field-vs-lab-doses-r-2

Jake Martin

17 April 2025

Table of Contents

[📕 README 1](#_Toc195811888)

[📧 Contact 1](#_Toc195811889)

[📑 Sharing/accessing and citing 1](#_Toc195811890)

[⚙️ Set-up and packages 1](#_Toc195811891)

[🔧 Custom functions 1](#_Toc195811892)

[📂 Directories 1](#_Toc195811893)

[Input 1](#_Toc195811894)

[Output 1](#_Toc195811895)

[💿 Databases 1](#_Toc195811896)

[All databases 1](#_Toc195811897)

[🧮 Analysis 1](#_Toc195811898)

[Samples/data 1](#_Toc195811899)

[Table S1 (Table 1) 1](#_Toc195811900)

[Cross-over 1](#_Toc195811901)

[Concentrations 1](#_Toc195811902)

[Modeling 1](#_Toc195811903)

[🌊 Surface water 1](#_Toc195811904)

[🚽 Waste water 1](#_Toc195811905)

[Concentration fold differences 1](#_Toc195811906)

[Results 1](#_Toc195811907)

[Surface water 1](#_Toc195811908)

[Waste water 1](#_Toc195811909)

[Occurance 1](#_Toc195811910)

[Occurance summary data 1](#_Toc195811911)

[Table S4 1](#_Toc195811912)

[Fig S7 1](#_Toc195811913)

[Fig S8 1](#_Toc195811914)

[Temporal trends data 1](#_Toc195811915)

[Fig S9 1](#_Toc195811916)

[Fig S10 1](#_Toc195811917)

[Fig S11 1](#_Toc195811918)

[Modeling 1](#_Toc195811919)

[Total occruance 1](#_Toc195811920)

[Model structure 1](#_Toc195811921)

[Re-load models 1](#_Toc195811922)

[Model diagnostics 1](#_Toc195811923)

[Model estimates 1](#_Toc195811924)

[Percentage detections 1](#_Toc195811925)

[Re-load models 1](#_Toc195811926)

[Model diagnostics 1](#_Toc195811927)

[Model estimates 1](#_Toc195811928)

[📈 Model results 1](#_Toc195811929)

[Total detections 1](#_Toc195811930)

[Fig S12 1](#_Toc195811931)

[Table S5 1](#_Toc195811932)

[Percentage detections 1](#_Toc195811933)

[Fig S13 1](#_Toc195811934)

[Table S6 1](#_Toc195811935)

[🧠 Supplementary discussion 1](#_Toc195811936)

[Compound occurrence comparison 1](#_Toc195811937)

# 📕 README

**Supplematry matrial 1** This File is write to act as supplmeratry matria 1 for the assocat

**PART 2 of 2** This script “field-vs-lab-doses-r-2” is the second of two used for this project. The second script, “field-vs-lab-doses-r-2”, deals with the analysis. The first script, “field-vs-lab-doses-r-1”, tidies the input databases and combines them into a single data frame for analysis. If you download the data files for this project from the Open Sciecne Framework (), you can run this script independently.

**ARTICLE** This R script is associated with **Martin et al (2025)** “Aligning Behavioural Ecotoxicology Tests with Real-World Water Concentrations: Current Minimum Tested Levels Far Exceed Environmental Reality” DOI: <*to be added*>

**AUTHORS**  
Jake M. Martin 1,2,3\*  
Erin S. McCallum 1  
Jack A. Band 2,4

**AFFILIATIONS**  
(1) School of Life and Environmental Sciences, Deakin University, Geelong, Australia  
(2) Department of Wildlife, Fish, and Environmental Studies, Swedish University of Agricultural Sciences, Umeå, Sweden  
(3) School of Biological Sciences, Monash University, Melbourne, Australia  
(4) Institute of Zoology, Zoological Society of London, London, United Kingdom  
(\*) Corresponding author

**AIM**  
The aim of this study is two-fold: (1) identify whether tested concentrations in behavioural ecotoxicology studies reflect those that are detected in surface waters and/or waste water; (2) to compare the target pharmaceuticals tested in behavioural ecotoxicology to what have been detected in the environment, highlighting potential targets for future investigation.

**METHODS**  
To achieve these aims, this study capitalises on four extensive open access databases, one of which focuses on peer-review research on the behavioural ecotoxicology of pharmaceuticals1, and the remaining focus on pharmaceutical environmental occurrence and concentrations2,3,4.

1. The Evidence of the Impacts of Pharmaceuticals on Aquatic Animal Behaviour (EIPAAB) database ([Martin et al. 2025](https://doi.org/10.32942/X2NG9R))

📥 EIPAAB Database

1. The Umwelt Bundesamt Pharmaceuticals in the environment (PHARMS-UBA) database

📥 PHARMS-UBA Database

1. Pharmaceutical pollution of the world’s rivers ([Wilkinson et al 2022](https://doi.org/10.1073/pnas.2113947119)), specifically “Dataset S4. Database of pharmaceutical concentrations at all the sampling locations monitored in this project”

📥 Wilkinson Database

1. The NORMAN EMPODAT Database - Chemical Occurrence Data.

📥 NORMAN EMPODAT Database

**SCRIPT**  
This is an R markdown script written in R studio (2023.09.0+463 “Desert Sunflower” Release). The ‘field-vs-lab-doses’ git repository hosts this script, and if downloaded (or pulled) it will reproduce all data tidy/filter, analysis, and visualisations used in Martin et al (2025). The GitHub repository includes all raw input data from the four open access databases, as well as all output data and figures. I have tried to structure this code with Open, Reliable, and Transparent (ORT) coding practices in mind. Feel free to reach out if anything is unclear.

The R script first cleans and filters each databases to make them comparable. In the gernalsit sensce In terms of filtering, we target pharmaceutical compounds, surface waters/waste water effluent, and environmental/lab samples measured in units of mass per volume of water (e.g. ug/L).

We then summarise occurrence and concentrations between EIPAAB the three environmental databases (UBA, Wilkson and NORMAN) separately, to highlight the consistency of different environmental databases.

Lastly we combined all environmental datasets for an overall assesses of occurrence and concentrations between EIPAAB and environmental data.

[**GitHub**](https://github.com/JakeMartinResearch/field-vs-lab-doses)

**DISCLAIMER**  
I (Jake Martin) am dyslexic. I have made an effort to review the script for grammatical errors, but some will likely remain. I apologise. Please reach out via the contact details below if anything is unclear.

# 📧 Contact

🐟 **Jake M. Martin**

📧 **Email:** [jake.martin@deakin.edu.au](mailto:jake.martin@deakin.edu.au)

📧 **Alt Email:** [jake.martin.research@gmail.com](mailto:jake.martin.research@gmail.com)

🌐 **Web:** [jakemartin.org](https://jakemartin.org/)

🐙 **GitHub**: [JakeMartinResearch](https://github.com/JakeMartinResearch)

# 📑 Sharing/accessing and citing

1. **Licenses/restrictions placed on the data:** CC-BY 4.0
2. **Link to the associated publication:**  
   🚧 ***To be added*** 🚧
3. **Recommended citation for this data:**  
   🚧 ***To be added*** 🚧

# ⚙️ Set-up and packages

Here we define our Knit settings, to make the output more user friendly, and to cache output for faster knitting.

#kniter setting  
knitr::opts\_chunk$set(  
message = FALSE,  
warning = FALSE, # no warnings  
cache = TRUE,# Cacheing to save time when kniting  
cache.lazy = FALSE, # I am running into issues with cache because the files are very large  
tidy = TRUE  
)  
  
# ---- Install pacman if it's not already installed ----  
if (!requireNamespace("pacman", quietly = TRUE)) install.packages("pacman")  
  
# ---- List of required packages ----  
pkgs <- c(  
 # ----- Data Visualisation -----  
 "ggthemes", "bayesplot", "gt", "gtsummary", "plotly", "qqplotr", "ggrepel",  
 "colorspace",  
   
 # ----- Tidy Data and Wrangling -----  
 "tidyverse", "janitor", "readxl", "broom.mixed", "data.table", "devtools", "scales",  
   
 # ----- Modelling and Statistical Analysis -----  
 "brms", "rstan", "cmdstanr", "marginaleffects", "performance", "emmeans",  
 "tidybayes","future", "coda"  
)  
  
# ---- Install and load all packages using pacman ----  
suppressPackageStartupMessages(  
 pacman::p\_load(char = pkgs, install = TRUE)  
)

Here’s a list of the package names

pkgs

## [1] "ggthemes" "bayesplot" "gt" "gtsummary"   
## [5] "plotly" "qqplotr" "ggrepel" "colorspace"   
## [9] "tidyverse" "janitor" "readxl" "broom.mixed"   
## [13] "data.table" "devtools" "scales" "brms"   
## [17] "rstan" "cmdstanr" "marginaleffects" "performance"   
## [21] "emmeans" "tidybayes" "future" "coda"

# 🔧 Custom functions

Here are some custom function used within this script.

sentence\_case() Changes a string into sentence case.

sentence\_case <- function(text) {  
 text <- tolower(text) # Convert the entire text to lowercase  
 text <- sub("^(\\s\*\\w)", "\\U\\1", text, perl = TRUE) # Capitalise the first letter  
 return(text)  
}

# 📂 Directories

Here we define the directories for the project. We also make the output directories if they don’t already exist.

## Input

These are the input directors for the databases used in this project.

**📥 data\_wd**: Directory for the input data.

wd <- getwd() # getwd tells us what the current wd is, we are using this to drop it in a variable called wd  
data\_wd <- paste0(wd, "./data") # creates a variable with the name of the wd we want to use

## Output

**fig\_wd**: Directory for the output figures.

fig\_wd <- paste0(wd, "./fig")  
if (!dir.exists("fig")) {  
 dir.create("fig")  
}

**mods\_wd**: Directory for the output models.

mods\_wd <- paste0(wd, "./mods")  
if (!dir.exists("mods")) {  
 dir.create("mods")  
}

# 💿 Databases

This is the database made in part one of this script martin-field-vs-lab-all-databases.csv

## All databases

all\_databases <- fread(paste0(data\_wd, "./martin-field-vs-lab-all-databases.csv"))

# 🧮 Analysis

## Samples/data

Let’s see how much data is present in the filtered and combined databases.

total\_compounds\_eipaab <- all\_databases %>%  
 dplyr::filter(source == "eipaab") %>%  
 nrow(.)  
  
total\_water\_samples <- all\_databases %>%  
 dplyr::filter(source != "eipaab") %>%  
 dplyr::reframe(n = sum(n\_units\_est, na.rm = TRUE)) %>%  
 dplyr::pull(n)  
  
paste0("There are ", total\_compounds\_eipaab, " behavioural ecotoxicology exposures (compounds by species) and ",  
 total\_water\_samples, " water samples")

## [1] "There are 767 behavioural ecotoxicology exposures (compounds by species) and 10009385 water samples"

How many water samples are surface water vs wastewater

all\_databases %>%  
 dplyr::filter(source != "eipaab") %>%  
 dplyr::group\_by(matrix\_group) %>%  
 dplyr::reframe(n = sum(n\_units\_est, na.rm = TRUE)) %>%  
 gt()

| matrix\_group | n |
| --- | --- |
| effluent | 281752 |
| surfacewater | 9727633 |

### Table S1 (Table 1)

A breakdown for data contributed by each data source.

In the manuscript this will be called **Table 1**

**Table 1**: The number of pharmaceutical compounds within each of the four dataset (EIPAAB, NORMAN, PHARMS-UBA, and Wilksom) used for this investigation, and number of individual data rows within each of the database (i.e. testsa for EIPAAB, and water samplesb for the environmental occurrence databases).

soruce\_data <- all\_databases %>%  
 dplyr::group\_by(source) %>%  
 dplyr::reframe(Data = sum(n\_units\_est, na.rm = TRUE), `Number of compounds` = length(unique(compound\_name\_corrected))) %>%  
 dplyr::rename(Source = source) %>%  
 dplyr::mutate(Source = case\_when(Source == "eipaab" ~ "EIPAAB", Source == "norman" ~  
 "NORMAN‡", Source == "uba" ~ "PHARMS-UBA", Source == "wilkson" ~ "Wilkson et al.",  
 ))  
  
total\_data <- all\_databases %>%  
 dplyr::reframe(Data = sum(n\_units\_est, na.rm = TRUE), `Number of compounds` = length(unique(compound\_name\_corrected))) %>%  
 dplyr::mutate(Source = "Total")  
  
n\_summary <- rbind(soruce\_data, total\_data)  
  
table\_1 <- n\_summary %>%  
 gt() %>%  
 fmt\_number(columns = "Data", decimals = 0) %>%  
 cols\_align(align = "center", columns = everything())  
  
table\_1

| Source | Data | Number of compounds |
| --- | --- | --- |
| EIPAAB | 767 | 184 |
| NORMAN‡ | 9,382,388 | 1379 |
| PHARMS-UBA | 562,865 | 911 |
| Wilkson et al. | 64,132 | 61 |
| Total | 10,010,152 | 1760 |

*Note: (a) each row of the EIPAAB database represents a unique species by compound exposure; (b) each row within the NORMAN and PHARMS-UBA represents one of more samples, as a row can be a measure of central tendency (e.g. mean or median) or a single value, all rows in the Wilkson database represent a single sample; ‡ NORMAN database was filtered to remove data from the German Environment Agency (UBA), and restricted to 2014-2022, as the PHARMS-UBA also included data from NORMAN prior to 2014. Thus, this number is not a true total number of pharmaceutical samples present in the NORMAN database.*

Number of aggregate values that had missing sample size.

all\_data <- all\_databases %>%  
 dplyr::filter(source != "eipaab") %>%  
 dplyr::reframe(total\_data\_n = length(compound\_cas)) %>%  
 dplyr::pull(total\_data\_n)  
  
source\_units\_nas <- all\_databases %>%  
 dplyr::filter(concentration\_type == "central") %>%  
 dplyr::group\_by(source) %>%  
 dplyr::reframe(total\_central = length(compound\_cas), total\_missing\_units = sum(is.na(n\_units)),  
 all\_data\_n = all\_data, percent\_aggregate = round((total\_central/all\_data\_n) \*  
 100, 2), percent\_missing = round((total\_missing\_units/all\_data\_n) \* 100,  
 2)) %>%  
 dplyr::rename(Source = source, `Total central data` = total\_central, `Total missing units` = total\_missing\_units,  
 `All data` = all\_data\_n, `Percent aggregate` = percent\_aggregate, `Percent missing` = percent\_missing) %>%  
 dplyr::mutate(Source = case\_when(Source == "eipaab" ~ "EIPAAB", Source == "norman" ~  
 "NORMAN‡", Source == "uba" ~ "PHARMS-UBA", Source == "wilkson" ~ "Wilkson et al.",  
 ))  
  
source\_units\_nas %>%  
 gt()

| Source | Total central data | Total missing units | All data | Percent aggregate | Percent missing |
| --- | --- | --- | --- | --- | --- |
| PHARMS-UBA | 31725 | 4023 | 9569449 | 0.33 | 0.04 |

## Cross-over

Checking the cross-over between the compounds assessed in behavioural test against all environmental occurrence data.

