The New ToxCast Analysis

Dayne Lewis Filer

filer.dayne@epa.gov

(919) 541-2439

Outline

1. Challenges
2. Project goals
3. Detailed overview of pipeline/new levels
4. Downloading the data
5. R package demo

Challenges

- Heterogeneous data formats
- Heterogeneous experimental design
- How to appropriately make a hit-call
- Identifying (systematically) false-positives (FP) and false negatives (FN)

Project Goals

- Efficiency
- Usability
- Generalized (vendor-independent)
- Centralization
- Transparency
- Reproducibility

New Pipeline Overview

LVL 0

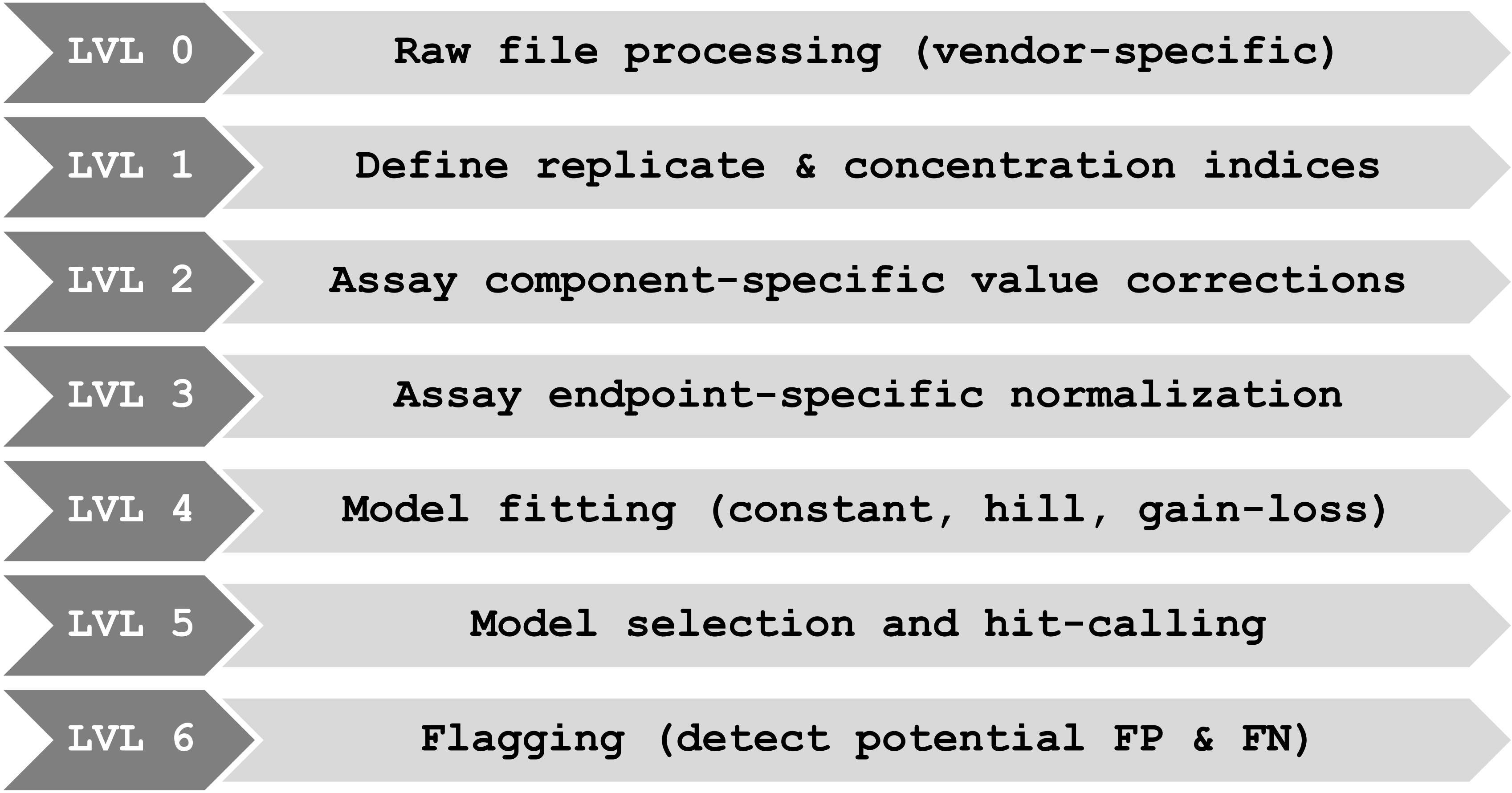
LVL 1

Intro information

- All data stored in **invitrodb**
- Pipeline interacts directly with db
- All processing done by assay component, or assay endpoint
- Independent of chemical information
- Does not store **assay names**

- **NOTE:** All table fields bolded, table names italicized

The new levels…



LVL 0 Raw file processing (vendor-specific)

LVL 1 Define replicate & concentration indices

LVL 2 Assay component-specific value corrections

LVL 3 Assay endpoint-specific normalization

LVL 4 Model fitting (constant, hill, gain-loss)

LVL 5 Model selection and hit-calling

LVL 6 Flagging (detect potential FP & FN)

The new levels…

LVL 0	Raw file processing (vendor-specific)
LVL 1	Define replicate & concentration indices
LVL 2	Assay component-specific value corrections
LVL 3	Assay endpoint-specific normalization
LVL 4	Model fitting (constant, hill, gain-loss)
LVL 5	Model selection and hit-calling
LVL 6	Flagging (detect potential FP & FN)

