READ ME

Table of Contents

1. What is the GBD Diet–NCD Model?

The GBD (Global Burden of Disease) Diet-NCD Model is a framework developed by the Institute for Health Metrics and Evaluation (IHME) used to quantify the impact of suboptimal diet on non-communicable disease (NCD) morbidity and mortality (1,2). It involves a comparative risk assessment approach that estimates the proportion, or the population attributable fraction (PAF) of disease-specific burden attributable to each of the 15 dietary risk factors, specified in Table 1 (2).

Table 1 – Dietary risk factors included in GBD 2017 and direction of risk

Dietary Risk Factors	**Direction of Risk**
Diet low in fruits	Low intake
Diet low in vegetables	Low intake
Diet low in legumes	Low intake
Diet low in whole grains	Low intake
Diet low in nuts and seeds	Low intake
Diet low in milk	Low intake
Diet high in red meat	High intake
Diet high in processed meat	High intake
Diet high in sugar-sweetened beverages	High intake
Diet low in fiber	Low intake
Diet low in calcium	Low intake
Diet low in seafood omega-3 fatty acids	Low intake
Diet low in polyunsaturated fatty acids	Low intake
Diet high in trans fatty acids	High intake
Diet high in sodium	High intake

The main inputs to this analysis include exposure in the form of **intake of each dietary factor**, the effect size in the form of the **relative risk (RR) of mortality and morbidity** of the dietary factor on **disease endpoints** and the level of intake associated with the lowest risk of mortality, or the **Theoretical Minimum Risk Exposure Level (TMREL).** 'Input exposure' extracted from dietary recall data and extrapolations, and 'input parameters', synthesized from long-term, prospective observational studies and short-term trials of intermediate outcomes, were used to construct the **PAF**. The PAF is a measure of the proportion of disease burden or prevalence that would be avoided in a hypothetical population, similar to the population of interest where a

particular risk factor (a dietary risk factor in this instance) is eliminated. Additional inputs to the model include **disease-specific mortality, and disability-adjusted life years (DALYs)**. By applying the constructed PAFs to the DALYs, the diet-attributable years of life lost (YLLs) and years lived with disability (YLDs) can be calculated for each disease outcome.

Python code for GBD Diet-NCD emulator (GBDDE2017)

The code implements an emulator of the IMHE 2017 GBD Diet–NCD Model in Python. Although newer versions of the GBD model have since been released by IMHE —and the GBD 2017 inputs are no longer publicly available, we provide a transparent, fully replicable data-processing pipeline that reproduces the core results of the GBD 2017 Diet-NCD model and extends it to support scenario-based analyses and marginal (partial derivative) disease-burden estimation. Some reasons why GBD 2017 is a preferred stand-alone dietary risk model over later GBD versions are given in Section 2.2.5 of (3). The overall workflow, including data preparation, scenario construction, and projection routines, is described in the accompanying Python scripts and summarized in the README.

Table 2. Descriptions of the analytical scenarios implemented by the emulator

Analytical Scenario	Description
Analytical Scenario 1	Estimation of disease burden from a given dietary exposure, expressed in YLLs and YLDs (summed to arrive at DALYs).
Analytical Scenario 2	Estimation of the difference in disease burden between exposure and a counterfactual exposure distribution generated through unilateral shifts in individual dietary risk factors; and
Analytical Scenario 3	Marginal disease-burden estimation around a given dietary exposure.

A unilateral shift in exposure refers to a shift in the intake distribution, such as increasing per-capita fruit intake (g/day) by a fixed amount (e.g., 5 g). In the accompanying code and subsequent explanations, this is referred to as the h-shift. A negative value for the shift allows for a reduction in intake (and for diets low in fruit this would be an increase in exposure). The shifts can be described individually for each or all dietary exposures, for each or all countries, and for each and all demographics subgroups (population stratified by sex and age). A simple assumption that every individual in a population increases or decreases their intake by the same amount of for one exposure variable results in a unilateral shift of the distribution of intake for all demographic subgroups. The marginal disease burdens are the partial derivative associated to unilateral shift in one dietary risk exposure, that is, the limit of the disease burden from unilateral shift divided by the size of the shift.

Together, these scripts provide all computational steps required to rerun the deterministic GBD Diet-NCD workflow, reproduce baseline and counterfactual disease-burden estimates, and extend the original framework to scenario-based and marginal analyses.

Extrapolations

The code supports iteration across multiple runs using input datasets that vary by scenario and time point enabling forward-looking analyses in which dietary exposures and total disease burden evolve over time. For their analysis, users may modify shift files (authorized users may access this in the *Shift* subfolder under *Data*), SSP means (e.g., by modifying narrative adjustments in *SSP_means.py*). However, input data must be structured properly (refer to data indexing below). The input data that can be downloaded contains a template for iterated runs under forward looking scenarios.

2. License and Data Use

The emulator Python code is available under the MIT License. It implements formulas for PAFs that are common goods available from journal publications. The code can be downloaded from GitHub.

To reconstruct the PAFs from the 2017 GBD study (2) and the associated disease burdens, or to project the PAFS and disease burdens based on GBD exposures and effects forward, requires data that was originally available by download and request from IMHE.

The input data required for the reconstruction comes in three kinds

- Exposure (mean and standard deviation of population intake)
- Effect parameters (relative risk parameters including TMREL)
- Overall disease burden by disease outcome (YLLs and YLDs)

The first two types of data are required to generate PAFs. The third type of data are required to arrive at disease burden estimates in DALYs from PAFs.

The input data for GBD 2017 estimate reconstruction falls under the *IHME Free-of-Charge Non-Commercial User Agreement*.

Any commercial use of GBD data, including integration into commercial software, tools, consultancy products, or paid services, requires a separate licensing agreement with IHME. Commercial users must obtain explicit permission from IHME prior to accessing, distributing, or modifying GBD data for commercial purposes. Availability of the emulator code does not grant permission for commercial use of the data required to reproduce or generate outputs compiled by IHME. Users of the code who also use GBD data are responsible for ensuring compliance with IHME's licensing requirements.

More information on IMHE licensing can be found here - https://www.healthdata.org/Data-tools-practices/data-practices/ihme-free-charge-non-commercial-user-agreement and https://www.healthdata.org/data-tools-practices/data-practices/terms-and-conditions.

Due to the difference in license between the emulator code and the input data required to reproduce disease burden results based GBD2017:

The emulator code is available in a public repository.

The input data to reconstruct GBD results is in a separate private repository.

If users wish to access the private repository of data to reproduce GBD 2017 results, they must first *register for an account to download data on the IHME website* and retain a screenshot confirming their registration and acceptance of the IHME Free-of-Charge Non-Commercial User Agreement. Once this evidence has been provided to the authors (contact details below), access to the data in the private Zenodo repository, the DOI of which will be specified under 'Data Access' within this GitHub repository will be granted. This process ensures that all users of the input data comply with IHME's data access requirements and any subsequent use of GBD exposure and other data remains consistent with IHME licensing conditions.