eipaab\_compounds\_list <- all\_databases %>%  
 dplyr::filter(source == "eipaab") %>%  
 dplyr::distinct(compound\_name\_corrected) %>%  
 dplyr::pull(compound\_name\_corrected)  
  
n\_eipaab\_compounds <- length(eipaab\_compounds\_list)  
  
enviro\_\_cross\_over\_compounds\_list <- all\_databases %>%  
 dplyr::filter(source != "eipaab" & compound\_name\_corrected %in% eipaab\_compounds\_list) %>%  
 dplyr::distinct(compound\_name\_corrected) %>%  
 dplyr::pull(compound\_name\_corrected)  
  
n\_cross\_over\_compounds <- length(enviro\_\_cross\_over\_compounds\_list)  
  
percent\_cross\_over = round(n\_cross\_over\_compounds/n\_eipaab\_compounds \* 100, 2)  
  
paste0(n\_cross\_over\_compounds, " of the ", n\_eipaab\_compounds, " pharmaceuticals tested in the EIPAAB database (with an environmental motivation) were present in the environmental databases (i.e. ",  
 percent\_cross\_over, "%)")

## [1] "167 of the 184 pharmaceuticals tested in the EIPAAB database (with an environmental motivation) were present in the environmental databases (i.e. 90.76%)"

## Concentrations

For questions around concentration, I will filtered for only cases where the levels are above zero or above the limit of detection (i.e. positive detections or above LOD). We will do this for two reasons, the first being that when researchers are trying to estimate the risk of environmental exposure, we assume that they are emulating potential exposure events at a contaminated site, not trying to replicate a theoretical global surface water average. The second is, by removing zero values and values below LODs, we are trying to be ***more conservative*** with the relative comparison between tested doses in behavioural ecotoxicology and observed environmental doses (as opposed to using some value substitute/estimated value for samples below detection limit). We are also using both single values and summary values of central tendency (means, median, ect). Fig

In this section we are calculating summary metrics for concentrations used in the behavioural test and detected in surface waters and effluent in the enviromental data. We will do this separately for effluent and surface waters samples. We are calculating, the mean, median, min, max, range, n (number of contributing data points), standard deviation (sd), standard error (se), and the upper and lower 95% credible intervals (ci\_upper, ci\_lower) using empirical quantiles.

conc\_summary <- all\_databases %>%   
 dplyr::filter(value > 0) %>%   
 dplyr::group\_by(matrix\_group, compound\_name\_corrected) %>%   
 dplyr::reframe(mean = mean(value, na.rm = TRUE),  
 median = median(value, na.rm = TRUE),  
 min = min(value, na.rm = TRUE),  
 max = max(value, na.rm = TRUE),  
 range = max-min,  
 n = length(unique\_row\_id), # Sample size  
 sd = sd(value, na.rm = TRUE), # Standard deviation  
 ci\_lower = quantile(value, 0.025, na.rm = TRUE), #95% credible interval using quantiles  
 ci\_upper = quantile(value, 0.975, na.rm = TRUE)  
 ) %>%   
 dplyr::mutate(ci\_lower = if\_else(is.na(ci\_lower), NA, ci\_lower),  
 ci\_upper = if\_else(is.na(ci\_upper), NA, ci\_upper)) %>%   
 tidyr::complete(  
 tidyr::expand(nesting(matrix\_group, compound\_name\_corrected))  
 )

Here we will add the summary statistics from above to the full database, so we can see how many EIPAAB tested concentrations fall under the median surface water and effluent concentrations for each compound.

rename\_cols <- conc\_summary %>%   
 dplyr::select(-matrix\_group, -compound\_name\_corrected) %>%   
 colnames()  
  
conc\_summary\_eff <- conc\_summary %>%   
 dplyr::filter(matrix\_group == "effluent") %>%   
 dplyr::rename\_with(  
 .cols = all\_of(rename\_cols), # Select columns to rename  
 .fn = ~ paste0("effluent\_", .) # Prefix with "effluent\_"  
 ) %>%   
 dplyr::select(-matrix\_group)  
  
  
conc\_summary\_surf <- conc\_summary %>%   
 dplyr::filter(matrix\_group == "surfacewater") %>%   
 dplyr::rename\_with(  
 .cols = all\_of(rename\_cols), # Select columns to rename  
 .fn = ~ paste0("surfacewater\_", .) # Prefix with "effluent\_"  
 ) %>%   
 dplyr::select(-matrix\_group)  
  
  
all\_databases\_with\_sum <- all\_databases %>%   
 dplyr::filter(source == "eipaab") %>%   
 dplyr::left\_join(., conc\_summary\_eff, by = "compound\_name\_corrected") %>%   
 dplyr::left\_join(., conc\_summary\_surf, by = "compound\_name\_corrected") %>%   
 dplyr::mutate(under\_med\_surf = if\_else(value < surfacewater\_median, 1, 0),  
 under\_hci\_surf = if\_else(value < surfacewater\_ci\_upper, 1, 0),  
 under\_med\_effl = if\_else(value < effluent\_median, 1, 0),  
 under\_hci\_effl = if\_else(value < effluent\_ci\_upper, 1, 0),  
 diff\_med\_surf = value/surfacewater\_median,  
 diff\_hic\_surf = value/surfacewater\_ci\_upper,  
 diff\_med\_effl = value/effluent\_median,  
 diff\_hic\_effl = value/effluent\_ci\_upper  
 )

Making a database for surface waters and waste waters

surfacewater\_conc <- all\_databases\_with\_sum %>%  
 dplyr::filter(!is.na(surfacewater\_median), !is.na(value)) %>%  
 dplyr::mutate(value\_log = log(value), surfacewater\_median\_log = log(surfacewater\_median),  
 relative\_year = year - min(year))  
  
wastewater\_conc <- all\_databases\_with\_sum %>%  
 dplyr::filter(!is.na(effluent\_median), !is.na(value)) %>%  
 dplyr::mutate(value\_log = log(value), effluent\_median\_log = log(effluent\_median),  
 relative\_year = year - min(year))

### Modeling

#### 🌊 Surface water

##### Model structure

concentration\_str <- bf(value\_log ~ surfacewater\_median\_log + relative\_year, family = gaussian())  
  
concentration\_str\_null <- bf(value\_log ~ relative\_year, family = gaussian())

These are the default priors.

suppressWarnings(get\_prior(concentration\_str, data = surfacewater\_conc, family = gaussian()))

## prior class coef group resp dpar  
## (flat) b   
## (flat) b relative\_year   
## (flat) b surfacewater\_median\_log   
## student\_t(3, -0.2, 4.4) Intercept   
## student\_t(3, 0, 4.4) sigma   
## nlpar lb ub source  
## default  
## (vectorized)  
## (vectorized)  
## default  
## 0 default

Run model. This has been hashed out as I have saved the model, and will re-load it instead of re-running it.

# options(brms.backend = 'cmdstanr') concentration\_mod\_log <-  
# brm(concentration\_str, data = surfacewater\_conc, cores = 4, chains = 4,  
# #prior = mod\_priors, #use defaults warmup = 1000, seed = 20250418, thin = 2,  
# iter = 8000, control = list(max\_treedepth = 20, adapt\_delta = 0.95),  
# save\_pars = save\_pars(all=TRUE), sample\_prior = TRUE, file = paste0(mods\_wd,  
# './concentration\_mod\_log')) concentration\_mod\_null\_log <-  
# brm(concentration\_str\_null, data = surfacewater\_conc, cores = 4, chains = 4,  
# #prior = mod\_priors, #use defaults warmup = 1000, seed = 20250418, thin = 2,  
# iter = 8000, control = list(max\_treedepth = 20, adapt\_delta = 0.95),  
# save\_pars = save\_pars(all=TRUE), sample\_prior = TRUE, file = paste0(mods\_wd,  
# './concentration\_mod\_null\_log')) print('Model complete')

##### Re-load models

Reload the model if necessary.

concentration\_mod\_log <- readRDS(file = paste0(mods\_wd, "./concentration\_mod\_log.rds"))  
concentration\_mod\_null\_log <- readRDS(file = paste0(mods\_wd, "./concentration\_mod\_null\_log.rds"))

##### Model diagnostics

Comparing null model with surface water model

loo\_compare(loo(concentration\_mod\_log, moment\_match = TRUE), loo(concentration\_mod\_null\_log,  
 moment\_match = TRUE))

## elpd\_diff se\_diff  
## concentration\_mod\_null\_log 0.0 0.0   
## concentration\_mod\_log -0.6 1.2

The null model is only slightly worse

bayes\_factor(concentration\_mod\_log, concentration\_mod\_null\_log)

## Iteration: 1  
## Iteration: 2  
## Iteration: 3  
## Iteration: 4  
## Iteration: 1  
## Iteration: 2  
## Iteration: 3  
## Iteration: 4

## Estimated Bayes factor in favor of concentration\_mod\_log over concentration\_mod\_null\_log: 0.55435

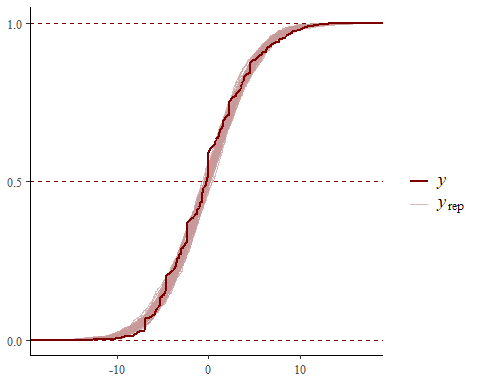
The R2 value is very low.

performance::r2\_bayes(concentration\_mod\_log, robust = FALSE, ci = 0.95)

## # Bayesian R2 with Compatibility Interval  
##   
## Conditional R2: 0.020 (95% CI [0.003, 0.039])

The mode roughly approximates the data

color\_scheme\_set("red")  
brms::pp\_check(concentration\_mod\_log, ndraws = 50, type = "ecdf\_overlay")



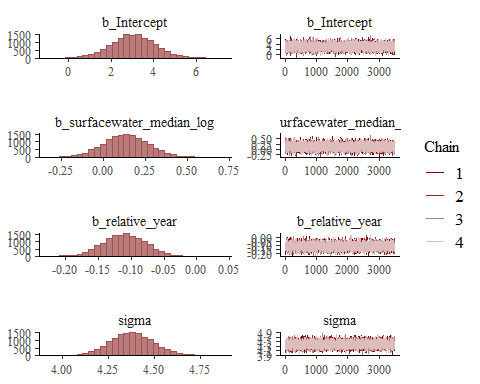
Checking R-hat values to confirm model convergence and ESS fo estimates.

print(summary(concentration\_mod\_log, prob = 0.95)) #All Rhat = 1 (good).

## Family: gaussian   
## Links: mu = identity; sigma = identity   
## Formula: value\_log ~ surfacewater\_median\_log + relative\_year   
## Data: surfacewater\_conc (Number of observations: 706)   
## Draws: 4 chains, each with iter = 8000; warmup = 1000; thin = 2;  
## total post-warmup draws = 14000  
##   
## Regression Coefficients:  
## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS  
## Intercept 3.04 1.00 1.07 5.00 1.00 13348  
## surfacewater\_median\_log 0.13 0.13 -0.12 0.39 1.00 13062  
## relative\_year -0.11 0.03 -0.17 -0.05 1.00 13519  
## Tail\_ESS  
## Intercept 11983  
## surfacewater\_median\_log 12261  
## relative\_year 12232  
##   
## Further Distributional Parameters:  
## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS  
## sigma 4.38 0.12 4.16 4.62 1.00 12903 11070  
##   
## Draws were sampled using sample(hmc). For each parameter, Bulk\_ESS  
## and Tail\_ESS are effective sample size measures, and Rhat is the potential  
## scale reduction factor on split chains (at convergence, Rhat = 1).

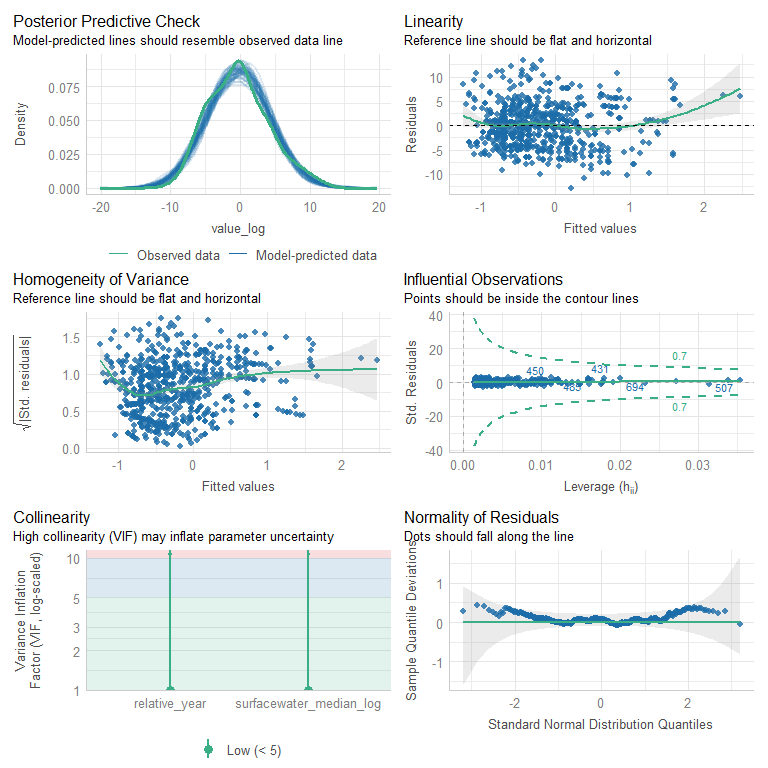
Diagnostic plots look fine. A smaller shape value indicates greater overdispersion (variance is much larger than the mean).

color\_scheme\_set("red")  
plot(concentration\_mod\_log)



Check model performance

performance::check\_model(concentration\_mod\_log)



##### Model estimates

Pulling the marginal effect estimates from the model for our plots.

# Extract marginal effects for surface water and year  
estimate\_data\_1 <- marginal\_effects(concentration\_mod\_log, effects = "surfacewater\_median\_log")[[1]]  
estimate\_data\_year\_1 <- marginal\_effects(concentration\_mod\_log, effects = "relative\_year")[[1]]

These estimated coefficients represent the average effects across the entire dataset, controlling for (not averaging over) the other variables in the model. Because this is a log–log model, the coefficient for log(surfacewater\_median) reflects an elasticity—that is, the percent change in the response (log(value)) for a percent change in the predictor.

The estimated effect of log(surfacewater\_median) was **β = 0.135** (95% CrI: –0.123 to 0.393), indicating that a **1% increase in surface water concentration** is associated with an average **0.135% increase in tested concentration**. However, because the 95% credible interval overlaps with zero, this **effect is not statistically well-supported**.