LVL 0

LVL 1

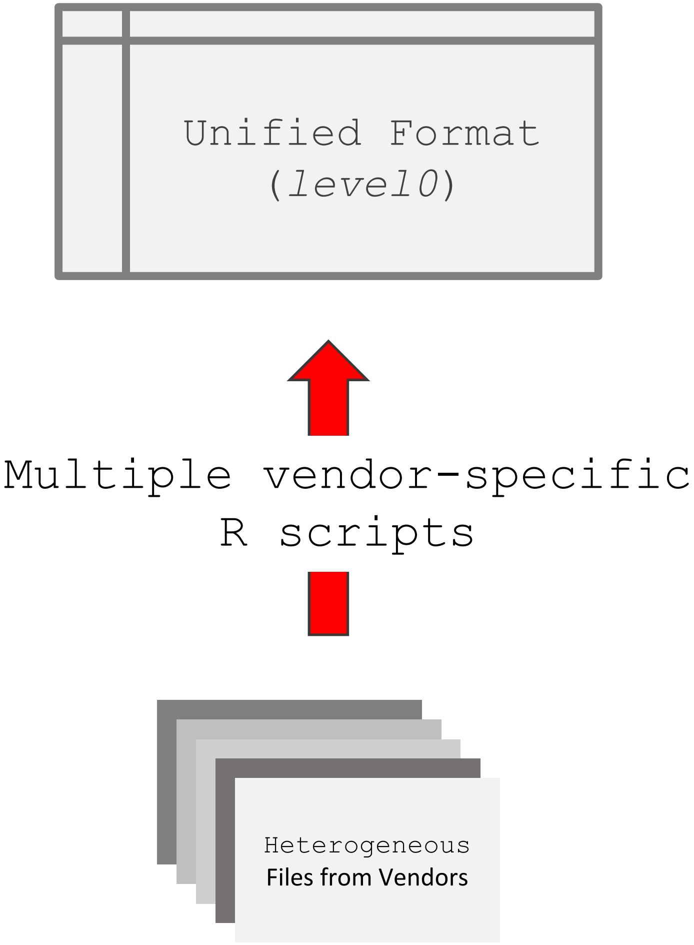
LVL 2

LVL 3

LVL 4

LVL 5

LVL 6



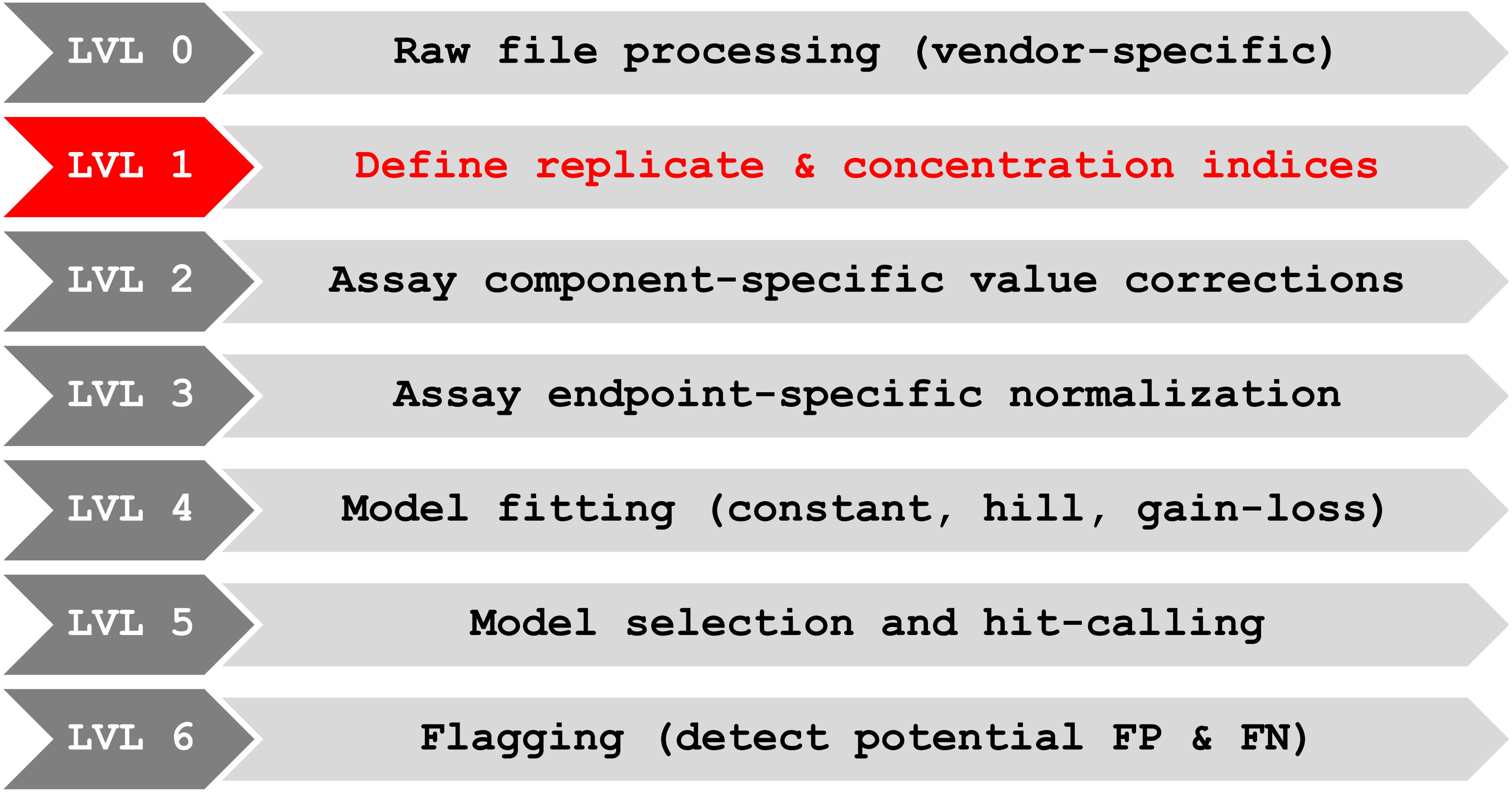
Unified Format (*level0*)

Multiple vendor-specific R scripts

Heterogeneous Files from Vendors

- ***ALL* vendor-specific processing**
- Each vendor script will act as "laboratory notebook" for that dataset
- Define well type (**wllt**) and well quality (**wllq**)
- Map the assay component source name (**acsn**) to the assay component id (**acid**)
- Store the raw value from the vendor (**rval**)

The new levels…



LVL 0 Raw file processing (vendor-specific)

LVL 1 Define replicate & concentration indices

LVL 2 Assay component-specific value corrections

LVL 3 Assay endpoint-specific normalization

LVL 4 Model fitting (constant, hill, gain-loss)

LVL 5 Model selection and hit-calling

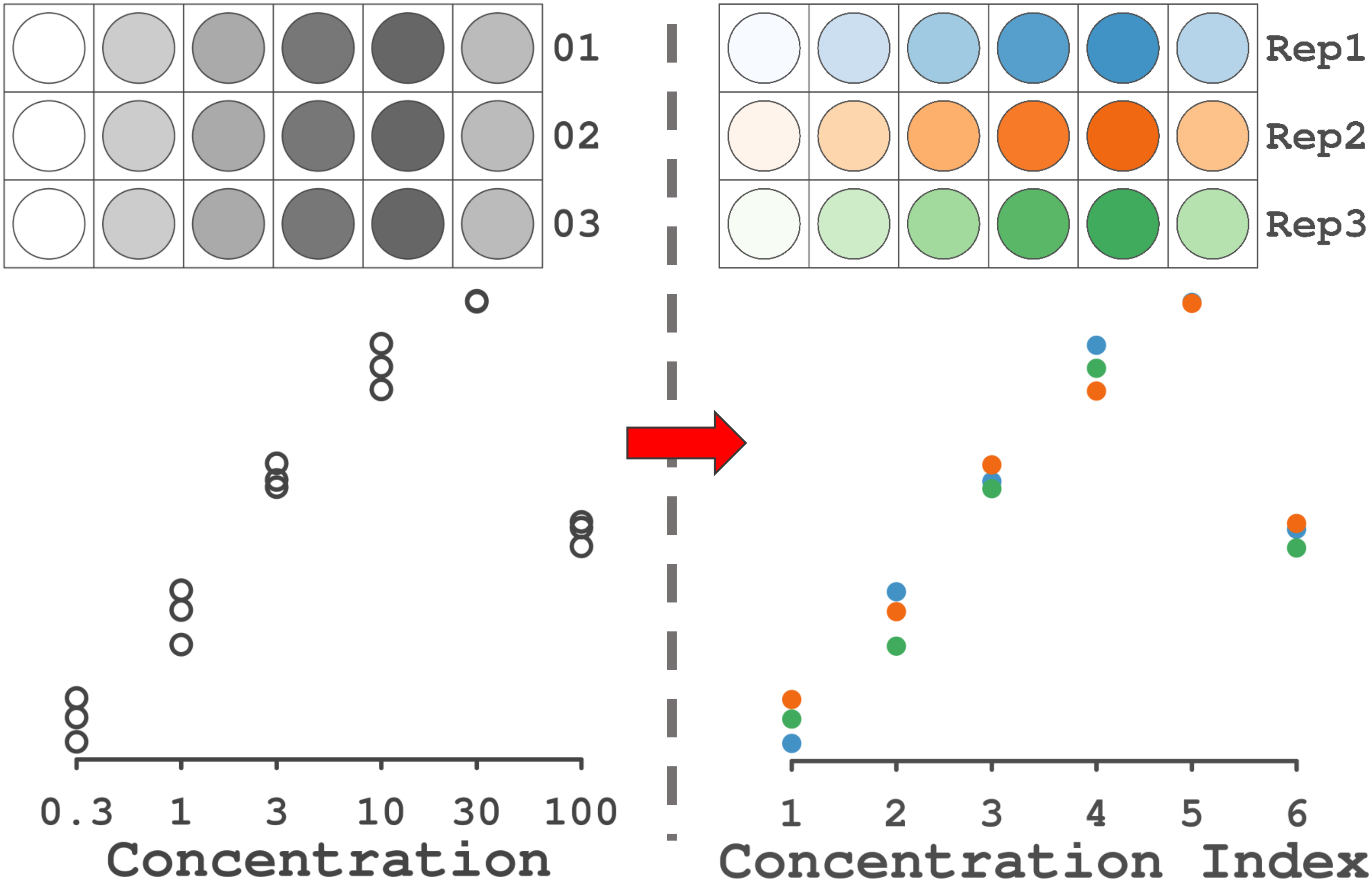
LVL 6 Flagging (detect potential FP & FN)