Data Availability and Access

To reiterate input datasets derived from Global Burden of Disease (GBD) data are subject to IHME data-use restrictions. As such, the full datasets are not stored within the public repository containing the emulator code.

The public repository includes a stub `Data/` directory containing empty placeholder folders only. No input data are distributed with this codebase.

Users wishing to run the emulator must first contact the authors and provide evidence that they have accepted the IHME Free-of-Charge Non-Commercial User Agreement and associated terms and conditions. Upon approval, users will be granted access to the full input dataset via a restricted Zenodo repository.

Once access has been granted:

1. Download the dataset from the Zenodo repository.
2. Extract the downloaded archive.
3. Replace the stub `Data/` directory with the extracted data directory at the project root, as described further under *Running the Emulator*.

3. Requirements

Python Environment

- *Python version:* 3.11.7 (Anaconda distribution)
- *Interactive environment:* IPython 8.20.0

The deterministic emulator relies on the following core scientific Python libraries:

- *NumPy* – numerical operations and array manipulation
- *SciPy* – statistical functions and numerical integration
- *Pandas* – data handling, cleaning, and transformations
- *SymPy* – symbolic mathematics used in analytical derivations
- *Matplotlib* – plotting and figure generation

These libraries provide all required numerical, statistical, and symbolic capabilities for running the emulator code and reproducing the analyses across all three scenarios discussed in Section 1.

4. What can the model do?

As mentioned earlier, the GBD Diet-NCD 2017 emulator provides a flexible, analytical framework for quantifying diet-related disease burden using GBD inputs. The three core python scripts include –

1) *Original_GBD.py:* This core script, in conjunction with the associated helper scripts and input data implements *Analytical Scenario 1* (Table 2; refer to *Original_GBD.png* in the *'Schematic Diagrams'* subfolder in the folder '*Additional Information';* also see Appendix A), by calculating disease burden (YLLs, YLDs, DALYs) under 2017 dietary exposure distributions. The input data on dietary exposure can replaced by a user's own data on baseline dietary exposure. This script serves as the baseline model run against which all counterfactual and marginal analyses are compared.
2) *Unilateral_Shift.py* and *Unilateral_Shift_PJ.py:* This core script implements *Analytical Scenario 2* (refer to by estimating the change in disease burden for each dietary risk through a unilateral h-shift relative to the baseline dietary exposure. For each risk, the script applies a user specified shift (how is the shift specified is described below), recalculates the corresponding disease burden, and reports the absolute change between the shifted and baseline DALY estimates. The output therefore provides a risk-by-risk decomposition of the isolated impact of each unilateral dietary adjustment. This script generates three types of outputs.
    a. The joint disease burden estimates from exposure to all 15 dietary risks. The joint DALY estimates aggregate the disease burden from individual exposure accounting for mediation effects between dietary risks, reflecting the interdependence of exposure-outcome pathways.
    b. Second, it produces estimates of disease burden from exposure to single dietary risks without mediation (non-joint DALYs), treating each risk independently.
    c. Third, it estimates the proportion of the joint disease burden that comes from single dietary risks.

    The sum of the disease burdens from proportional single dietary risks in c. equals the joint disease burden in a. The sum of the disease burdens from single dietary risks in b. exceeds the joint disease burden in a.

    Refer to *Unilateral_Shift.png* in the subfolder 'Schematic Diagrams' in the 'Additional Information' folder. (we put these diagrams in Appendix A of this README as well).

3) *Partial_Derivative_Calculation.py:* This script implements *Analytical Scenario* (refer to Table 2) by assessing how small, incremental changes in each dietary exposure would affect disease burden around the baseline. It generates marginal PAF estimates for each risk and disease and then applies the mediation matrix to account for overlaps between risks. The resulting output provides marginal, mediation-adjusted estimates of how small changes in each dietary exposure would affect the joint disease burden of dietary exposure at the baseline. Refer to *Marginal_Change.png* in the subfolder *'Schematic Diagrams'* in the '*Additional Information'* folder.

Scenario studies

The code supports scenario-based analyses. As discussed under 'Extrapolations', exogenous projections of dietary exposures and overall disease burden (YLLs and YLDs) can be incorporated to reflect how diets and disease burden evolve over time (Schematic Diagram 4, Appendix A). The structure and indexing of these inputs are described in the 'Extrapolations' section and illustrated through the accompanying example files, which include –

- The user specified shift files containing h-shift values. These can be found in the *Shift* subfolder in the *Data* folder.
- YLL and YLD projections of population total disease burdens from 2020 to 2050 are in the *Projections* subfolder of the larger *Data* folder.
- Shared Socio-Economic (SSP) scenarios for dietary exposure from 2020 to 2050 that can be found in the *SSP Means* subfolder of the larger *Data* folder.

Data are structured and indexed by time point, location, sex, age group, and disease outcome for disease rate projections, while mean dietary exposures are indexed by time point, location, sex, age group, and dietary risk.

The index dictionary specifies the set of dimensions over which the analysis is run. It defines which years, locations, demographic groups (e.g. age and sex), and dietary risks are included, and is used throughout the codebase to control iteration and alignment of inputs and outputs. The index dictionary is constructed in helpers.py. This file contains all index definitions required for modelling, as the indices used in the nested loops of the main scripts are drawn directly from this dictionary. When running custom analyses, users must update the index dictionary accordingly and ensure that all input data conform to the specified indexing structure. Any mismatch between the index dictionary and the input data will result in an error during execution of the main scripts.

As an illustration, the authors present an example implementation of Analytical Scenario 2 (refer to Table 2). Two scenarios "CT" and "FT" were explored for the time points 2025 and 2030 in the example data files. The helpers.py module, which contains the index dictionary, was modified accordingly, and the corresponding input files, including the shift and disease-rate projection files, were structured to match the required indexing. The shift file was indexed by scenario, time point, country, age, sex, and dietary risk, while the disease-rate projection file was indexed by time point, country, age, sex, and disease outcome. An example of this analysis is provided in the input data as a template for users conducting forward-looking analyses using the emulator.

The example provided for Analytical Scenario 3 differs from that for Analytical Scenario 2 (see Table 2). The data files made available to authorized users are not illustrative only; they correspond to an example implementation of Analytical Scenario 3 used by the authors to generate the reported results in (3). In this case, the nested loop in the final script Partial_Derivative_Calculation.py does not iterate over scenarios, as the code is executed separately for each Shared Socioeconomic Pathway (SSP), which serves as the scenario dimension. Mean dietary exposures and disease rates therefore vary by only year rather than by year and scenario within a single model run.

