US EPA TOXCAST DATA RELEASE SEPTEMBER 2020 ASSAY INFORMATION & QUALITY STATISTICS

This file describes the contents of the September 2020 ToxCast Assay Quality Summary release. The zip file contains the following assay quality statistic and summary file, not including this README file:

```
## [1] "assay_annotation_information_invitrodb_v3_3.RData"
## [2] "assay_annotation_information_invitrodb_v3_3.xlsx"
## [3] "assay_methods_invitrodb_v3_3.RData"
## [4] "assay_methods_invitrodb_v3_3.xlsx"
## [5] "Assay_Quality_Detailed_Stats.Rdata"
## [6] "Assay_Quality_Detailed_Stats_200819.csv"
## [7] "Assay_Quality_Summary_Stats.Rdata"
## [8] "Assay_Quality_Summary_Stats_200819.csv"
```

In addition to the above listed files, the ToxCast program also released a MySQL dump file containing all data and a beta version of the R package (tcpl) that interacts with the MySQL database used to process all of the data for this release. For information/data not included in the listed summary files, users will need to download and interact with the MySQL database. We also encourage the database users to utilize the 'tcpl' R package containing numerous queries and functionality for easily loading and visualizing the data. At the bottom of this file is an R script to produce all of the listed files, utilizing the MySQL database and 'tcpl' R package.

The assay annotation information file contains all of the annotation fields used to describe an assay, assay component, and assay component endpoint, including assay citation, assay reagent, and assay target information.

The definition of an "assay" is, for the purposes of this package, broken into:

- assay_source – the vendor/origination of the data
- assay – the procedure to generate the component data
- assay_component – the raw data readout(s)
- assay_component_endpoint – the normalized component data

Each assay element is represented by a separate table in the tcpl database. In general, we refer to an "assay_component_endpoint" as an "assay endpoint." As we move down the hierarchy, each additional layer has a one-to-many relationship with the previous layer. For example, an assay component can have multiple assay endpoints, but an assay endpoint can derive only from a single assay component.

All processing occurs by assay component or assay endpoint, depending on the processing type (single-concentration or multiple-concentration) and level. No data are stored at the assay or assay source level. The "assay" and "assay_source" tables store annotations to help in the processing and down-stream understanding/analysis of the data.

Source: https://cran.r-project.org/web/packages/tcpl/vignettes/Introduction_Appendices.html

The assay methods invitrodb file contains all of the methods appliead to each of the assays during each step of the pipeline.

The assay endpoint detailed statistics are derived from the raw concentration response data and provide assay-plate-wise statistics common to the high throughput screening community, including z-prime and ssmd (strictly standardized mean difference). The detailed file provides the median and median absolute deviation across all plates, where applicable. These calculations are performed at the assay component level (i.e., assay readout) rather than the endpoint (i.e., direction of analysis).

- acid = assay component id (unique id for each assay readout)
- acnm = assay component name
- nmed = neutral control well median value, by plate
- nmad = neutral control median absolute deviation, by plate
- pmed = positive control well median value, by plate
- pmad = positive control well median absolute deviation, by plate
- mmed = negative control well median, by plate
- mmad = negative control well median absolute deviation value, by plate
- zprm.p = robust z-prime, median across all plates using positive control wells
- zprm.m = robust z-prime, median across all plates using negative control wells
- ssmd.p = robust ssmd, median across all plates using positive control wells
- ssmd.m = robust ssmd, median across all plates using negative control wells
- cv = median coefficient of variation across all plate
- sn.p = median signal-to-noise across all plates based on positive control wells
- sn.m = median signal-to-noise across all plates based on negative control wells
- sb.p = median signal-to-background across all plates based on positive control wells
- sb.m = median signal-to-background across all plates based on negative control wells

Many of these calculations result in NA values because there may not be plate-level details provided to us or because the analysis process precludes us from making the calculation. This initial release of the quality statistics are for general and relative reference only. Due to the diverse assay technologies and study designs deployed, a highly generalized and robust (median and mad vs mean and sd) set of calculations were performed.

