

ReadMe

Population PFAS TK Model

Updated 2026-06-26

This document describes key processes and functions of the Population PFAS Toxicokinetic Model discussed in the Journal of Environmental Exposure Assessment publication *Estimating Drinking Water PFAS Exposures Associated with Serum Clinical Action Levels Using a Probabilistic Toxicokinetic Model* by Nielsen et al., 2026. Model limitations and disclaimers regarding the model are also presented and should be read carefully before the model is used.

Limitations and Disclaimers. This code is made available to support transparency, reproducibility, scientific review, and appropriate implementation of the methods described in the publication.

This model and code are research materials only. They are not a validated clinical, diagnostic, or medical decision-making tool. They are not intended to provide medical advice, diagnosis, treatment recommendations, or individualized exposure or health determinations. The model should not be used as the sole basis for determining whether any individual should receive clinical testing, treatment, medical monitoring, or follow-up. Any health-related interpretation or application of model outputs should be conducted by qualified professionals using appropriate clinical, exposure, toxicological, and public health information.

Publication of this code does not establish, revise, or supersede any regulatory standard, guidance value, imminent hazard value, policy, enforcement position, permit condition, or other official determination of the Massachusetts Department of Environmental Protection (MassDEP), the Executive Office of Energy and Environmental Affairs (EEA), or the Commonwealth of Massachusetts. Any regulatory, enforcement, site-specific, or public health use of the model requires independent professional judgment, review of applicable law and guidance, and consideration of relevant facts, assumptions, uncertainties, and limitations.

Model outputs are estimates and are subject to uncertainty. Users should review the associated publication, documentation, assumptions, limitations, and uncertainty analyses before applying or interpreting the model or code.

The code and associated materials are provided “as is”, without warranty of any kind, express or implied, including but not limited to warranties of accuracy, completeness, reliability, security, compatibility, merchantability, fitness for a particular purpose, or non-infringement. Potential incompatibilities with other software, systems, packages, or computing environments have not been fully assessed. Users are responsible for evaluating the code, validating results, complying with applicable laws and requirements, and determining whether and how the code is appropriate for their intended use.

The views, conclusions, and model outputs generated by users of this code do not necessarily represent the views, policies, or positions of MassDEP, EEA, or the Commonwealth of Massachusetts. Peer review of the associated publication and code does not constitute approval, certification, or validation of the model for any clinical, regulatory, legal, commercial, or site-specific application.

Overview

The population PFAS TK model predicts the distribution of serum PFOA, PFOS, PFHxS, PFNA, PFDA, and PFHpA levels for defined populations from drinking water levels. Model inputs include including drinking water PFAS concentrations over time, maternal serum PFAS levels, population information, exposure factors, non-drinking water or ‘background’ exposure, and chemical-specific toxicokinetic parameters. These inputs are organized in an Excel spreadsheet and calculations are performed in R.

The model main directory “Population-PFAS-TK-Model/V2.4” contains an R project, a “scripts” directory for the underlying functions, the working script for a model demo, and directories for the demo inputs and outputs. The folder tree is below, with descriptions in italics.

- /scripts
 - analyze_tk_output.R
 - *Functions creating model runs and analysis from input spreadsheet*
 - monte_carlo_vars.R
 - *Functions creating distributional parameters for Monte Carlo process*
 - serum_calcs.R
 - *Functions to calculate serum PFAS concentrations over time*
- /Set00_MaternalSerumDemo
 - LogsplineFits_Set00_MaternalSerumDemo.RData
 - *Maternal serum distribution data for the model demo (Set01_Demo)*
- /Set01_Demo
 - inputdata_Set01_Demo.xlsx
 - *Population, exposure, and parameter data, plus analysis information for demo model set*
- 2026-06-25_Set01_Demo.R
 - *Script to execute model demo*
- ReadMe_Population-PFAS-TK-Model_V2.4.pdf
 - *Provides information about model functions and structure*
- Demo_Instructions_Population-PFAS-TK-Model_V2.4.pdf
 - *Provides instructions for a worked example to run the model*
- PFAS_TK_Model_MassDEP_V2.4.Rproj
 - *Project file organizing model execution in R Studio*

Model Demo

A worked model example is provided with instructions here:

Population-PFAS-TK-Model/ V2.4

- Demo_Instructions_Population-PFAS-TK-Model_V2.4.pdf

Modeling Process

When modeling a population over time, serum PFAS distributions are calculated at regular intervals as the population ages. By default, calculations are performed when parameters change, typically monthly below the age of 1 and annually beyond this age. The model tracks time by date in order to reference exposure events such as the date when drinking water concentrations change.