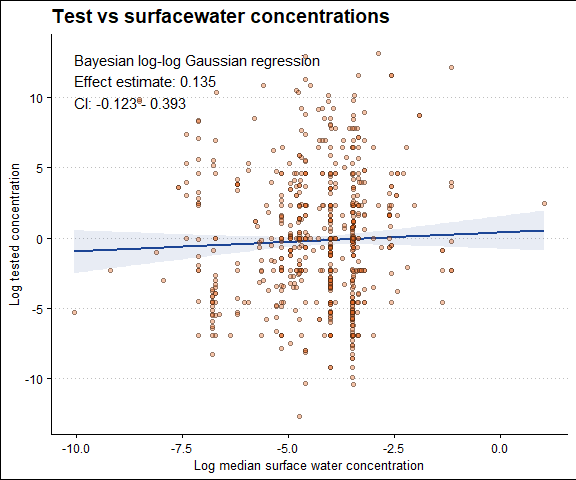
For relative\_year, the estimated coefficient was **β = –0.110** (95% CrI: –0.173 to –0.046), suggesting a clear temporal decline in tested concentrations. Because the predictor is linear and the response is log-transformed, this implies that **each additional year** is associated with an average **10.4% decrease in tested concentration**, holding surface water concentration constant (exp(–0.110) ≈ 0.896, meaning the response is multiplied by ~0.896 each year).

model\_summary\_1 <- as.data.frame(fixef(concentration\_mod\_log, summary = TRUE)) %>%  
 rownames\_to\_column() %>%  
 clean\_names()  
  
estimate\_1 <- model\_summary\_1$estimate[2]  
L95CI\_1 <- model\_summary\_1$q2\_5[2]  
H95CI\_1 <- model\_summary\_1$q97\_5[2]  
  
estimate\_year\_1 <- model\_summary\_1$estimate[3]  
L95CI\_year\_1 <- model\_summary\_1$q2\_5[3]  
H95CI\_year\_1 <- model\_summary\_1$q97\_5[3]  
  
  
model\_summary\_1 %>%  
 dplyr::mutate(across(where(is.double), ~round(.x, 3))) %>%  
 gt()

|  | estimate | est\_error | q2\_5 | q97\_5 |
| --- | --- | --- | --- | --- |
| Intercept | 3.038 | 1.003 | 1.068 | 4.999 |
| surfacewater\_median\_log | 0.135 | 0.132 | -0.123 | 0.393 |
| relative\_year | -0.110 | 0.033 | -0.173 | -0.046 |

##### Fig S1 (Fig 1a)

text\_coord <- surfacewater\_conc %>%  
 summarise(y\_max = max(value\_log, na.rm = TRUE), x\_max = max(surfacewater\_median\_log,  
 na.rm = TRUE), x\_min = min(surfacewater\_median\_log, na.rm = TRUE), x = mean(c(x\_max,  
 x\_min)))  
  
y\_coord <- text\_coord %>%  
 pull(y\_max)  
  
x\_coord <- text\_coord %>%  
 pull(x\_min)  
  
  
fig\_1a <- ggplot() + geom\_line(data = estimate\_data\_1, aes(x = surfacewater\_median\_log,  
 y = estimate\_\_), color = "#1c4595", linewidth = 1) + geom\_ribbon(data = estimate\_data\_1,  
 aes(x = surfacewater\_median\_log, ymin = lower\_\_, ymax = upper\_\_), fill = "#1c4595",  
 alpha = 0.1) + geom\_point(data = surfacewater\_conc, aes(x = surfacewater\_median\_log,  
 y = value\_log), alpha = 0.4, shape = 21, color = "#01080A", fill = "#E76A24") +  
 annotate("text", x = x\_coord, y = y\_coord, label = paste0("Bayesian log-log Gaussian regression",  
 "\n", "Effect estimate: ", round(estimate\_1, 3), "\n", "CI: ", round(L95CI\_1,  
 3), " - ", round(H95CI\_1, 3)), hjust = 0, vjust = 1, size = 4) + labs(title = "Test vs surfacewater concentrations",  
 x = "Log median surface water concentration", y = "Log tested concentration") +  
 theme\_clean() + theme(legend.position = "bottom", legend.justification = "center")  
  
fig\_1a



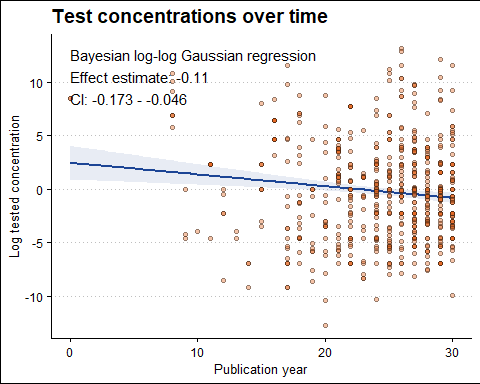
Save this figure for the MS

# ggsave(filename = paste0(fig\_wd, './figure-1a.pdf'), plot = fig\_1a, width =  
# 210, height = 297/1.5, # Specify the height of the plot units = 'mm')

##### Fig S2

Plot surface water concentrations by year

text\_coord <- surfacewater\_conc %>%  
 summarise(y\_max = max(value\_log, na.rm = TRUE), x\_max = max(relative\_year, na.rm = TRUE),  
 x\_min = min(relative\_year, na.rm = TRUE), x\_mean = mean(c(x\_max, x\_min)))  
  
y\_coord <- text\_coord %>%  
 pull(y\_max)  
  
x\_coord <- text\_coord %>%  
 pull(x\_min)  
  
  
fig\_s2 <- ggplot() + geom\_line(data = estimate\_data\_year\_1, aes(x = relative\_year,  
 y = estimate\_\_), color = "#1c4595", linewidth = 1) + geom\_ribbon(data = estimate\_data\_year\_1,  
 aes(x = relative\_year, ymin = lower\_\_, ymax = upper\_\_), fill = "#1c4595", alpha = 0.1) +  
 geom\_point(data = surfacewater\_conc, aes(x = relative\_year, y = value\_log), alpha = 0.4,  
 shape = 21, color = "#01080A", fill = "#E76A24") + annotate("text", x = x\_coord,  
 y = y\_coord, label = paste0("Bayesian log-log Gaussian regression", "\n", "Effect estimate: ",  
 round(estimate\_year\_1, 3), "\n", "CI: ", round(L95CI\_year\_1, 3), " - ", round(H95CI\_year\_1,  
 3)), hjust = 0, vjust = 1, size = 4) + labs(title = "Test concentrations over time",  
 x = "Publication year", y = "Log tested concentration") + theme\_clean() + theme(legend.position = "bottom",  
 legend.justification = "center")  
  
fig\_s2



#### 🚽 Waste water

##### Model structure

eff\_concentration\_str <- bf(value\_log ~ effluent\_median\_log + relative\_year, family = gaussian())  
  
eff\_concentration\_str\_null <- bf(value\_log ~ relative\_year, family = gaussian())

These are the default priors.

suppressWarnings(get\_prior(eff\_concentration\_str, data = wastewater\_conc, family = gaussian()))

## prior class coef group resp dpar nlpar lb  
## (flat) b   
## (flat) b effluent\_median\_log   
## (flat) b relative\_year   
## student\_t(3, -0.2, 4.5) Intercept   
## student\_t(3, 0, 4.5) sigma 0  
## ub source  
## default  
## (vectorized)  
## (vectorized)  
## default  
## default

Run model. This has been hashed out as I have saved the model, and will re-load it instead of re-running it.

# options(brms.backend = 'cmdstanr') eff\_concentration\_mod\_log <-  
# brm(eff\_concentration\_str, data = wastewater\_conc, cores = 4, chains = 4,  
# #prior = mod\_priors, #use defaults warmup = 1000, seed = 20250418, thin = 2,  
# iter = 8000, control = list(max\_treedepth = 20, adapt\_delta = 0.95),  
# save\_pars = save\_pars(all=TRUE), sample\_prior = TRUE, file = paste0(mods\_wd,  
# './eff\_concentration\_mod\_log')) eff\_concentration\_mod\_null\_log <-  
# brm(eff\_concentration\_str\_null, data = wastewater\_conc, cores = 4, chains =  
# 4, #prior = mod\_priors, #use defaults warmup = 1000, seed = 20250418, thin =  
# 2, iter = 8000, control = list(max\_treedepth = 20, adapt\_delta = 0.9999),  
# save\_pars = save\_pars(all=TRUE), sample\_prior = TRUE, file = paste0(mods\_wd,  
# './eff\_concentration\_mod\_null\_log')) print('Model complete')

##### Re-load models

Reload the model if necessary.

eff\_concentration\_mod\_log <- readRDS(file = paste0(mods\_wd, "./eff\_concentration\_mod\_log.rds"))  
eff\_concentration\_mod\_null\_log <- readRDS(file = paste0(mods\_wd, "./eff\_concentration\_mod\_null\_log.rds"))

##### Model diagnostics

Comparing null model with effluent water model

loo\_compare(loo(eff\_concentration\_mod\_log, moment\_match = TRUE), loo(eff\_concentration\_mod\_null\_log,  
 moment\_match = TRUE))

## elpd\_diff se\_diff  
## eff\_concentration\_mod\_log 0.0 0.0   
## eff\_concentration\_mod\_null\_log -31.3 8.0

The model with effluent is much better

bayes\_factor(eff\_concentration\_mod\_log, eff\_concentration\_mod\_null\_log)

## Iteration: 1  
## Iteration: 2  
## Iteration: 3  
## Iteration: 4  
## Iteration: 1  
## Iteration: 2  
## Iteration: 3  
## Iteration: 4  
## Iteration: 5

## Estimated Bayes factor in favor of eff\_concentration\_mod\_log over eff\_concentration\_mod\_null\_log: 22579493748979.09375

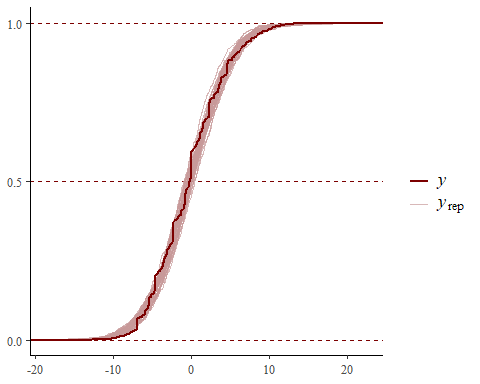
The R2 value is very low.

performance::r2\_bayes(eff\_concentration\_mod\_log, robust = FALSE, ci = 0.95)

## # Bayesian R2 with Compatibility Interval  
##   
## Conditional R2: 0.106 (95% CI [0.065, 0.145])

The mode roughly approximates the data

color\_scheme\_set("red")  
brms::pp\_check(eff\_concentration\_mod\_log, ndraws = 50, type = "ecdf\_overlay")



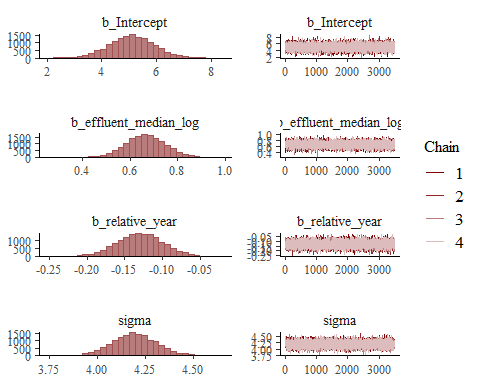
Checking R-hat values to confirm model convergence and ESS fo estimates.

print(summary(eff\_concentration\_mod\_log, prob = 0.95)) #All Rhat = 1 (good).

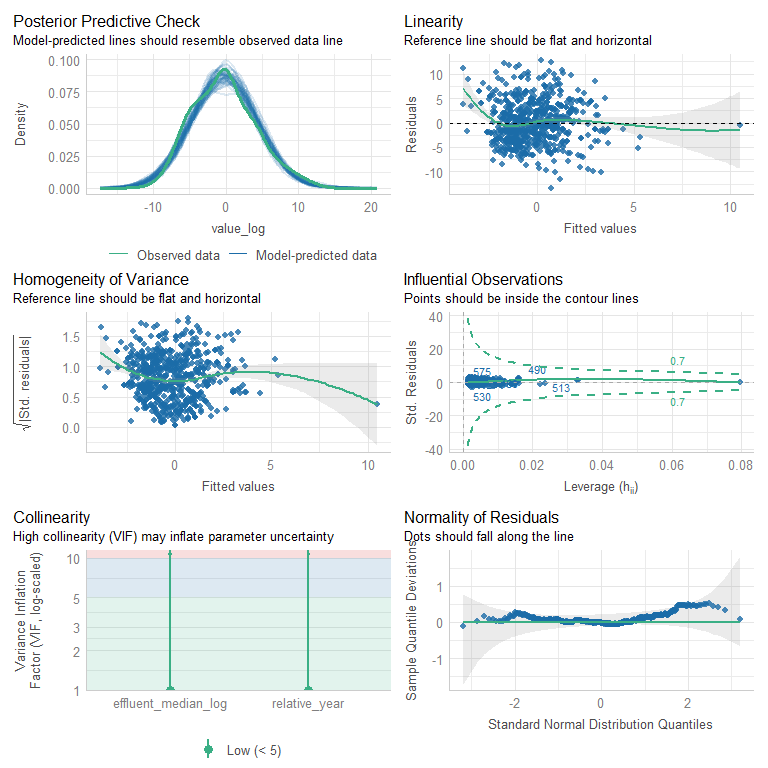
## Family: gaussian   
## Links: mu = identity; sigma = identity   
## Formula: value\_log ~ effluent\_median\_log + relative\_year   
## Data: wastewater\_conc (Number of observations: 714)   
## Draws: 4 chains, each with iter = 8000; warmup = 1000; thin = 2;  
## total post-warmup draws = 14000  
##   
## Regression Coefficients:  
## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS  
## Intercept 5.10 0.81 3.52 6.67 1.00 13295 12494  
## effluent\_median\_log 0.67 0.08 0.50 0.82 1.00 13126 12732  
## relative\_year -0.13 0.03 -0.19 -0.07 1.00 13370 12175  
##   
## Further Distributional Parameters:  
## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS  
## sigma 4.19 0.11 3.98 4.42 1.00 13300 12153  
##   
## Draws were sampled using sample(hmc). For each parameter, Bulk\_ESS  
## and Tail\_ESS are effective sample size measures, and Rhat is the potential  
## scale reduction factor on split chains (at convergence, Rhat = 1).

Diagnostic plots look fine. A smaller shape value indicates greater overdispersion (variance is much larger than the mean).

color\_scheme\_set("red")  
plot(eff\_concentration\_mod\_log)



performance::check\_model(eff\_concentration\_mod\_log)



##### Model estimates

Pulling the marginal effect estimates from the model for our plots.

# Extract marginal effects for surface water and year  
estimate\_data\_2 <- marginal\_effects(eff\_concentration\_mod\_log, effects = "effluent\_median\_log")[[1]]  
estimate\_data\_year\_2 <- marginal\_effects(eff\_concentration\_mod\_log, effects = "relative\_year")[[1]]

These estimated coefficients represent the average effects across the entire dataset, controlling for (not averaging over) the other variables in the model. Because this is a log–log model, the coefficient for log(effluent\_median) reflects an elasticity: the percent change in the response (log(value)) for a percent change in the predictor (log(effluent\_median)).

The estimated effect of log(effluent\_median) was **β = 0.665** (95% CrI: 0.505 to 0.823), indicating that a **1% increase in effluent median concentration** is associated with an average **0.665% increase in tested concentration**. Since the credible interval does not include zero, this is a statistically meaningful effect.

