Technical Memorandum:
Methodology for Developing Default Dispersion Factors for MassDEP's Air Toxics Risk Screening Tool

Submitted to:
Air and Climate Programs Division
Attn: Joanne Morin
Massachusetts Department of Environmental Protection
1 Winter Street
Boston, MA 02108

Submitted by:
Abt Associates Inc.
6130 Executive Boulevard
Rockville, MD 20852

Contact(s):
Ambrish Sharma
Jonathan Dorn
Lisa McDonald

June 2023

Table of Contents

Introduction 1
Model Specifications and Input/Output Options 1
AERMOD Batch Runs – Setup and Automation 3
Model Output Analysis and Dispersion Factor Table Generation 6
R Code and Additional Statistical Analysis 13

Introduction

To assist the Massachusetts Department of Environmental Protection (MassDEP) in the evaluation of cumulative cancer and non-cancer risks from toxic air pollutants associated with new facilities or modifications at existing facilities, Abt Associates (Abt) supported the development of the Massachusetts Air Toxics Risk Screening Tool (MATRiST). The tool is based on the Minnesota Pollution Control Agency's (MPCA) Risk Assessment Screening Spreadsheet (RASS) with appropriate modifications to support MassDEP CIA Rule objectives. One modification was to update the default dispersion factor table incorporated into the RASS tool. Two methodologies were evaluated for developing the default dispersion factors – the one developed by MPCA and the other developed by Oregon Department of Environmental Quality to create lookup dispersion tables for the Cleaner Air Oregon Rule. The approach used by MPCA utilized AERSCREEN and MAKEMET to create meteorological data files that were then used in AERMOD screening mode to loop through various combinations of stack parameter inputs. The MPCA approach created worst case meteorology and potentially produced overly conservative results. In contrast, the Oregon approach used site-specific meteorological data collected in-state and utilized AERMOD in refined mode to model ten combinations of stack parameters each keyed to stack heights. Abt, under the direction of MassDEP adopted a hybrid approach that utilized the basic MPCA processing scheme but incorporated Massachusetts-specific meteorological data and ran AERMOD in refined mode. This approach simulated thousands of hypothetical input scenarios and generated a lookup table of conservative default dispersion factors.

As a first step in understanding risks from air toxic emissions, facilities can apply these dispersion factors to various combinations of stack height, receptor distance, and pollutant emission rates to generate screening level estimates of toxic air pollutant concentrations and associated risk values. Default dispersion factors were modeled for three averaging periods, 1-hr, 24-hr and annual, during the development of dispersion factor lookup tables. Dispersion factors for all three averaging periods were evaluated while deciding on the most appropriate averaging periods for dispersion factors to be used in the risk screening tool. This document provides the details of the batch processing scheme and inputs used to generate the default dispersion factor table included in MATRiST.

Model Specifications and Input/Output Options

This section describes the dispersion model and input/output options used to develop the default dispersion factors.

Dispersion model. AERMOD version 22112 was used for the dispersion modeling as it was the latest U.S. Environmental Protection Agency (EPA) approved version at the time the modeling was conducted in 2022. The model was run using EPA default regulatory options, including use of ADJ_U* (an option in AERMET for adjusting surface friction velocity for low wind/stable conditions), and assuming flat terrain and rural dispersion.

Receptor grid. A polar grid array was used for receptor location definition. The grid comprised of 36 radials at 10-degree increments with rings of receptors at the following distances from the stack:

- 10 to 100 meters (10-meter increments)
- 120 to 200 meters (20-meter increments)
- 250 to 400 meters (50-meter increments)
- 500 to 1,000 meters (100-meter increments)
- 1,500 meters
- 2,500 to 10,000 meters (2,500-meter increments)

> *The receptor distances are the same as those used by MPCA; however, instead of a linear cartesian grid, the receptors were placed in a polar grid arrangement. All receptor heights were kept at ground level, consistent with flat terrain.*

Figure 1 shows the polar array, with modeling receptors located at each modeled distance and radial for a total of 1,080 (36 × 30) receptors.

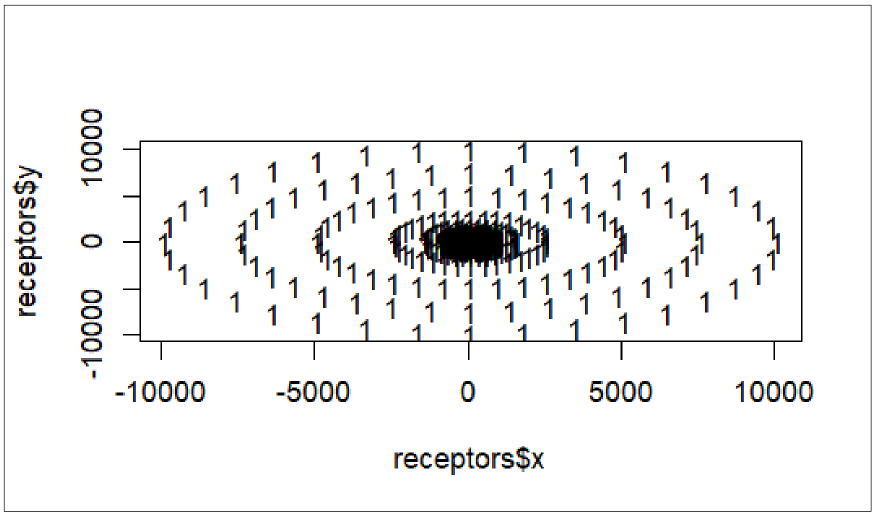


Figure 1. Representation of modeled receptors in polar grid arrangement.

Meteorological data. Meteorological data for calendar year 2018 for two representative Automated Surface Observing System (ASOS) sites in Massachusetts — Boston and Worcester – were used in the modeling. Boston Logan Airport (KBOS) surface data were coupled with Gray, Maine upper air data and Worcester Airport (KORH) surface data were coupled with Albany, New York upper air data and processed through AERMET to derive the AERMOD-ready datasets. Meteorological data collected at KBOS represents eastern Massachusetts while data collected at KORH represents central and western Massachusetts.

U-Star option. The modeling used meteorological datasets created using AERMET (meteorological preprocessor for AERMOD) with the ADJ_U* option turned on.

Surface type (e.g., urban vs. forested). This information was incorporated in the meteorological data processing step, so no input options were necessary in the hybrid scheme.

Building preprocessor. Building Profile Input Program for PRIME (BPIPPRM) inputs:

- Building height: 2 inputs: 90% stack height and 70% stack height

- Building distance: 2 inputs: 0 meters from stack and 25 meters from stack
- Maximum and minimum horizontal building dimension: 50 meters
- Maximum building dimension angle to true north (0°−179°): Set to 0
- Direction of stack from building center (0°−360°): Set to 0.

Dispersion Type. All scenarios were modeled with both rural and urban dispersion.

Emission Rate. Unit emission rate of 1 gram per second (g/s) was used to achieve the objective of producing dispersion factors in units of $\mu g/m^3$ per g/s.

Stack Height. Abt modeled a range of 1 to 60 meters (~3 to 200 feet) at 1-meter (~3 feet) increments for a total of 60 stack heights.

Stack Diameter. Abt modeled seven inputs (0.1 to 1.52 meters) representative of the process and combustion equipment typically found in Massachusetts.

Stack Exit Velocity. Abt modeled seven inputs (0.91 to 38.11 meters per second) representative of the process and combustion equipment typically found in Massachusetts.

Stack Temperature. Abt modeled five inputs (293 to 700°K) representative of the process and combustion equipment typically found in Massachusetts.

Terrain Height. Flat terrain was assumed. In AERMOD, base elevation (Z_s) was set to 0 meters for LOCATION keyword.

Averaging Period for Modeled Dispersion Factors. 1-hour, 24-hour, and annual.

Final Dispersion Factor Matrices (lookup tables). Two final lookup tables were generated, one each for rural and urban dispersion type. Each table consisted of 1800 data points (60 stack heights × 30 receptor distances) for each averaging period. For input to the risk tool, average results at each distance over the two site-specific meteorological locations was used.

Executing this modeling scheme required 235,200 AERMOD runs. 117,600 runs (58,800 each for rural and urban dispersion) per meteorological location times the two meteorological locations (KBOS and KORH).

AERMOD Batch Runs – Setup and Automation

Since the development of the default dispersion table required simulation of thousands of air dispersion scenarios, Abt used a prebuilt AERMOD package in R programming language to execute AERMOD batch runs. The package was developed by MPCA and is publicly available through a Github code repository (GitHub - dKvale/aermod). The R based AERMOD package provides a data frame interface for performing AERMOD batch runs from R and contains libraries of various functions utilized in AERMOD simulations. Abt modified the MPCA base code to perform the intended dispersion modeling with the Massachusetts-specific inputs and parameterization described in the section above.

The major modifications to the MPCA base code included:

- **Removed AERSCREEN module from MPCA base code.** MPCA utilized AERSCREEN to create land surface and building specific MAKEMET meteorological data files that were then used in AERMOD (in screening mode) to loop through various combinations of stack parameter inputs. However, MAKEMET uses default meteorological data and does not allow the user to define site-specific meteorological data. To generate 1-hour, 24-hour and annual dispersion factors using site-specific meteorology and a polar receptor grid, Abt modified the MPCA base code to use AERMOD ready site-specific meteorology with integrated surface characteristic information.