For forward-looking analyses, conducted to assess future dietary and disease-burden scenarios, additional processing was undertaken to generate projections of YLLs and YLDs and to modify 2017 mean dietary exposures to construct SSP-based future dietary trajectories. Example input data and accompanying code, illustrating exposure changes indexed by scenario and time, and disease-burden inputs indexed over time as exogenous series (see Figure 4, Appendix A) are provided in the repository to serve as templates for this use case. A brief description of the adjustment methods used is provided in Appendix C *Extrapolations*.

5. Data Description.

Data in the 'Data' folder of the private repository can be grouped into three categories. Refer to the diagrams in 'Schematic Diagrams' for a visual overview of how each category of data is used within the emulator –

1) Input Exposure Data: This category of data includes all datasets containing population-level dietary exposure data (both raw and processed data) used as inputs to the model. These files provide age, sex, and dietary risk specific exposure data (means and standard deviations) for the 2017 GBD baseline. They also include any shifts in mean *h* required for *Analytical Scenario 2*. These data are used to generate two-parameter intake distributions for each dietary risk across countries, ages, sexes, and time-periods.
2) Input Effect Parameters: This category includes all datasets necessary to construct PAFs. These files contain TMREL values for each dietary risk, relative risk parameters for each risk–outcome and age-group pairing, unit specifications, and the classification of each dietary risk as "high" or "low." Together, these inputs define the risk–disease relationships and underpin all PAF computations across analytical scenarios.
3) Input Disease Data: This category contains the demographic and disease-burden datasets required to translate PAFs into YLLs, YLDs, and DALYs. These files include baseline disease burden estimates (by country, sex, age, and disease outcome) and may also include projected YLL and YLD values for future time periods. These datasets enable the emulator to compute total and scenario-specific disease-burden estimates across all analytical scenarios.

6. Key Modules

Some key modules for the full GBD 2017 emulator code include –

- *Setup_file.py:* Specifies the number of runs and the number of samples drawn for the uncertainty calculation (to be included in a future iteration). Contains file paths used during modelling. The *Setup_file.py* for the second and third analytical scenarios may differ from the first, as they include paths for shift files (Analytical Scenario 2), mean exposures under different Socioeconomic Pathways (SSPs), and projections of total disease burdens (Analytical Scenario 3).
- *helpers.py:* Specifies the index dictionary, containing scenarios, time-points, country names, diseases, risks, ages, and genders.
- *helpers_data_and_setup.py:* Contains functions for loading and formatting data for the emulator, including calculation of the mediation matrix. The

helpers_data_and_setup.py file in the second analytical scenario includes modified mediation matrices for joint and non-joint PAFs.

- *helpers_varaibles_calculation.py:* Contains functions for calculating parameters (alpha, beta, k, lambda, etc.) for the different probability distributions based on means and standard deviations. Some parameters are calculated using optimization.
- *Variable_creater_class.py:* Defines the *VariableCreator* class for calculating distribution parameters when they cannot be derived from means and standard deviations. Creates a data-frame of parameters based on means, standard deviations, and minimum-maximum values for each risk, age, and gender.
  Uses: *helpers_varaibles_calculation.py, helpers.py*
- *Distribution_creater_class.py:* Generates probability distributions of dietary intake from mean and standard deviation values. Distributions are created directly or using parameters from the *VariableCreator* class and combined through resampling. Provides methods to retrieve distributions and PDFs.
  Uses: helpers.py, Variable_creater_class.py
- *helpers_PAF_calculation.py*: Contains functions to calculate relative risks (RRs), individual PAFs, and mediation-adjusted PAFs. For Analytical Scenarios 2 and 3, includes functions for shifted PAFs, RR derivatives, and PAF derivatives (joint, mediation-adjusted).

For more detailed documentation of code modules, refer to the 'Documentation' file in the 'Additional Information' section. This has also been included as an appendix (Appendix B) to this document.

7. Running the Emulator

Before running the emulator, users must obtain the required input data from the restricted Zenodo repository in the same format and directory structure. This repository includes a stub *Data/* directory containing empty placeholder folders only.

After obtaining access to the full dataset, users must replace the stub *Data/* directory with the complete data directory at the project root, alongside the three folders containing the main Python scripts. The emulator relies on relative file paths that assume this directory structure. Placing the *Data/* directory elsewhere will require manual modification of file paths in the setup and helper scripts.

The three main Python scripts associated with Analytical Scenarios 1, 2, and 3 are housed in separate folders in the root directory. These are named 'Original GBD Emulator', 'Unilateral Shift Intake', and 'Marginals Calculation' respectively. Associated setup and helper files are contained within their respective folders.

Relative file paths in the setup scripts are formatted such that the main scripts are intended to be run from within the scenario folders. Helper modules are loaded and required .csv data files are imported accordingly. This version of the emulator uses central values of dietary risk exposure means, standard deviations, TMRELs, relative risk parameters, and YLLs and YLDs; therefore, *num_runs* in *Setup_file.py* must be set to 1.

For Analytical Scenario 2, *Setup_file.py* specifies a relative file path to the file containing the shifts, *h*. This path must be modified if a user wishes to implement their own scenarios, and associated shift values. An example input file is available in the *Data* folder (*Shift* subfolder).

For this scenario, the script *Unilateral_Shift.py,* DALY estimates can be computed either with or without accounting for mediation and aggregation of dietary risks. When the flag *calculate_NJ_DALYs is set to True,* the script computes individual disease burdens (non-joint DALYs); however, when this flag is set to *False,* the script computes the joint disease burden, which account for mediation between risks using mediation matrix. The script *Unilateral_Shift_PJ.py*, estimates proportional individual disease burdens and does not contain a flag. It must be run normally, as in Analytical Scenario 1.

In a similar fashion, for Analytical Scenario 3, the file paths specifying SSP-specific means, and YLL and YLD projections can be modified. Input files are provided for these as well.

Any modifications in *Setup_file.py* need to be reflected in *helpers.py* and, consequently, in the index dictionary that is constructed. This index dictionary is used to control the nested loop structure in the main scripts.

After making the required changes in *Setup_file.py,* the main scripts can be run. By default, the full model is run for all 160 countries and outputs are saved. However, the emulator can also be run using command-line arguments, allowing users to specify a subset of countries by position (e.g., countries 40 to 60) and generate subset results.

8. Model Output

The results (outputs) generated by the final scripts are stored as CSV files in the relevant Predictions/ subfolders within the Data/ directory. For Analytical Scenario 1, which emulates the original GBD Diet-NCD (2017) model results, DALYs are aggregated by summing across disease outcomes, dietary risks, age groups, and gender to arrive at a total per country.

For Analytical Scenarios 2 and 3, DALYs are aggregated by summing across diseases outcomes, gender, and age to obtain values per dietary risk. In *Unilateral_Shift.py,* DALY estimates can be computed either with or without accounting for mediation between dietary risks. When the flag *calculate_NJ_DALYs = True,* the script computes non-joint DALYs, treating each dietary risk independently and ignoring any overlap with other risks. When this flag is set to *False,* the script computes joint DALYs, which account for mediation between risks using mediation matrix, thereby avoiding double counting of shared causal pathways.