LVL 0
LVL 1
LVL 2
LVL 3
LVL 4
LVL 5
LVL 6

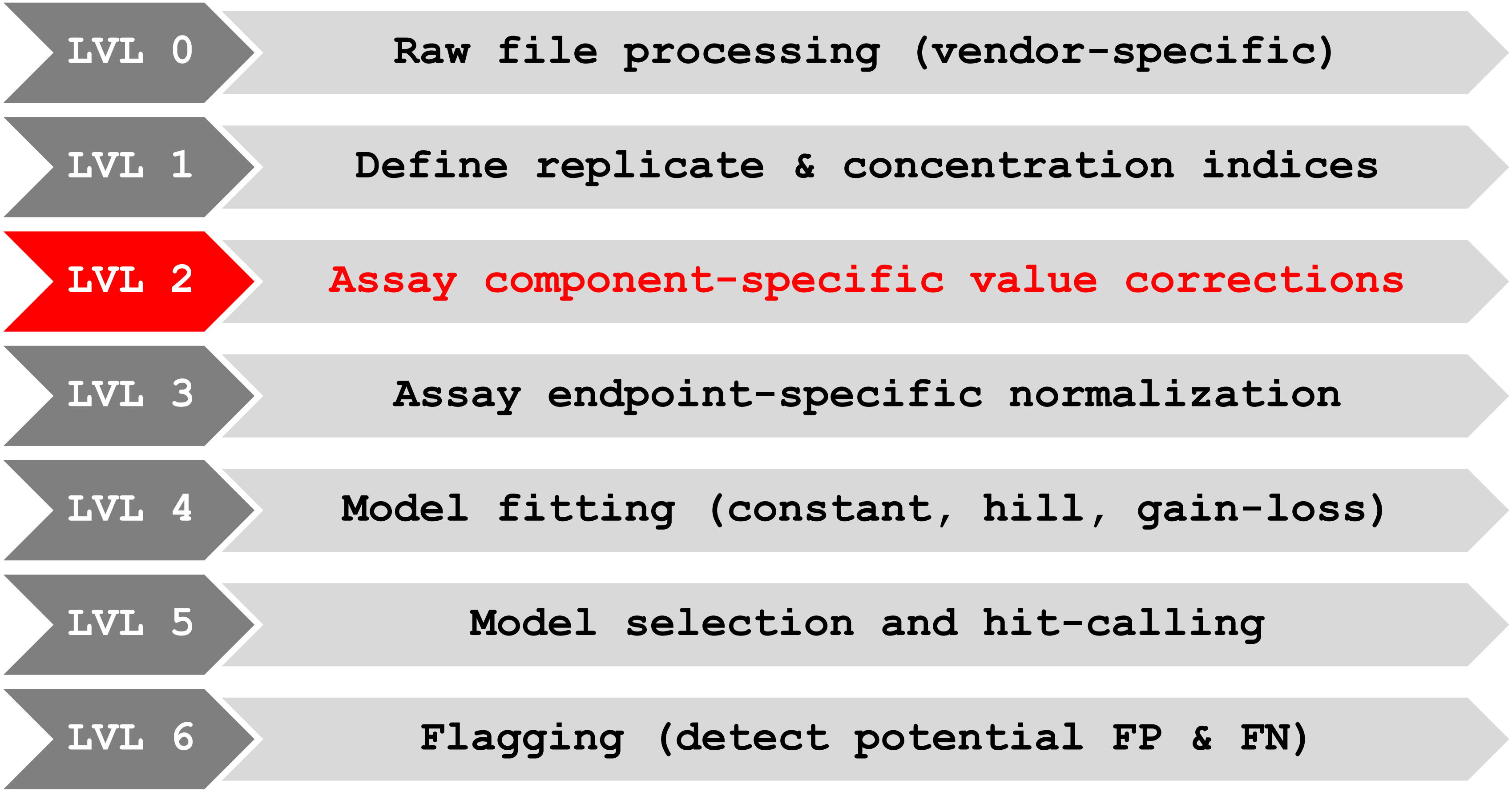
- Create concentration index (**cndx**) field
- Create replicate index (**repi**) field

```
# Order by the following columns
setkeyv(dat, c('acid', 'srcf', 'apid', 'coli', 'rowi', 'spid', 'conc'))

# Define rpid column for test compound wells
nconc <- dat[wllt == "t",
             list(n = lu(conc)),
             by = list(acid, apid, spid)][ , list(nconc = min(n)), by = acid]
dat[wllt == "t" & acid %in% nconc[nconc > 1, acid],
    rpid := paste(acid, spid, wllt, srcf, apid, cpid,
                  "rep1", conc, sep = "_")]
dat[wllt == "t" & acid %in% nconc[nconc == 1, acid],
    rpid := paste(acid, spid, wllt, srcf, cpid,
                  "rep1", conc, sep = "_")]

# Define concentration index
indexfunc <- function(x) as.integer(rank(unique(x))[match(x, unique(x))])
dat[ , cndx := indexfunc(conc), by = list(rpid)]
```

LVL 0 | LVL 1 | LVL 2 | LVL 3 | LVL 4 | LVL 5 | LVL 6



The new levels…



LVL 0 Raw file processing (vendor-specific)

LVL 1 Define replicate & concentration indices

LVL 2 Assay component-specific value corrections

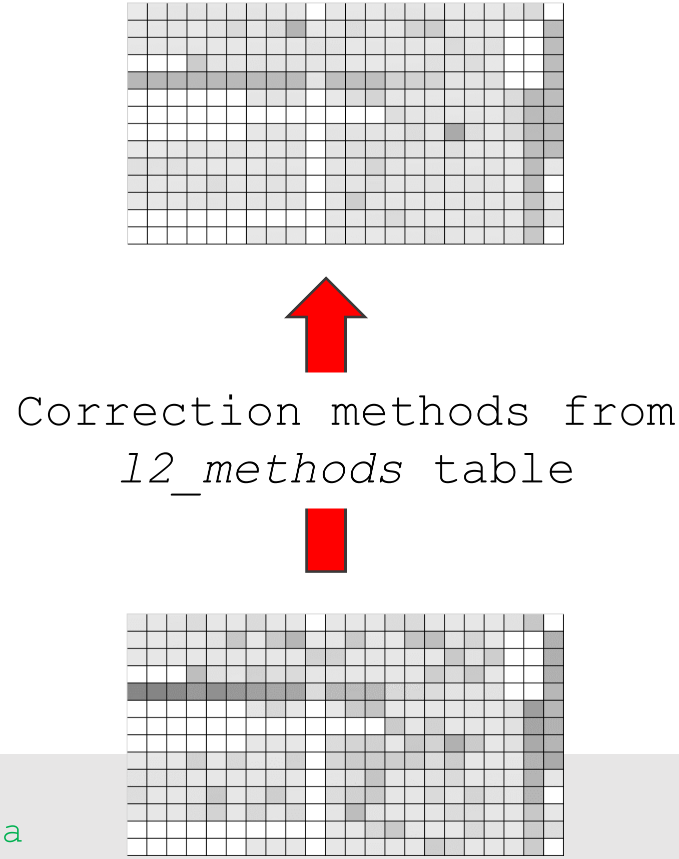
LVL 3 Assay endpoint-specific normalization

LVL 4 Model fitting (constant, hill, gain-loss)

LVL 5 Model selection and hit-calling

LVL 6 Flagging (detect potential FP & FN)

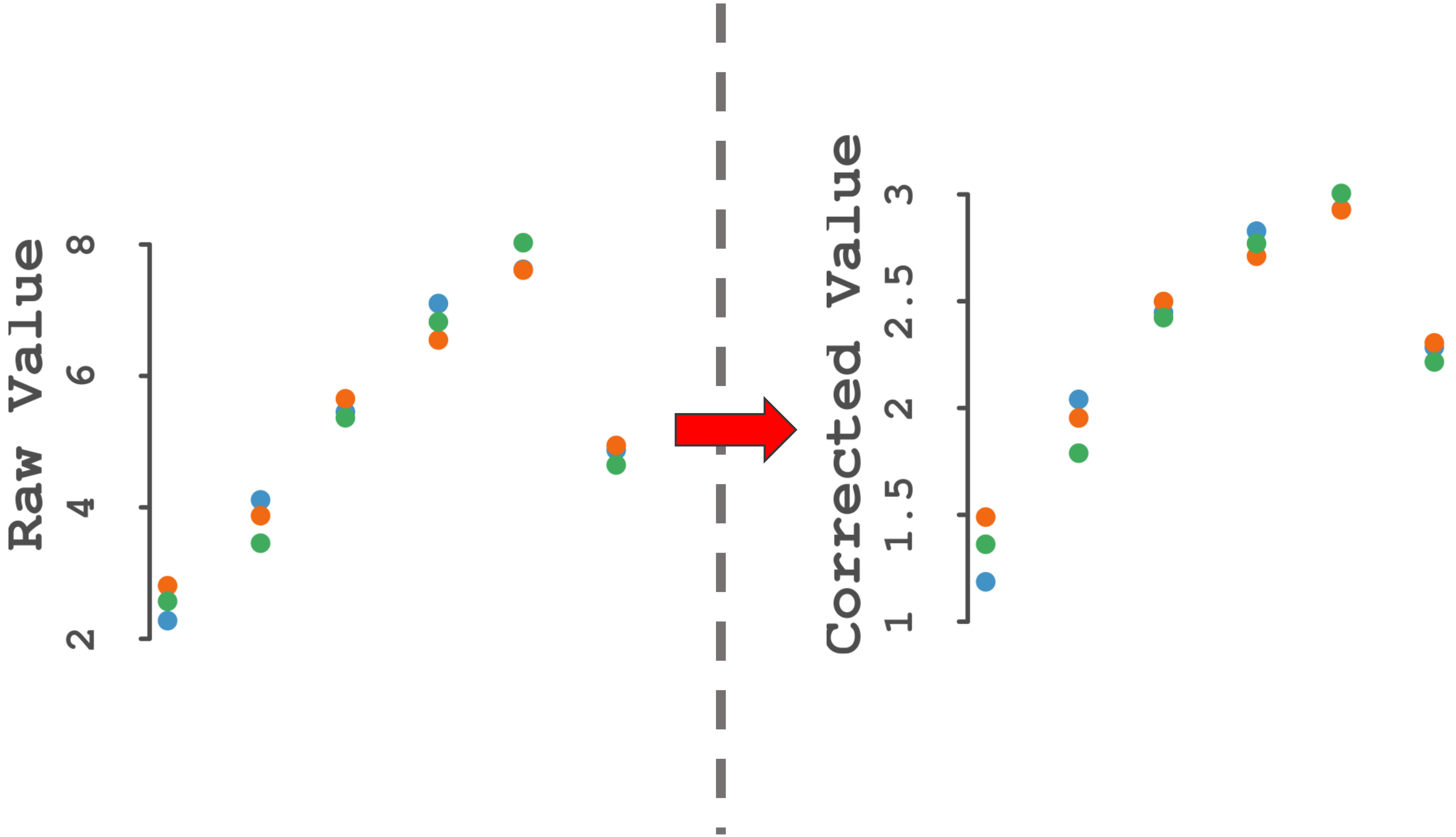
- Remove bad data (**wllq** = 0)
- Uses modular list of methods to generate the corrected value (**cval**) from **rval**
- Call methods by listing the method id (**l2_mthd_id**) and the execute order (**exec_ordr**) in *l2_acid*
- Methods are written as expressions to prevent making copies of the data