- **Utilized R modules to run BPIPPRM.** The AERMOD preprocessor BPIPPRM includes an algorithm for calculating a matrix of five unique building downwash parameters (input direction specific building heights (BUILDHGT), widths (BUILDWID), lengths (BUILDLEN), and XBADJ and YBADJ - the input direction-specific along-flow and across-flow distances from the stack to the center of the upwind face of the projected building respectively) for input into the PRIME algorithm contained in AERMOD for plume rise calculations. Since AERSCREEN (which runs BPIPPRM) was removed from the code, Abt used an R sub-package contained in the AERMOD package from MPCA to run BPIPPRM. This sub-package was installed, and specific functions were written into the MassDEP code to (1) run BPIPPRM; (2) read and extract the BPIPPRM outputs; and (3) input the BPIPPRM output values into AERMOD before looping through stack variations. An example of the BPIPPRM output is provided in Figure 2. The information contained in the red box is needed by AERMOD for PRIME calculations. The MassDEP code extracts this portion from the BPIPPRM output file and pastes it into the AERMOD input file.

```
  * Results are based on Determinants 1 & 2 on pages 1 & 2 of the GEP
    Technical Support Document.  Determinant 3 may be investigated for
    additional stack height credit.  Final values result after
    Determinant 3 has been taken into consideration.
 ** Results were derived from Equation 1 on page 6 of GEP Technical
    Support Document.  Values have been adjusted for any stack-building
    base elevation differences.

    Note:  Criteria for determining stack heights for modeling emission
    limitations for a source can be found in Table 3.1 of the
    GEP Technical Support Document.



                           BPIP (Dated: 04274)
DATE :  9/27/2022
TIME :  3:26:51

bpip

 BPIP output is in meters
```

```
SO BUILDHGT STACK      13.50   13.50   13.50   13.50   13.50   13.50
SO BUILDHGT STACK      13.50   13.50   13.50   13.50   13.50   13.50
SO BUILDHGT STACK      13.50   13.50   13.50   13.50   13.50   13.50
SO BUILDHGT STACK      13.50   13.50   13.50   13.50   13.50   13.50
SO BUILDHGT STACK      13.50   13.50   13.50   13.50   13.50   13.50
SO BUILDHGT STACK      13.50   13.50   13.50   13.50   13.50   13.50
SO BUILDWID STACK      57.92   64.09   68.30   70.44   70.44   68.30
SO BUILDWID STACK      64.09   57.92   50.00   57.92   64.09   68.30
SO BUILDWID STACK      70.44   70.44   68.30   64.09   57.92   50.00
SO BUILDWID STACK      57.92   64.09   68.30   70.44   70.44   68.30
SO BUILDWID STACK      64.09   57.92   50.00   57.92   64.09   68.30
SO BUILDWID STACK      70.44   70.44   68.30   64.09   57.92   50.00
SO BUILDLEN STACK      57.92   64.09   68.30   70.44   70.44   68.30
SO BUILDLEN STACK      64.09   57.92   50.00   57.92   64.09   68.30
SO BUILDLEN STACK      70.44   70.44   68.30   64.09   57.92   50.00
SO BUILDLEN STACK      57.92   64.09   68.30   70.44   70.44   68.30
SO BUILDLEN STACK      64.09   57.92   50.00   57.92   64.09   68.30
SO BUILDLEN STACK      70.44   70.44   68.30   64.09   57.92   50.00
SO XBADJ    STACK      -4.34   -8.55  -12.50  -16.07  -19.15  -21.65
SO XBADJ    STACK     -23.49  -24.62  -25.00  -33.30  -40.59  -46.65
SO XBADJ    STACK     -51.29  -54.37  -55.80  -55.54  -53.58  -50.00
SO XBADJ    STACK     -53.58  -55.54  -55.80  -54.37  -51.29  -46.65
SO XBADJ    STACK     -40.59  -33.30  -25.00  -24.62  -23.49  -21.65
SO XBADJ    STACK     -19.15  -16.07  -12.50   -8.55   -4.34    0.00
SO YBADJ    STACK       4.34    8.55   12.50   16.07   19.15   21.65
SO YBADJ    STACK      23.49   24.62   25.00   24.62   23.49   21.65
SO YBADJ    STACK      19.15   16.07   12.50    8.55    4.34    0.00
SO YBADJ    STACK      -4.34   -8.55  -12.50  -16.07  -19.15  -21.65
SO YBADJ    STACK     -23.49  -24.62  -25.00  -24.62  -23.49  -21.65
SO YBADJ    STACK     -19.15  -16.07  -12.50   -8.55   -4.34    0.00
```

Figure 2. Snippet from a BPIPPRM output file generated using R-based BPIPPRM module during AERMOD batch simulations.

- **Linking building heights with stack heights in BPIPPRM.** The R-based BPIPPRM module provides an option to supply user defined inputs to BPIPPRM such as building height, maximum and minimum building horizontal dimension, distance between stack and building center, etc. One important consideration was to ensure the accurate linkage between modeled stack heights and building heights. This was achieved by adding a FOR loop in the MassDEP code encompassing all stack height variations before the BPIPPRIM code and linking the building height input parameter required by BPIPRIM to the current stack height in the loop using predefined fractions of 70% and 90%.

- **Polar grid receptors.** Instead of linear cartesian grid receptors used in the MPCA base code, Abt used polar grid receptors with the receptor locations described above in Figure 1. Since the receptor definition remains static across all simulations in the batch run, the definition was added to the default AERMOD input file. This file is read by the MassDEP code when creating an AERMOD input object. This object is later used to modify building parameters and stack variations. The receptor definition is shown below:

```
RE STARTING
RE GRIDPOLR UPOL1 STA
          ORIG STACK
          DIST 10 20 30 40 50 60 70 80 90 100 120 140 160 180 200
          DIST 250 300 350 400 500 600 700 800 900 1000 1500 2500 5000
          DIST 7500 10000
          GDIR 36 0.00 10.00
  GRIDPOLR UPOL1 END
RE FINISHED
```

- **Utilized MA-specific meteorological data instead of MAKEMET.** Instead of MAKEMET, Abt used site-specific real-world meteorology when developing the default air dispersion factors. Two AERMOD batch simulations were run, one using Worcester surface meteorology and the other using Boston surface meteorology. Since the meteorological inputs are static across all simulations, meteorological input definitions were added to the respective default AERMOD input files. An example of the meteorological input definition is shown below:

```
ME STARTING
  SURFFILE  WORALB_2018.sfc
  PROFFILE  WORALB_2018.pfl
  SURFDATA  94746  2018  WORCESTER,MA
  UAIRDATA  54775  2018  ALBANY,NY
  PROFBASE  0.0  METERS
ME FINISHED
```

- **Modeled 1-hour, 24-hr and annual concentrations.** The MPCA base code ran AERMOD in screening mode which only allows modeling of a 1-hour averaging period. The 24-hour and annual dispersion values used in the MPCA approach are based on scaling factors applied to modeled 1-hour values. In contrast, Abt used AERMOD in refined mode which allows the calculation of 24-hour and annual averages for modeled concentrations, in addition to 1-hour values. In the refined mode, this was accomplished using the AVERTIME card in the control option of the AERMOD input file.

- **Generated unique output file for each scenario.** A total of 235,200 scenarios were modeled using AERMOD, corresponding to variations in the following input parameters:
    - 60 stack heights: [1 to 60 in 1m increments] m
    - 7 stack diameters: [0.10, 0.20, 0.30, 0.46, 0.61, 0.91, 1.52] m
    - 5 stack exit temperatures: [293, 311, 366, 450, 700] Kelvin
    - 7 stack exit velocities: [0.91, 3.05, 9.15, 15.24, 22.87, 30.49, 38.11] m/s
    - 2 building distances: [0, 25] m
    - 2 building height fractions: [70%, 90%] of stack height
    - 2 dispersion types: rural and urban
    - 2 meteorological sites: Boston, MA (KBOS) and Worcester, MA (KORH)

  The MPCA base code did not store the output of each scenario modeled; instead, it read the output of each modeled scenario and extracted information to update the final lookup table as part of code execution. Abt added a unique capability to the code to store the AERMOD output of each modeled scenario. This enables future quality assurance / quality control (QA/QC) analysis and detailed accuracy checks of the modeling outputs. The naming convention of each output file is explained in Figure 3:

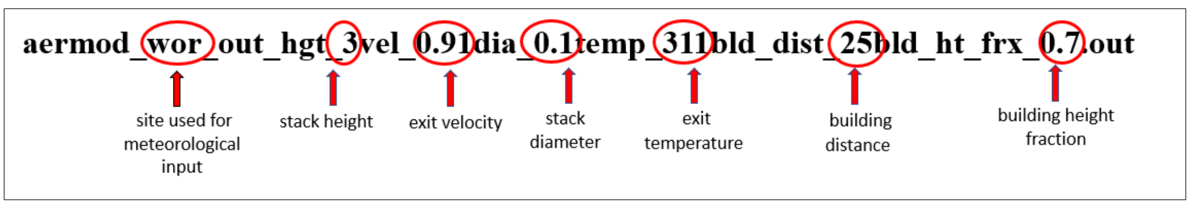


Figure 3. Output file naming convention.