In addition, another script named *Unilateral_Shift_PJ.py,* decomposes the total joint DALY change across all risks into risk-specific contributions using proportional joint PAFs. This is particularly relevant for correlated risks – such as 'Diet low in calcium' and 'Diet low in milk'. Example outputs illustrating these distinctions are provided in the *Predictions* folder.

Lastly, the final marginal change script estimates marginal changes in DALYs resulting from a marginal change in the joint disease burden of dietary risks from change in a single dietary exposure. Results are produced for all time points, countries, and risks with a given SSP, and DALYs are disaggregated by dietary risk and age group (below 70 years and all ages).

9. Contact

For questions regarding the emulator, or to request access to the private code repository and associated data, please contact:

Dr Steven Lord

Senior Researcher Food System Economics

Food Systems Transformation Group

Environmental Change Institute, University of Oxford

Email: steven.lord@ouce.ox.ac.uk

Shaun Solomon

Research Programmer and Data Analyst for Food System Economic Cost Modelling

Food Systems Transformation Group

Environmental Change Institute, University of Oxford

Email: shaun.solomon@ouce.ox.ac.uk

10. Citation

Lord, S., Solomon, S., & Paulus, E. (2026). *GBD Diet–NCD Model Emulator** [Computer software]. GitHub. URL forthcoming.