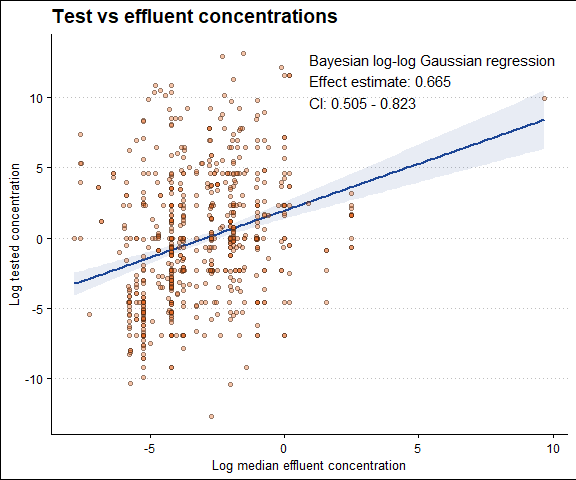
For relative\_year, the estimated coefficient was **β = –0.130** (95% CrI: –0.189 to –0.070), suggesting a temporal decline in tested concentrations. Because the predictor is linear and the response is log-transformed, this indicates that each additional year is associated with an average **12.2% decrease in tested concentration, holding effluent concentration constant** (exp(–0.130) ≈ 0.878).

model\_summary\_2 <- as.data.frame(fixef(eff\_concentration\_mod\_log, summary = TRUE)) %>%  
 rownames\_to\_column() %>%  
 clean\_names()  
  
estimate\_2 <- model\_summary\_2$estimate[2]  
L95CI\_2 <- model\_summary\_2$q2\_5[2]  
H95CI\_2 <- model\_summary\_2$q97\_5[2]  
  
estimate\_year\_2 <- model\_summary\_2$estimate[3]  
L95CI\_year\_2 <- model\_summary\_2$q2\_5[3]  
H95CI\_year\_2 <- model\_summary\_2$q97\_5[3]  
  
  
model\_summary\_2 %>%  
 dplyr::mutate(across(where(is.double), ~round(.x, 3))) %>%  
 gt()

|  | estimate | est\_error | q2\_5 | q97\_5 |
| --- | --- | --- | --- | --- |
| Intercept | 5.099 | 0.806 | 3.522 | 6.672 |
| effluent\_median\_log | 0.665 | 0.082 | 0.505 | 0.823 |
| relative\_year | -0.130 | 0.030 | -0.189 | -0.070 |

##### Fig S3 (Fig 1b)

text\_coord <- wastewater\_conc %>%  
 summarise(y = max(value\_log, na.rm = TRUE), x\_max = max(effluent\_median\_log,  
 na.rm = TRUE), x\_min = min(effluent\_median\_log, na.rm = TRUE), x = mean(c(x\_max,  
 x\_min)))  
  
y\_coord <- text\_coord %>%  
 pull(y)  
  
x\_coord <- text\_coord %>%  
 pull(x)  
  
  
fig\_1b <- ggplot() + geom\_line(data = estimate\_data\_2, aes(x = effluent\_median\_log,  
 y = estimate\_\_), color = "#1c4595", linewidth = 1) + geom\_ribbon(data = estimate\_data\_2,  
 aes(x = effluent\_median\_log, ymin = lower\_\_, ymax = upper\_\_), fill = "#1c4595",  
 alpha = 0.1) + geom\_point(data = wastewater\_conc, aes(x = effluent\_median\_log,  
 y = value\_log), alpha = 0.4, shape = 21, color = "#01080A", fill = "#E76A24") +  
 annotate("text", x = x\_coord, y = y\_coord, label = paste0("Bayesian log-log Gaussian regression",  
 "\n", "Effect estimate: ", round(estimate\_2, 3), "\n", "CI: ", round(L95CI\_2,  
 3), " - ", round(H95CI\_2, 3)), hjust = 0, vjust = 1, size = 4) + labs(title = "Test vs effluent concentrations",  
 x = "Log median effluent concentration", y = "Log tested concentration") + theme\_clean() +  
 theme(legend.position = "bottom", legend.justification = "center")  
  
fig\_1b



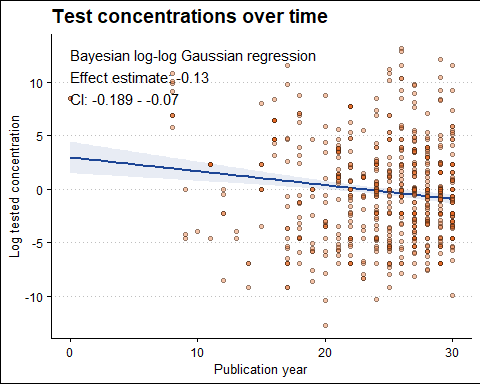
Saving figure

# ggsave(filename = paste0(fig\_wd, './figure-1b.pdf'), plot = fig\_1b, width =  
# 210, height = 297/1.5, # Specify the height of the plot units = 'mm')

##### Fig S4

Plot surface water concentrations by year

text\_coord <- wastewater\_conc %>%  
 summarise(y\_max = max(value\_log, na.rm = TRUE), x\_max = max(relative\_year, na.rm = TRUE),  
 x\_min = min(relative\_year, na.rm = TRUE), x\_mean = mean(c(x\_max, x\_min)))  
  
y\_coord <- text\_coord %>%  
 pull(y\_max)  
  
x\_coord <- text\_coord %>%  
 pull(x\_min)  
  
  
fig\_s4 <- ggplot() + geom\_line(data = estimate\_data\_year\_2, aes(x = relative\_year,  
 y = estimate\_\_), color = "#1c4595", linewidth = 1) + geom\_ribbon(data = estimate\_data\_year\_2,  
 aes(x = relative\_year, ymin = lower\_\_, ymax = upper\_\_), fill = "#1c4595", alpha = 0.1) +  
 geom\_point(data = wastewater\_conc, aes(x = relative\_year, y = value\_log), alpha = 0.4,  
 shape = 21, color = "#01080A", fill = "#E76A24") + annotate("text", x = x\_coord,  
 y = y\_coord, label = paste0("Bayesian log-log Gaussian regression", "\n", "Effect estimate: ",  
 round(estimate\_year\_2, 3), "\n", "CI: ", round(L95CI\_year\_2, 3), " - ", round(H95CI\_year\_2,  
 3)), hjust = 0, vjust = 1, size = 4) + labs(title = "Test concentrations over time",  
 x = "Publication year", y = "Log tested concentration") + theme\_clean() + theme(legend.position = "bottom",  
 legend.justification = "center")  
  
fig\_s4



## Concentration fold differences

Here we are summarising how many test were below the median surface water and effluent concentrations as well as the upper 95%CI. We are also calculating the mean-fold difference in concentrations.

conc\_summary\_overall <- all\_databases\_with\_sum %>%   
 dplyr::reframe(  
 eipaab\_n = length(unique\_row\_id),  
 n\_compounds\_sw = length(unique(compound\_name\_corrected[!is.na(surfacewater\_median)])),  
 n\_compounds\_ef = length(unique(compound\_name\_corrected[!is.na(effluent\_median)])),  
 n\_sw = sum(!is.na(under\_med\_surf)),  
 less\_med\_sw = sum(under\_med\_surf, na.rm = TRUE),  
 less\_hci\_sw = sum(under\_hci\_surf, na.rm = TRUE),  
 percent\_less\_med\_sw = (less\_med\_sw / n\_sw) \* 100,  
 percent\_less\_hic\_sw = (less\_hci\_sw / n\_sw) \* 100,  
 n\_ef = sum(!is.na(under\_med\_effl)),  
 less\_med\_ef = sum(under\_med\_effl, na.rm = TRUE),  
 less\_hci\_ef = sum(under\_hci\_effl, na.rm = TRUE),  
 percent\_less\_med\_ef = (less\_med\_ef / n\_ef) \* 100,  
 percent\_less\_hic\_ef = (less\_hci\_ef / n\_ef) \* 100,  
   
 # Median values  
 fdiff\_med\_sw = median(diff\_med\_surf, na.rm = TRUE),  
 fdiff\_hci\_sw = median(diff\_hic\_surf, na.rm = TRUE),  
 fdiff\_med\_ef = median(diff\_med\_effl, na.rm = TRUE),  
 fdiff\_hci\_ef = median(diff\_hic\_effl, na.rm = TRUE),  
   
 # Standard Deviation  
 sd\_fdiff\_med\_sw = sd(diff\_med\_surf, na.rm = TRUE),  
 sd\_fdiff\_hci\_sw = sd(diff\_hic\_surf, na.rm = TRUE),  
 sd\_fdiff\_med\_ef = sd(diff\_med\_effl, na.rm = TRUE),  
 sd\_fdiff\_hci\_ef = sd(diff\_hic\_effl, na.rm = TRUE)  
 ) %>%   
 dplyr::mutate(across(everything(), ~ ifelse(is.nan(.), NA, .))) # Convert NaNs to NAs

#### Results

For the 167 compounds that were present in the EIPAAB database and environmental detections (in UBA, NORMAN, or Wilkson) representing 767 tests, only 18.70% and 37.96% employed test concentrations lower then median and upper 95% credible interval of surface water concentrations, respectively (Table S5). Further, only 23.39% and 53.08% employed test concentrations lower then median and upper 95% credible interval of wastewater effluent (Table S5). On average, concentrations used in behavioural ecotoxicology were 43 times higher then median surf water concentrations, and 10 times higher then those in effluent.

##### Table S2

This is called Table 2 in the manuscript

**Table 2:** The number of exposures in the EIPAAB database that used concentrations less then median and upper 95% credible interval of surface water and effluent concentrations

surf <- conc\_summary\_overall %>%  
 dplyr::mutate(fdiff\_med\_sw = round(fdiff\_med\_sw, 2), fdiff\_hci\_sw = round(fdiff\_hci\_sw)) %>%  
 dplyr::select(n\_sw, less\_med\_sw, less\_hci\_sw, fdiff\_med\_sw, fdiff\_hci\_sw, percent\_less\_med\_sw,  
 percent\_less\_hic\_sw) %>%  
 dplyr::rename(total\_n = n\_sw, less\_med = less\_med\_sw, less\_hci = less\_hci\_sw,  
 fdiff\_med = fdiff\_med\_sw, fdiff\_hci = fdiff\_hci\_sw, percent\_less\_med = percent\_less\_med\_sw,  
 percent\_less\_hic = percent\_less\_hic\_sw) %>%  
 dplyr::mutate(matrix = "Surface water")  
  
effluent <- conc\_summary\_overall %>%  
 dplyr::mutate(fdiff\_med\_ef = round(fdiff\_med\_ef, 2), fdiff\_hci\_ef = round(fdiff\_hci\_ef,  
 2)) %>%  
 dplyr::select(n\_ef, less\_med\_ef, less\_hci\_ef, fdiff\_med\_ef, fdiff\_hci\_ef, percent\_less\_med\_ef,  
 percent\_less\_hic\_ef) %>%  
 dplyr::rename(total\_n = n\_ef, less\_med = less\_med\_ef, less\_hci = less\_hci\_ef,  
 fdiff\_med = fdiff\_med\_ef, fdiff\_hci = fdiff\_hci\_ef, percent\_less\_med = percent\_less\_med\_ef,  
 percent\_less\_hic = percent\_less\_hic\_ef) %>%  
 dplyr::mutate(matrix = "Effluent")  
  
table\_3 <- rbind(surf, effluent) %>%  
 dplyr::select(matrix, everything()) %>%  
 dplyr::mutate(percent\_less\_med = round(percent\_less\_med, 2), percent\_less\_hic = round(percent\_less\_hic,  
 2)) %>%  
 dplyr::rename(Matrix = matrix, `Total tests` = total\_n, `Dose less than median` = less\_med,  
 `Dose less than upper 95%CI` = less\_hci, `Mean-fold differene from median` = fdiff\_med,  
 `Mean-fold differene from upper 95%CI` = fdiff\_hci, `Percent less than median (%)` = percent\_less\_med,  
 `Percent less than upper 95%CI (%)` = percent\_less\_hic) %>%  
 gt() %>%  
 fmt\_number(columns = "Total tests", decimals = 0) %>%  
 cols\_align(align = "center", columns = everything())  
  
table\_3

| Matrix | Total tests | Dose less than median | Dose less than upper 95%CI | Mean-fold differene from median | Mean-fold differene from upper 95%CI | Percent less than median (%) | Percent less than upper 95%CI (%) |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Surface water | 706 | 133 | 269 | 42.70 | 2.00 | 18.84 | 38.10 |
| Effluent | 714 | 167 | 379 | 10.11 | 0.78 | 23.39 | 53.08 |

Now we will make a summary for each compound

sample\_counts <- all\_databases\_with\_sum %>%  
 dplyr::select(compound\_name\_corrected, surfacewater\_n, effluent\_n) %>%  
 group\_by(compound\_name\_corrected) %>%  
 dplyr::slice(1) %>%  
 ungroup()  
  
under\_levels <- all\_databases\_with\_sum %>%  
 dplyr::group\_by(compound\_name\_corrected) %>%  
 dplyr::reframe(eipaab\_n = length(unique\_row\_id), n\_sw = sum(!is.na(under\_med\_surf)),  
 less\_med\_sw = sum(under\_med\_surf, na.rm = TRUE), less\_hci\_sw = sum(under\_hci\_surf,  
 na.rm = TRUE), prop\_less\_med\_sw = less\_med\_sw/n\_sw, prop\_less\_hic\_sw = less\_hci\_sw/n\_sw,  
 n\_ef = sum(!is.na(under\_med\_effl)), less\_med\_ef = sum(under\_med\_effl, na.rm = TRUE),  
 less\_hci\_ef = sum(under\_hci\_effl, na.rm = TRUE), prop\_less\_med\_ef = less\_med\_ef/n\_ef,  
 prop\_less\_hic\_ef = less\_hci\_ef/n\_ef, fdiff\_med\_sw = median(diff\_med\_surf,  
 na.rm = TRUE), fdiff\_hci\_sw = median(diff\_hic\_surf, na.rm = TRUE), fdiff\_med\_ef = median(diff\_med\_effl,  
 na.rm = TRUE), fdiff\_hci\_ef = median(diff\_hic\_effl, na.rm = TRUE), ) %>%  
 dplyr::left\_join(., sample\_counts, by = "compound\_name\_corrected") %>%  
 dplyr::mutate(across(everything(), ~ifelse(is.nan(.), NA, .)))