- **Developed code for extracting data from output files and creating dispersion tables.** Due to significant modification to the MPCA base code, and multiple differences in output files (e.g., three averaging periods, modeled concentrations in polar grid format), an output analysis code in R was developed for parsing out relevant data from the output files and storing values in the dispersion factor table. The details of the code are discussed below.

Model Output Analysis and Dispersion Factor Table Generation

Abt developed an R module for analyzing the AERMOD batch run outputs and generating the final dispersion factor lookup table to be used in the MATRiST. Two blocks of AERMOD batch runs were performed – one each for the two meteorological stations with each block comprising 117,600 (58,800 rural and 58,800 urban dispersion) model runs.

Each model output file contained the 438th highest 1-hour average (95th percentile of 1-hr concentrations), 18th highest 24-hour average (95th percentile of 24-hr concentrations) values, and annual average values at each modeled receptor location in the polar grid format. The R code contains the following steps for generating the final dispersion factor lookup tables by processing the AERMOD outputs generated by the two blocks of AERMOD batch runs.

Step 1: Reading model output for each scenario

- From each output file that was processed, a single value (from 36 corresponding modeled directions) was extracted at each receptor distance for each averaging period.
- For the 1-hour and 24-hour modeled concentrations, the maximum of the 36 95th percentile values were extracted at each receptor distance (see Figure 4 for a representative output)
- For the annual modeled concentrations, and the maximum of the 36 annual average values was extracted at each receptor distance

Figure 4 shows the extraction of the maximum 95th percentile (18th highest) 24-hour concentration value from a representative AERMOD output. The highest of the 36 values shown in the red box was extracted for the receptor distance of 10 meters from this specific AERMOD output file. The process was repeated for all remaining 29 receptor distances to get the maximum value of the 18th highest 24-hour average concentrations across all modeled distances and directions.

```
                         *** THE  18TH HIGHEST 24-HR AVERAGE CONCENTRATION   VALUES FOR SOURCE GROUP:  ALL      ***
                             INCLUDING SOURCE(S):     STACK      ,

                              *** NETWORK ID: UPOL1    ;  NETWORK TYPE: GRIDPOLR ***

                                   ** CONC OF OTHER    IN MICROGRAMS/M**3                          **

DIRECTION |                                          DISTANCE (METERS)
(DEGREES) |           10.00                    20.00                    30.00                    40.00                    50.00
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

   360.0 |   3969.73392 (18092824)   1659.14295 (18071824)   1144.59027 (18062824)   1139.97903 (18061124)   1170.20517 (18090224)
    10.0 |   3871.73061 (18101124)   1834.75728 (18030324)   1316.23284 (18102324)   1190.66950 (18102324)   1252.61920 (18020424)
    20.0 |   3522.66851 (18022224)   1900.93189 (18051224)   1367.90951 (18051024)   1201.16808 (18060124)   1188.20619 (18011024)
    30.0 |   2978.47223 (18022224)   1686.79576 (18012224)   1193.26261 (18032524)   1009.82999 (18100924)   1007.24026 (18041224)
    40.0 |   2422.18568 (18022224)   1311.83983 (18053024)    844.69115 (18021424)    827.74804 (18070324)    840.81061 (18082524)
    50.0 |   1998.18387 (18052824)    937.60465 (18012124)    753.69691 (18082624)    692.35778 (18070524)    691.16282 (18061224)
    60.0 |   1637.69016 (18053024)    746.10249 (18042724)    616.35229 (18050224)    565.15133 (18012824)    652.45750 (18100924)
    70.0 |   1456.79195 (18081724)    636.17326 (18102224)    561.78685 (18111724)    566.72527 (18092324)    505.38695 (18101424)
    80.0 |   1464.26463 (18120424)    610.61942 (18022724)    669.80772 (18092324)    576.38228 (18022624)    402.40118 (18041724)
    90.0 |   1663.22440 (18093024)    819.77463 (18041924)    927.84542 (18020524)    573.27656 (18030924)    417.61918 (18120324)
   100.0 |   2031.18515 (18092924)   1189.22104 (18012824)   1103.31346 (18010224)    583.83806 (18041824)    450.79796 (18041824)
   110.0 |   2204.50445 (18102424)   1333.17903 (18120424)   1460.09788 (18122524)    643.68119 (18080524)    482.46240 (18020524)
   120.0 |   2511.70688 (18102424)   1692.09127 (18120224)   1864.47278 (18021224)    637.64665 (18102624)    466.63209 (18012524)
   130.0 |   3005.15222 (18113024)   2188.24958 (18061924)   1898.14157 (18112224)    640.42202 (18031824)    454.10872 (18031824)
   140.0 |   3405.67947 (18090824)   3159.52028 (18102124)   1550.55175 (18021124)    640.29206 (18021824)    437.05233 (18021824)
   150.0 |   4024.25111 (18020724)   3301.69316 (18122524)   1310.24393 (18032424)    630.14538 (18122524)    443.07961 (18021124)
   160.0 |   4615.61433 (18100624)   3233.05696 (18120424)   1169.63147 (18020724)    635.37805 (18100624)    455.73679 (18092324)
   170.0 |   4742.55545 (18012324)   2704.05822 (18022224)    943.87281 (18032724)    632.57087 (18020724)    453.29796 (18032724)
   180.0 |   5050.65862 (18121024)   1996.37955 (18090124)    977.53490 (18030524)    690.75902 (18082124)    496.19059 (18082124)
   190.0 |   4894.55509 (18051024)   2619.70079 (18022224)   1029.72360 (18051024)    711.69953 (18022224)    513.11007 (18022224)
   200.0 |   4701.93570 (18022224)   3402.04160 (18112624)   1122.54133 (18022224)    726.43207 (18051024)    525.99920 (18051024)
   210.0 |   3938.92049 (18081224)   2873.75023 (18022224)   1140.54961 (18052824)    697.77433m(18100824)    526.43433m(18100824)
   220.0 |   3242.30906 (18022224)   2478.31454 (18032124)   1149.47080 (18100124)    615.33845 (18052824)    473.17460 (18051324)
   230.0 |   2782.81926 (18091324)   1685.90970 (18092824)   1292.46263 (18092424)    566.13048 (18021124)    428.64598 (18021124)
   240.0 |   2143.91050 (18012924)   1272.92494 (18070224)   1348.30832 (18042724)    569.53519 (18082024)    446.07750 (18083124)
   250.0 |   1860.93540 (18081324)    993.85275 (18042724)   1128.28934 (18081224)    556.14140 (18061724)    415.24771 (18071124)
   260.0 |   1717.70317 (18091924)    966.90402 (18110524)   1021.92978 (18022024)    559.24165 (18091424)    434.41962 (18042524)
   270.0 |   1488.40484 (18120224)    782.67495 (18051824)    929.34463 (18080824)    576.25675 (18091124)    422.02459 (18060324)
   280.0 |   1480.12395 (18120124)    631.96224 (18051624)    642.79068 (18060324)    579.96123 (18060224)    406.48918 (18050824)
   290.0 |   1632.75565 (18061924)    744.33318 (18122524)    572.87059 (18071524)    532.73557 (18042824)    527.50941 (18042824)
   300.0 |   1942.90155 (18120224)    957.62940 (18032824)    622.15466 (18080524)    487.42264 (18081524)    545.72597 (18081524)
   310.0 |   2188.80459 (18021124)   1223.05208 (18120824)    763.07931 (18120824)    539.54467 (18063024)    419.53931b(18121324)
   320.0 |   2554.04095c(18123024)   1468.53067 (18113024)    962.02521 (18061924)    626.81513 (18021224)    501.18539 (18032924)
   330.0 |   3105.57827 (18020724)   1798.32276 (18120224)   1160.28141 (18120224)    752.73279 (18092224)    561.43523 (18102324)
   340.0 |   3531.47658b(18121324)   2269.77072 (18011724)   1392.41432 (18090824)    955.99240 (18121124)    761.50960c(18122724)
   350.0 |   3662.71569 (18070424)   2085.55979 (18061924)   1309.83839 (18112524)   1011.80709 (18060124)    849.45776 (18101524)
 *** AERMOD - VERSION 22112  ***   *** A Simple Example Problem for the AERMOD Model with PRIME            ***        12/22/22
 *** AERMET - VERSION  19191 ***    ***                                                                    ***        04:58:09
```

Figure 4. Extraction of the maximum of the 95th percentile (18th highest) 24-hr concentration values at one of the receptor distances from a representative AERMOD output.

Step 2: Storing values calculated in step 1 in extended dispersion lookup tables

- Separate extended format lookup tables were generated for each of the three averaging periods, employing the methodology described in step 1, first using the Boston output datasets (rural and urban), and then using the Worcester output datasets (rural and urban).
- Each extended table stored the extracted output information from step 1 and included the corresponding input parameter information.
- As a result, each extended format table contained 58,800 rows (58,800 scenarios were modeled for each meteorological site-dispersion type combination).
- For better processing efficiency, the analysis code was broken down into 12 modules to handle a unique combination of averaging period (three values), meteorological sites used (two values), and dispersion type (two values).