11. Authors

Dr Steven Lord, Environmental Change Institute, University of Oxford

Estelle Paulus, Environmental Change Institute, University of Oxford

Shaun Solomon, Environmental Change Institute, University of Oxford

12. Appendices

Appendix A – Schematic Diagrams

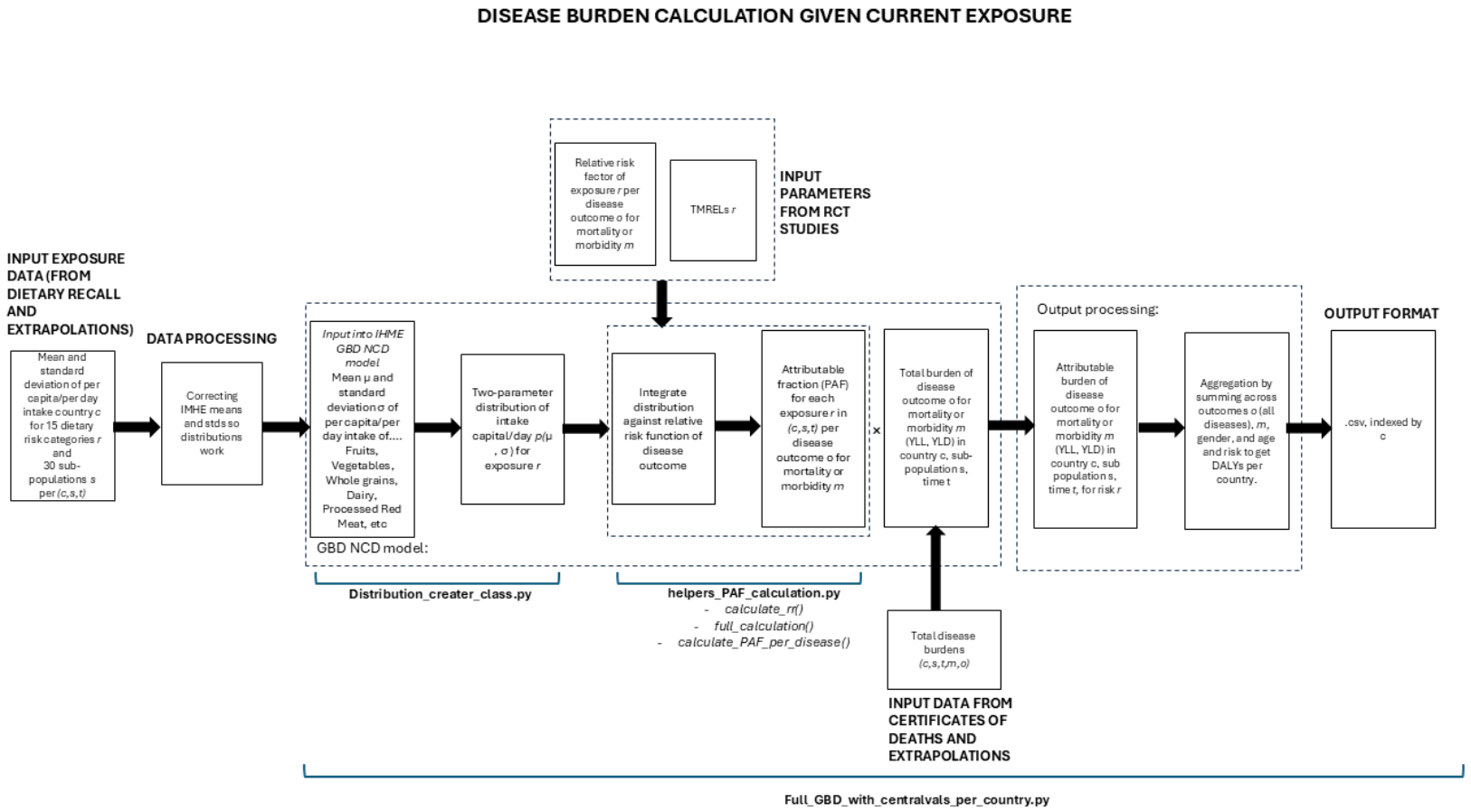

Figure 1. Original GBD; Analytical Scenario 1

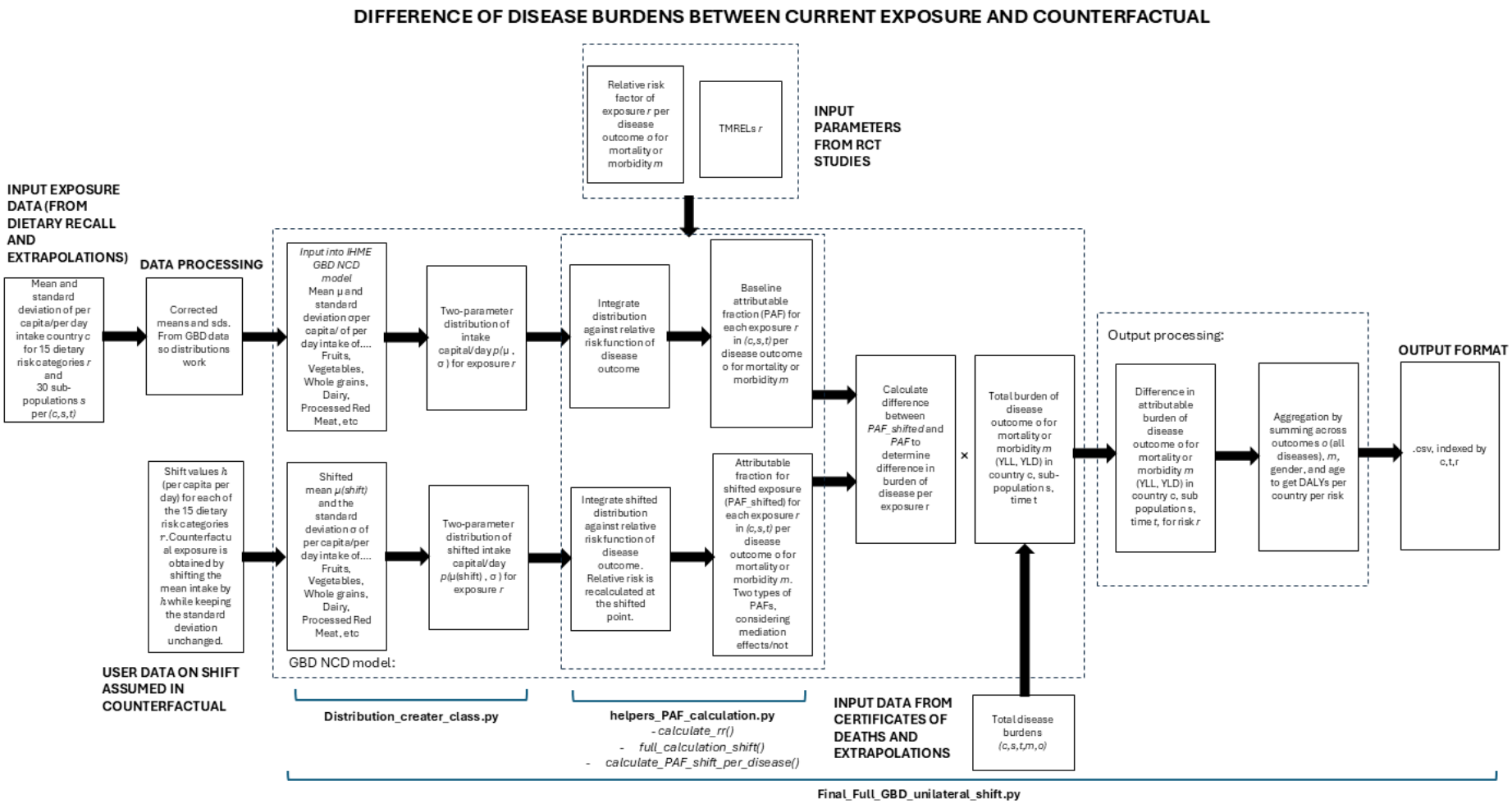

Figure 2. Unilateral Shift in Intake; Analytical Scenario 2

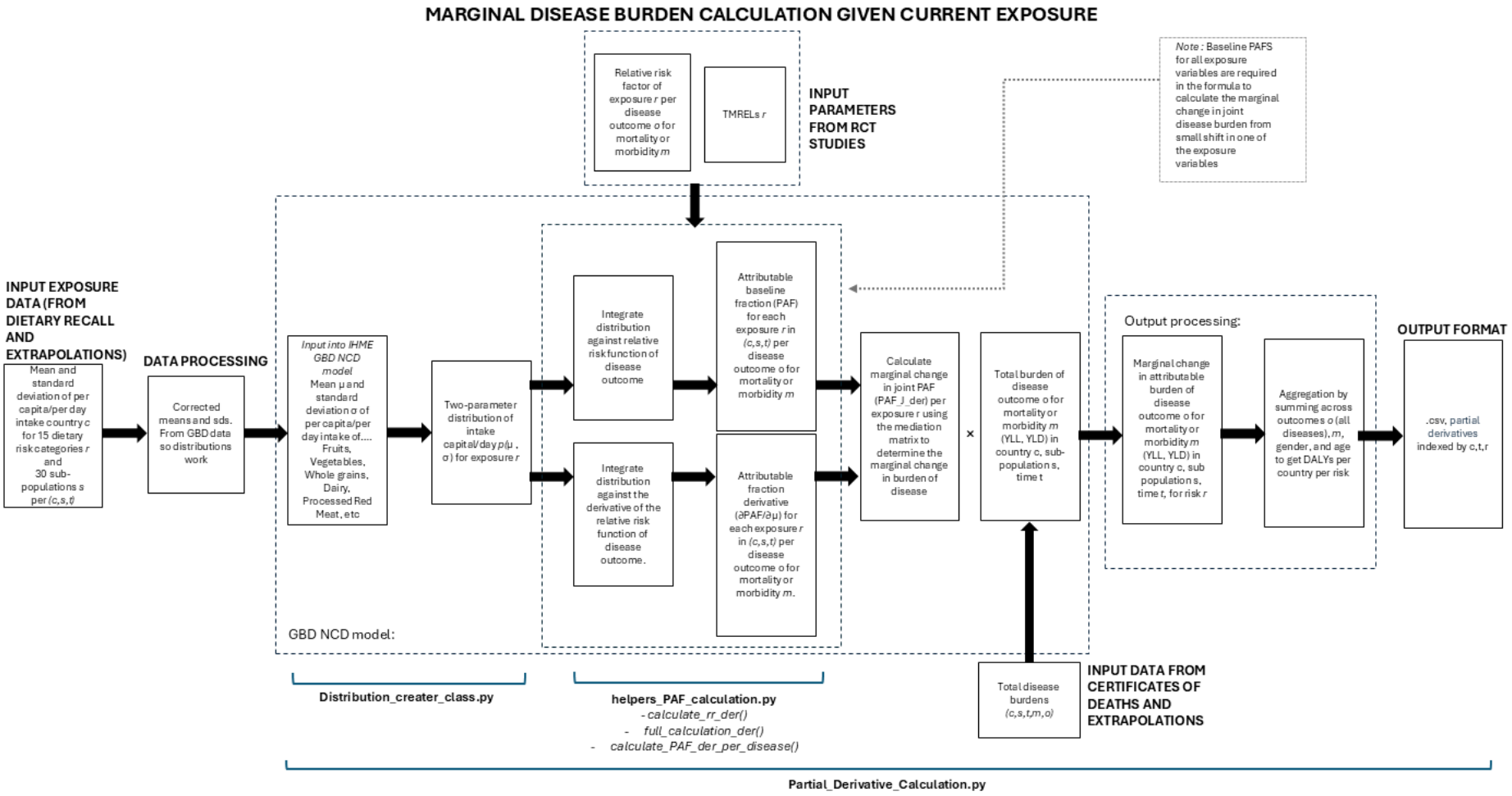

Figure 3. Marginal Disease Calculation; Analytical Scenario 3

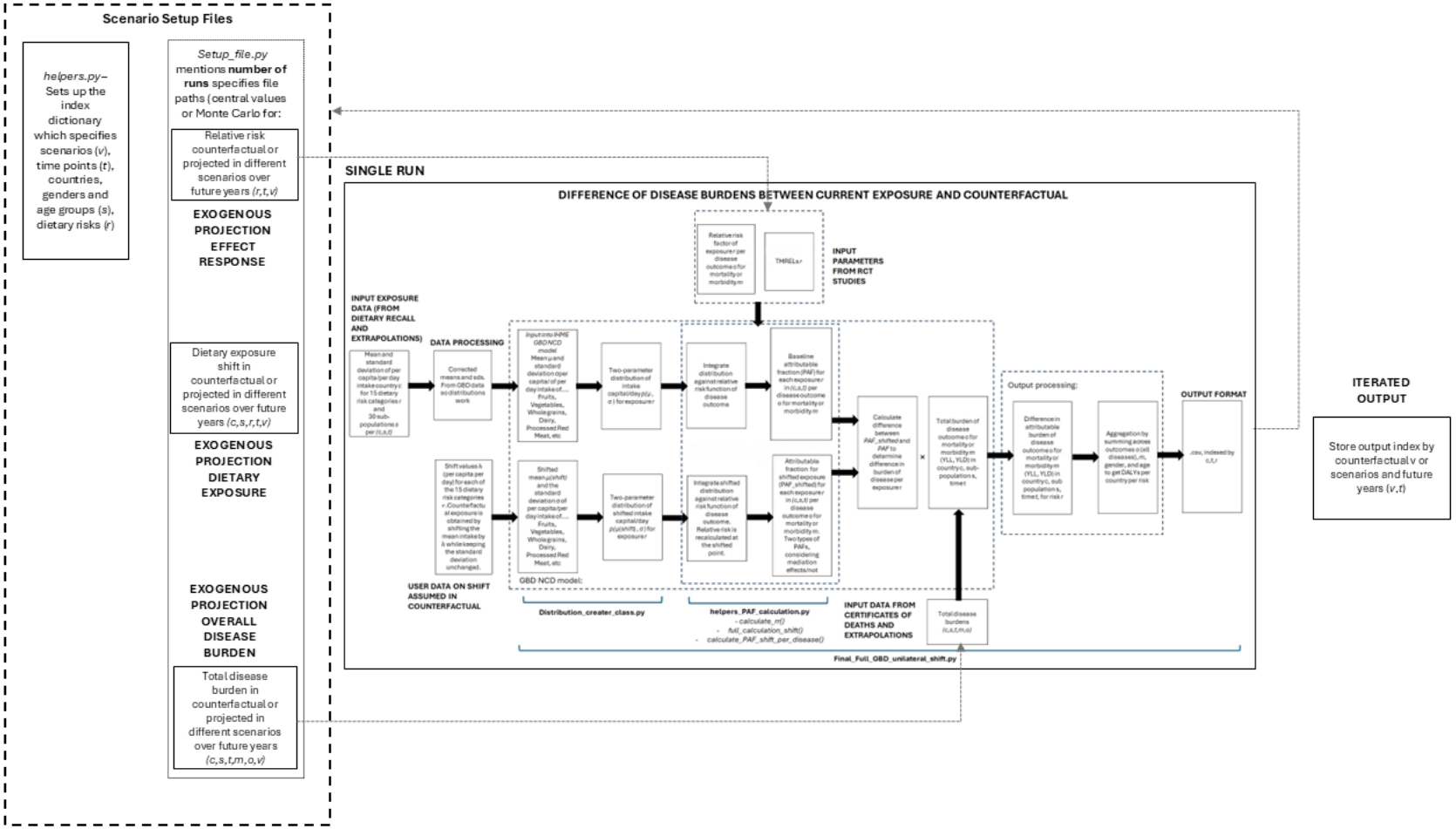

Figure 4. One-time run of the emulator

Appendix B – Documentation

The GBD model calculates the expected DALYs per disease, country, age, and sex from the intake values provided through the regression model. This version runs the model without doing a sampling for uncertainty assessments, to have fast results, which can be compared to the original results available through the GBD.

Helper Modules

Setup.py

Specifies the number of runs and the number of samples drawn for the uncertainty calculation. Contains paths for files to be loaded during the modelling process. Specifies the setup for a unilateral shift in intake, as well as the setup for a unilateral shift in supply. The shift in intake was mainly for testing reasons.

helpers.py

Specifies the index dictionary, which contains all important information needed for modelling. Specifies country names, diseases, risks, ages, and genders, as well as categories for marginal change scenarios.

helpers_data_and_setup.py

It contains functions for loading the relevant data and preparing it into the right format from the GBD and the regression model. Contains the function calculating the mediation matrix.

helpers_variables_calculation.py

Contains functions for calculating parameters (alpha, beta, k, and lambda, etc.) for the different probability distributions (Beta, Fisk/Log-Logistic, Weibull, Inverse Weibull) based on the mean and standard deviation values. Some parameters are calculated using optimization. It provides helper functions for specific distribution parameter calculations under certain conditions.

Variable_creater_class.py