Correction methods from *l2_methods* table

```
log2 = function (acids) {
  ## This method takes the log base 2 of the data
  e1 <- bquote(dat[.(acids), cval := log2(cval)])
  list(e1)
}
```

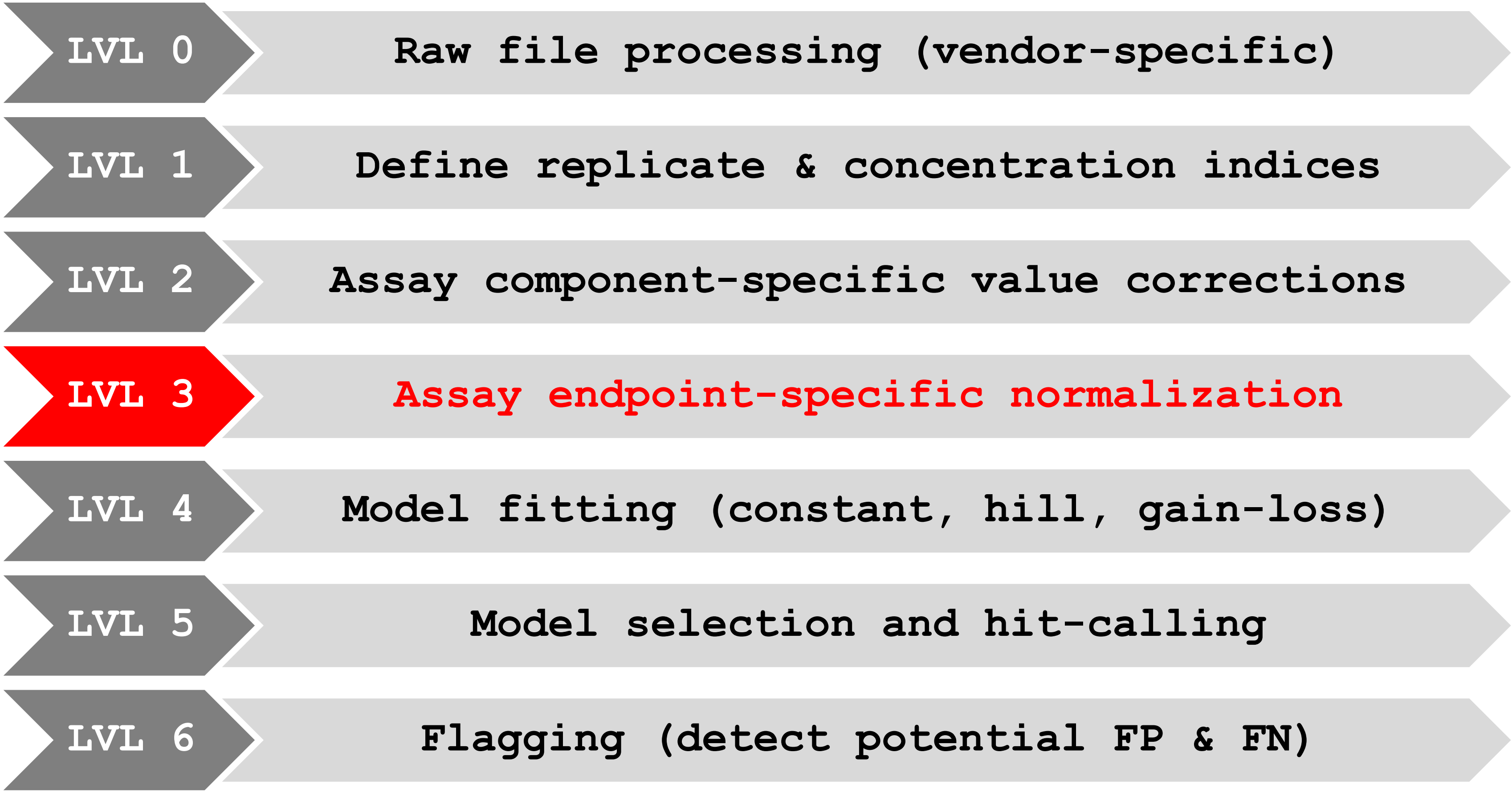
LVL 0 LVL 1 LVL 2 LVL 3 LVL 4 LVL 5 LVL 6



LVL 0 LVL 1 LVL 2 LVL 3 LVL 4 LVL 5 LVL 6

l2_mthd_id	l2_mthd	desc
1	none	apply no level 2 method
2	log2	log2 all raw data
3	rmneg	remove negative values prior to logging values
4	rmzero	remove 0 values prior to logging values
5	mult25	multiply values by 25
7	mult100	multiply values by 100
10	log10	log10 the raw data

The new levels…



LVL 0 Raw file processing (vendor-specific)

LVL 1 Define replicate & concentration indices

LVL 2 Assay component-specific value corrections

LVL 3 Assay endpoint-specific normalization

LVL 4 Model fitting (constant, hill, gain-loss)

LVL 5 Model selection and hit-calling

LVL 6 Flagging (detect potential FP & FN)

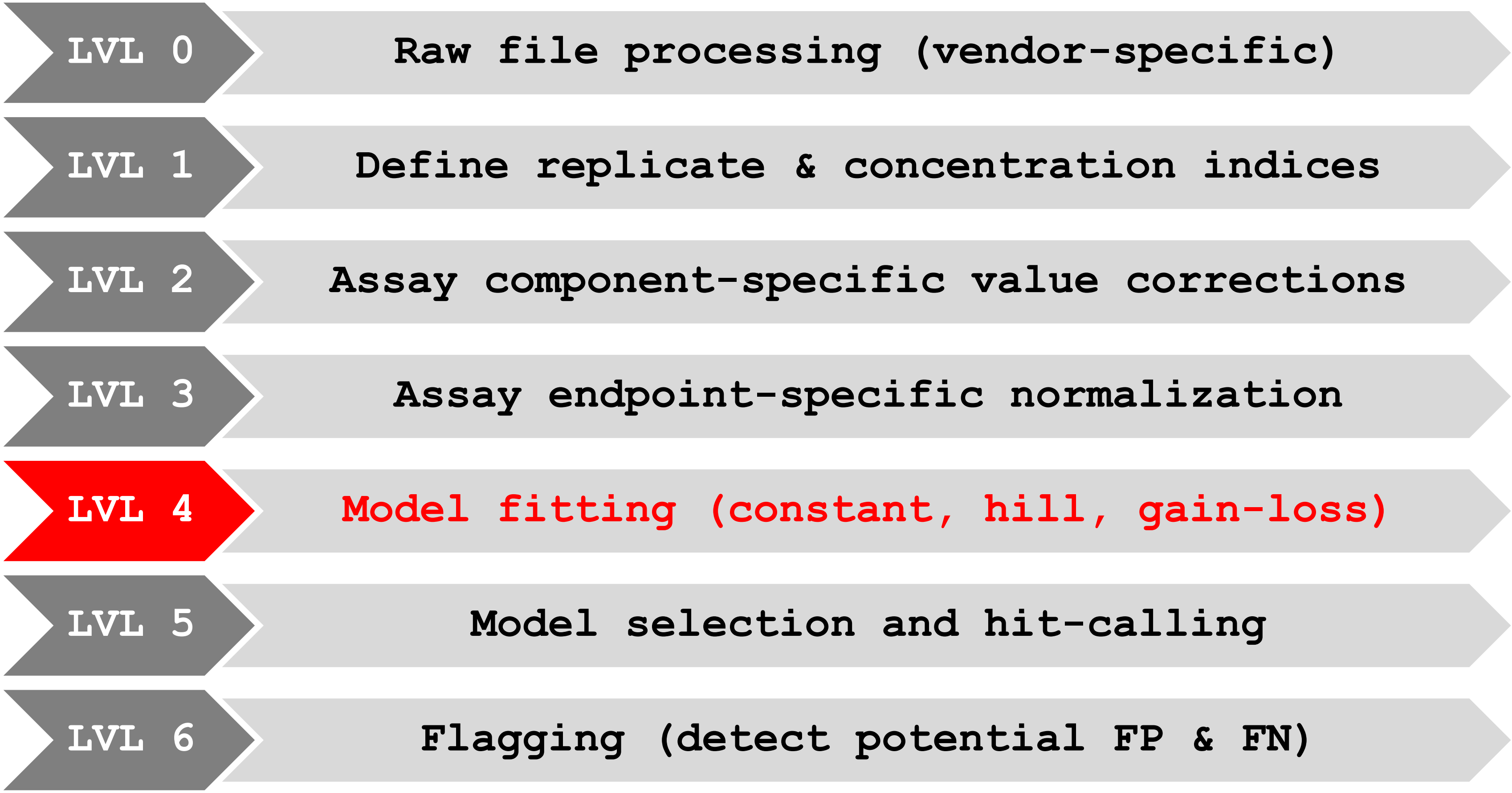
- Similar to level 2, except based on assay endpoint
- Create response values (**resp**) using **cval**
- Define baseline value (**bval**), pos ctrl value (**pval**) if necessary, and log$_{10}$ concentration (**logc**)
- Define methods for assay endpoints in *l3_aeid*
- ***ALL* fold-change values must be logged**

```
bval.apid.nwlls.med = function (aeids) {
  ## Take the median of all the well type "n" values, by apid
  e1 <- bquote(dat[J(.(aeids)),
                   bval := median(cval[wllt == "n"], na.rm = TRUE),
                   by = list(aeid, apid)])
  list(e1)
}
resp.fc = function (aeids) {
  ## Calculate the response as a fold change over baseline
  e1 <- bquote(dat[J(.(aeids)), resp := cval/bval])
  list(e1)
}
```