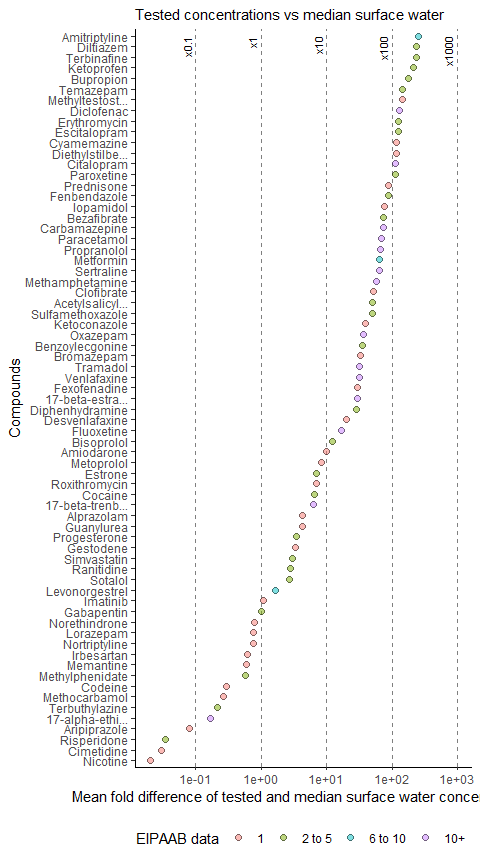
#### Surface water

##### Fig S5 (Fig 2)

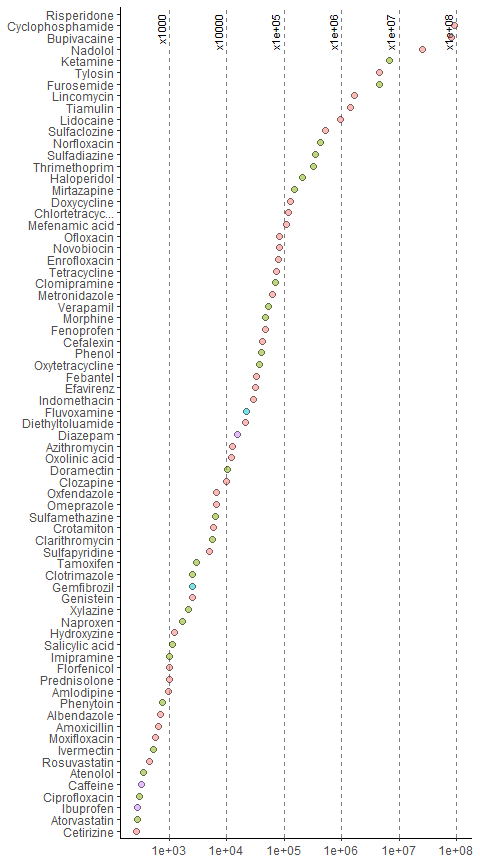
Lets plot the mean-fold difference Building the plot in two halves so that it can fit in our manuscript

**Fig 2a.** The median fold difference in tested concentrations in EIPAAB and median surface water concentration (i.e. tested dose / median surface water concentration)

plot\_left



plot\_right



***Note:*** *the size of the points represent the number of occurrence in EIPAAB, and colour of the point represents the number of detections in the environmental databases*

Save the figure

# invisible(fig\_2 <- plot\_left | plot\_right) ggsave(filename = paste0(fig\_wd,  
# './figure-2.pdf'), plot = fig\_2, width = 210, height = 297/1.25, # Specify  
# the height of the plot units = 'mm')

#### Waste water

##### Fig S6

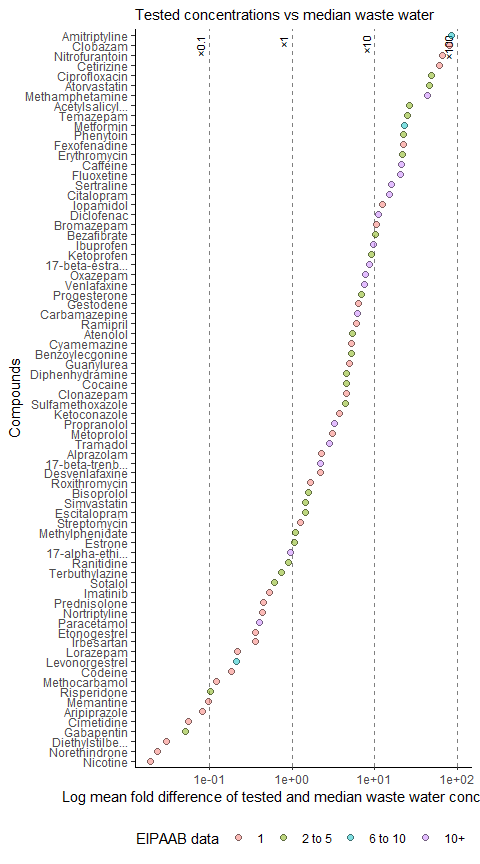
Let’s plot mean-fold difference fo effluent

Building the plot in two halves so that it can fit in our manuscript

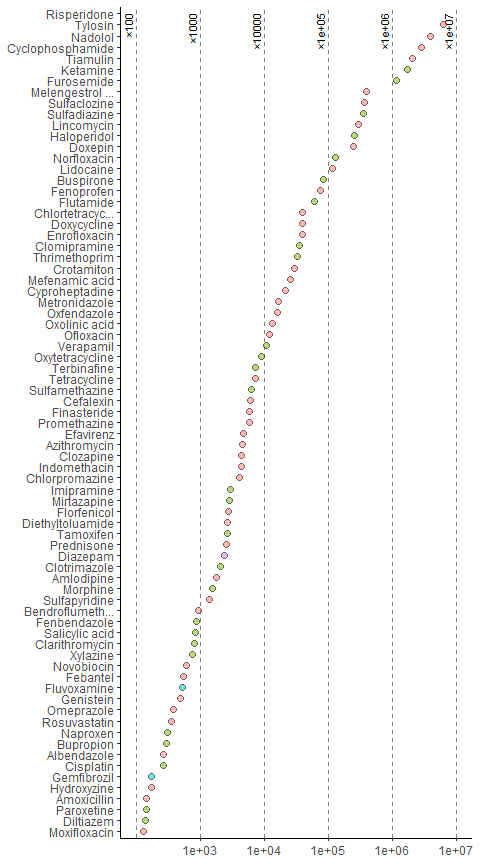
plot\_ef\_data <- under\_levels %>%   
 dplyr::filter(!is.na(fdiff\_med\_ef)) %>%   
 dplyr::arrange(desc(fdiff\_med\_ef)) %>%   
 dplyr::mutate(  
 order = 1:nrow(.),  
 compound\_name\_corrected = sentence\_case(compound\_name\_corrected),  
 compound\_name\_short = if\_else(  
 nchar(as.character(compound\_name\_corrected)) > 13,   
 str\_trunc(compound\_name\_corrected, 16, ellipsis = "..."),   
 compound\_name\_corrected  
 ),  
 compound\_name\_short = factor(compound\_name\_short, levels = rev(compound\_name\_short)), # Fixed the parentheses  
 compound\_name\_short = factor(compound\_name\_short, levels = rev(compound\_name\_short)), # Fixed the parentheses  
 eipaab\_n\_cat = case\_when(  
 eipaab\_n == 1 ~ "1",  
 eipaab\_n >= 2 & eipaab\_n <= 5 ~ "2 to 5",  
 eipaab\_n >= 6 & eipaab\_n <= 10 ~ "6 to 10",  
 eipaab\_n > 10 ~ "10+",  
 TRUE ~ "ERROR" # This catches unexpected values  
 ),  
 eipaab\_n\_cat = factor(eipaab\_n\_cat, levels = c("1", "2 to 5", "6 to 10", "10+"))  
 )  
  
vline\_data\_left <- data.frame(xintercept = log(10^(-1:2)),  
 xlabs = as.character(10^(-1:2))) %>%   
 dplyr::mutate(xlabs = paste0("×",xlabs))  
  
vline\_data\_right <- data.frame(xintercept = log(10^(2:7)),  
 xlabs = as.character(10^(2:7))) %>%   
 dplyr::mutate(xlabs = paste0("×",xlabs))  
  
  
n\_half <- ceiling(nrow(plot\_ef\_data) / 2)  
  
plot\_ef\_data\_right <- plot\_ef\_data %>%   
 dplyr::slice(1:n\_half)  
  
plot\_ef\_data\_left <- plot\_ef\_data %>%   
 dplyr::slice((n\_half + 1):nrow(plot\_ef\_data))  
  
plot\_left <- plot\_ef\_data\_left %>%   
 ggplot(aes(x = log(fdiff\_med\_ef), y = compound\_name\_short, fill = eipaab\_n\_cat)) +  
 geom\_vline(data = vline\_data\_left, aes(xintercept = xintercept),   
 linetype = "dashed", colour = "black", alpha = 0.5) +  
 geom\_text(data = vline\_data\_left, aes(x = xintercept, y = "Amitriptyline", label = xlabs),   
 inherit.aes = FALSE, angle = 90, vjust = -0.5, hjust = 1, size = 3) +  
 geom\_point(alpha = 0.5, shape = 21, size = 2) +  
 theme\_classic() +  
   
 # Adjusted log scale with 10-fold increments  
 scale\_x\_continuous(  
 breaks = log(10^(-1:2)), # Explicitly set log-scale breaks  
 labels = function(x) format(exp(x), scientific = TRUE)  
 ) +  
  
 labs(  
 subtitle = "Tested concentrations vs median waste water",  
 x = "Log mean fold difference of tested and median waste water concentrations",  
 y = "Compounds",  
 fill = "EIPAAB data"  
 ) +  
 theme(  
 legend.position = "bottom",  
 legend.justification = "right"  
 )  
  
  
plot\_right <- plot\_ef\_data\_right %>%   
 ggplot(aes(x = log(fdiff\_med\_ef), y = compound\_name\_short, fill = eipaab\_n\_cat)) +  
 geom\_vline(data = vline\_data\_right, aes(xintercept = xintercept),   
 linetype = "dashed", colour = "black", alpha = 0.5) +  
 geom\_text(data = vline\_data\_right, aes(x = xintercept, y = "Risperidone", label = xlabs),   
 inherit.aes = FALSE, angle = 90, vjust = -0.5, hjust = 1, size = 3) +  
 geom\_point(alpha = 0.5, shape = 21, size = 2) +  
 theme\_classic() +  
   
 # Adjusted log scale: removes 10^2 while keeping 10^3 to 10^7  
 scale\_x\_continuous(  
 breaks = log(c(10^(3:7))), # Explicitly setting ticks without 10^2  
 labels = function(x) format(exp(x), scientific = TRUE)  
 ) +  
  
 theme(  
 axis.title = element\_blank(), # Removes axis labels  
 legend.position = "none"  
 )

**Fig S6**: The median fold difference in tested concentrations in EIPAAB and median waste water concentration (i.e. tested dose / median waste water concentration)

plot\_left



plot\_right



***Note:*** *the size of the points represent the number of occurrence in EIPAAB, and colour of the point represents the number of detections in the environmental databases*

Save the figure

# invisible(fig\_S6 <- plot\_left | plot\_right) ggsave(filename = paste0(fig\_wd,  
# './figure-S5.pdf'), plot = fig\_S5, width = 210/2, height = 297, # Specify the  
# height of the plot units = 'mm')

Looking at the number of compounds in the database that have data at concentrations below median and upper 95 CI

##### Table S3

**Table S3:** The percentage of compounds in the EIPAAB database that have data at concentrations less then median and upper 95% credible interval of surface water and effluent concentrations

compound\_sum\_table\_surf <- under\_levels %>%  
 dplyr::reframe(total\_compounds = sum(!is.na(prop\_less\_med\_sw)), count\_less\_med\_sw = sum(prop\_less\_med\_sw ==  
 0, na.rm = TRUE), count\_less\_hic\_sw = sum(prop\_less\_hic\_sw == 0, na.rm = TRUE),  
 percent\_less\_med\_sw = (count\_less\_med\_sw/total\_compounds) \* 100, percent\_less\_hic\_sw = (count\_less\_hic\_sw/total\_compounds) \*  
 100, `Percent with no data under median concentration` = paste0(round(percent\_less\_med\_sw,  
 2), "%", " (", count\_less\_med\_sw, " of ", total\_compounds, ")"), `Percent with no data under upper 95 CI concentration` = paste0(round(percent\_less\_hic\_sw,  
 2), "%", " (", count\_less\_hic\_sw, " of ", total\_compounds, ")")) %>%  
 dplyr::mutate(Matrix = "Surface water") %>%  
 dplyr::select(Matrix, "Percent with no data under median concentration", "Percent with no data under upper 95 CI concentration")  
  
compound\_sum\_table\_eff <- under\_levels %>%  
 dplyr::reframe(total\_compounds = sum(!is.na(prop\_less\_med\_ef)), count\_less\_med\_ef = sum(prop\_less\_med\_ef ==  
 0, na.rm = TRUE), count\_less\_hic\_ef = sum(prop\_less\_hic\_ef == 0, na.rm = TRUE),  
 percent\_less\_med\_ef = (count\_less\_med\_ef/total\_compounds) \* 100, percent\_less\_hic\_ef = (count\_less\_hic\_ef/total\_compounds) \*  
 100, `Percent with no data under median concentration` = paste0(round(percent\_less\_med\_ef,  
 2), "%", " (", count\_less\_med\_ef, " of ", total\_compounds, ")"), `Percent with no data under upper 95 CI concentration` = paste0(round(percent\_less\_hic\_ef,  
 2), "%", " (", count\_less\_hic\_ef, " of ", total\_compounds, ")")) %>%  
 dplyr::mutate(Matrix = "Effluent") %>%  
 dplyr::select(Matrix, "Percent with no data under median concentration", "Percent with no data under upper 95 CI concentration")  
  
compound\_sum\_table <- rbind(compound\_sum\_table\_surf, compound\_sum\_table\_eff)  
  
compound\_sum\_table %>%  
 gt() %>%  
 fmt\_number(columns = c("Percent with no data under median concentration", "Percent with no data under upper 95 CI concentration"),  
 decimals = 0) %>%  
 cols\_align(align = "center", columns = everything())

| Matrix | Percent with no data under median concentration | Percent with no data under upper 95 CI concentration |
| --- | --- | --- |
| Surface water | 74.82% (104 of 139) | 50.36% (70 of 139) |
| Effluent | 64.86% (96 of 148) | 44.59% (66 of 148) |

## Occurance

### Occurance summary data

Making a summary data frame for each compound, **all\_surfacewater**, that includes: (1) the number of samples, (2) the number of positive detections (based on single samples, not summary statistics), (3) the rank order of detections, (4) the percentage of positive detections (only for compounds with more then 10 sample units), and, (5) the rank order of percentage of positive detection.

all\_surfacewater <- all\_databases %>%  
 dplyr::filter(source != "eipaab", matrix\_group == "surfacewater") %>%  
 dplyr::group\_by(compound\_name\_corrected) %>%  
 dplyr::reframe(combind\_samples = sum(n\_units\_est, na.rm = TRUE), combind\_rows = length(unique\_row\_id),  
 combind\_singles = sum(concentration\_type == "single", na.rm = TRUE), combind\_detection = sum(value >  
 0, na.rm = TRUE), combind\_single\_detection = sum(value > 0 & concentration\_type ==  
 "single", na.rm = TRUE), combind\_percent\_detection = if\_else(combind\_singles >  
 10, (combind\_single\_detection/combind\_singles) \* 100, NA)) %>%  
 ungroup() %>%  
 dplyr::mutate(combind\_rank\_samples = rank(-combind\_samples, ties.method = "max",  
 na.last = "keep"), combind\_rank\_detection = rank(-combind\_detection, ties.method = "max",  
 na.last = "keep"), combind\_rank\_detection\_percent = rank(-combind\_percent\_detection,  
 ties.method = "max", na.last = "keep"))

There’s a total of 1654 compounds in this surface water summary

all\_surfacewater %>%  
 dplyr::distinct(compound\_name\_corrected) %>%  
 nrow(.)