- aeid = assay endpoint id (unique id)
- ocnc = overall concordance among chemical replicates calculated as the percentage of time all samples for a chemical were either negative or positive (e.g., 0 out of 3 or 3 out of 3) over the total number of chemicals with replicates.
- hcnc = hit concordance among chemical replicates calculated as the percentage of time all samples for a chemical were positive (e.g., 3 out of 3) over the total number of chemicals with any replicate being positive (e.g., 1 out of 3 or 2 out of 3). *It should be noted that most of these chemical replicates were separately procured and that these concordance values are highly influenced by the number of replicates.*
- aenm = assay endpoint name (i.e., assay_component_endpoint_name)
- resp_unit = response unit (fold induction or percent activity)
- bmad = baseline median absolute deviation for the assay (based on the response values at the 2 lowest tested concentrations)
- nconc = nominal number of tested concentrations
- coff = the response cutoff used to derive the hit calls (e.g., 5*bmad, 10*bmad)
- test = total number of samples tested
- acnt = number of active samples
- apct = percent active samples
- icnt = number of inactive samples
- ipct = percent of inactive samples
- ncnt = number of samples that could not be modeled (e.g., having less than 4 concs)
- npct = percent not modeled
- mmed = maximum observed response across the assay
- cmax = target (nominal) maximal tested concentration
- cmin = target (nominal) minimal tested concentration
- mtop = maximum modeled response across the assay (max top of curve)
- nrep = target (nominal) number of replicates
- npts = target (nominal) number of points (nconc //* nrep)
- cnst = percent constant model winner (based on having lowest AIC value)
- hill = percent hill model winner (based on having lowest AIC value)
- gnls = percent gain-loss model winner (based on having lowest AIC value)

- rmse = median root mean squared error across all winning models

The summary quality statistics file provides a nice overview of the target study design for each assay endpoint as well as summary statistics around active prevalence and hit-calling criteria.

For questions or concerns, please contact Monica Linnenbrink at: linnenbrink.monica@epa.gov.