When using this model, population and exposure information are organized into a batch called a “set”. Within each set, several “exposure scenarios” can be modeled and analyzed, either individually or in relation to one another. Each set has a specified ID, e.g. “Set01_Demo”, called the setID. For the model to run, there must be a folder in the R project directory with a name that exactly matches the setID. The setID will generally be appended to model-related files inside that directory.

The inputs and analyses for each set are organized into a spreadsheet tagged with the setID formatted as “inputdata_setID.xlsx”. Different tabs manage different parts of the modeling and analysis. The role of each tab is described in the “ReadMe” tab of the inputdata spreadsheet, along with the purpose of each column or row in the spreadsheet. The key tab for users is the ‘Scenario’ tab, where population and exposure information is entered to model the desired exposure scenario.

Key Model Functions

Functions in the “scripts” directory are generally well-commented but are described briefly below in order of use.

`Read.Input.Data()`: Reads and organizes the Excel file to be compatible with subsequent functions.

`Model.All.Scenarios()`: Organizes the rows of the inputdata file “Scenario” tab, each of which represents an exposure scenario, and runs them through the TK model process. Each exposure scenario has an ID to organize it. The process saves when each exposure

scenario is complete. If the process is interrupted, re-running `Model.All.Scenarios()` will continue from exposure scenarios without an associated output. Therefore, re-running this function for the same exposure scenario ID with new input parameters will NOT overwrite existing outputs. If a new input for the same scenario ID is desired, the corresponding outputs in the RawData file must be manually removed. This can be achieved by removing the corresponding RawData file from the set directory.

`monte_carlo_vars()`: Draws parameters from the defined distributions to prepare for the modeling process. It creates a dataframe where each row represents a Monte Carlo iteration, and each column a parameter (or set of parameters) that will be used in the TK model. It uses a helper function `make_rnd()`, which reads the parameter entries and distribution shape to create random values from the specified distribution.

`calc_Serum_PFAS()`: The wrapper function for calculating a single iteration of serum changing over time. Running this function produces a list of dataframes. Each list index represents an iteration. Each dataframe has rows representing dates and columns representing values such as consumed liquid concentration, background serum PFAS, and total serum PFAS at each date. This list is saved as the “results” object in the “RawData” files.

`create_frame()`: Sets up the dates for calculating serum PFAS levels and changing parameters. Serum levels are calculated monthly by default before age 1 and annually after age 1. Dates also include any time the PFAS concentration in water changes during the receptor’s lifetime. The time zoom functionality can generate extra dates to consider around a timeframe of interest. The function also generates a rounded-down age for eventual matching to liquid intakes. Until age 1, it rounds down to the nearest month, then divides by 12. Afterwards, it rounds down to the nearest year.

`pop_Liquid_Intake()`: Fills in the liquid intake in mL/(kg-day) at each date, multiplying by the liquid intake adjustment factor for the appropriate age group.

`pop_Liquid_Conc()`: Fills in water or breast milk PFAS concentrations as appropriate by date. Water is filled in with a simple matching based on dates. Dates where liquid intake occurs as breastmilk instead of water are identified using `fed_mo`. The PFAS concentration in the milk is calculated using `MilkDecline`, `LactXfer`, and `MatStart` (appearing in the function as `momserum`).

`pop_Bkgd_Serum()`: Fills in background serum PFAS for each date. It also makes the background PFAS phase in linearly starting from 0 until age 3.

`calc_Total_Serum()`: Uses entries in each date to calculate `Total_Serum` at the next date. Modeling begins with `Total_Serum` based on placental transfer, with no background. The

second row needs to be made manually because there isn't a value for bkgdold before birth (so all the placental transfer PFAS is available for elimination). The helper function `nextserum()` performs the calculation using the integrated rate law.

`Extract.Serum.at.Dates()`: Combines all iterations' serum values into a single dataframe, where rows are iterations and columns are scenario IDs with dates. RawData files are loaded one by one. Processing time depends on the number of iterations and modeling scenarios.

Functions to analyze model output

`Fit.Serum.Distributions()`: For when model output will be used as the maternal serum distribution in another exposure scenario.

`Evaluate.Against.Observed.Sera()`: For comparing modeled serum percentiles against observed serum percentiles.

`Project.from.Serum()`: For calculating CGWCs and other linear regressions. Connects exposure scenario parameters (for example, water concentration) to serum concentration at a specified percentile.

`Prep.Projection.Plot.Data()`: Saves data in the format required to plot the CGWC projection process.

`Prep.Lifetime.Plot.Data()`: Saves data in the format required to plot serum percentiles over the lifetime of a population.

`Check.Stability()`: Estimates between-run error. Helps determine if enough iterations were run.

`Calculate.RMSE()`: Calculates RMSE between observed and modeled percentiles.