## [1] 1650

Making occurrence count in EIPAAB, 📊 **eipaab\_counts**.

all\_surfacewater\_list <- all\_surfacewater %>%  
 dplyr::distinct(compound\_name\_corrected) %>%  
 dplyr::pull(compound\_name\_corrected)  
  
eipaab\_counts <- all\_databases %>%  
 dplyr::filter(source == "eipaab" & compound\_name\_corrected %in% all\_surfacewater\_list) %>%  
 dplyr::group\_by(compound\_name\_corrected) %>%  
 dplyr::reframe(eipaab\_n = sum(n\_units\_est, na.rm = TRUE), eipaab\_n\_log = log(eipaab\_n)) %>%  
 ungroup() %>%  
 dplyr::mutate(eipaab\_rank = rank(-eipaab\_n, ties.method = "max", na.last = "keep")) %>%  
 tidyr::complete(., compound\_name\_corrected)

Combining the data together to make 📊 **test\_surfacewater**

test\_surfacewater <- left\_join(all\_surfacewater, eipaab\_counts, by = "compound\_name\_corrected") %>%  
 dplyr::mutate(rank\_samples\_dif = eipaab\_rank - combind\_rank\_samples, rank\_detection\_dif = eipaab\_rank -  
 combind\_rank\_detection, rank\_detection\_percent\_dif = eipaab\_rank - combind\_rank\_detection\_percent) %>%  
 dplyr::arrange(desc(rank\_detection\_dif), desc(combind\_detection)) %>%  
 dplyr::mutate(rank\_detection\_dif = 1:nrow(.)) %>%  
 dplyr::arrange(desc(rank\_detection\_percent\_dif), desc(combind\_percent\_detection)) %>%  
 dplyr::mutate(rank\_detection\_percent\_dif = 1:nrow(.))  
  
uba\_rank\_detection\_max <- test\_surfacewater %>%  
 dplyr::arrange(desc(combind\_rank\_detection)) %>%  
 dplyr::slice(1) %>%  
 dplyr::pull(combind\_rank\_detection)  
  
uba\_rank\_detection\_percent\_max <- test\_surfacewater %>%  
 dplyr::arrange(desc(combind\_rank\_detection\_percent)) %>%  
 dplyr::slice(1) %>%  
 dplyr::pull(combind\_rank\_detection\_percent)

Figuring out what the max rank occurrence and positive detections could be.

uba\_rank\_detection\_max <- test\_surfacewater %>%  
 dplyr::arrange(desc(combind\_rank\_detection)) %>%  
 dplyr::slice(1) %>%  
 dplyr::pull(combind\_rank\_detection)  
  
uba\_rank\_detection\_percent\_max <- test\_surfacewater %>%  
 dplyr::arrange(desc(combind\_rank\_detection\_percent)) %>%  
 dplyr::slice(1) %>%  
 dplyr::pull(combind\_rank\_detection\_percent)  
  
print(paste0("The max rank in terms of the number of occurrences is ", uba\_rank\_detection\_max,  
 ". The max rank for the percentage of postive detections is ", uba\_rank\_detection\_percent\_max))

## [1] "The max rank in terms of the number of occurrences is 1650. The max rank for the percentage of postive detections is 1424"

#### Table S4

The 10 most common compounds in EIPAAB versus occurrence and detection frequency in UBA surface waters UBA samples.

test\_surfacewater %>%  
 dplyr::arrange(desc(eipaab\_n)) %>%  
 dplyr::slice(1:10) %>%  
 dplyr::mutate(compound\_name = sentence\_case(compound\_name\_corrected), combind\_percent\_detection = round(combind\_percent\_detection,  
 2)) %>%  
 dplyr::select(compound\_name, eipaab\_n, combind\_samples, combind\_detection, combind\_percent\_detection,  
 eipaab\_rank, combind\_rank\_detection, combind\_rank\_detection\_percent) %>%  
 dplyr::rename(Name = compound\_name, `EIPAAB tests` = eipaab\_n, `Environmental samples` = combind\_samples,  
 `All detections` = combind\_detection, `Precent single detections` = combind\_percent\_detection,  
 `EIPAAB rank (1–870)` = eipaab\_rank, `Rank total detections (1–1654)` = combind\_rank\_detection,  
 `Rank percent detections (1–1428)\*` = combind\_rank\_detection\_percent) %>%  
 gt() %>%  
 fmt\_number(columns = c("Environmental samples", "All detections"), decimals = 0) %>%  
 cols\_align(align = "center", columns = everything())

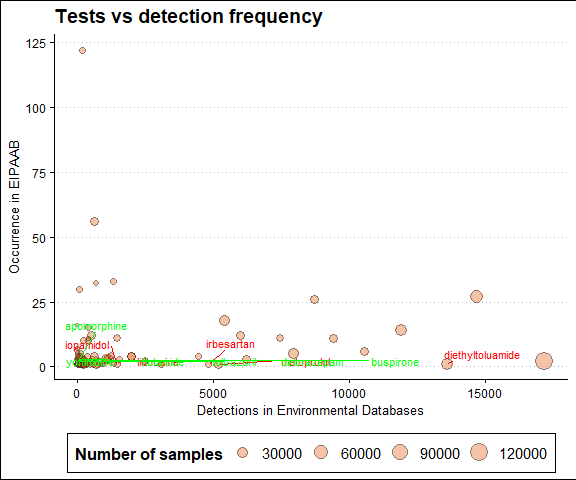
| Name | EIPAAB tests | Environmental samples | All detections | Precent single detections | EIPAAB rank (1–870) | Rank total detections (1–1654) | Rank percent detections (1–1428)\* |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Fluoxetine | 122 | 6,642 | 197 | 2.92 | 1 | 185 | 367 |
| 17-alpha-ethinylestradiol | 56 | 18,120 | 626 | 3.04 | 2 | 113 | 361 |
| Venlafaxine | 33 | 5,333 | 1,308 | 46.68 | 3 | 77 | 79 |
| Citalopram | 32 | 3,471 | 697 | 27.62 | 4 | 108 | 125 |
| Sertraline | 30 | 4,928 | 56 | 1.04 | 5 | 299 | 471 |
| Carbamazepine | 27 | 56,084 | 14,637 | 54.90 | 6 | 16 | 65 |
| Oxazepam | 26 | 18,858 | 8,704 | 49.00 | 7 | 23 | 74 |
| Ibuprofen | 18 | 27,447 | 5,389 | 25.04 | 8 | 30 | 134 |
| 17-beta-trenbolone | 16 | 247 | 4 | 0.00 | 9 | 647 | 1424 |
| Diazepam | 15 | 7,192 | 399 | 6.68 | 10 | 143 | 261 |

*Note: the rank for total percentage detections does not range from 1 to the number of compounds (i.e. 1–1428) because percentage detections were only calculated where the number of single samples are greater than ten*

#### Fig S7

There’s massive skew in occurrence for specific compounds in both the UBA and EIPAAB database, in terms of modelling it, there is likely going to be issues with overdispersion. We will use a negative binomial distribution.

max\_rank <- test\_surfacewater %>%  
 dplyr::filter(!is.na(eipaab\_n) & !is.na(rank\_detection\_dif)) %>%  
 dplyr::summarise(max = max(rank\_detection\_dif)) %>%  
 pull(max)  
  
max\_rank\_5 <- max\_rank - 5  
  
fig\_s7 <- test\_surfacewater %>%  
 dplyr::filter(!is.na(eipaab\_n) & !is.na(rank\_detection\_dif)) %>%  
 dplyr::arrange(rank\_detection\_dif) %>%  
 dplyr::mutate(is\_top5 = rank\_detection\_dif %in% 1:5) %>%  
 ggplot() + geom\_point(aes(x = combind\_detection, y = eipaab\_n, size = combind\_samples),  
 alpha = 0.4, shape = 21, color = "#01080A", fill = "#E76A24") + geom\_text\_repel(data = test\_surfacewater %>%  
 filter(!is.na(eipaab\_n) & !is.na(rank\_detection\_dif) & rank\_detection\_dif %in%  
 1:5), aes(x = combind\_detection, y = eipaab\_n, label = compound\_name\_corrected),  
 colour = "red", size = 3, box.padding = 0.7, min.segment.length = 0, hjust = 0,  
 vjust = 5) + geom\_text\_repel(data = test\_surfacewater %>%  
 filter(!is.na(eipaab\_n) & !is.na(rank\_detection\_dif) & rank\_detection\_dif %in%  
 max\_rank\_5:max\_rank), aes(x = combind\_percent\_detection, y = eipaab\_n, label = compound\_name\_corrected),  
 colour = "green", size = 3, box.padding = 0.7, min.segment.length = 0, hjust = 0,  
 vjust = 5) + scale\_fill\_gradientn(colours = colorspace::diverge\_hcl(7)) + scale\_colour\_manual(values = c(`TRUE` = "#01080A",  
 `FALSE` = "#01080A"), guide = "none") + labs(title = "Tests vs detection frequency",  
 x = "Detections in Environmental Databases", y = "Occurrence in EIPAAB", size = "Number of samples") +  
 theme\_clean() + theme(legend.position = "bottom", legend.justification = "center")  
  
fig\_s7

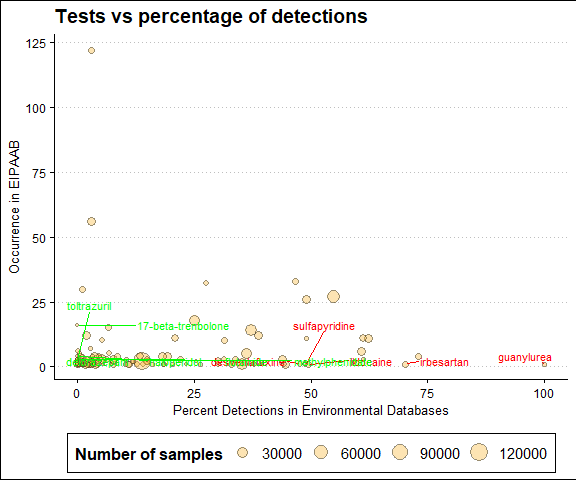


***Note:*** *the size of the points represents the total number of samples in the environmental databases. The five marked points represent those that are under represented in EIPAAB given the number of detections (i.e. rank EIPAAB - rank environmental detections).*

#### Fig S8

There’s massive skew in occurrence for specific compounds in both the UBA and EIPAAB database, in terms of modelling it, there is likely going to be issues with overdispersion. We will use a negative binomial distribution.

max\_rank <- test\_surfacewater %>%  
 dplyr::filter(!is.na(eipaab\_n) & !is.na(combind\_percent\_detection)) %>%  
 dplyr::summarise(max = max(rank\_detection\_percent\_dif)) %>%  
 pull(max)  
  
max\_rank\_5 <- max\_rank - 5  
  
  
fig\_s8 <- test\_surfacewater %>%  
 dplyr::filter(!is.na(eipaab\_n) & !is.na(combind\_percent\_detection)) %>%  
 dplyr::arrange(rank\_detection\_percent\_dif) %>%  
 dplyr::mutate(is\_top5 = rank\_detection\_percent\_dif %in% 1:5) %>%  
 ggplot() + geom\_point(aes(x = combind\_percent\_detection, y = eipaab\_n, size = combind\_samples),  
 alpha = 0.4, shape = 21, color = "#01080A", fill = "#FBBC42") + geom\_text\_repel(data = test\_surfacewater %>%  
 filter(!is.na(eipaab\_n & !is.na(combind\_percent\_detection)), rank\_detection\_percent\_dif %in%  
 1:5), aes(x = combind\_percent\_detection, y = eipaab\_n, label = compound\_name\_corrected),  
 colour = "red", size = 3, box.padding = 0.7, min.segment.length = 0, hjust = 0,  
 vjust = 5) + geom\_text\_repel(data = test\_surfacewater %>%  
 filter(!is.na(eipaab\_n & !is.na(combind\_percent\_detection)), rank\_detection\_percent\_dif %in%  
 max\_rank\_5:max\_rank), aes(x = combind\_percent\_detection, y = eipaab\_n, label = compound\_name\_corrected),  
 colour = "green", size = 3, box.padding = 0.7, min.segment.length = 0, hjust = 0,  
 vjust = 5) + scale\_fill\_gradientn(colours = colorspace::diverge\_hcl(7)) + scale\_colour\_manual(values = c(`TRUE` = "#01080A",  
 `FALSE` = "#01080A"), guide = "none") + labs(title = "Tests vs percentage of detections",  
 x = "Percent Detections in Environmental Databases", y = "Occurrence in EIPAAB",  
 size = "Number of samples") + theme\_clean() + theme(legend.position = "bottom",  
 legend.justification = "center")  
  
fig\_s8



***Note:*** *the size of the points represents the total number of samples in the environmental databases. The five marked points represent those that are under represented in EIPAAB given the percentage of positive detections (i.e. rank EIPAAB - rank environmental detections).*

### Temporal trends data

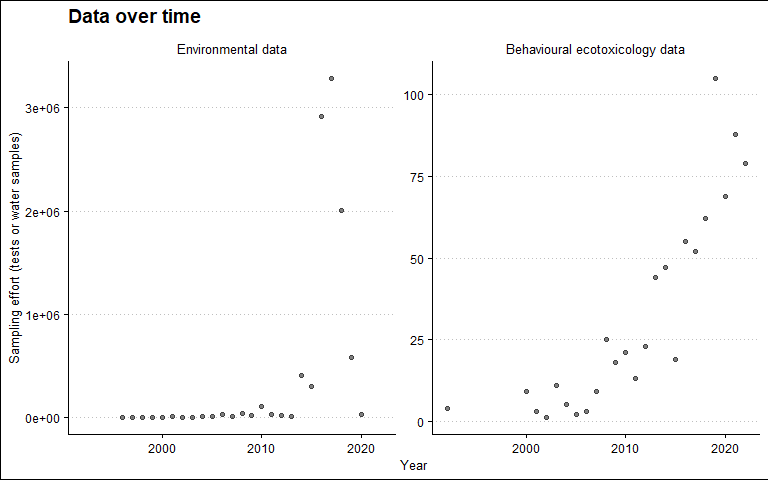
Here we will look at general temporal trends using the EIPAAB, UBA and NORMAN data.

temporal\_trends <- all\_databases %>%  
 dplyr::filter(source != "wilkson") %>%  
 dplyr::mutate(group\_type = if\_else(source == "eipaab", "test", "environmental")) %>%  
 dplyr::group\_by(group\_type, year) %>%  
 dplyr::reframe(n\_samples = sum(n\_units\_est, na.rm = TRUE), n\_compounds = length(unique(compound\_name\_corrected)),  
 rel\_compounds = n\_compounds/n\_samples, compound\_list = list(unique(compound\_name\_corrected))) %>%  
 tidyr::complete(group\_type, year) %>%  
 dplyr::filter(!is.na(year)) %>%  
 dplyr::group\_by(group\_type) %>%  
 dplyr::mutate(compound\_list = purrr::map(compound\_list, ~if (is.null(.x)) character(0) else .x),  
 cumulative\_compounds = purrr::accumulate(compound\_list, ~union(.x, .y)) %>%  
 purrr::map\_int(length), n\_samples\_z = scale(n\_samples)) %>%  
 dplyr::ungroup()

#### Fig S9

Let’s look at the total number of samples per year in these combined databases

fig\_s9 <- temporal\_trends %>%  
 # dplyr::filter(year > 1999 & year < 2021) %>%  
ggplot(aes(x = year, y = n\_samples)) + geom\_point(alpha = 0.5) + facet\_wrap(~group\_type,  
 scales = "free", labeller = labeller(group\_type = c(environmental = "Environmental data",  
 test = "Behavioural ecotoxicology data"))) + labs(title = "Data over time",  
 x = "Year", y = "Sampling effort (tests or water samples)") + theme\_clean() +  
 theme(legend.position = "bottom", legend.justification = "center")  
  
fig\_s9



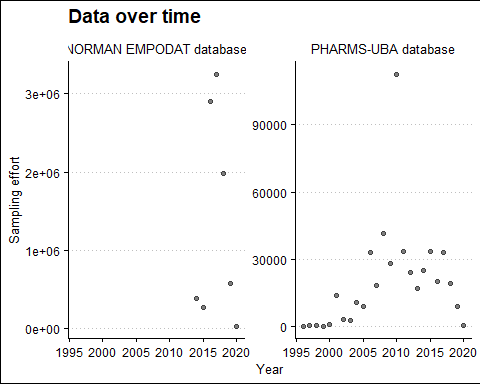
For a clearer temporal trend we look within the NORMAN and UBA-PHAMA databases separately.