R Script to produce September 2020 ToxCast Tox21 Data Release

```
# Connect to database using tcplConf before running
rm(list = ls())
library(tcpl)
library(data.table)
library(parallel)
library(stringr) #used to extract database info
## setup directory

post <- format(Sys.Date(), "_%y%m%d.csv")
mainDir <- paste0(toupper(format(Sys.time(), "%b%Y")),"_TOXCAST_EXTERNAL_RELEASE")
subDir <- paste0(toupper(str_extract(tcplConfList()$TCPL_DB,"invitrodb.*")),"_SUMMARY")
dir.create(file.path(getwd(),mainDir))
dir.create(file.path(getwd(),mainDir,subDir))


## Write the assay and chemical summary files
assay_info <- tcplQuery("SELECT *
                        FROM assay_source, assay, assay_component,
                        assay_component_endpoint
                        WHERE assay_source.asid=assay.asid AND assay.aid =
                        assay_component.aid AND
                        assay_component.acid =
                        assay_component_endpoint.acid;")
assay_info <- assay_info[, which(!duplicated(names(assay_info))),
  with =
    FALSE
]
tt_info <- tcplQuery("SELECT assay_component.acid, gene.* FROM
                     assay_component, technological_target, gene
                     WHERE assay_component.acid = technological_target.acid
                     AND technological_target.target_id = gene.gene_id;")
tt_info_agg <- tt_info[, lapply(.SD, paste, collapse = "|"), by = acid]
setnames(tt_info_agg, names(tt_info_agg)[-
  1], paste0("technological_target_", names(tt_info_agg)[-1]))
it_info <- tcplQuery("SELECT assay_component_endpoint.aeid, gene.* FROM
                     assay_component_endpoint, intended_target, gene
                     WHERE assay_component_endpoint.aeid =
                     intended_target.aeid
                     AND intended_target.target_id = gene.gene_id;")
it_info_agg <- it_info[, lapply(.SD, paste, collapse = "|"), by = aeid]
setnames(it_info_agg, names(it_info_agg)[-
  1], paste0("intended_target_", names(it_info_agg)[-1]))
citations <- tcplQuery("SELECT assay.aid, citations.* FROM assay,
```

```
                       assay_reference, citations
                       WHERE assay.aid = assay_reference.aid AND
                       assay_reference.citation_id = citations.citation_id;")
citations_agg <- citations[, lapply(.SD, paste, collapse = "|"), by = aid]
setnames(citations_agg, names(citations_agg)[-
  1], paste0("citations_", names(citations_agg)[-1]))
reagents <- tcplQuery("SELECT assay.aid, assay_reagent.* FROM assay,
                      assay_reagent
                      WHERE assay.aid = assay_reagent.aid;")
reagents_agg <- reagents[, lapply(.SD, paste, collapse = "|"), by = aid]
setnames(reagents_agg, names(reagents_agg)[-
  1], paste0("reagent_", names(reagents_agg)[-1]))
setkey(assay_info, "aeid")
setkey(it_info_agg, "aeid")
assay_info <- merge(x = assay_info, y = it_info_agg, all.x = TRUE)
setkey(assay_info, "acid")
setkey(tt_info_agg, "acid")
assay_info <- merge(x = assay_info, y = tt_info_agg, all.x = TRUE)
setkey(assay_info, "aid")
setkey(citations_agg, "aid")
assay_info <- merge(x = assay_info, y = citations_agg, all.x = TRUE)
setkey(assay_info, "aid")
setkey(reagents_agg, "aid")
assay_info <- merge(x = assay_info, y = reagents_agg, all.x = TRUE)

write.csv(assay_info,
  file.path(
    getwd(),
    mainDir,
    subDir,
    paste0("Assay_Summary", post)
  ),
  row.names = FALSE
)
### ASSAY QUALITY SUMMARY
dat <- tcplLoadData(5L)
dat <- tcplPrepOtpt(dat)
agg <- dat[, list(
  # baseline median absolute deviation (mad around the first 2 tested concentrations
  bmad = max(bmad, na.rm = TRUE),
  # nominal number of  concentrations tested for the assay endpoint
  nconc = as.double(median(nconc, na.rm = TRUE)),
  # global response cutoff established for   the assay (methods available within pipeline)
  coff = max(coff, na.rm = TRUE),
  # total number of samples tested in concentration response
  test = .N,
  acnt = as.double(lw(hitc == 1)), # active count
  apct = lw(hitc == 1) / .N, # active percentage