- 12 extended dispersion tables were generated at the end of step 2, six for rural dispersion type (1-hour Boston, 1-hour Worcester, 24-hour Boston, 24-hour Worcester, Annual Boston, Annual Worcester) and similar six combinations for urban dispersion. See Figure 5 for a subset of the 24-hour Boston rural dispersion table.

  Figure 5 shows a preview of the extended format lookup table generated for 24-hour values extracted from the 58,800 scenarios modeled with Boston meteorology and rural dispersion. Each of the 58,800 rows in the table have information on the input parameters of the modeled scenario and the maximum of the modeled 18th highest 24-hour average dispersion factors across the 36 directions at each receptor distance for that scenario.

	A	B	C	D	E	F	G	H	I	J
1	stack_height	stack_exit_velocity	stack_diameter	stack_exit_temp	building_dist	building_ht_frx	Aver_Period	10	20	30
2	3	0.91	0.1	293	0	0.7	24-hr	3160.51332	2627.26391	2455.82324
3	3	0.91	0.1	293	0	0.9	24-hr	2565.8864	2181.04697	2058.10658
4	3	0.91	0.1	293	25	0.7	24-hr	6242.55419	3757.36522	1668.1329
5	3	0.91	0.1	293	25	0.9	24-hr	5050.65862	3402.0416	1898.14157
6	3	0.91	0.1	311	0	0.7	24-hr	3143.17807	2619.96675	2448.07419
7	3	0.91	0.1	311	0	0.9	24-hr	2547.39391	2170.14024	2056.6571
8	3	0.91	0.1	311	25	0.7	24-hr	5421.71382	3667.48049	1558.98835
9	3	0.91	0.1	311	25	0.9	24-hr	4382.41438	3232.58341	1830.29996
10	3	0.91	0.1	366	0	0.7	24-hr	3109.32859	2589.76778	2417.47535
11	3	0.91	0.1	366	0	0.9	24-hr	2503.81655	2158.54038	2048.78658
12	3	0.91	0.1	366	25	0.7	24-hr	4989.74829	3520.60073	1517.85659
13	3	0.91	0.1	366	25	0.9	24-hr	3815.17611	2922.34135	1756.45091
14	3	0.91	0.1	450	0	0.7	24-hr	3072.73882	2574.9812	2409.96185
15	3	0.91	0.1	450	0	0.9	24-hr	2462.42526	2148.3232	2017.09359

Figure 5. Preview of extended format lookup table generated for the 24-hour values extracted from the 58,800 scenarios modeled with Boston meteorology and rural dispersion.

Step 3: Creating two-dimensional dispersion tables with statistical analyses

- Each extended table generated in step 2 gave 980 dispersion factor options (from 7 stack diameters x 5 stack exit temperatures x 7 stack exit velocities x 2 building distances x 2 building height fractions modeled) for each stack height- boundary distance combination.
- Statistical analyses were conducted on each block of such 980 factors to calculate the maximum, minimum, average, 25th, 50th, 75th, and 90th percentile values for each stack height boundary distance combination from the 12 extended format tables

- This produced 12 two-dimensional tables representing the statistical information for each block analyzed in step 3 from the 12 respective extended tables from step 2. See Figure 6 for a preview of the two-dimensional dispersion factor table of summary statistics for the modeled 24-hour dispersion factors using Boston meteorological data and rural dispersion. Figure 7 shows a box plot of the minimum, maximum, average, 25th, 50th, and 75th percentile of the modeled 24-hour maximum dispersion factors for the Boston meteorological station using rural dispersion for eight receptor distances using a stack height of 3 meters.

	A	B	C	D	E	F	G	H	I
1	stack_height	distance	max	min	quant25	quant50	quant75	quant90	avg
2	3	10	6242.55419	18.2981	611.767995	1056.02307	1712.257378	2425.521076	1262.728446
3	3	20	3757.36522	20.15104	698.77675	1209.279945	1763.916408	2214.87783	1278.826395
4	3	30	2455.82324	27.07869	642.3327025	1031.766515	1437.733403	1839.639033	1066.816462
5	3	40	2329.18807	17.44177	517.99098	924.59135	1273.530765	1641.196077	931.2516389
6	3	50	1442.98874	16.47156	320.5788275	482.58988	689.72577	1105.881002	555.5431062
7	3	60	1390.69169	13.04968	265.5368325	398.77939	690.8624275	1085.792285	507.365525
8	3	70	501.48992	12.77017	188.9928975	271.58176	372.0477975	446.578282	276.0680591
9	3	80	402.56085	10.88518	156.1724225	223.026585	304.6384075	349.187406	224.2449848
10	3	90	335.92892	9.76092	133.8057675	193.97288	260.67793	296.767908	192.1177927
11	3	100	290.76331	9.18417	116.817285	170.92274	226.9014725	257.417689	167.7080303
12	3	120	226.23728	6.8935	89.754455	136.679885	180.4681825	201.604171	132.3041467
13	3	140	184.63366	5.27986	72.3765875	111.68917	147.52986	164.982239	107.6860008
14	3	160	152.34228	4.2114	59.7343375	93.13392	123.3237375	139.229447	89.3964764
15	3	180	131.30292	3.62192	49.817685	79.24684	103.8570675	118.61305	75.34423658
16	3	200	113.38205	3.19734	41.777535	67.99996	90.1418875	101.694087	64.4922465
17	3	250	81.8724	2.4812	29.40981	48.5988	66.272245	74.766183	46.62804503
18	3	300	62.95874	2.0782	22.44333	36.895935	50.3948425	56.965632	35.66048673
19	3	350	49.5486	1.82147	17.872105	29.209545	40.1645925	45.109082	28.25814539
20	3	400	39.79732	1.78568	14.4383725	23.686605	32.869095	36.03035	23.0443255

Figure 6. Preview of the two-dimensional dispersion factor table of summary statistics for the modeled 24-hr dispersion factors using Boston meteorological data and rural dispersion.

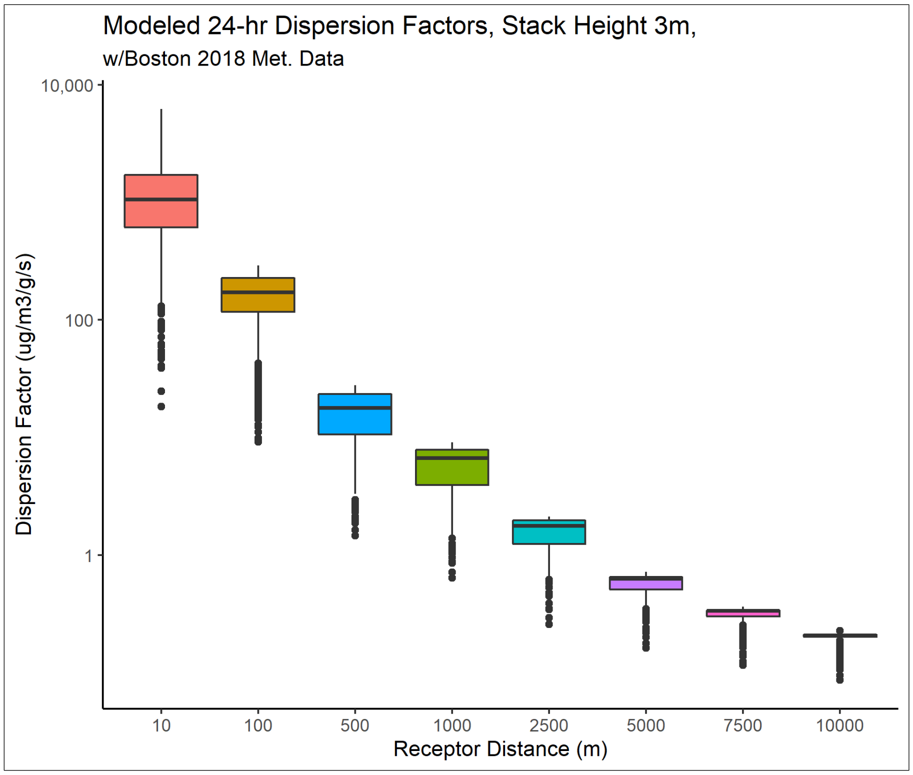


Figure 7. Box plot showing min, max, 25th, 50th and 75th percentile of the modeled 24-hour max dispersion factors (from Boston station rural output) for a stack height of 3 meters.

Step 4: Creating a consolidated lookup table with maximum dispersion factors for each station

- Maximum dispersion factors calculated in step 3, that represent the highest estimated pollutant concentration for each receptor distance-stack height combination, were then extracted from the 12 statistical tables from step 3 for further processing. By grouping all three averaging periods for a given meteorological station and dispersion type combination in one table, the 12 tables from step 3 were consolidated into four tables in step 4 - Boston rural (maximum dispersion), Boston urban (maximum dispersion), Worcester rural (maximum dispersion) and Worcester urban (maximum dispersion). Each of the newly generated four consolidated tables contain dispersion factors for all three averaging periods (see Figure 8).