LVL 0 LVL 1 LVL 2 LVL 3 LVL 4 LVL 5 LVL 6

l3_mthd_id	l3_mthd	desc
1	none	apply no level 2 method
2	bval.apid.1owconc.med	plate-wise baseline based on low conc median value
3	pval.apid.medpcbyconc.max	plate-wise median response of positive control (max)
4	pval.apid.medpcbyconc.min	plate-wise median response of positive control (min)
5	resp.pc	response percent activity
6	resp.multneg1	multiply the response by -1
7	resp.log2	take the log base 2 of the response
8	resp.mult25	multiply the response by 25
9	resp.fc	calculate response as fold-change
11	bval.apid.nwlls.med	plate-wise baseline based on neutral ctrl median value
12	bval.spid.lowconc.med	sample-wise baseline based on low conc median value
13	pval.apid.pwlls.med	plate-wise meidan based on positive control, single dose
14	pval.apid.mwlls.med	plate-wise meidan based on negative control, single dose
15	pval.apid.medncbyconc.min	plate-wise meidan based on negative control, (min)
16	bval.apid.twlls.med	Take the median cval of the t wells, by apid
17	bval.apid.nwllslowconc.med	Take the median cval of the n wells and the first two concentrations, by apid
18	resp.shiftneg.3bmad	Make values below baseline zero.
19	resp.blineshift.3bmad.repi	Do baseline correction by repi, with a window of 3*bmad
20	resp.blineshift.50.repi	Do baseline correction by repi, with a window of 50
21	resp.blineshift.50.spid	Do baseline correction by repi, with a window of 50
23	resp.blineshift.3bmad.spid	Do baseline correction by repi, with a window of 3*bmad
24	bval.apid.tn.med	Take the median cval of the t and n wells, by apid
25	pval.apid.pmv.min	Calculate the median p, m, and v values by concentration, then take the minimum by apid.
26	pval.apid.pmv.max	Calculate the median p, m, and v values by concentration, then take the maximum by apid.
27	pval.apid.f.max	Calculate the median of f values by concentration, then take the maximum by apid
28	pval.apid.f.min	Calculate the median of f values by concentration, then take the minimum by apid
29	pval.apid.p.min	Calculate the median of p values by concentration, then take the minimum by apid
30	pval.apid.p.max	Calculate the median of p values by concentration, then take the maximum by apid
31	pval.apid.v.min	Calculate the median of v values by concentration, then take the minimum by apid
32	pval.zero	Set pval to 0.
33	resp.shiftneg.6bmad	Shift response values falling below -6 * bmad to 0.
34	resp.shiftneg.10bmad	Shift response values falling below -10 * bmad to 0.

The new levels…



LVL 0 Raw file processing (vendor-specific)

LVL 1 Define replicate & concentration indices

LVL 2 Assay component-specific value corrections

LVL 3 Assay endpoint-specific normalization

LVL 4 Model fitting (constant, hill, gain-loss)

LVL 5 Model selection and hit-calling

LVL 6 Flagging (detect potential FP & FN)

- Fit three models, by sample id (**spid**)
  - **cnst** – constant model (slope and intercept equal 0)
  - **hill** – three parameter hill model with bottom equal to 0
  - **gnls** – gain-loss model (product of two three-parameter hill models with bottoms equal to 0)
- Use maximal likelihood to model the data, each model has an additional error term (**er**)
- Calculate the baseline MAD (**bmad**)
- Calculate model and data summary values, such as AIC (**aic**), RMSE (**rmse**), parameter sds (**sd**), and the max median response by concentration (**max_med**)

Let $t(z, \nu)$ be the Studen's t-distribution with ν degrees of freedom and y_i be the log response at the i^{th} observation. We calculate z_i as:

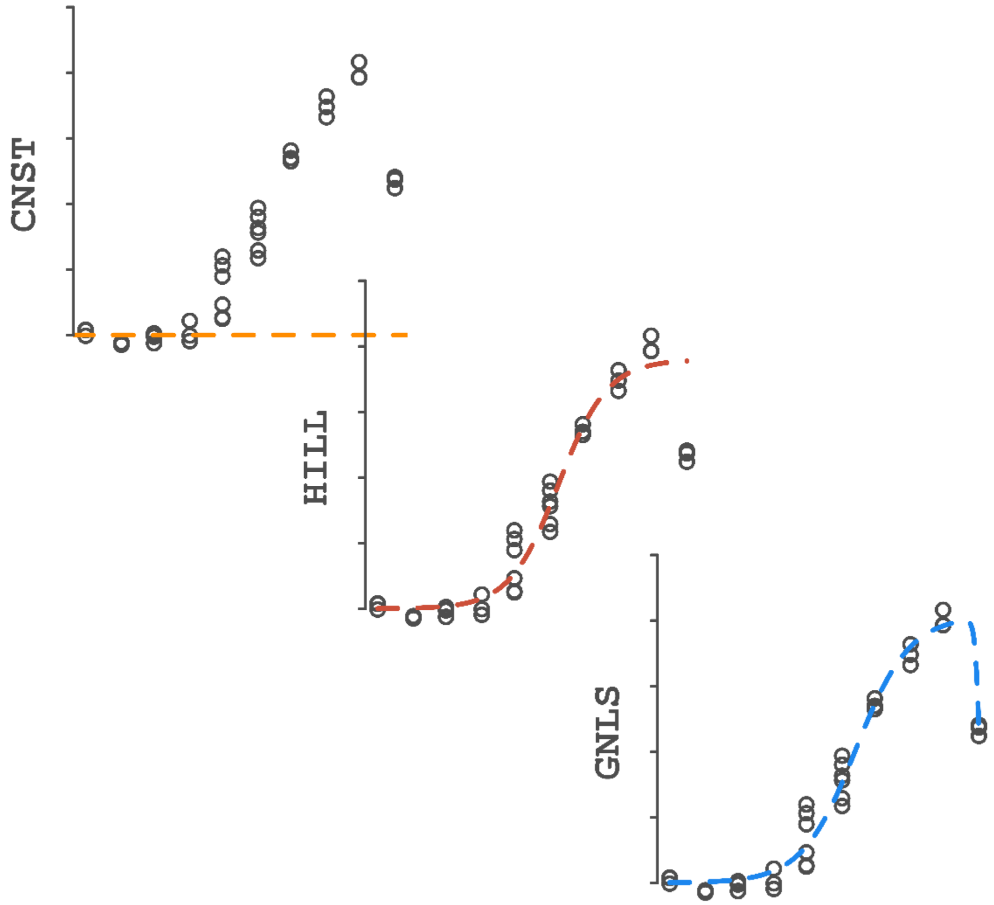
$$z_i = \frac{y_i - \mu_i}{e^{\sigma}}$$

Where σ is the scale term. Then the log-likelyhood is:

$$\sum_{i=1}^{n} \ln\left(t(z_i, 4)\right) - \sigma$$

Where n is the number of observations.

