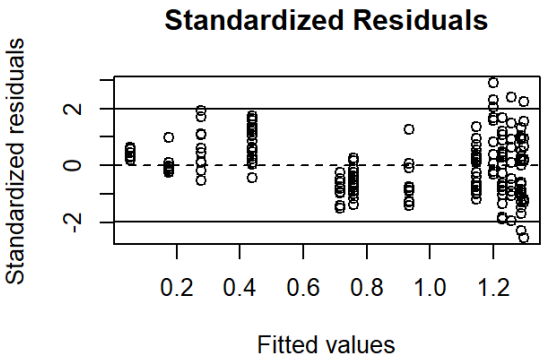

Residuals
Residuals
Fitted values

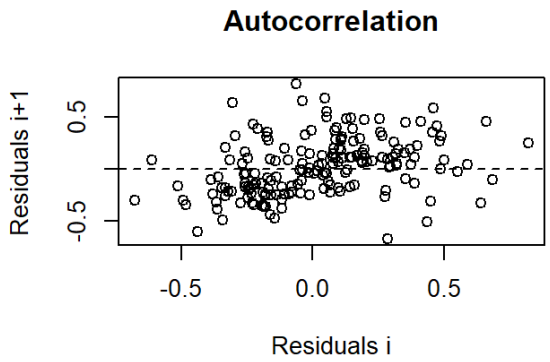

Standardized Residuals
Standardized residuals
Fitted values

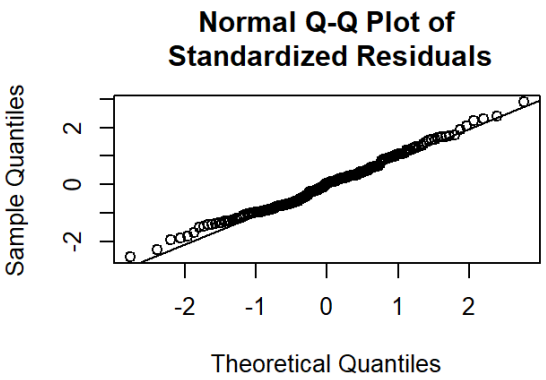

Autocorrelation
Residuals i+1
Residuals i

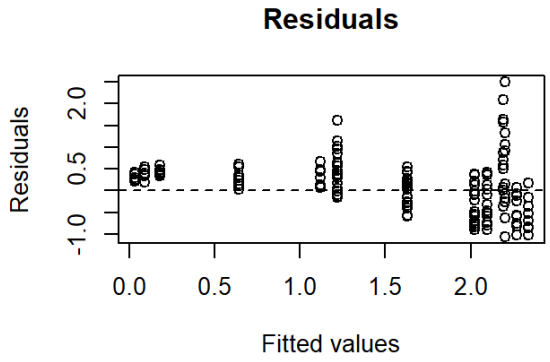

Normal Q-Q Plot of
Standardized Residuals
Sample Quantiles
Theoretical Quantiles

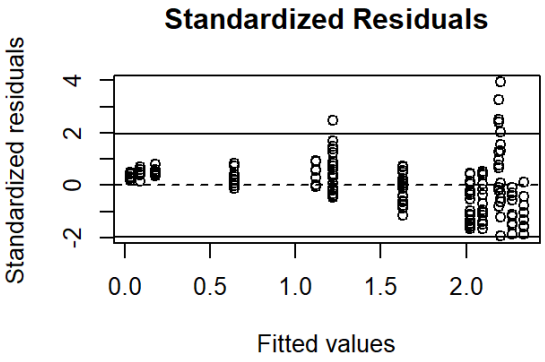

Residuals
Residuals
Fitted values

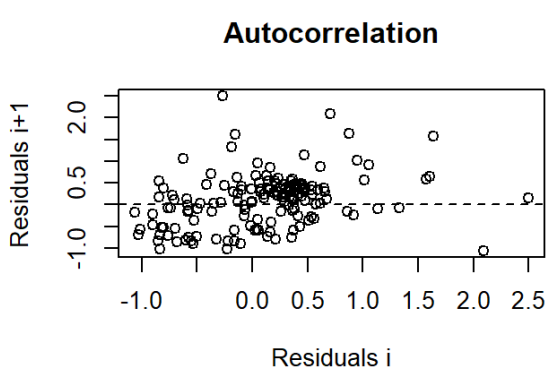

Standardized Residuals
Standardized residuals
Fitted values

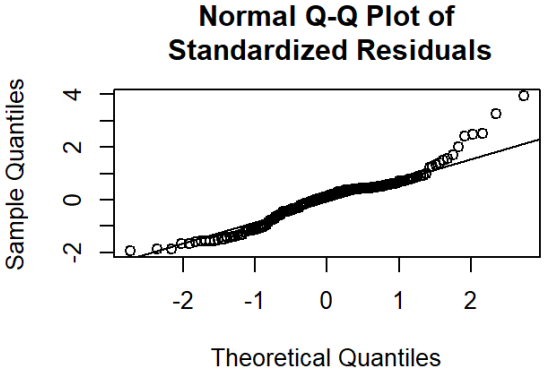

Autocorrelation
Residuals i+1
Residuals i

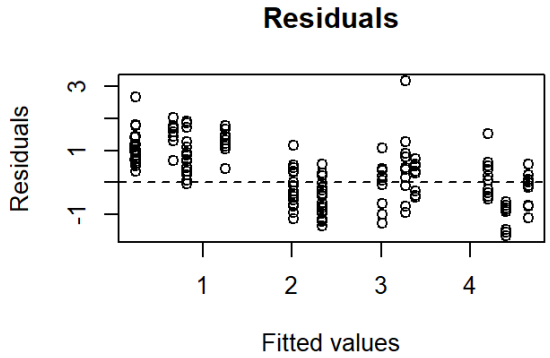

Normal Q-Q Plot of
Standardized Residuals
Sample Quantiles
Theoretical Quantiles

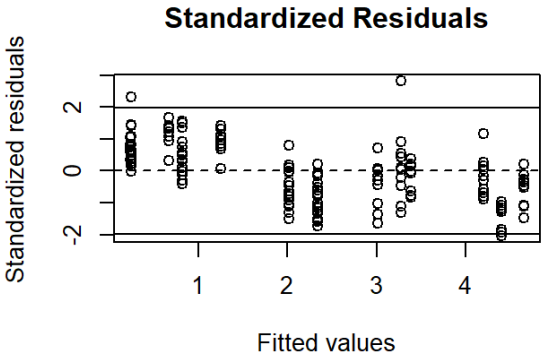


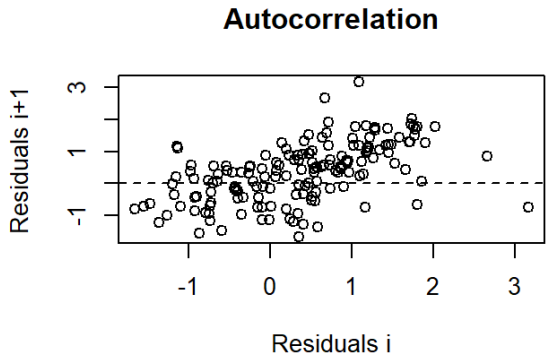


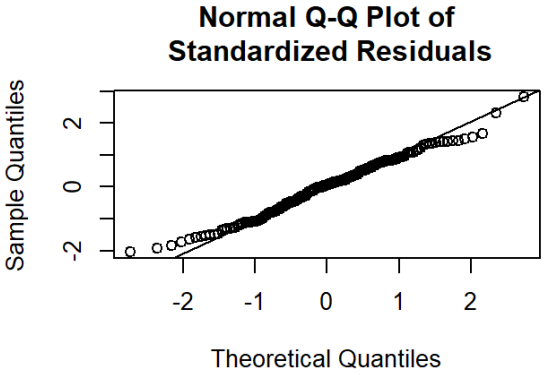













```
predictDb <- lapply(seq_along(m2), function(i){
  predictDb <- cmpdDb[[i]] %>%
```

```
    distinct(CMPD, MW, SITE, Dose, M, HAS_IMM) %>%