Aver_Period	Stack_Height	Distance =	10	20	30	40	50	60	70	80
1-hr	46		70.60617	77.77217	83.97505	82.08926	76.88695	72.88617	69.85117	66.23845
1-hr	47		68.26517	75.34599	81.28591	79.65219	74.29074	70.52402	67.33676	63.99444
1-hr	48		66.09249	72.77078	78.78845	77.17543	71.8049	67.99659	65.01413	61.65737
1-hr	49		64.03286	70.43114	76.37978	74.48467	69.23565	65.69899	62.77375	59.50327
1-hr	50		61.89882	68.28175	73.86872	72.10395	66.82676	63.38787	60.70686	57.51401
1-hr	51		59.95903	65.96911	71.28167	69.84824	64.68395	61.24587	58.59517	55.57402
1-hr	52		57.94279	63.91962	69.25858	67.43946	62.72311	59.33841	56.68743	53.80673
1-hr	53		56.25948	61.86148	67.12903	65.55722	60.73404	57.48334	54.78797	52.20832
1-hr	54		54.33697	59.94137	65.1822	63.56579	58.7459	55.65852	53.05999	50.47593
1-hr	55		52.78171	58.1199	63.07246	61.64159	56.963	53.87288	51.27251	48.86535
1-hr	56		51.17836	56.42527	61.2411	59.87713	55.20872	52.15763	49.64871	47.33256
1-hr	57		49.78631	54.7082	59.41258	58.00012	53.52657	50.47722	48.01344	45.79685
1-hr	58		48.392	53.21352	57.67804	56.16342	51.83263	49.01022	46.67506	44.45742
1-hr	59		46.98272	51.61214	55.92537	54.56999	50.25366	47.45554	45.28558	43.07282
1-hr	60		45.70509	50.13471	54.3985	53.01621	48.78139	45.98225	43.87399	41.72971
24-hr	1		19392.11684	10093.45152	6407.56539	3972.15277	2952.4769	1914.00482	780.34987	629.46963
24-hr	2		11985.12086	5119.80739	3815.72876	2661.0309	1986.91678	1887.34605	605.33363	478.02861
24-hr	3		6242.55419	3757.36522	2455.82324	2329.18807	1442.98874	1390.69169	501.48992	402.56085
24-hr	4		4114.09687	2958.7083	1932.48923	1608.37772	1164.24885	1128.45688	767.24058	332.98824
24-hr	5		2939.2787	2116.85987	1614.60848	1373.84961	946.50975	918.44805	742.73991	294.12602
24-hr	6		2148.34975	1669.14516	1501.36425	1183.67101	913.67162	789.16372	731.61207	286.03709
24-hr	7		1708.42424	1394.11822	1275.52516	1036.99873	802.19677	698.91291	655.48744	374.45689
24-hr	8		1366.15029	1191.81449	1060.66736	924.16901	749.58155	626.85962	564.06316	398.70916
24-hr	9		1095.59104	961.51474	915.58381	825.57787	682.1547	554.9508	506.5936	394.78304
24-hr	10		919.10199	844.36371	817.88698	735.47925	624.22406	494.46108	453.5155	366.49511
24-hr	11		813.08664	753.33862	735.10631	653.61752	567.62402	447.65061	408.96755	358.79282
24-hr	12		715.2332	662.49844	658.4648	584.67925	513.31068	416.75932	371.34791	333.61697
24-hr	13		630.46676	586.36882	599.83392	529.3255	466.73835	378.85937	336.01969	313.12427
24-hr	14		562.78658	522.68321	533.15457	468.57998	420.1696	342.55231	304.25137	281.5018

Figure 8. Preview of consolidated Boston station rural dispersion factor table with 1-hr, 24-hr and annual dispersion factors stored in a 2-D format for all stack-heights and receptor distance combinations.

Step 5: Using the average of consolidated max dispersion lookup tables for final dispersion look-up tables

- The final rural dispersion table was created by averaging the dispersion factors from the two rural tables generated in step 4 (i.e., Boston rural and Worcester rural tables). Figure 9 shows the preview of the final rural dispersion table. Similarly, the final urban dispersion table was created by averaging the dispersion factors from the two urban tables generated in step 4.
- For averaging the two rural and two urban tables from step 4, all dispersion factors were grouped by specific receptor distance, stack height, and averaging period combination.

Aver_Period	Stack_Height	Distance =	10	20	30	40	50	60	70	80
1-hr	46		75.34005	82.60086	89.3473	88.52509	83.2858	78.79576	75.3391	71.70506
1-hr	47		72.76894	79.96964	86.37417	85.64321	80.55226	76.18003	72.74839	69.28523
1-hr	48		70.39028	77.24088	83.49758	82.86696	77.81916	73.65413	70.18362	66.99125
1-hr	49		68.1547	74.8165	80.7529	79.92303	75.05208	71.19354	67.86888	64.75258
1-hr	50		65.93136	72.46676	78.08564	77.28012	72.4427	68.73306	65.6767	62.58946
1-hr	51		63.76236	70.05533	75.47949	74.80911	70.15884	66.40394	63.50796	60.60021
1-hr	52		61.69708	67.76212	73.21634	72.33292	67.96102	64.28677	61.51918	58.69211
1-hr	53		59.87082	65.52594	71.03276	70.24219	65.82018	62.24063	59.51131	56.90148
1-hr	54		57.91868	63.55601	68.93812	68.15	63.77209	60.36002	57.68691	55.06995
1-hr	55		56.25349	61.5188	66.79498	66.05468	61.79934	58.43681	55.91162	53.31537
1-hr	56		54.52072	59.75364	64.76019	64.11251	59.90434	56.6312	54.13811	51.6304
1-hr	57		52.98589	57.9595	62.7503	62.13201	58.1082	54.92526	52.54258	50.05405
1-hr	58		51.42092	56.34924	60.89854	60.24167	56.33014	53.34349	51.0919	48.65706
1-hr	59		49.92626	54.68481	59.07002	58.52153	54.60777	51.80859	49.61091	47.1567
1-hr	60		48.56696	53.07415	57.3886	56.91286	53.05974	50.28642	48.1347	45.78715
24-hr	1		19037.2	10872.29	7609.345	4710.286	2477.23	1892.032	813.2976	651.8116
24-hr	2		11990.32	6044.235	4596.91	3854.581	1702.942	1690.175	597.6083	483.3033
24-hr	3		6698.222	4935.253	3032.941	2839.272	1328.88	1334.031	512.4455	411.6858
24-hr	4		4760.451	3925.984	2424.64	2050.663	1383.706	1121.99	767.0027	369.0501
24-hr	5		3428.451	2769.381	2209.695	1655.056	1202.004	946.2687	742.8008	342.6101
24-hr	6		2636.369	2218.32	2038.415	1397.997	1189.926	829.8904	708.3007	322.061
24-hr	7		2124.024	1868.626	1771.08	1210.887	1028.976	735.8194	638.1566	371.8872
24-hr	8		1704.056	1563.178	1476.921	1128.762	923.4444	689.8634	584.3541	401.8379
24-hr	9		1387.158	1269.156	1218.473	989.8077	821.1825	649.712	539.2108	398.028
24-hr	10		1154.747	1077.049	1017.116	889.5576	741.0757	593.8381	488.6216	374.2768
24-hr	11		975.7153	918.5411	861.3883	785.5702	669.794	574.7717	446.5431	367.0834
24-hr	12		818.8179	778.2074	745.9949	680.6182	603.6289	531.9664	402.2467	348.9978
24-hr	13		709.7193	672.7294	667.489	607.1156	544.9337	479.5378	379.6389	330.0552
24-hr	14		631.7212	601.2689	593.9931	539.2874	488.4691	429.4579	358.7684	302.4857

Figure 9. Preview of the final 1-hr, 24-hour and annual rural dispersion factor lookup values evaluated for inclusion in the Air Toxics Risk Screening Tool.

R Code and Additional Statistical Analysis

The R code used to generate the final dispersion factor table and additional statistical analyses are included in Appendix 1 and Appendix 2, respectively.

Appendix 1

R Codes Used to Develop Default Dispersion Factors for the Massachusetts Air Toxics Risk Screening Tool (MATRiST)

Description

This document contains a series of R codes used for performing batch runs of the air dispersion model AERMOD, analysis of AERMOD output and generation of dispersion factor tables for MassDEP's air toxics risk screening tool, MATRiST v2.0.

A1.1 is used for performing AERMOD batch runs using MA specific input parameters. The code shown here executes simulation of 58,800 scenarios run with Worcester 2018 Meteorology and urban dispersion.

A1.2-A1.4 are used for reading model output and generating extended lookup tables for 1-hr, 24-hr and annual concentrations respectively (execute step 1 and step 2 of model output analysis described in the technical memorandum). The codes shown here were used to process the model output generated using Worcester 2018 meteorology with urban dispersion. The same codes with minimal modifications could be applied to process the Worcester rural dispersion output and the counterpart rural and urban dispersion output sets generated using Boston 2018 meteorology.

A1.5 is used for creating two-dimensional dispersion tables with statistical analyses, then creating a consolidated lookup table with maximum dispersion factors for each averaging period for each station, and finally generating a final dispersion factor look-up table to be used in the risk screening tool (executes step 3, 4 and 5 of model output analysis described in the technical memorandum). This code was used to generate the final urban dispersion table used in the risk screening tool, MATRiST v2.0, same code could be applied to the extended rural dispersion tables to generate the final rural dispersion table.