LVL 0 LVL 1 LVL 2 LVL 3 LVL 4 LVL 5 LVL 6



$$\mu_i = 0$$

$$\mu_i = \frac{1}{1 + 10^{(ga - x_i)gw}}$$

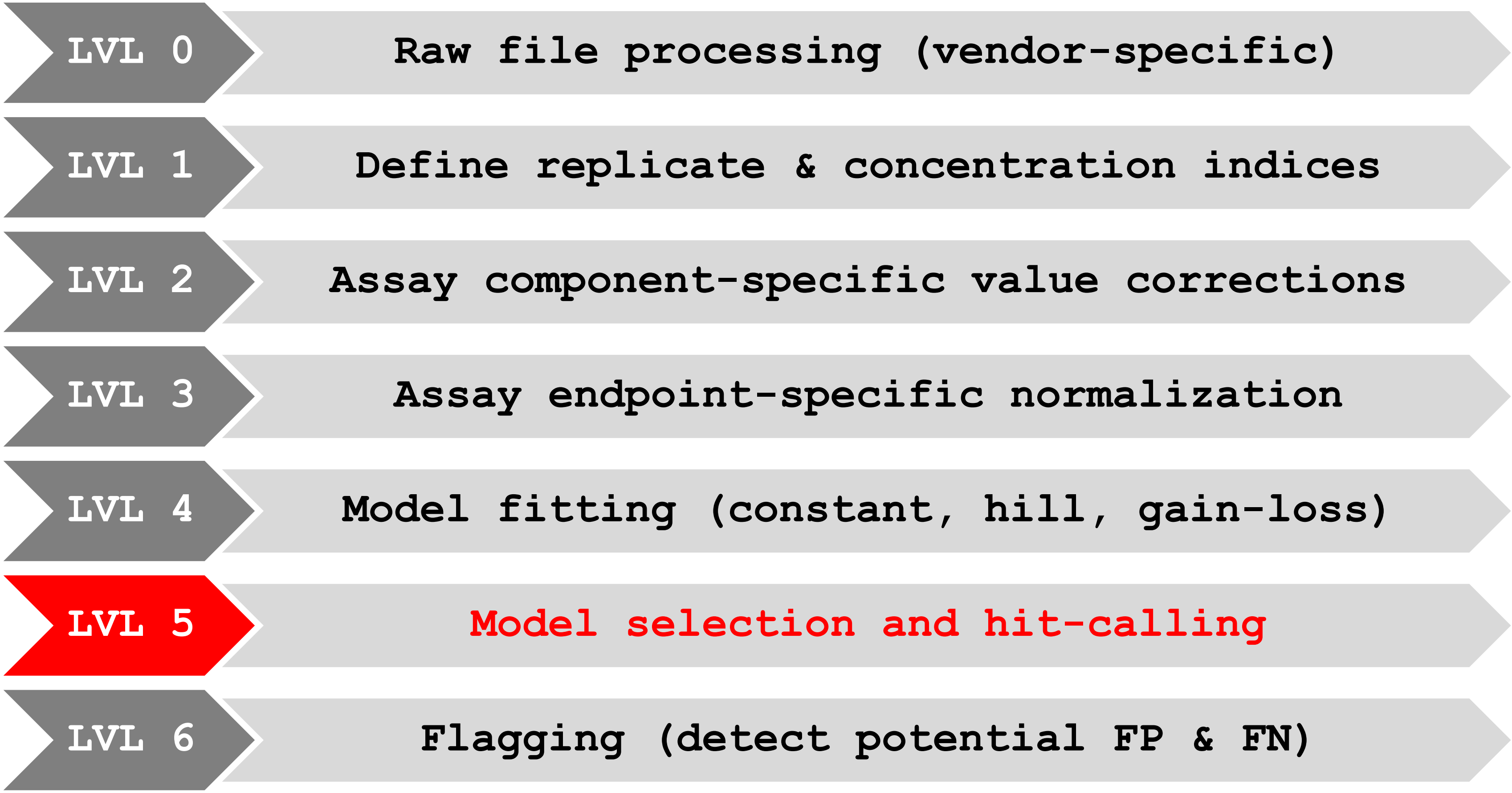
$$g_i = \frac{1}{1 + 10^{(ga - x_i)gw}}$$

$$l_i = \frac{1}{1 + 10^{(x_i - la)lw}}$$

$$\mu_i = tp * g_i * l_i$$

Where u_i and x_i are the modeled response and log concentration at the i^{th}observation, respectively, ga is the gain log(AC50), gw is the gain Hill coefficient, la is the loss log(AC50), and la is the loss Hill coefficent.

The new levels…



LVL 0 Raw file processing (vendor-specific)

LVL 1 Define replicate & concentration indices

LVL 2 Assay component-specific value corrections

LVL 3 Assay endpoint-specific normalization

LVL 4 Model fitting (constant, hill, gain-loss)

LVL 5 Model selection and hit-calling

LVL 6 Flagging (detect potential FP & FN)

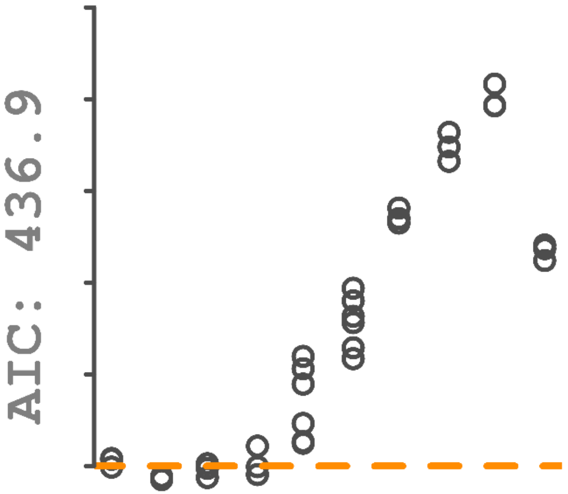
LVL 0 LVL 1 LVL 2 LVL 3 LVL 4 LVL 5 LVL 6

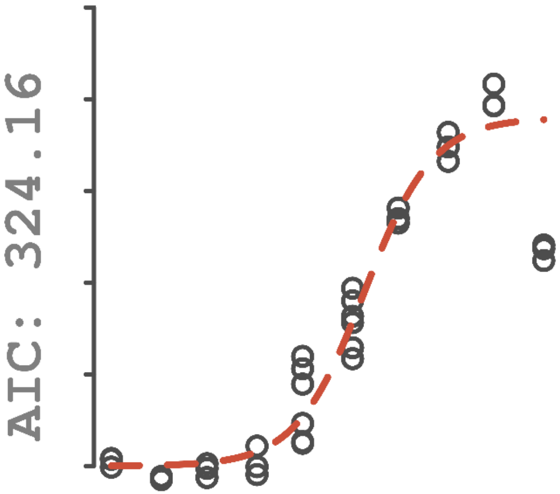
- Select the winning model(**modl**), bin the fits (**fitc**), and make a hit-call (**hitc**)
- Define activity cutoff (**coff**)
  - Always at least 3*bmad or 20% change
  - Can be increased with an additional cutoff method from *l5_methods*

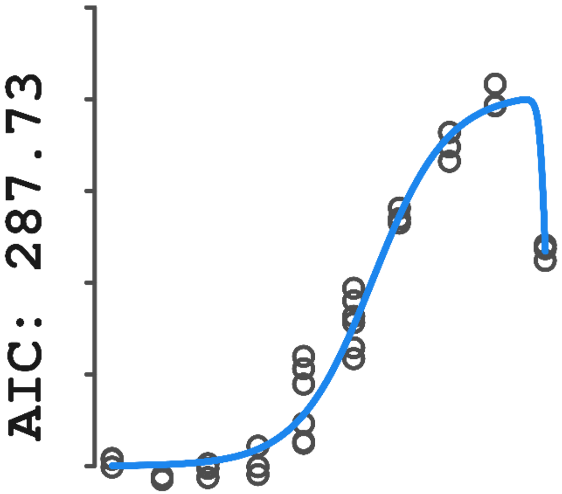
```
## Determine winning model
dat[ , maic := pmin(cnst_aic, hill_aic, gnls_aic, na.rm = TRUE)]
# Order matters here, because in the case of a tie the simpler model will
# overwrite the more complex model as the winner.
dat[gnls_aic == maic, modl := "gnls"]
dat[hill_aic == maic, modl := "hill"]
dat[cnst_aic == maic, modl := "cnst"]

## Make the hitcall
dat[ , hitc := FALSE]
dat[modl == "hill" & hill_tp >= coff & max_med >= coff, hitc := TRUE]
dat[modl == "gnls" & gnls_tp >= coff & max_med >= coff, hitc := TRUE]
```

LVL 0 LVL 1 LVL 2 LVL 3 LVL 4 LVL 5 LVL 6

Select the winning model (lowest AIC):