    expand_grid(TIME=seq(0, 50, by=0.01))
  predictDb$RFU <- predict(m2[[i]], newdata=predictDb)
  limitDb <- cmpdDb[[i]] %>% filter(HAS_IMM) %>%
    distinct(CMPD, MW, SITE, Dose, M, HAS_IMM) %>%
    mutate(TIME=99999)
  limitDb$RFU_LIM <- predict(m2[[i]], newdata=limitDb)

  predictDb %>% left_join(limitDb %>% select(-TIME))
}) %>% bind_rows %>%
  left_join(data.frame(
    CMPD=factor(c("S-CY3A", "S-CY5.5A", "S-CY5A", "FAMA", "TAMRA", "R6GA",
"CY3A")),
    MIN=c(0.10, 0.10, 0.10, 0.10, 0.3, 0.3, 0.3)
  )) %>% filter(RFU > MIN)
```

```
## Joining, by = c("CMPD", "MW", "SITE", "HAS_IMM", "Dose", "M")
## Joining, by = c("CMPD", "MW", "SITE", "HAS_IMM", "Dose", "M")
## Joining, by = c("CMPD", "MW", "SITE", "HAS_IMM", "Dose", "M")
## Joining, by = c("CMPD", "MW", "SITE", "HAS_IMM", "Dose", "M")
## Joining, by = c("CMPD", "MW", "SITE", "HAS_IMM", "Dose", "M")
## Joining, by = c("CMPD", "MW", "SITE", "HAS_IMM", "Dose", "M")
## Joining, by = c("CMPD", "MW", "SITE", "HAS_IMM", "Dose", "M")
## Joining, by = "CMPD"
```