  icnt = as.double(lw(hitc == 0)), # inactive count
  ipct = lw(hitc == 0) / .N, # inactive percentage
  # could not model count (<=3 concentrations with viable data)
  ncnt = as.double(lw(hitc == -1)),
  npct = lw(hitc == -1) / .N, # could not model percentage
```

```
  # maximum response (median at any given concentration) across entire assay endpoint
  mmed = max(max_med, na.rm = TRUE),
  # nominal maximum tested concentration (target concentration)
  cmax = 10^median(logc_max, na.rm = TRUE),
  # nominal minimum tested concentration (target concentration)
  cmin = 10^median(logc_min, na.rm = TRUE),
  # maximum modeled response (top of curve) across entire assay endpoint
  mtop = max(modl_tp, na.rm = TRUE),
  # nominal number of replicates per sample (target number of replicates)
  nrep = as.double(median(nrep, na.rm = TRUE)),
  # nominal number of data points per sample
  npts = as.double(median(npts, na.rm = TRUE)),
  # percentage of sample-assayendpoints where the constant model won (may not all be 'actives')
  cnst = lw(modl == "cnst") / .N,
  # percentage of sample-assayendpoints where the hill model won (may not all be 'actives')
  hill = lw(modl == "hill") / .N,
  # percentage of sample-assayendpoints where the gain-loss model won (may not all be 'actives')
  gnls = lw(modl == "gnls") / .N,
  # median root mean squared error across all model winners for an assay endpoint
  rmse = median(modl_rmse, na.rm = TRUE)
), by = list(aeid, aenm, resp_unit)]
setkeyv(agg, "aenm")
agg2 <- dat[hitc >= 0,
  list(
    n = .N,
    acnt = sum(hitc)
  )
  ,
  by = list(aeid, chid)
]
agg3 <- agg2[n > 1, list(
  ocnc = lw(acnt == n | acnt == 0) / .N, # overall concordance among chemicalreplicates

  hcnc = lw(acnt == n) / lw(acnt > 0) # hit concordance among chemical-replicates
  # (may be samples from different sources)
), by = aeid]
setkey(agg3, "aeid")
setkey(agg, "aeid")
agg <- agg3[agg]
write.csv(agg, file.path(
  getwd(),
  mainDir,
  subDir,
  paste0("Assay_Quality_Summary_Stats", post)
),
row.names = FALSE
)
acids <- tcplLoadAcid()$acid
aq <- function(ac) {
  dat <- tcplPrepOtpt(tcplLoadData(1L, "acid", ac, type = "mc"))
  dat <- dat[wllq == 1]
  agg <- dat[, list(
    nmed = median(rval[wllt == "n"], na.rm = TRUE),
```

```
    nmad = mad(rval[wllt == "n"], na.rm = TRUE),
    pmed = median(rval[wllt == "p"], na.rm = TRUE),
    pmad = mad(rval[wllt == "p"], na.rm = TRUE),
    mmed = median(rval[wllt == "m"], na.rm = TRUE),
    mmad = mad(rval[wllt == "m"], na.rm = TRUE)
  ), by = list(acid, acnm, apid)]

  agg[, zprm.p := 1 - ((3 * (pmad + nmad)) / abs(pmed - nmed))]
  agg[, zprm.m := 1 - ((3 * (mmad + nmad)) / abs(mmed - nmed))]
  agg[, ssmd.p := (pmed - nmed) / sqrt(pmad^2 + nmad^2)]
  agg[, ssmd.m := (mmed - nmed) / sqrt(mmad^2 + nmad^2)]
  agg[, cv := nmad / nmed]
  agg[, sn.p := (pmed - nmed) / nmad]
  agg[, sn.m := (mmed - nmed) / nmad]
  agg[, sb.p := pmed / nmed]
  agg[, sb.m := mmed / nmed]

  agg[zprm.p < 0, zprm.p := 0]
  agg[zprm.m < 0, zprm.m := 0]

  acqu <- agg[, list(
    nmed = signif(median(nmed, na.rm = TRUE)),
    nmad = signif(median(nmad, na.rm = TRUE)),
    pmed = signif(median(pmed, na.rm = TRUE)),
    pmad = signif(median(pmad, na.rm = TRUE)),
    mmed = signif(median(mmed, na.rm = TRUE)),
    mmad = signif(median(mmad, na.rm = TRUE)),
    zprm.p = round(median(zprm.p, na.rm = TRUE), 2),
    zprm.m = round(median(zprm.m, na.rm = TRUE), 2),
    ssmd.p = round(median(ssmd.p, na.rm = TRUE), 0),
    ssmd.m = round(median(ssmd.m, na.rm = TRUE), 0),
    cv = round(median(cv, na.rm = TRUE), 2),
    sn.p = round(median(sn.p, na.rm = TRUE), 2),
    sn.m = round(median(sn.m, na.rm = TRUE), 2),
    sb.p = round(median(sb.p, na.rm = TRUE), 2),
    sb.m = round(median(sb.m, na.rm = TRUE), 2)
  ), by = list(acid, acnm)]

  return(acqu)
} # per acid
aqList <- mclapply(acids, aq, mc.cores = 1)
aqd <- rbindlist(aqList)
write.csv(aqd, file.path(
  getwd(),
  mainDir,
  subDir,
  paste0("Assay_Quality_Detailed_Stats", post)
),
row.names = FALSE
)
```