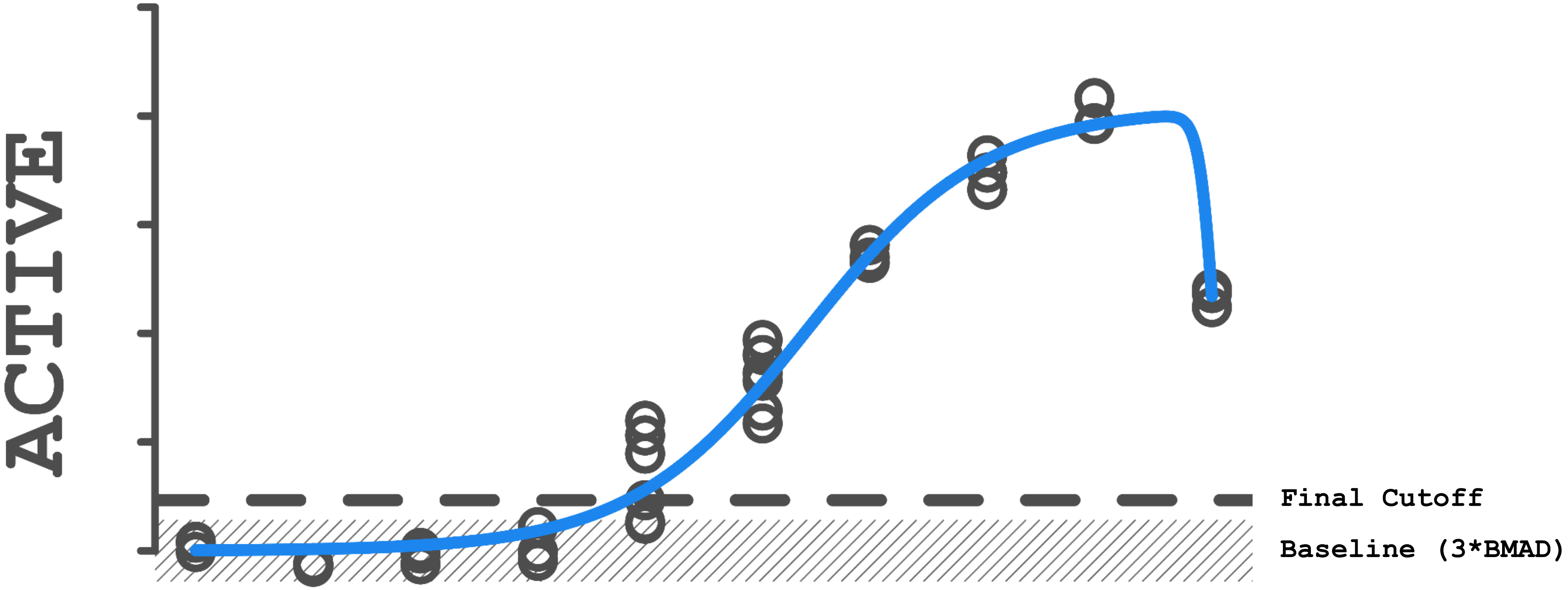




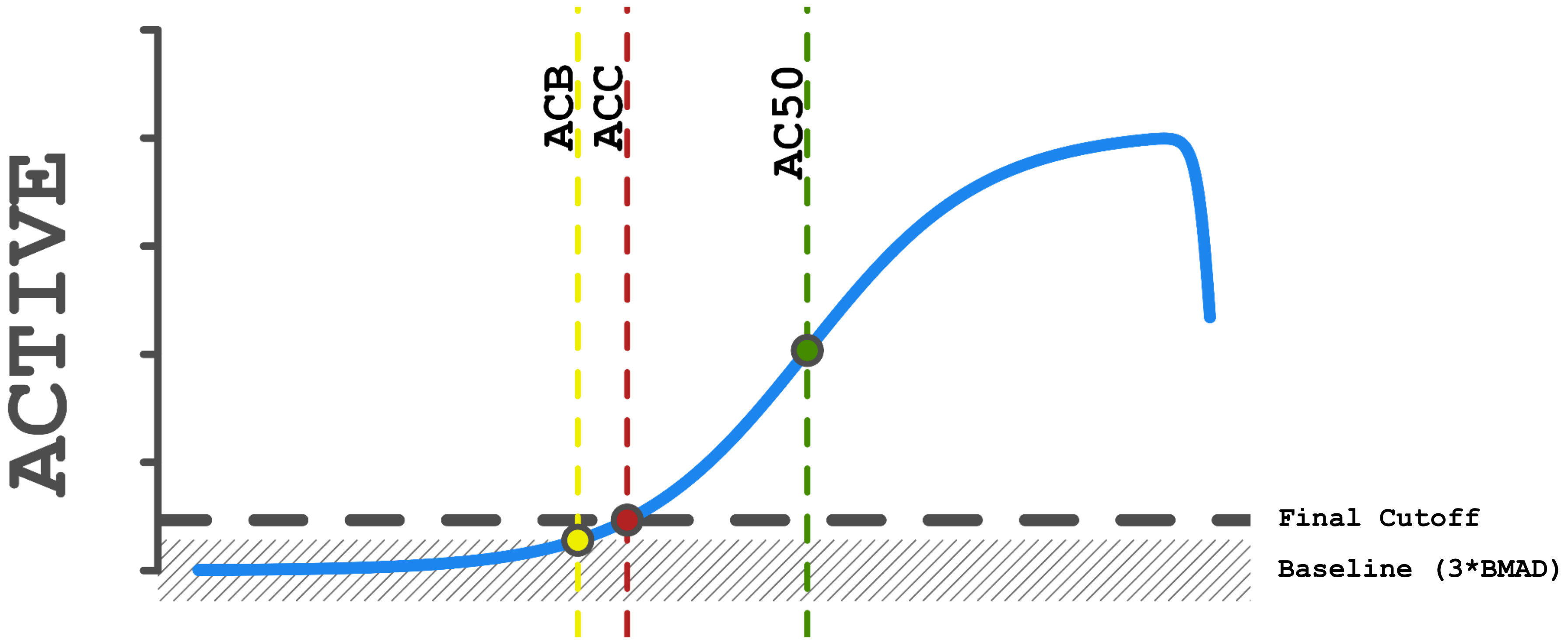


LVL 0 LVL 1 LVL 2 LVL 3 LVL 4 **LVL 5** LVL 6

Make activity call (hit-call):

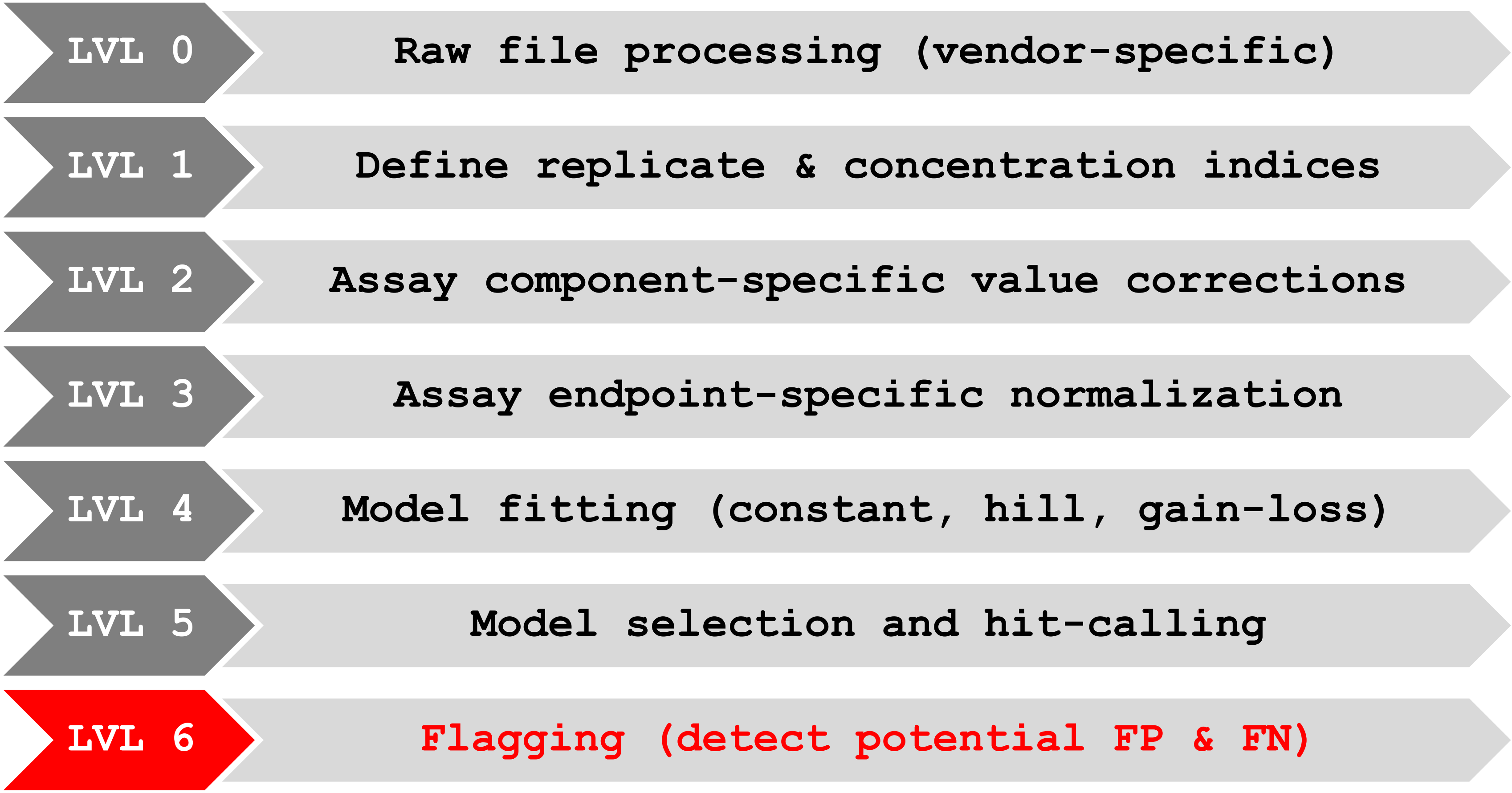


Point of departure estimates:



l5_mthd_id	l5_mthd	desc
1	none	Add no additional cutoff. Will default to 3*bmad
2	bmad5	Use 5*bmad
3	bmad10	Use 10*bmad
4	bmad6	Use 6 * bmad