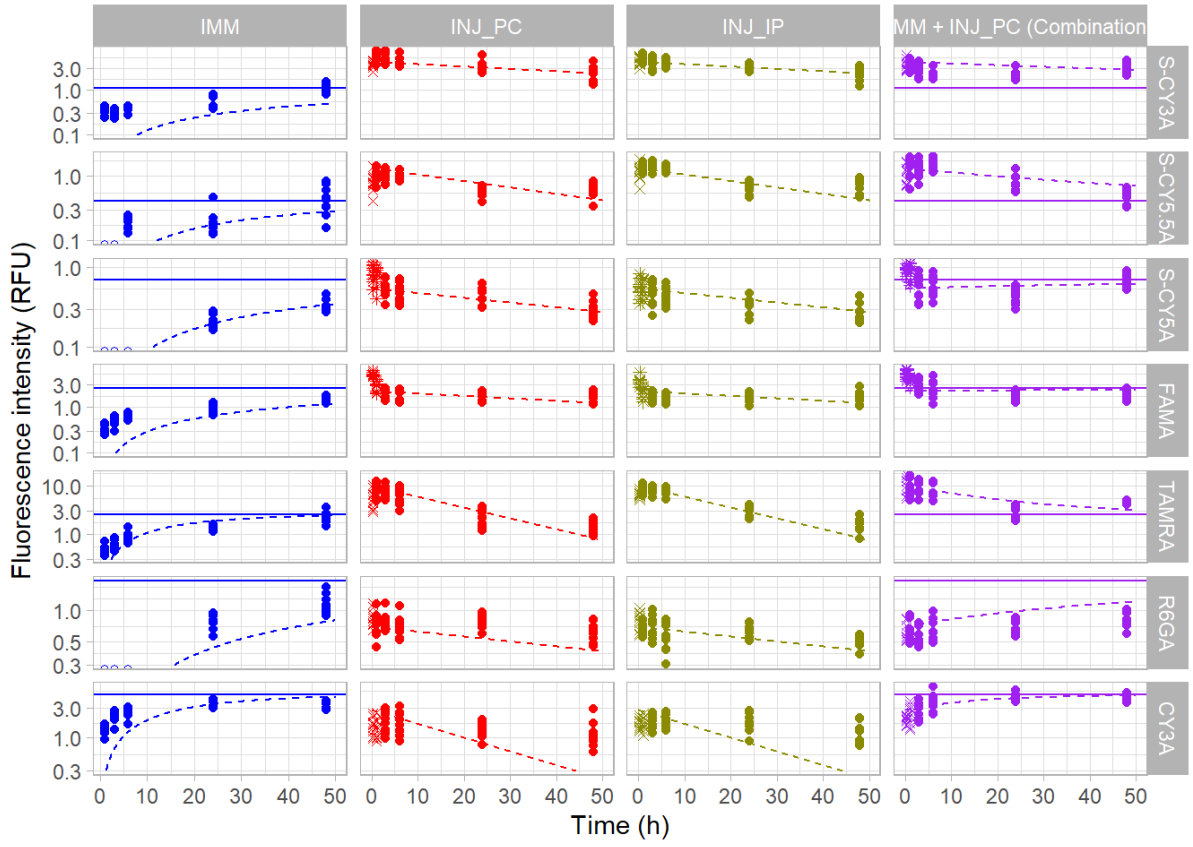
```
result <- bind_rows(cmpdDb)

resultIres <- result %>%
  group_by(CMPD) %>%
  filter(CENSOR == "NO") %>%
  summarize(SIGMA = summary(MODEL[1][[1]])$sigma,
            M_IRES = mean(IRES, na.rm=TRUE))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
# ggplot(result, aes(x=TIME, y=RFU, color=SITE)) +
#   geom_point() +
#   geom_line(aes(y=PRED)) +
#   facet_grid(CMPD~SITE) +
#   scale_y_log10()
```

```
ggplot(result, aes(x=TIME, y=RFU, color=SITE)) +
  geom_hline(data=predictDb, aes(yintercept=RFU_LIM, color=SITE), linetype=
1) +
  geom_point(aes(shape=CENSOR)) +
  geom_line(data=predictDb, linetype=2) +
  facet_grid(CMPD~SITE, scales="free") +
  #facet_wrap(CMPD~., scales="free") +
  scale_y_log10() +
  #scale_x_continuous(minor_breaks=c(1, 3, 6, 24, 48, 0.25)) +
  scale_shape_manual(values=c(NO=16, AGGREG=4, UNEXPL=8, BLOQ=1, `IMM+INJ`=
3)) +
  labs(x="Time (h)", y="Fluorescence intensity (RFU)") +
  theme_light() +
  theme(legend.position="none") +
  scale_color_manual(values=c(IMM="blue", INJ_PC="red", INJ_IP="yellow4", `
IMM + INJ_PC (Combination)`="purple"))
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Removed 68868 rows containing missing values (geom_hline).
```




```
ggplot(result %>% filter(CENSOR != "BLOQ"), aes(x=TIME, y=IRES, color=SITE)
) +
```

```
  geom_point(aes(shape=CENSOR)) +
  facet_grid(CMPD~SITE, scales="free_y") +
  geom_hline(yintercept=0) +
  geom_hline(data=resultIres, aes(yintercept=M_IRES), linetype=2) +
  geom_hline(data=resultIres, aes(yintercept=M_IRES+1.96*SIGMA), linetype=2
) +
  geom_hline(data=resultIres, aes(yintercept=M_IRES-1.96*SIGMA), linetype=2
) +
  #geom_smooth(se=F) +
  labs(x="Time (h)", y="Residuals") +
  scale_shape_manual(values=c(NO=16, AGGREG=4, UNEXPL=8, BLOQ=1, `IMM+INJ`=
3)) +
  theme_light() +
  theme(legend.position="none") +
  scale_color_manual(values=c(IMM="blue", INJ_PC="red", INJ_IP="yellow4", `
IMM + INJ_PC (Combination)`="purple"))
```