The VariableCreator calculates the parameters for the probability distributions which cannot be computed from mean and standard deviation directly (Beta, Weibull and Fisk distributions). Those parameters are calculated using optimization methods. Final parameter values for each country, risk, age and sex are stored in a data frame.

1. **Constructor (__init__):** The constructor initializes the `VaribaleCreator` object with risks, age groups and genders. It automatically initializes and calculates the parameters from the mean, standard deviation, and upper and lower limit and stores the resulting data frame as an attribute.
2. **Getting the parameters:** The _get_variables_dataframe returns the previously calculated attribute.

Uses: helpers.py
helpers_variables_caluculation.py

Distribution_creater_class.py

The DistributionCreator generates the probability distributions of dietary intake from given mean and standard deviation values. The class is initialized with the country, risks, age groups, genders, and sample size. The distributions are either created by taking input means and standard deviations directly (e.g., Exponential, Gamma, Gumbel, etc.) or by first calculating the parameters using the VariableCreator class (Beta, Fisk, Weibull): The probability distributions are combined to obtain the final distribution by resampling.

1. **Constructor (__init__):** The constructor initializes the `DistributionCreator` object with several parameters, including country, risks, age groups, genders, run number and optional sample size. It sets up empty data frames for distributions, parameters, and coefficients.
2. **Getting the parameters:** The _get_variables method uses the VariableCreator class to calculate the parameters for the distributions, which cannot be calculated from mean and standard deviations directly.
   The _get_parameters uses those variables as well as mean, standard deviation and upper and lower limit to store calculate the parameters of all distributions and store them.
3. **Getting the distributions:** The get_distributions method calls on private functions to calculate the final distribution by first calculating the individual probability distributions from their parameters and then performing a weighted resampling to obtain the final distribution.
4. **Getting the PDFs:** The get_pdfs method retrieves the probability density functions (PDFs) for distributions corresponding to a specific risk, age group, and gender.