Making a new dataset just for NORMAN EMPODAT and PHARMS-UBA database

uba\_norman\_temporal\_trends <- all\_databases %>%  
 dplyr::filter(source == "uba" | source == "norman") %>%  
 dplyr::group\_by(source, year) %>%  
 dplyr::reframe(n\_samples = sum(n\_units\_est, na.rm = TRUE), n\_compounds = length(unique(compound\_name\_corrected)),  
 rel\_compounds = n\_compounds/n\_samples, compound\_list = list(unique(compound\_name\_corrected))) %>%  
 tidyr::complete(source, year) %>%  
 dplyr::filter(!is.na(year)) %>%  
 dplyr::group\_by(source) %>%  
 dplyr::mutate(compound\_list = purrr::map(compound\_list, ~if (is.null(.x)) character(0) else .x),  
 cumulative\_compounds = purrr::accumulate(compound\_list, ~union(.x, .y)) %>%  
 purrr::map\_int(length), n\_samples\_z = scale(n\_samples)) %>%  
 dplyr::ungroup()

#### Fig S10

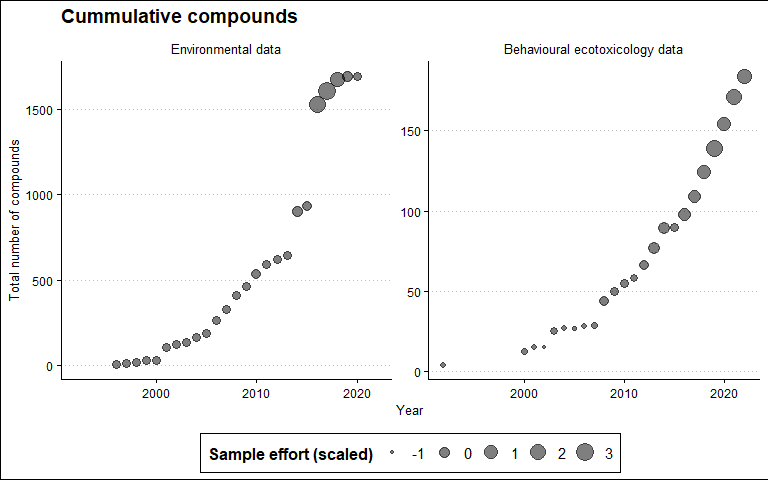
fig\_s10 <- uba\_norman\_temporal\_trends %>%  
 # dplyr::filter(year > 1999 & year < 2021) %>%  
ggplot(aes(x = year, y = n\_samples)) + geom\_point(alpha = 0.5) + facet\_wrap(~source,  
 scales = "free", labeller = labeller(source = c(uba = "PHARMS-UBA database",  
 norman = "NORMAN EMPODAT database"))) + labs(title = "Data over time", x = "Year",  
 y = "Sampling effort") + theme\_clean() + theme(legend.position = "bottom", legend.justification = "center")  
  
fig\_s10



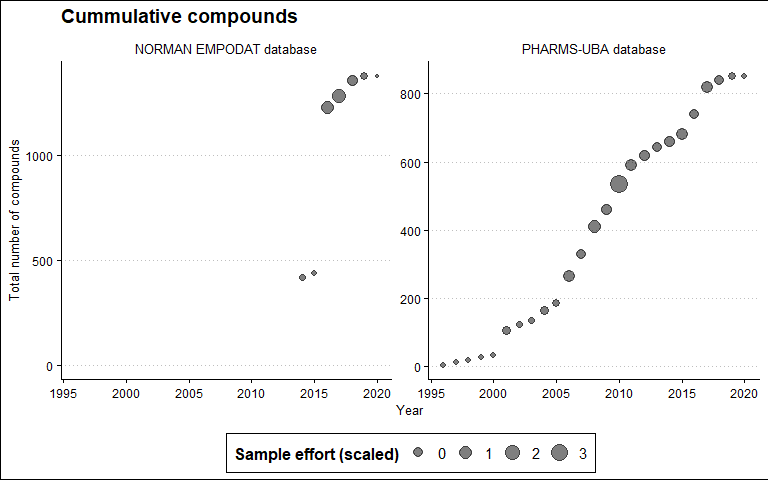
#### Fig S11

Now the cumulative number of compounds over time.

fig\_s11 <- temporal\_trends %>%  
 # dplyr::filter(year > 1999 & year < 2021) %>%  
ggplot(aes(x = year, y = cumulative\_compounds, size = n\_samples\_z)) + geom\_point(alpha = 0.5) +  
 facet\_wrap(~group\_type, scales = "free", labeller = labeller(group\_type = c(environmental = "Environmental data",  
 test = "Behavioural ecotoxicology data"))) + labs(title = "Cummulative compounds",  
 x = "Year", y = "Total number of compounds", size = "Sample effort (scaled)") +  
 theme\_clean() + theme(legend.position = "bottom", legend.justification = "center")  
  
fig\_s11



uba\_norman\_temporal\_trends %>%  
 # dplyr::filter(year > 1999 & year < 2021) %>%  
ggplot(aes(x = year, y = cumulative\_compounds, size = n\_samples\_z)) + geom\_point(alpha = 0.5) +  
 facet\_wrap(~source, scales = "free", labeller = labeller(source = c(uba = "PHARMS-UBA database",  
 norman = "NORMAN EMPODAT database"))) + labs(title = "Cummulative compounds",  
 x = "Year", y = "Total number of compounds", size = "Sample effort (scaled)") +  
 theme\_clean() + theme(legend.position = "bottom", legend.justification = "center")



## Modeling

### Total occruance

Here we will build a simple Bayesian negative binomial (log-link) regression model to look at the relationship between the two parameters.

#### Model structure

detection\_negbin\_str <- bf(eipaab\_n ~ combind\_detection, family = negbinomial())

These are the default priors.

suppressWarnings(get\_prior(detection\_negbin\_str, data = test\_surfacewater, family = negbinomial()))

## prior class coef group resp dpar nlpar lb ub  
## (flat) b   
## (flat) b combind\_detection   
## student\_t(3, 0, 2.5) Intercept   
## inv\_gamma(0.4, 0.3) shape 0   
## source  
## default  
## (vectorized)  
## default  
## default

Run model. This has been hashed out as I have saved the model, and will re-load it instead of re-running it.

# detection\_negbin\_mod <- brm(detection\_negbin\_str, data = test\_surfacewater,  
# cores = 4, chains = 4, #prior = mod\_priors, #use defaults warmup = 1000, seed  
# = 20250418, thin = 2, iter = 8000, control = list(max\_treedepth = 20,  
# adapt\_delta = 0.9999), save\_pars = save\_pars(all=TRUE), sample\_prior = TRUE,  
# file = paste0(mods\_wd, './detection\_negbin\_mod')) print('Model complete')

#### Re-load models

Reload the model if necessary.

detection\_negbin\_mod <- readRDS(file = paste0(mods\_wd, "./detection\_negbin\_mod.rds"))

#### Model diagnostics

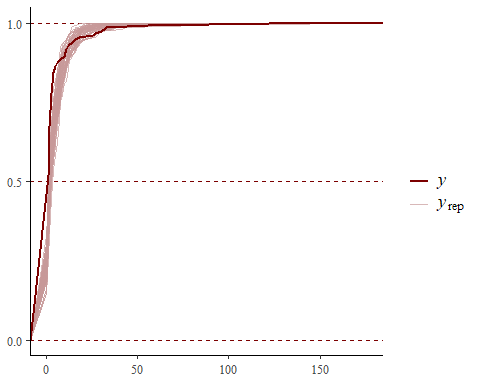
The R2 value is very low, but should be fine for the purposes of this investigation.

performance::r2\_bayes(detection\_negbin\_mod, robust = FALSE, ci = 0.95)

## # Bayesian R2 with Compatibility Interval  
##   
## Conditional R2: 0.077 (95% CI [3.720e-07, 0.255])

The mode generally follows the observed data, but isn’t the best fit. Again, this should be fine for the purposes of this investigation.

color\_scheme\_set("red")  
brms::pp\_check(detection\_negbin\_mod, ndraws = 50, type = "ecdf\_overlay")



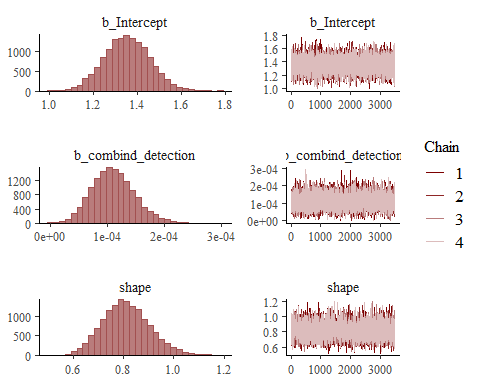
Checking R-hat values to confirm model convergence and ESS fo estimates.

print(summary(detection\_negbin\_mod, prob = 0.95)) #All Rhat = 1 (good).

## Family: negbinomial   
## Links: mu = log; shape = identity   
## Formula: eipaab\_n ~ combind\_detection   
## Data: test\_surfacewater (Number of observations: 163)   
## Draws: 4 chains, each with iter = 8000; warmup = 1000; thin = 2;  
## total post-warmup draws = 14000  
##   
## Regression Coefficients:  
## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS  
## Intercept 1.35 0.10 1.15 1.56 1.00 8541 9244  
## combind\_detection 0.00 0.00 0.00 0.00 1.00 13563 10167  
##   
## Further Distributional Parameters:  
## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS  
## shape 0.81 0.09 0.64 1.01 1.00 7742 7445  
##   
## Draws were sampled using sampling(NUTS). For each parameter, Bulk\_ESS  
## and Tail\_ESS are effective sample size measures, and Rhat is the potential  
## scale reduction factor on split chains (at convergence, Rhat = 1).

Diagnostic plots look fine. A smaller shape value indicates greater overdispersion (variance is much larger than the mean).

color\_scheme\_set("red")  
plot(detection\_negbin\_mod)



#### Model estimates

Pulling the conditional effect estimates from the model. The estimate corresponds to the effect of the predictor on the log of the expected response variable. A 1-unit increase (i.e. 1 detection) in the number of environmental detections increases the expected occurrence in EIPAAB database by 0.011% (i.e. (exp(0.0001105696)-1)×100 = 0.01105757). In other words, for each 1000 detections we would expect a 5% increase in test in the EIPAAB database.

# Extract estimates  
esitmates\_1 <- conditional\_effects(detection\_negbin\_mod)  
estimate\_data\_1 <- esitmates\_1$combind\_detection  
  
# Extract the summary as a data frame  
model\_summary\_1 <- as.data.frame(summary(detection\_negbin\_mod, prob = 0.95)$fixed) %>%  
 rownames\_to\_column() %>%  
 clean\_names()  
  
estimate\_1 <- model\_summary\_1$estimate[2]  
L95CI\_1 <- model\_summary\_1$l\_95\_percent\_ci[2]  
H95CI\_1 <- model\_summary\_1$u\_95\_percent\_ci[2]  
estimate\_percent\_1 <- (exp(estimate\_1) - 1) \* 100  
L95CI\_percent\_1 <- (exp(L95CI\_1) - 1) \* 100  
H95CI\_percent\_1 <- (exp(H95CI\_1) - 1) \* 100  
  
model\_summary\_1 %>%  
 gt()

|  | estimate | est\_error | l\_95\_percent\_ci | u\_95\_percent\_ci | rhat | bulk\_ess | tail\_ess |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Intercept | 1.3528460139 | 1.028764e-01 | 1.153918e+00 | 1.5552078171 | 1.000282 | 8540.574 | 9244.45 |
| combind\_detection | 0.0001105696 | 3.742394e-05 | 4.242594e-05 | 0.0001899142 | 1.000127 | 13563.245 | 10166.72 |

### Percentage detections

Model structure.

percent\_detection\_negbin\_str <- bf(eipaab\_n ~ combind\_percent\_detection, family = negbinomial())

Run model. This has been hashed out as I have saved the model, and will re-load it instead of re-running it.

# percent\_detection\_negbin\_mod <- brm(percent\_detection\_negbin\_str, data =  
# test\_surfacewater, cores = 4, chains = 4, #prior = mod\_priors, #use defaults  
# warmup = 1000, seed = 20250418, thin = 2, iter = 8000, control =  
# list(max\_treedepth = 20, adapt\_delta = 0.9999), save\_pars =  
# save\_pars(all=TRUE), sample\_prior = TRUE, file = paste0(mods\_wd,  
# './percent\_detection\_negbin\_mod')) print('Model complete')

#### Re-load models

Reload the model if necessary.

percent\_detection\_negbin\_mod <- readRDS(file = paste0(mods\_wd, "./percent\_detection\_negbin\_mod.rds"))

#### Model diagnostics

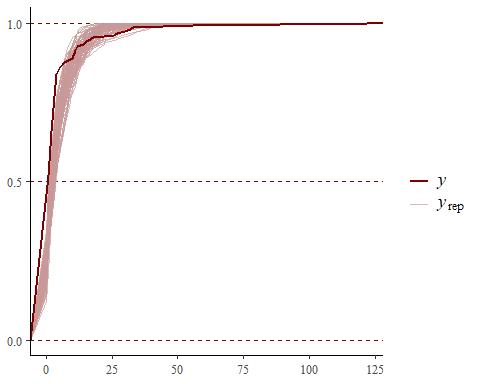
The R2 value is very low, but should be fine for the purposes of this investigation.

performance::r2\_bayes(percent\_detection\_negbin\_mod, robust = FALSE, ci = 0.95)

## # Bayesian R2 with Compatibility Interval  
##   
## Conditional R2: 0.019 (95% CI [3.045e-10, 0.063])

Check fit.

color\_scheme\_set("red")  
brms::pp\_check(percent\_detection\_negbin\_mod, ndraws = 100, type = "ecdf\_overlay")



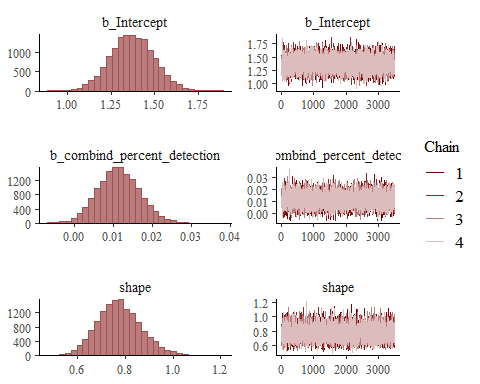
Checking R-hat values to confirm model convergence.

print(summary(percent\_detection\_negbin\_mod, prob = 0.95)) #All Rhat = 1 (good).

## Family: negbinomial   
## Links: mu = log; shape = identity   
## Formula: eipaab\_n ~ combind\_percent\_detection   
## Data: test\_surfacewater (Number of observations: 156)   
## Draws: 4 chains, each with iter = 8000; warmup = 1000; thin = 2;  
## total post-warmup draws = 14000  
##   
## Regression Coefficients:  
## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS  
## Intercept 1.37 0.12 1.13 1.62 1.00 13384  
## combind\_percent\_detection 0.01 0.01 0.00 0.02 1.00 13218  
## Tail\_ESS  
## Intercept 11593  
## combind\_percent\_detection 11943  
##   
## Further Distributional Parameters:  
## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS  
## shape 0.78 0.09 0.61 0.97 1.00 12857 12214  
##   
## Draws were sampled using sampling(NUTS). For each parameter, Bulk\_ESS  
## and Tail\_ESS are effective sample size measures, and Rhat is the potential  
## scale reduction factor on split chains (at convergence, Rhat = 1).