AUC_0,48h calculation

```
predictDb <- lapply(seq_along(m2), function(i){
  predictDb <- cmpdDb[[i]] %>%
    distinct(CMPD, MW, SITE, Dose, M, HAS_IMM) %>%
    expand_grid(TIME=seq(0, 48, by=0.01))
  predictDb$RFU <- predict(m2[[i]], newdata=predictDb)
  predictDb
}) %>% bind_rows


sumDb <- predictDb %>% group_by(CMPD, SITE) %>%
  summarize(AUC = sum(RFU*0.01)) %>%
  pivot_wider(names_from=SITE, values_from=AUC)
## `summarise()` regrouping output by 'CMPD' (override with `.groups` argum
ent)
sumDb$RE_0_48 = sumDb$IMM / sumDb$INJ_PC
sumDb$Fcom_0_48 = sumDb$`IMM + INJ_PC (Combination)` / sumDb$IMM


sumDb$CMPD <- 1:7
sumDb1 <- sumDb
```

```
pander::pander(sumDb)
```

CMPD	IMM	INJ_PC	INJ_IP	IMM + INJ_PC (Combination)	I
1	12.75	151.1	151.1	163.8	
2	7.839	38.33	38.33	46.17	
3	8.97	19.13	19.13	28.1	
4	29.45	79.28	79.28	108.7	
5	78.17	174.1	174.1	252.2	
6	20.23	26.01	26.01	46.25	
7	144.3	49.66	49.66	193.9	