Uses: helpers.py
Variable_creater_class.py

helpers_PAF_calculation.py

Helper file containing the functions needed for RR and PAF calculation. In Analytical Scenario 2 and 3, this helper file is used to calculate shifts in PAFs (given a unilateral shift in intake), derivatives of RRs and PAFs for calculation of marginals, respectively.

helpers_marginal_calculation.py

Contains the helper functions for marginal change calculations, loading or creating the relevant data frames for calculating and storing the DALY changes.

Main Scripts

Original_GBD.py

This code calculates the Disability-Adjusted Life Years (DALYs) attributable to the 15 dietary risks across countries, diseases, age groups, and genders for the original GBD (2017).

1. **General Setup:** Imports country code file and sets the risks, diseases, age groups and sex information.
2. **Country Setup**: It handles command-line arguments for specifying the range of countries to process.
3. **Loading Data**: Creates a dictionary including all risks and if they differ for mortality or morbidity. Loads data files including risk factors, TMRELs, mean values of YLDs and YLLs, mediation matrix 'MF', and input files.

4. **Start the Actual Calculation**: Initiates the core calculation by iterating through different countries. For each country and each run (number of runs set to 1 for this version):
    a. Loads YLD and YLL data.
    b. Loads mean and standard deviation values and calculates distribution.
    c. Loads risk factor distributions.
    d. Iterates through diseases and calculates Population Attributable Fractions (PAFs) for both morbidity and mortality.
    e. Calculates DALYs attributable to specific diseases based on PAFs.
    f. Saves DALYs across disease outcomes per country.
5. **Saving Results**: The code saves the results and other related data to CSV files.

Uses: *Distribution_creater_class.py*
helpers_data_and_setup.py
helpers_PAF_calculation.py

Unilateral_Shift.py

This code calculates changes in Disability-Adjusted Life Years (DALYs) attributable to individual dietary risks across time-points, countries, diseases, age groups, and genders for various scenarios given a unilateral shift in intake.

1. **General Setup:** Imports country code file and sets the scenarios, time-points, risks, diseases, age groups and sex information.
2. **Country Setup**: It handles command-line arguments for specifying the range of countries to process.
3. **Loading Data**: Creates a dictionary of all dietary risks and specifies whether they apply to morbidity, mortality, or both. The script loads all required input files, including risk factor parameters, TMRELs, shift values, mean YLD and YLL values, the mediation matrix (MF), and other model inputs. A configuration flag controls whether DALYs are calculated with or without accounting for mediation effects between dietary risks.
4. **Start the Actual Calculation**: Initiates the core calculation by iterating through different scenarios, time-points, and countries. For each scenario, time-point, country and run (number of runs set to 1 for this version):
    a. Loads mean and standard deviation values and calculates distribution.
    b. Loads risk factor distributions.
    c. Iterates through diseases and computes baseline PAFs, shifted PAFs, and the resulting differences for both morbidity and mortality.
    d. Leaves the risk loop once PAF arrays are populated. Re-enters the age-group loop and loads YLD and YLL data. These could be disease rate projections. Sets up the modified mediation matrix in accordance with the flag at the beginning of the nested loop.
    e. Either calculates joint or non-joint individual PAFs (per risk).
    f. The code calculates changes in DALYs resulting from the specified unilateral shift in intake for each scenario and time-point.
    g. Saves DALY changes per scenario, time point, country, and risk.
5. **Saving Results**: The code saves the results, including DALY changes and other related data to CSV files.

Uses*: Distribution_creater_class.py*

helpers_data_and_setup.py, helpers_PAF_calculation.py particularly *full_calculation_shift()* and *calculation_PAF_shift_per_disease()*

Unilateral_Shift_PJ.py

Unlike the previous script, this script calculates proportional joint DALY (PJ DALYs) and PJ DALY changes for individual dietary risks under a unilateral shift in intake, by decomposing the total joint disease burden change across all risks into risk-specific contributions. Outputs are produced across scenarios, time-points, countries, diseases, age groups, and sexes.

1. **General Setup:** Imports country code file and sets the scenarios, time-points, risks, diseases, age groups and sex information.
2. **Country Setup**: It handles command-line arguments for specifying the range of countries to process.
3. **Loading Data**: Creates a dictionary of all dietary risks and specifies whether they apply to morbidity, mortality, or both. The script loads all required input files, including risk factor parameters, TMRELs, shift values, mean YLD and YLL values, the mediation matrix (MF), and other model inputs.
4. **Start the Actual Calculation**: Initiates the core calculation by iterating through different scenarios, time-points, and countries. For each scenario, time-point, country and run (number of runs set to 1 for this version):
    a. Loads mean and standard deviation values and calculates distribution.
    b. Loads risk factor distributions.
    c. Iterates through diseases and computes baseline PAFs, shifted PAFs, and the resulting differences for both morbidity and mortality.
    d. Leaves the risk loop once PAF arrays are populated. Re-enters the age-group loop and loads YLD and YLL data. These could be disease rate projections. Sets up the modified mediation matrix.
    e. Calculates joint PAFs per risk (original and changes). Calculates a combined PAF for all 15 risks and computes proportional joint PAFs (per risk).
    f. Re-enters the risk loop, extracts the relevant proportional PAFs from the array, computes DALYs and stores it in an array.
    g. The code calculates changes in DALYs resulting from the specified unilateral shift in intake for each scenario and time-point.
    h. Saves DALY changes per scenario, time point, country, and risk.
5. **Saving Results**: The code saves the results, including DALY changes and other related data to CSV files

Uses: *Distribution_creater_class.py*
helpers_data_and_setup.py
helpers_PAF_calculation.py particularly *full_calculation_shift(), calculation_PAF_shift_per_disease()* and *calculate_PJ_PAFs()*

Partial_Derivative_Calculation.py

This code computes the marginal change in Disability Adjusted Life Years (DALYs) for the 15 dietary risks from 2020 to 2050 in 5-year time steps, using SSP specific mean exposures and disease burden projections.

1. **General Setup:** Imports country code file and sets the time-points, risks, diseases, age groups and sex information.
2. **Country Setup**: It handles command-line arguments for specifying the range of countries to process.

3. **Loading Data:** Loads all required inputs, including risk factor parameters, TMRELs, and the mediation matrix (MF). Creates a risk–disease mapping dictionary to restrict calculations to valid risk–outcome pairs.
4. **Start the Actual Calculation -** Iterates over years, and countries. For each year, country and run (set to 1 for central values in this version):
    a. Loads SSP specific means (while keeping SDs the same) for each country and generates intake distributions.
    b. Iterates over diseases, age groups, sexes, and risks and calculates baseline PAFs and PAF derivatives.
    c. Leaves the risk loop and re-enters the age-group loop. Loads SSP-specific YLL/YLD projections. Here the fully populated PAF arrays are used to compute mediation-adjusted marginal PAFs.
    d. Calculates marginal changes in DALYs for all 15 risks.
5. **Saving Results:** The code saves marginal changes in DALYs for all the time-points, risk, and country for the selected SSP.