The new levels…



LVL 0	Raw file processing (vendor-specific)
LVL 1	Define replicate & concentration indices
LVL 2	Assay component-specific value corrections
LVL 3	Assay endpoint-specific normalization
LVL 4	Model fitting (constant, hill, gain-loss)
LVL 5	Model selection and hit-calling
LVL 6	Flagging (detect potential FP & FN)

LVL 0 LVL 1 LVL 2 LVL 3 LVL 4 LVL 5 LVL 6

- Flag fits as potential false positive/negative
- Call methods from *l6_methods*
  - Additional field, **nddr**, indicates whether the method needs the dose-response data loaded
- *level6* is long, with one fit-flag combination per line. Each fit can have multiple flags. Fits without any flags are not listed in the table.

```
border.hit = function (mthd) {
  flag <- "Borderline active"
  out  <- c("l5id", "l4id", "aeid", "l6_mthd_id",
            "flag", "fval", "fval_unit")
  init <- bquote(list(.(mthd), .(flag), NA_real_, NA_character_))
  e1 <- bquote(ft[ , .(out[4:7]) := .(init), with = FALSE])
  e2 <- bquote(ft[ ,
                  test := hitc == 1L & (actp < 0.9 | modl_tp <= 1.2*coff)])
  e3 <- bquote(f[[.(mthd)]] <- ft[which(test), .SD, .SDcols = .(out)])
  cr <- c("l6_mthd_id", "flag", "test")
  e4 <- bquote(ft[ , .(cr) := NULL, with = FALSE])
  list(e1, e2, e3, e4)
}
```

LVL 0 LVL 1 LVL 2 LVL 3 LVL 4 LVL 5 LVL 6

l6_mthd_id	l6_mthd	desc	nddr
1	row.dev.up	Look for row-wise plate effects, increase	1
2	row.dev.dn	Look for row-wise plate effects, decrease	1
3	col.dev.dn	Look for column-wise plate effects, decrease	1
4	col.dev.up	Look for column-wise plate effects, increase	1
5	plate.flare	Look for plate flare effects	1
6	singlept.high.hit	Look for single point hits with activity only at the highest conc tested	0
7	singlept.mid.hit	Look for signle point hits with activity not at highest conc tested	0
8	multipoint.neg	Look for inactives with multiple medians above baseline	0
9	pintool	Look for pintool carryover issues	0
10	noise	Look for noisy curves, relative to the assay	0
11	border.hit	Look for actives with borderline activity	0
12	border.miss	Look for inactives with borderline activity	0
13	plate.interlace	Look for interlaced chemical-plate effects	1
14	rep.mismatch	Look for mismatched techinal replicates	1
15	gnls.lowconc	Look for low concentration gnls winners	0
16	overfit.hit	Flag hit-calls that would get changed after doing the small N correction to the aic values.	0
17	efficacy.50	Flag hit-calls with efficacy values less than 50% -- intended for biochemical assays.	0

Project Goals

- ✓ Efficiency – ~120x faster
- ✓ Usability – completely functionalized
- ✓ Generalized – vendor-independent
- ✓ Centralization – all in relational database
- ✓ Transparency – well commented R package
- ✓ Reproducibility – processing based on database parameters

Downloading the Data

LVL 0

LVL 1

What Will Be Available?

- **R package (tcpl)**
- **MySQL database dump -- including all ToxCast PhI and PhII data from level 0 to level 6**
- Summary matrix files
- Level 5 table
- Level 6 table
- http://epa.gov/ncct/toxcast/data.html

Summary Matrices

- All parameters for the winning model
  - modl_ga – gain log(AC50)
  - modl_tp – top
  - modl_gw – gain Hill coefficient
  - …
  - modl_acc – activity concentration at cutoff
  - modl_acb – activity concentration at baseline
  - modl_ac10 – activity concentration at 10%
  - **DOES NOT INDICATE HIT-CALL**
- Hit-call matrix
- Tested/not-tested matrix
- **CHEMICALS NOT TESTED IN DOSE-RESPONSE WILL BE NA IN ALL FILES EXCEPT THE TESTED/NOT TESTED MATRIX**

Acknowledgements

- **Parth Kothiya**
- Matt Martin
- Richard Judson
- Woody Setzer
- Jeff Edwards
- John Wambaugh
- Jimmy Phuong

Demo

LVL 0

LVL 1

Questions?

LVL 0

LVL 1

Field/Variable Index

LVL 0

LVL 1

LVL 0 LVL 1 LVL 2 LVL 3 LVL 4 LVL 5 LVL 6

- **l0id** = level 0 id
- **acid** = assay component id
- **spid** = sample id
- **cpid** = chemical plate id
- **apid** = assay plate id
- **rowi** = row index
- **coli** = column index
- **wllq** = well quality (bool)
- **wllt** = well type
  - ***t*** = test compound
  - ***c*** = pos ctrl (dose-resp)
  - ***p*** = pos ctrl (1 conc)
  - ***n*** = neutral control
  - ***m*** = neg ctrl (1 conc)
  - ***o*** = neg ctrl (dose-resp)
  - ***b*** = blank
  - ***v*** = viab ctrl (1 conc)
- **conc** = concentration
- **rval** = raw value
- **srcf** = source file from vendor

LVL 0 LVL 1 LVL 2 LVL 3 LVL 4 LVL 5 LVL 6

- **l1id** = level 1 id
- **l0id** = level 0 id
- **acid** = assay component id
- **repi** = replicate index
- **cndx** = concentration index

LVL 0 LVL 1 LVL 2 LVL 3 LVL 4 LVL 5 LVL 6

- **l2id** = level 2 id
- **l1id** = level 1 id
- **l0id** = level 0 id
- **acid** = assay component id
- **cval** = corrected value

LVL 0 LVL 1 LVL 2 LVL 3 LVL 4 LVL 5 LVL 6

- **l3id** = level 3 id
- **l2id** = level 2 id
- **l1id** = level 1 id
- **l0id** = level 0 id
- **acid** = assay component id
- **aeid** = assay endpoint id
- **bval** = baseline value
- **pval** = positive control value
- **logc** = $\log_{10}$ concentration
- **resp** = normalized response value

LVL 0
LVL 1
LVL 2
LVL 3
LVL 4
LVL 5
LVL 6

- **l4id** = level 3 id
- **aeid** = assay endpoint id
- **spid** = sample id
- **bmad** = baseline median absolute deviation
- **resp_max** = max resp value
- **resp_min** = min resp value
- **max_mean** = max mean value, by concentration
- **max_mean_conc** = concentration of the max_mean
- **max_med** = max median value, by concentration
- **max_med_conc** = concentration of the max_med
- **logc_max** = max $\log_{10}$ concentration
- **logc_min** = min $\log_{10}$ concentration
- **nconc** = number of concentrations
- **npts** = number of values
- **nrep** = number of technical replicates
- **nmed_gtbl** = number of median values > 3*bmad

LVL 0 | LVL 1 | LVL 2 | LVL 3 | **LVL 4** | LVL 5 | LVL 6

- **cnst** = TRUE/FALSE did cnst fit
- **hill** = TRUE/FALSE did hill fit
- **hcov** = TRUE/FALSE did hill hessian matrix invert
- **gnls** = TRUE/FALSE did gnls fit
- **gcov** = TRUE/FALSE did gnls hessian matrix invert
- **er** = model error term
- **tp** = model top
- **ga** = model gain AC50
- **gw** = model gain hill coefficient (slope)
- **la** = model loss AC50
- **lw** = model loss hill coefficient (slope)
- **sd** = model parameter standard deviation
- **aic** = model Akaike information criterion
- **prob** = model probability (derived from all aics)
- **rmse** = model root mean square error

NOTE: The fields for the model values are concatenated, for example **cnst_er**, **gnls_lw_sd**, **hill_prob**

LVL 0
LVL 1
LVL 2
LVL 3
LVL 4
LVL 5
LVL 6

- **l5id** = level 5 id
- **l4id** = level 3 id
- **aeid** = assay endpoint id
- **modl** = winning model
- **fitc** = fit category
- **coff** = final cutoff
- **actp** = activity probability
- **modl_er** = error of winning model
- **modl_tp** = top of winning model
- **modl_ga** = gain AC50 of winning model
- **modl_gw** = gain hill coefficient of winning model
- **modl_la** = loss AC50 of winning model
- **modl_lw** = loss hill coefficient of winning model
- **modl_prob** = probability of winning model
- **modl_rmse** = root mean square error of winning model
- **modl_acc** = activity concentration at cutoff
- **modl_acb** = activity concentration at baseline

LVL 0 LVL 1 LVL 2 LVL 3 LVL 4 LVL 5 LVL 6

- **l5id** = level 5 id
- **l4id** = level 3 id
- **aeid** = assay endpoint id
- **l6_mthd_id** = error of winning model
- **flag** = flag text
- **fval** = flag value
- **fval_unit** = flag value unit