A1.1 MADEP AERMOD batch runs code

Author: Ambrish Sharma, Abt Associates Inc.
Modified from base code developed by Dorian Kvale for MPCA, 2017
Purpose: Executing AERMOD batch runs with MA specific inputs and storing output of each unique scenario
Year: 2023

```
#Install AERMOD R packages from Github and load libraries

install.packages("remotes")

remotes::install_github(c("dKvale/installEPA",
"dKvale/bpip",
"dKvale/receptors",
"dKvale/aermod"))
library(installEPA)
library(bpip)
library(aermod)
library(knitr)
library(dplyr)
library(stringr)

# Define input parameter variables
sk_heights   <- c((1:60)
sk_diameters <- c(0.10, 0.20, 0.30, 0.46, 0.61, 0.91, 1.52)
sk_velocity  <- c(0.91, 3.05, 9.15, 15.24, 22.87, 30.49, 38.11)
tempsK <- c(293, 311, 366, 450, 700)
build_distance <- c(0, 25)
build_ht_frx   <- c(0.70, 0.90)

n_scenarios <- length(sk_heights) *
length(sk_diameters) *
length(sk_velocity) *
length(tempsK) *
length(build_distance) *
length(build_ht_frx)

# Set up receptor distances for final output dataframe

boundary_distances <- c(seq(10, 100, 10),
seq(120, 200, 20),
seq(250, 400, 50),
seq(500, 1000, 100),
seq(1500, 1500, 0),
seq(2500, 2500, 0),
seq(5000, 10000, 2500))

results_table=
expand.grid(stack_height=sk_heights,boundary_distance=boundary_distances,
stack_diameters=sk_diameters, exit_velocity=sk_velocity, exit_temp=tempsK,
disp_1hr          = NA,
disp_24hr         = NA,
```

```
disp_annual       = NA)

print(results_table)


# Begin loops for batch runs

for(bld_dist in build_distance) {

for(bld_ht_frx in build_ht_frx) {

for(height in sk_heights) {


# BPIP module section begins

test_build <- new_bpip()
test_build

# Update inputs of hypothetical building before bpip run
test_build$bld_rotation      <- 0
test_build$length_y          <- 50
test_build$width_x           <- 50
test_build$angle_from_source <- 0
test_build$dist_from_source  <- bld_dist
test_build$bld_height   <- height * bld_ht_frx  #link stack height with
building height
test_build$source_height   <- height
test_build$source_name   <- "STACK"


test_build

# Run BPIP

plot_bpip(test_build)
write_bpip(test_build,"bpip.inp")
run_bpip("bpip.inp", "building_results", exe_folder = "/user defined
directory with bpip executable/")

# Extract required section from bpip output
out <- readLines("building_results.out")
so_start <-min(grep("BUILDHGT", out))
out <- out[so_start:length(out)]
out<-str_trim(out, side = c("left"))

# Paste bpip output to AERMOD input object
aermod_inp    <- readLines("aermod_test.inp")
source_line_bld   <-min(grep("SO BUILDHGT stack   ", aermod_inp))
aermod_inp[source_line_bld:(source_line_bld+length(out)-1)]<-paste0(out)

aermod_inp

# BPIP module section ends
```

```
# Set loop counter
count <- 1

# Loop through remaining stack parameter variations


for(diam in sk_diameters) {

for(velocity in sk_velocity) {

for(tempk in tempsK) {



# Print iteration count
print(count)
count <- count + 1


# Update AERMOD input with stack parameters in loop
source_line   <- grep("   SRCPARAM  STACK  1.00   ", aermod_inp)

aermod_inp[source_line] <- paste0("   SRCPARAM  STACK  1.00    ",
height, "    ",

tempk, "    ",

velocity, "    ",

diam, "   ")


# Run AERMOD on input file and store unique output
      results <- run_aermod(aermod_inp,
      out_file   = paste0("aermod_wor_out_hgt_", height, "vel_", velocity,
      "dia_", diam, "temp_", tempk,"bld_dist_",bld_dist,"bld_ht_frx_",
      bld_ht_frx),
      exe_folder = "/user defined directory with aermod executable/")



}}}
}}
}
```

****

A1.2, A1.3, A1.4 AERMOD Output Analysis Codes
Authors: Ambrish Sharma and Caroline Watson, Abt Associates Inc.
Purpose: AERMOD output analysis – Reads model output and creates extended dispersion table for specific averaging period
Year: 2023

A1.2: Processes 1-hr output – uses output files from AERMOD batch runs using Worcester 2018 meteorology and urban dispersion.

```
library(dplyr)
library(stringr)
library(readr)
library(tidyr)
library(tidyverse)

setwd("/user defined working directory/")

# Read filenames of output files
filenames <- list.files(path = "/directory where output files are
stored/")

# Replace variable expressions in filenames for constant vector length
names_2 <- gsub("dia_0.1", "dia_0.10",
gsub("dia_0.2", "dia_0.20",
gsub("dia_0.3", "dia_0.30",
gsub("dist_0", "dist_00",
gsub("hgt_3vel", "hgt_03vel",
gsub("hgt_6vel", "hgt_06vel",
gsub("hgt_9vel", "hgt_09vel",
gsub("vel_0.91", "vel_00.91",
gsub("vel_9.15", "vel_09.15",
gsub("vel_3.05", "vel_03.05",filenames))))))))))

# Create dataframe for storing input parameter information from filenames
results_table <- tibble(stack_height = vector(mode = "numeric", length =
length(filenames)),
stack_exit_velocity = NA,
stack_diameter = NA,
stack_exit_temp = NA,
building_dist = NA,
building_ht_frx = NA)

# Create dataframe for extracted output information
df <- data.frame(matrix(ncol = 31, nrow = length(filenames)))

colnames(df) <- c("Aver_Period",
seq(10,100,10),
seq(120,200, 20),
seq(250,400,50),
seq(500,1000,100),
seq(1500,1500,0),
seq(2500,2500,0),
seq(5000,10000,2500))
```

```
# Start loop to extract input information from each filename and read
output
for(i in 1:length(filenames)){

results_table$stack_height[i] <- as.numeric(substr(names_2[i],20,21))
results_table$stack_exit_velocity[i] <-
as.numeric(substr(names_2[i],26,30))
results_table$stack_diameter[i] <- as.numeric(substr(names_2[i], 35,38))
results_table$stack_exit_temp[i] <- as.numeric(substr(names_2[i],44,46))
results_table$building_dist[i] <- as.numeric(substr(names_2[i],56,57))
results_table$building_ht_frx[i] <- as.numeric(substr(names_2[i],69,71))

print(i)

output <- readLines(paste0("directory with output files/", filenames[i]))


# 1-hour max extraction – Total 6 blocks for 1-hr output in model output

#block 1

onehr_output_blk1 <- output[576:611]