Check chains, and estimates.

color\_scheme\_set("red")  
plot(percent\_detection\_negbin\_mod)



#### Model estimates

Pulling the conditional effect estimates from the model. The estimate corresponds to the effect of the predictor on the log of the expected response variable. A 1-unit increase (i.e. 1%) in the percentage of environmental detections increases the expected occurrence in EIPAAB database by 1.17% (i.e. (exp(0.01160477)-1)\*100 = 1.167237).

# Extract estimates  
esitmates\_2 <- conditional\_effects(percent\_detection\_negbin\_mod)  
estimate\_data\_2 <- esitmates\_2$combind\_percent\_detection  
  
# Extract the summary as a data frame  
model\_summary\_2 <- as.data.frame(summary(percent\_detection\_negbin\_mod, prob = 0.95)$fixed) %>%  
 rownames\_to\_column() %>%  
 clean\_names()  
  
estimate\_2 <- model\_summary\_2$estimate[2]  
L95CI\_2 <- model\_summary\_2$l\_95\_percent\_ci[2]  
H95CI\_2 <- model\_summary\_2$u\_95\_percent\_ci[2]  
estimate\_percent\_2 <- (exp(estimate\_2) - 1) \* 100  
L95CI\_percent\_2 <- (exp(L95CI\_2) - 1) \* 100  
H95CI\_percent\_2 <- (exp(H95CI\_2) - 1) \* 100  
  
model\_summary\_2 %>%  
 gt()

|  | estimate | est\_error | l\_95\_percent\_ci | u\_95\_percent\_ci | rhat | bulk\_ess | tail\_ess |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Intercept | 1.37193582 | 0.124771054 | 1.129258961 | 1.61932209 | 1.000070 | 13383.85 | 11593.22 |
| combind\_percent\_detection | 0.01142544 | 0.005311951 | 0.001332697 | 0.02204329 | 1.000163 | 13218.02 | 11942.79 |

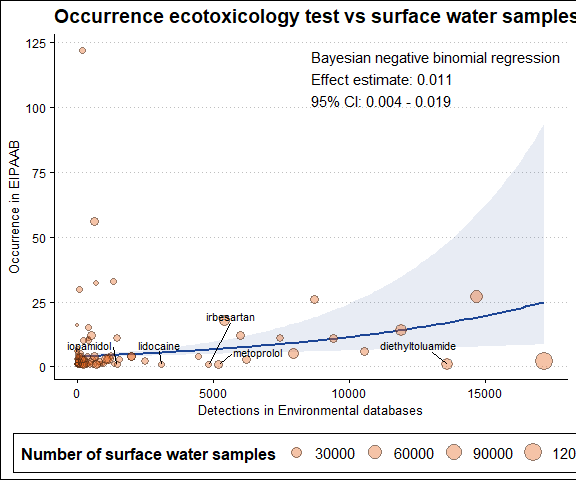
## 📈 Model results

### Total detections

There is generally a small positive correlation between the total environmental detections in the surface water data and occurrence in the EIAPAAB database (Fig 1; Bayesian negative binomial regression estimate: 0.011 [0.001, 0.022]), although there is a lot of uncertainty around these estimates. With that said, there are a number of compounds that occurrence frequently in the surface waters that are not well represented in the EIPAAB database (see Table 3 below). These could be sensible targets for future behavioural tests.

#### Fig S12

text\_coord <- test\_surfacewater %>%  
 filter(!is.na(eipaab\_n)) %>%  
 summarise(x = max(combind\_detection, na.rm = TRUE)/2, y = max(eipaab\_n, na.rm = TRUE))  
  
y\_coord <- text\_coord %>%  
 pull(y)  
  
x\_coord <- text\_coord %>%  
 pull(x)  
  
  
fig\_s12 <- ggplot() + geom\_line(data = estimate\_data\_1, aes(x = combind\_detection,  
 y = estimate\_\_), color = "#1c4595", linewidth = 1) + geom\_ribbon(data = estimate\_data\_1,  
 aes(x = combind\_detection, ymin = lower\_\_, ymax = upper\_\_), fill = "#1c4595",  
 alpha = 0.1) + geom\_point(data = test\_surfacewater %>%  
 filter(!is.na(eipaab\_n)), aes(x = combind\_detection, y = eipaab\_n, size = combind\_samples),  
 alpha = 0.4, shape = 21, color = "#01080A", fill = "#E76A24") + annotate("text",  
 x = x\_coord, y = y\_coord, label = paste0("Bayesian negative binomial regression",  
 "\n", "Effect estimate: ", round(estimate\_percent\_1, 3), "\n", "95% CI: ",  
 round(L95CI\_percent\_1, 3), " - ", round(H95CI\_percent\_1, 3)), hjust = 0,  
 vjust = 1, size = 4) + geom\_text\_repel(data = test\_surfacewater %>%  
 filter(!is.na(eipaab\_n), rank\_detection\_dif %in% 1:5), aes(x = combind\_detection,  
 y = eipaab\_n, label = compound\_name\_corrected), size = 3, box.padding = 0.7) +  
 # Adjust scales  
scale\_fill\_gradientn(colours = colorspace::diverge\_hcl(7)) + scale\_colour\_manual(values = c(`TRUE` = "#01080A",  
 `FALSE` = "#01080A"), guide = "none") + labs(title = "Occurrence ecotoxicology test vs surface water samples",  
 x = "Detections in Environmental databases", y = "Occurrence in EIPAAB", fill = "Number of surface water samples",  
 size = "Number of surface water samples") + theme\_clean() + theme(legend.position = "bottom",  
 legend.justification = "center")  
  
fig\_s12



**Fig S11:** Each pharmaceutical occurrence in the behavioural database plotted against the number of positive detections in the environmental databases. The five compounds highlighted in panels A and B represent those that are currently underrepresented in the behaviour test database based on their rank detections (i.e. rank occurrence in behaviour test databases vs rank detections). The point size represents the total number of samples across all environmental databases.

This code saves the figure above as PDF.

# ggsave(filename = paste0(fig\_wd, './figure-3a.pdf'), plot = fig\_3a, width =  
# 210, height = 297/1.5, # Specify the height of the plot units = 'mm')

#### Table S5

**Table S5:** Compounds with the most environmental detections that are not present in EIPAAB.

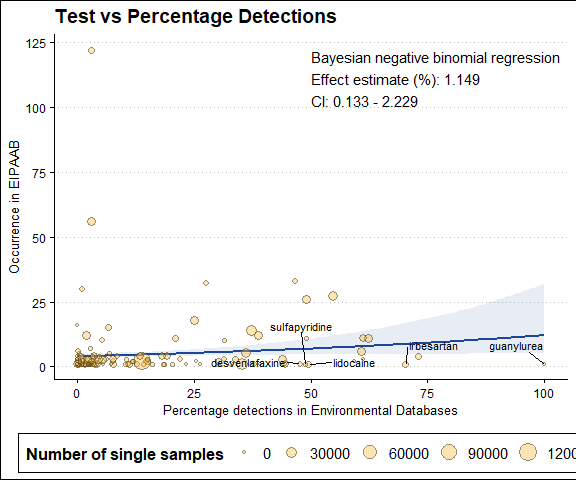
table\_S5 <- test\_surfacewater %>%  
 dplyr::filter(is.na(eipaab\_n)) %>%  
 dplyr::arrange(desc(combind\_detection)) %>%  
 dplyr::slice(1:15) %>%  
 dplyr::mutate(compound\_name = sentence\_case(compound\_name\_corrected)) %>%  
 dplyr::select(compound\_name, combind\_samples, combind\_detection, combind\_percent\_detection,  
 combind\_rank\_samples, combind\_rank\_detection) %>%  
 dplyr::rename(Name = compound\_name, Samples = combind\_samples, Detections = combind\_detection,  
 `Percentage detections` = combind\_percent\_detection, `Rank samples` = combind\_rank\_samples,  
 `Rank detections` = combind\_rank\_detection) %>%  
 gt() %>%  
 fmt\_number(columns = c("Samples", "Detections"), decimals = 0) %>%  
 cols\_align(align = "center", columns = everything())  
  
table\_S5

| Name | Samples | Detections | Percentage detections | Rank samples | Rank detections |
| --- | --- | --- | --- | --- | --- |
| Glyphosate | 79,682 | 33,956 | 42.614392 | 28 | 1 |
| Atrazine | 121,709 | 29,480 | 24.221709 | 3 | 2 |
| Bentazone | 98,363 | 28,315 | 28.786231 | 19 | 3 |
| Carbendazim | 110,395 | 22,410 | 20.299832 | 9 | 4 |
| Lead | 57,662 | 21,463 | 37.222087 | 56 | 5 |
| Tebuconazole | 106,651 | 20,664 | 19.375346 | 12 | 6 |
| Imidacloprid | 90,527 | 20,383 | 22.515934 | 21 | 7 |
| Boscalid | 68,930 | 19,456 | 28.225736 | 39 | 8 |
| Chlorotoluron | 114,605 | 18,155 | 15.841368 | 4 | 9 |
| Diflufenican | 70,085 | 17,074 | 24.361846 | 38 | 11 |
| Isoproturon | 128,238 | 16,944 | 13.212932 | 1 | 12 |
| Azoxystrobin | 83,421 | 16,516 | 19.798372 | 25 | 13 |
| Propyzamide | 105,962 | 16,302 | 15.384761 | 14 | 14 |
| Chloridazon | 98,996 | 15,203 | 15.357186 | 17 | 15 |
| Ethofumesate | 111,478 | 10,649 | 9.552557 | 8 | 19 |

### Percentage detections

#### Fig S13

text\_coord <- test\_surfacewater %>%  
 filter(!is.na(eipaab\_n)) %>%  
 summarise(x = max(combind\_percent\_detection, na.rm = TRUE)/2, y = max(eipaab\_n,  
 na.rm = TRUE))  
  
y\_coord <- text\_coord %>%  
 pull(y)  
  
x\_coord <- text\_coord %>%  
 pull(x)  
  
fig\_s13 <- ggplot() + geom\_line(data = estimate\_data\_2, aes(x = combind\_percent\_detection,  
 y = estimate\_\_), color = "#1c4595", linewidth = 1) + geom\_ribbon(data = estimate\_data\_2,  
 aes(x = combind\_percent\_detection, ymin = lower\_\_, ymax = upper\_\_), fill = "#1c4595",  
 alpha = 0.1) + geom\_point(data = test\_surfacewater %>%  
 filter(!is.na(eipaab\_n)) %>%  
 dplyr::arrange(rank\_detection\_percent\_dif) %>%  
 mutate(is\_top5 = rank\_detection\_percent\_dif %in% 1:5), aes(x = combind\_percent\_detection,  
 y = eipaab\_n, size = combind\_singles), alpha = 0.4, shape = 21, color = "#01080A",  
 fill = "#fbbc42") + annotate("text", x = x\_coord, y = y\_coord, label = paste0("Bayesian negative binomial regression",  
 "\n", "Effect estimate (%): ", round(estimate\_percent\_2, 3), "\n", "CI: ", round(L95CI\_percent\_2,  
 3), " - ", round(H95CI\_percent\_2, 3)), hjust = 0, vjust = 1, size = 4) +  
 geom\_text\_repel(data = test\_surfacewater %>%  
 filter(!is.na(eipaab\_n), rank\_detection\_percent\_dif %in% 1:5), aes(x = combind\_percent\_detection,  
 y = eipaab\_n, label = compound\_name\_corrected), size = 3, box.padding = 0.7) +  
 scale\_fill\_gradientn(colours = colorspace::diverge\_hcl(7)) + scale\_colour\_manual(values = c(`TRUE` = "#01080A",  
 `FALSE` = "#01080A"), guide = "none") + labs(title = "Test vs Percentage Detections",  
 x = "Percentage detections in Environmental Databases", y = "Occurrence in EIPAAB",  
 size = "Number of single samples") + theme\_clean() + theme(legend.position = "bottom",  
 legend.justification = "center")  
  
fig\_s13



**Fig S12:** Pharmaceutical occurrence in the behavioural test database plotted against the percentage of positive detections (i.e. positive detections relative to total samples for that compound) in environmental databases. The five compounds highlighted in panels A and B represent those that are currently underrepresented in the behaviour test database based on their rank percentage detections (i.e. rank occurrence in behaviour test databases vs rank percentage detections in environmental databases). The point size represents the total number of samples across all environmental databases.

# ggsave(filename = paste0(fig\_wd, './figure-3b.pdf'), plot = fig\_3b, width =  
# 210, height = 297/1.5, # Specify the height of the plot units = 'mm')

#### Table S6

table\_s6 <- test\_surfacewater %>%  
 dplyr::filter(is.na(eipaab\_n) & !is.na(combind\_percent\_detection)) %>%  
 dplyr::arrange(desc(combind\_percent\_detection)) %>%  
 dplyr::slice(1:15) %>%  
 dplyr::mutate(compound\_name = sentence\_case(compound\_name\_corrected)) %>%  
 dplyr::select(compound\_name, combind\_singles, combind\_single\_detection, combind\_percent\_detection) %>%  
 dplyr::rename(Name = compound\_name, `Single samples` = combind\_singles, `Single detections` = combind\_single\_detection,  
 `Percent detections` = combind\_percent\_detection) %>%  
 gt() %>%  
 fmt\_number(columns = c("Percent detections"), decimals = 2) %>%  
 cols\_align(align = "center", columns = everything())  
  
table\_s6

| Name | Single samples | Single detections | Percent detections |
| --- | --- | --- | --- |
| Sulisobenzone | 101 | 101 | 100.00 |
| 10,11-dihydro-10,11-dihydroxy carbamazepine | 13 | 13 | 100.00 |
| Alpha-hydroxy metoprolol | 24 | 24 | 100.00 |
| Acetaminophen sulfate | 24 | 24 | 100.00 |
| 17-beta-estradiol-17-glucuronide | 14 | 14 | 100.00 |
| Estriol-16-glucuronide | 14 | 14 | 100.00 |
| 2-hydroxy ibuprofen | 41 | 40 | 97.56 |
| 4-acetaminoantipyrine | 21 | 20 | 95.24 |
| Methenamine | 811 | 768 | 94.70 |
| 1-hydroxy ibuprofen | 61 | 57 | 93.44 |
| Carbamazepine-3oh | 26 | 24 | 92.31 |
| Carboxyibuprofen | 39 | 36 | 92.31 |
| O-desmethyltramadol | 34 | 31 | 91.18 |
| Perindopril | 112 | 101 | 90.18 |
| Lamotrigine | 1555 | 1378 | 88.62 |

# 🧠 Supplementary discussion

## Compound occurrence comparison

We found a weak positive relationship between the compounds studied in behavioural assays and their detection frequency in environmental databases—consistent with the idea that environmental occurrence may be partly motivating the behavioural research. The intended therapeutic effects or mechanism of action also appears to play a significant role in overall representation in the behavioural test database, with most of the most common compounds being psychoactive/neuroactive pharmaceuticals (e.g. antidepressants, antiepileptics and anxiolytics; Table S4).