Uses: *Distribution_creater_class.py*
helpers_data_and_setup.py
helpers_PAF_calculation.py particularly *calculate_rr_der(), full_calculation_der(),* and *calculation_PAF_der_per_disease()*

Appendix C – Extrapolations

Projections of Mean Dietary Risk Exposures

These projections were used as inputs for the marginal disease burden calculations by defining scenario-specific exposure trajectories against which marginal changes in disease burden were evaluated. In our calculations, the scenarios represent the five Shared Socioeconomic Pathways (SSPs), a set of narratives that describe future changes in human development, including changes in demographics, economy, technology and the environment (4) To project dietary risk exposures beyond the 2020 Global Burden of Disease (GBD) baseline, we used the dietary demand projections from a study (5) which generated SSP-consistent food demand trajectories using the MAgPIE integrated assessment model. These projections reflect income and population driven dietary transitions across three major food groups, including 'animal source foods', 'vegetables, fruits and nuts', and 'empty calories'.

Since the demand projections did not match the exposure variables of the GBD dietary composition model either in intake food group or nutrient detail or in disaggregation into age and sex groups in the population, translation from the demand module to exposure input was required.

Dietary intake trajectories are projected from 2020 to 2050 using proportional adjustment of present intake obtained from the demand projections. To operationalise the adjustment approach, a matrix was constructed to translate demand for agricultural products, provided by the MAgPIE model into the dietary risk exposure variables specified by the GBD framework (Table A1.). Demand, provided in 5-year time steps, includes projections for three broad food demand categories. These categories do not directly align with the GBD dietary risk exposures. The matrix reconciles differences in food group definitions between the two systems and was applied consistently across all time-points. As part of this approach, proportional change factors were calculated by comparing demand at each future time-point to the baseline year, for each category. These factors represent relative shifts in per capita food demand. Once mapped, these factors were used to scale the corresponding GBD baseline mean exposures, resulting in adjusted intake values for each dietary risk. This allowed for the generation of projected exposure inputs by country, age group, sex, and time-point, aligned with demand trends.

Table A1. Mapping from dietary risk factor to MAgPIE food demand category (β)

MAgPIE food demand categories	**GBD dietary risk factors**
Demand for animal source foods	Diet high in red meat
Demand for animal source foods	Diet low in milk
Demand for animal source foods	Diet low in calcium
Demand for empty calories	Diet high in processed meat
Demand for empty calories	Diet high in sodium
Demand for empty calories	Diet high in sugar-sweetened beverages
Demand for empty calories	Diet high in trans fatty acids
Demand for vegetables, fruits and nuts	Diet low in fiber
Demand for vegetables, fruits and nuts	Diet low in fruits
Demand for vegetables, fruits and nuts	Diet low in legumes
Demand for vegetables, fruits and nuts	Diet low in nuts and seeds
Demand for vegetables, fruits and nuts	Diet low in whole grains
Demand for vegetables, fruits and nuts	Diet low in vegetables

To calculate the proportional change in per capita demand for MAgPIE food group g, in country c, at time t, relative to the 2020 baseline:

$$\propto_{gct} = \frac{D_{gct}}{D_{gc,2020}}$$

Where, D_{gct} and $D_{gc,2020}$ are per capita demand values for year t and 2020 respectively.

To reflect heterogeneity in dietary transitions across SSPs, these proportional demand change factors were further adjusted by World Bank income group and SSP. SSP specific adjustment targets were applied to each food category as linear scaling factors over time, with magnitudes varying by income group. These adjusted proportional changes were then applied to the mapped GBD baseline exposures, yielding income and SSP-consistent projected dietary risk exposures. The SSP-specific adjustments used are specified in the accompanying code, and users may modify these parameters to generate alternative marginal change results.

To calculate the mean exposure μ to a dietary risk factor j, in country c, age group a, sex s, and year t, based on proportional changes in food demand g, we use:

$$\mu_{jasc,t} = \mu_{jasc,2020} \cdot \propto_{g=\beta(j),c,t}$$

Where $g = \beta(j)$ denotes the MAgPIE food demand category g corresponding to GBD risk category j in Table A1.

Disease Rate Projections

Upon request, the Institute for Health Metrics and Evaluation (IHME) at the University of Washington provides mean projections of Years of Life Lost (YLL) and Years Lived with Disability (YLD) for each country, disease outcome, and age group for the years 2022 to 2050. Since the GBD dietary composition model requires sex-disaggregated inputs, sex-specific projections were derived by redistributing total YLL and YLD values using male and female proportions from GBD 2021 estimates.

Projected YLLs and YLDs were subsequently rescaled using SSP-specific population projections. SSP2 was treated as the business-as-usual (BAU) scenario, and projections for all other SSPs were scaled relative to SSP2 by multiplying each future YLL and YLD value by the ratio of SSP population to SSP2 population for the corresponding country and year. This aligns disease-burden estimates with the demographic assumptions embedded in each SSP.

Together, these steps produced a harmonised set of future disease burden projections, stratified by country, age group, sex, disease outcome, and scenario year, that are internally consistent with both IHME's epidemiological projections and the SSP demographic framework required for the emulator.

13. References

1. Global Burden of Disease (GBD) [Internet]. [cited 2026 Jan 6]. Available from: https://www.healthdata.org/research-analysis/gbd

2. Afshin A, Sur PJ, Fay KA, Cornaby L, Ferrara G, Salama JS, et al. Health effects of dietary risks in 195 countries, 1990–2017: a systematic analysis for the Global Burden of Disease Study 2017. The Lancet. 2019 May 11;393(10184):1958–72.

3. Lord S, Solomon S. Notes on a global dataset of national marginal costs of environmental externalities and dietary risk internalities associated to food system activities under the shared socioeconomic pathways: FOODCoST SSPs Dataset. 2025 [cited 2026 Jan 7]; Available from: https://ora.ox.ac.uk/objects/uuid:4ae3f458-68c4-402c-9a8a-9ed7abf8cca8

4. O'Neill BC, Kriegler E, Ebi KL, Kemp-Benedict E, Riahi K, Rothman DS, et al. The roads ahead: Narratives for shared socioeconomic pathways describing world futures in the 21st century. Global Environmental Change. 2017 Jan 1;42:169–80.

5. Bodirsky BL, Dietrich JP, Martinelli E, Stenstad A, Pradhan P, Gabrysch S, et al. The ongoing nutrition transition thwarts long-term targets for food security, public health and environmental protection. Sci Rep. 2020 Nov 18;10(1):19778.