onehr_output_blk1 <- str_trim(onehr_output_blk1, side = c("both"))

onehr_output_blk1_final<-str_split_fixed(onehr_output_blk1," \\s+",6)
onehr_output_blk1_final <- as.data.frame(onehr_output_blk1_final)
column_names <- c("Direction", "10.00", "20.00", "30.00", "40.00",
"50.00")
names(onehr_output_blk1_final) <- column_names

onehr_output_blk1_final <- onehr_output_blk1_final %>%
mutate(Direction = str_trim(Direction, side = "right"),
Direction = sub("\\|", "", Direction)) %>%
pivot_longer(cols = c(2:6), names_to = "distance", values_to =
"one_hour_avg") %>%
mutate(one_hour_avg = str_replace(one_hour_avg, " \\s*\\([^\\)]+\\)", ""))
%>%
pivot_wider(names_from = distance, values_from = one_hour_avg) %>%
mutate_all(as.numeric)


#block 2
onehr_output_blk2 <- output[628:663]
onehr_output_blk2 <- str_trim(onehr_output_blk2, side = c("both"))

onehr_output_blk2_final<-str_split_fixed(onehr_output_blk2," \\s+",6)
onehr_output_blk2_final <- as.data.frame(onehr_output_blk2_final)
column_names <- c("Direction", "60.00", "70.00", "80.00", "90.00",
"100.00")
names(onehr_output_blk2_final) <- column_names

onehr_output_blk2_final <- onehr_output_blk2_final %>%
```

```
mutate(Direction = str_trim(Direction, side = "right"),
Direction = sub("\\|", "", Direction)) %>%
pivot_longer(cols = c(2:6), names_to = "distance", values_to =
"one_hour_avg") %>%
mutate(one_hour_avg = str_replace(one_hour_avg, " \\s*\\([^\\)]+\\)", ""))
%>%
pivot_wider(names_from = distance, values_from = one_hour_avg) %>%
mutate_all(as.numeric)


#block 3
onehr_output_blk3 <- output[680:715]
onehr_output_blk3 <- str_trim(onehr_output_blk3, side = c("both"))
onehr_output_blk3_final<-str_split_fixed(onehr_output_blk3," \\s+",6)

onehr_output_blk3_final <- as.data.frame(onehr_output_blk3_final)
column_names <- c("Direction", "120.00", "140.00", "160.00", "180.00",
"200.00")
names(onehr_output_blk3_final) <- column_names

onehr_output_blk3_final <- onehr_output_blk3_final %>%
mutate(Direction = str_trim(Direction, side = "right"),
Direction = sub("\\|", "", Direction)) %>%
pivot_longer(cols = c(2:6), names_to = "distance", values_to =
"one_hour_avg") %>%
mutate(one_hour_avg = str_replace(one_hour_avg, " \\s*\\([^\\)]+\\)", ""))
%>%
pivot_wider(names_from = distance, values_from = one_hour_avg) %>%
mutate_all(as.numeric)


#block 4
onehr_output_blk4 <- output[732:767]
onehr_output_blk4 <- str_trim(onehr_output_blk4, side = c("both"))
onehr_output_blk4_final<-str_split_fixed(onehr_output_blk4," \\s+",6)

onehr_output_blk4_final <- as.data.frame(onehr_output_blk4_final)
column_names <- c("Direction", "250.00", "300.00", "350.00", "400.00",
"500.00")
names(onehr_output_blk4_final) <- column_names

onehr_output_blk4_final <- onehr_output_blk4_final %>%
mutate(Direction = str_trim(Direction, side = "right"),
Direction = sub("\\|", "", Direction)) %>%
pivot_longer(cols = c(2:6), names_to = "distance", values_to =
"one_hour_avg") %>%
mutate(one_hour_avg = str_replace(one_hour_avg, " \\s*\\([^\\)]+\\)", ""))
%>%
pivot_wider(names_from = distance, values_from = one_hour_avg) %>%
mutate_all(as.numeric)


#block 5
onehr_output_blk5 <- output[784:819]
```

```
onehr_output_blk5 <- str_trim(onehr_output_blk5, side = c("both"))
onehr_output_blk5_final<-str_split_fixed(onehr_output_blk5," \\s+",6)

onehr_output_blk5_final <- as.data.frame(onehr_output_blk5_final)
column_names <- c("Direction", "600.00", "700.00", "800.00", "900.00",
"1000.00")
names(onehr_output_blk5_final) <- column_names

onehr_output_blk5_final <- onehr_output_blk5_final %>%
mutate(Direction = str_trim(Direction, side = "right"),
Direction = sub("\\|", "", Direction)) %>%
pivot_longer(cols = c(2:6), names_to = "distance", values_to =
"one_hour_avg") %>%
mutate(one_hour_avg = str_replace(one_hour_avg, "[a-z]", " "),
one_hour_avg = str_replace(one_hour_avg, " \\s*\\([^\\)]+\\)", "")) %>%
pivot_wider(names_from = distance, values_from = one_hour_avg) %>%
mutate_all(as.numeric)


#block 6
onehr_output_blk6 <- output[836:871]
onehr_output_blk6 <- str_trim(onehr_output_blk6, side = c("both"))
onehr_output_blk6_final<-str_split_fixed(onehr_output_blk6," \\s+",6)

onehr_output_blk6_final <- as.data.frame(onehr_output_blk6_final)

column_names <- c("Direction", "1500.00", "2500.00", "5000.00", "7500.00",
"10000.00")
names(onehr_output_blk6_final) <- column_names

onehr_output_blk6_final <- onehr_output_blk6_final %>%
mutate(Direction = str_trim(Direction, side = "right"),
Direction = sub("\\|", "", Direction)) %>%
pivot_longer(cols = c(2:6), names_to = "distance", values_to =
"one_hour_avg") %>%
mutate(one_hour_avg = str_replace(one_hour_avg, "[a-z]", " "),
one_hour_avg = str_replace(one_hour_avg, " \\s*\\([^\\)]+\\)", "")) %>%
pivot_wider(names_from = distance, values_from = one_hour_avg) %>%
mutate_all(as.numeric)


#Create data frame combining individual blocks and find max at each
distance
onehour_df <- left_join(onehr_output_blk1_final,
onehr_output_blk2_final, by = "Direction") %>%
left_join(onehr_output_blk3_final, by = "Direction") %>%
left_join(onehr_output_blk4_final, by = "Direction") %>%
left_join(onehr_output_blk5_final, by = "Direction") %>%
left_join(onehr_output_blk6_final, by = "Direction") %>%
pivot_longer(names_to = "distance", values_to = "one_hr_avg", cols =
c(2:31)) %>%
group_by(distance) %>%
summarize(one_hr_max = max(one_hr_avg)) %>%
mutate(distance = as.numeric(distance)) %>%
```

```
arrange(distance) %>%
pivot_wider(names_from = distance, values_from = one_hr_max) %>%
mutate(Aver_Period = "1-hr max") %>%
relocate(Aver_Period, .before = "10")

df[i,] <- onehour_df
}

#create final data frame with the stack parameter information
final_df <- cbind(results_table, df)

#write file to csv
write.csv(final_df, "user defined directory to store output dispersion
table/dispersion_table_wor_1hr_urban.csv", row.names = FALSE)
```