AUC_0,3h calculation

```
predictDb <- lapply(seq_along(m2), function(i){
  predictDb <- cmpdDb[[i]] %>%
    distinct(CMPD, MW, SITE, Dose, M, HAS_IMM) %>%
    expand_grid(TIME=seq(0, 3, by=0.01))
  predictDb$RFU <- predict(m2[[i]], newdata=predictDb)
  predictDb
}) %>% bind_rows


sumDb <- predictDb %>% group_by(CMPD, SITE) %>%
  summarize(AUC = sum(RFU*0.01)) %>%
  pivot_wider(names_from=SITE, values_from=AUC)
```

```
## `summarise()` regrouping output by 'CMPD' (override with `.groups` argum
ent)
```

```
sumDb$RE_0_3 = sumDb$IMM / sumDb$INJ_PC
sumDb$Fcom_0_3 = sumDb$`IMM + INJ_PC (Combination)` / sumDb$IMM


sumDb$CMPD <- 1:7
```

```
sumDb2 <- sumDb


pander::pander(sumDb)
```

CMPD	IMM	INJ_PC	INJ_IP	IMM + INJ_PC (Combination)
1	0.05934	12.26	12.26	12.32
2	0.04196	3.829	3.829	3.871
3	0.0427	1.605	1.605	1.647
4	0.1371	6.432	6.432	6.569
5	0.5739	27.46	27.46	28.03
6	0.09252	2.055	2.055	2.148
7	1.038	7.61	7.61	8.648

Multiple Linera Regression

```
physico <- readxl::read_excel("CMPDs_20200901.xlsx")


fitSet <- lapply(m2, coef) %>% bind_rows() %>%
  mutate(CMPD = seq_len(7),
         T12 = log(2) / Ke) %>%
  left_join(physico) %>%
  left_join(sumDb, by="CMPD")
## Joining, by = "CMPD"
summary(base)
##
## Call:
## lm(formula = Q ~ 1, data = fitSet)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
```

```
## -1.736e-09 -1.149e-09 -3.845e-10  4.796e-10  3.460e-09
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.005e-09  6.919e-10   2.898   0.0274 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.831e-09 on 6 degrees of freedom
m1 <- step(lm(RE_0_3 ~ 0, data=fitSet),
     scope= ~ `Log P` + MW + HBA + HBD + TPSA + rotor + MR,
     direction="both",
     k=2,
     trace=9999 )
## Start:  AIC=-40.23
## RE_0_3 ~ 0
##
##           Df Sum of Sq      RSS     AIC
## + MR       1 0.0091346 0.013223 -41.902
## + rotor    1 0.0072523 0.015105 -40.971
## + MW       1 0.0071810 0.015176 -40.938
## + HBD      1 0.0056798 0.016677 -40.277
## <none>                 0.022357 -40.226
## + `Log P`  1 0.0051271 0.017230 -40.049
## + TPSA     1 0.0019726 0.020385 -38.872
## + HBA      1 0.0014877 0.020869 -38.708
##
## Step:  AIC=-41.9
## RE_0_3 ~ MR - 1
##
##           Df Sum of Sq       RSS     AIC
## + `Log P`  1 0.0099352 0.0032873 -49.645
## + HBA      1 0.0053132 0.0079094 -43.499
## + TPSA     1 0.0050338 0.0081888 -43.256
## + MW       1 0.0043494 0.0088732 -42.694
## <none>                 0.0132226 -41.902
## + rotor    1 0.0014744 0.0117482 -40.730
```

```
## - MR       1 0.0091346 0.0223572 -40.226
## + HBD      1 0.0001184 0.0131042 -39.965
##
## Step:  AIC=-49.65
## RE_0_3 ~ MR + `Log P` - 1
##
##           Df Sum of Sq       RSS     AIC
## <none>                 0.0032873 -49.645
## + rotor    1 0.0007943 0.0024930 -49.581
## + HBA      1 0.0001523 0.0031351 -47.977
## + TPSA     1 0.0000730 0.0032144 -47.802
## + HBD      1 0.0000411 0.0032462 -47.733
## + MW       1 0.0000175 0.0032698 -47.683
## - `Log P`  1 0.0099352 0.0132226 -41.902
## - MR       1 0.0139427 0.0172301 -40.049
```

```
summary(m1)
```

```
##
## Call:
## lm(formula = RE_0_3 ~ MR + `Log P` - 1, data = fitSet)
##
## Residuals:
##         1         2         3         4         5         6         7
##  0.015642 -0.003146 -0.001514 -0.003694 -0.029571 -0.026485  0.037960
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## MR      2.699e-04  5.861e-05   4.605  0.00581 **
## `Log P` 3.035e-02  7.808e-03   3.887  0.01156 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02564 on 5 degrees of freedom
## Multiple R-squared:  0.853,  Adjusted R-squared:  0.7941
## F-statistic:  14.5 on 2 and 5 DF,  p-value: 0.00829
```