****

A1.3: Processes 24-hr output – uses output files from AERMOD batch runs using Worcester 2018 meteorology and urban dispersion.

```
library(dplyr)
library(stringr)
library(readr)
library(tidyr)
library(tidyverse)


setwd("/user defined working directory/")

# read filenames of output files
filenames <- list.files(path = "/directory where output files are
stored/")


# Replace variable expressions in filenames for constant vector length

names_2 <- gsub("dia_0.1", "dia_0.10",
           gsub("dia_0.2", "dia_0.20",
           gsub("dia_0.3", "dia_0.30",
           gsub("dist_0", "dist_00",
           gsub("hgt_3vel", "hgt_03vel",
           gsub("hgt_6vel", "hgt_06vel",
           gsub("hgt_9vel", "hgt_09vel",
           gsub("vel_0.91", "vel_00.91",
           gsub("vel_9.15", "vel_09.15",
           gsub("vel_3.05", "vel_03.05",filenames))))))))))

# Create dataframe for input parameter information
results_table <- tibble(stack_height = vector(mode = "numeric", length =
length(filenames)),
                        stack_exit_velocity = NA,
                        stack_diameter = NA,
                        stack_exit_temp = NA,
                        building_dist = NA,
```

```
                        building_ht_frx = NA)

# Create dataframe for extracted output information

df <- data.frame(matrix(ncol = 31, nrow = length(filenames)))
#df[1:length(names)] <- NA
colnames(df) <- c("Aver_Period",
                  seq(10,100,10),
                  seq(120,200, 20),
                  seq(250,400,50),
                  seq(500,1000,100),
                  seq(1500,1500,0),
                  seq(2500,2500,0),
                  seq(5000,10000,2500))

# Start loop to extract input information from each filename and read
output

for(i in 1:length(filenames)) {

    results_table$stack_height[i] <- as.numeric(substr(names_2[i],20,21))
    results_table$stack_exit_velocity[i] <-
as.numeric(substr(names_2[i],26,30))
    results_table$stack_diameter[i] <- as.numeric(substr(names_2[i],
35,38))
    results_table$stack_exit_temp[i] <-
as.numeric(substr(names_2[i],44,46))
    results_table$building_dist[i] <- as.numeric(substr(names_2[i],56,57))
    results_table$building_ht_frx[i] <-
as.numeric(substr(names_2[i],69,71))

    print(i)

  output <- readLines(paste0("directory with output files"/filenames[i]))

# 24-hour max extraction – Total 6 blocks for 24-hr output in model output

#block 1

twentyfourhr_output_blk1 <- output[888:923]
twentyfourhr_output_blk1 <- str_trim(twentyfourhr_output_blk1, side =
c("both"))
twentyfourhr_output_blk1_final<-str_split_fixed(twentyfourhr_output_blk1,"
\\s+",6)

twentyfourhr_output_blk1_final <-
as.data.frame(twentyfourhr_output_blk1_final)
column_names <- c("Direction", "10.00", "20.00", "30.00", "40.00",
"50.00")
names(twentyfourhr_output_blk1_final) <- column_names

twentyfourhr_output_blk1_final <- twentyfourhr_output_blk1_final %>%
mutate(Direction = str_trim(Direction, side = "right"),
Direction = sub("\\|", "", Direction)) %>%
```

```
pivot_longer(cols = c(2:6), names_to = "distance", values_to =
"twentyfour_hour") %>%
mutate(twentyfour_hour = str_replace(twentyfour_hour, "[a-z]", " "),
twentyfour_hour = str_replace(twentyfour_hour, " \\s*\\([^\\)]+\\)", ""))
%>%
pivot_wider(names_from = distance, values_from = twentyfour_hour) %>%
mutate_all(as.numeric)


#block 2
twentyfourhr_output_blk2 <- output[940:975]
twentyfourhr_output_blk2 <- str_trim(twentyfourhr_output_blk2, side =
c("both"))


twentyfourhr_output_blk2_final<-str_split_fixed(twentyfourhr_output_blk2,"
\\s+",6)
twentyfourhr_output_blk2_final <-
as.data.frame(twentyfourhr_output_blk2_final)

column_names <- c("Direction", "60.00", "70.00", "80.00", "90.00",
"100.00")
names(twentyfourhr_output_blk2_final) <- column_names

twentyfourhr_output_blk2_final <- twentyfourhr_output_blk2_final %>%
mutate(Direction = str_trim(Direction, side = "right"),
Direction = sub("\\|", "", Direction)) %>%
pivot_longer(cols = c(2:6), names_to = "distance", values_to =
"twentyfour_hour") %>%
mutate(twentyfour_hour = str_replace(twentyfour_hour, "[a-z]", " "),
twentyfour_hour = str_replace(twentyfour_hour, " \\s*\\([^\\)]+\\)", ""))
%>%
pivot_wider(names_from = distance, values_from = twentyfour_hour) %>%
mutate_all(as.numeric)



#block 3
twentyfourhr_output_blk3 <- output[992:1027]
twentyfourhr_output_blk3 <- str_trim(twentyfourhr_output_blk3, side =
c("both"))

twentyfourhr_output_blk3_final<-str_split_fixed(twentyfourhr_output_blk3,"
\\s+",6)

twentyfourhr_output_blk3_final <-
as.data.frame(twentyfourhr_output_blk3_final)

column_names <- c("Direction", "120.00", "140.00", "160.00", "180.00",
"200.00")
names(twentyfourhr_output_blk3_final) <- column_names

twentyfourhr_output_blk3_final <- twentyfourhr_output_blk3_final %>%
mutate(Direction = str_trim(Direction, side = "right"),
```

```
Direction = sub("\\|", "", Direction)) %>%
pivot_longer(cols = c(2:6), names_to = "distance", values_to =
"twentyfour_hour") %>%
mutate(twentyfour_hour = str_replace(twentyfour_hour, "[a-z]", " "),
twentyfour_hour = str_replace(twentyfour_hour, " \\s*\\([^\\)]+\\)", ""))
%>%
pivot_wider(names_from = distance, values_from = twentyfour_hour) %>%
mutate_all(as.numeric)

#block 4
twentyfourhr_output_blk4 <- output[1044:1079]
twentyfourhr_output_blk4 <- str_trim(twentyfourhr_output_blk4, side =
c("both"))
twentyfourhr_output_blk4_final<-str_split_fixed(twentyfourhr_output_blk4,"
\\s+",6)

twentyfourhr_output_blk4_final <-
as.data.frame(twentyfourhr_output_blk4_final)

column_names <- c("Direction", "250.00", "300.00", "350.00", "400.00",
"500.00")
names(twentyfourhr_output_blk4_final) <- column_names

twentyfourhr_output_blk4_final <- twentyfourhr_output_blk4_final %>%
mutate(Direction = str_trim(Direction, side = "right"),
Direction = sub("\\|", "", Direction)) %>%
pivot_longer(cols = c(2:6), names_to = "distance", values_to =
"twentyfour_hour") %>%
mutate(twentyfour_hour = str_replace(twentyfour_hour, "[a-z]", " "),
twentyfour_hour = str_replace(twentyfour_hour, " \\s*\\([^\\)]+\\)", ""))
%>%
pivot_wider(names_from = distance, values_from = twentyfour_hour) %>%
mutate_all(as.numeric)


#block 5
Twentyfourhr_output_blk5 <- output[1096:1131]
twentyfourhr_output_blk5 <- str_trim(twentyfourhr_output_blk5, side =
c("both"))

twentyfourhr_output_blk5_final<-str_split_fixed(twentyfourhr_output_blk5,"
\\s+",6)

twentyfourhr_output_blk5_final <-
as.data.frame(twentyfourhr_output_blk5_final)

column_names <- c("Direction", "600.00", "700.00", "800.00", "900.00",
"1000.00")
names(twentyfourhr_output_blk5_final) <- column_names

twentyfourhr_output_blk5_final <- twentyfourhr_output_blk5_final %>%
mutate(Direction = str_trim(Direction, side = "right"),
Direction = sub("\\|", "", Direction)) %>%
```

```
pivot_longer(cols = c(2:6), names_to = "distance", values_to =
"twentyfour_hour") %>%
mutate(twentyfour_hour = str_replace(twentyfour_hour, "[a-z]", " "),
twentyfour_hour = str_replace(twentyfour_hour, " \\s*\\([^\\)]+\\)", ""))
%>%
pivot_wider(names_from = distance, values_from = twentyfour_hour) %>%
mutate_all(as.numeric)


#block 6
twentyfourhr_output_blk6 <- output[1148:1183]
twentyfourhr_output_blk6 <- str_trim(twentyfourhr_output_blk6, side =
c("both"))

twentyfourhr_output_blk6_final<-str_split_fixed(twentyfourhr_output_blk6,"
\\s+",6)

twentyfourhr_output_blk6_final <-
as.data.frame(twentyfourhr_output_blk6_final)

column_names <- c("Direction", "1500.00", "2500.00", "5000.00", "7500.00",
"10000.00")
names(twentyfourhr_output_blk6_final) <- column_names

twentyfourhr_output_blk6_final <- twentyfourhr_output_blk6_final %>%
mutate(Direction = str_trim(Direction, side = "right"),
Direction = sub("\\|", "", Direction)) %>%
pivot_longer(cols = c(2:6), names_to = "distance", values_to =
"twentyfour_hour") %>%
mutate(twentyfour_hour = str_replace(twentyfour_hour, "[a-z]", " "),
twentyfour_hour = str_replace(twentyfour_hour, " \\s*\\([^\\)]+\\)", ""))
%>%
pivot_wider(names_from = distance, values_from = twentyfour_hour) %>%
mutate_all(as.numeric)


 #create data frame combining individual blocks and find max at each
distance

  twentyfourhour_df <- left_join(twentyfourhr_output_blk1_final,
twentyfourhr_output_blk2_final, by = "Direction") %>%
    left_join(twentyfourhr_output_blk3_final, by = "Direction") %>%
    left_join(twentyfourhr_output_blk4_final, by = "Direction") %>%
    left_join(twentyfourhr_output_blk5_final, by = "Direction") %>%
    left_join(twentyfourhr_output_blk6_final, by = "Direction") %>%
    pivot_longer(names_to = "distance", values_to = "twentyfour_hr_avg",
cols = c(2:31)) %>%
    group_by(distance) %>%
    summarize(twentyfour_hr_max = max(twentyfour_hr_avg)) %>%
    mutate(distance = as.numeric(distance)) %>%
    arrange(distance) %>%
    pivot_wider(names_from = distance, values_from = twentyfour_hr_max)
%>%
    mutate(Aver_Period = "24-hr max") %>%
```

```
    relocate(Aver_Period, .before = "10")

  df[i,] <- twentyfourhour_df


}
#create final data frame with the stack parameter information

final_df <- cbind(results_table, df)

#write file to csv
write.csv(final_df, "user defined directory to store output dispersion
table/dispersion_table_wor_24hr_urban.csv", row.names = FALSE)
```

****

A1.4: Processes annual average output – uses output files from AERMOD batch runs using Worcester 2018 meteorology and urban dispersion.

```
library(dplyr)
library(stringr)
library(readr)
library(tidyr)
library(tidyverse)

setwd("/user defined working directory/")

# read filenames of output files
filenames <- list.files(path = "/directory where output files are
stored/")


# Replace variable expressions in filenames for constant vector length
names_2 <- gsub("dia_0.1", "dia_0.10",
gsub("dia_0.2", "dia_0.20",
gsub("dia_0.3", "dia_0.30",
gsub("dist_0", "dist_00",
gsub("hgt_3vel", "hgt_03vel",
gsub("hgt_6vel", "hgt_06vel",
gsub("hgt_9vel", "hgt_09vel",
gsub("vel_0.91", "vel_00.91",
gsub("vel_9.15", "vel_09.15",
gsub("vel_3.05", "vel_03.05",filenames))))))))))

# Create dataframe for input parameter information
results_table <- tibble(stack_height = vector(mode = "numeric", length =
length(filenames)),
stack_exit_velocity = NA,
stack_diameter = NA,
stack_exit_temp = NA,
building_dist = NA,
building_ht_frx = NA)
```

```
# Create dataframe for extracted output information

df <- data.frame(matrix(ncol = 31, nrow = length(filenames)))
#df[1:length(names)] <- NA
colnames(df) <- c("Aver_Period",
seq(10,100,10),
seq(120,200, 20),
seq(250,400,50),
seq(500,1000,100),
seq(1500,1500,0),
seq(2500,2500,0),
seq(5000,10000,2500))


# Start loop to extract input information from each filename and read
output

for(i in 1:length(filenames)){

results_table$stack_height[i] <- as.numeric(substr(names_2[i],20,21))
results_table$stack_exit_velocity[i] <-
as.numeric(substr(names_2[i],26,30))
results_table$stack_diameter[i] <- as.numeric(substr(names_2[i], 35,38))
results_table$stack_exit_temp[i] <- as.numeric(substr(names_2[i],44,46))
results_table$building_dist[i] <- as.numeric(substr(names_2[i],56,57))
results_table$building_ht_frx[i] <- as.numeric(substr(names_2[i],69,71))

print(i)

output <- readLines(paste0("directory where output files are stored/",
filenames[i]))


# Annual average extraction – Total 4 blocks for annual output in model
output

#block 1

annualoutput_blk1 <- output[368:403]
annualoutput_blk1 <- str_trim(annualoutput_blk1, side = c("both"))
annualoutput_blk1_final<-str_split_fixed(annualoutput_blk1," \\s+",10)

annualoutput_blk1_final <- as.data.frame(annualoutput_blk1_final)
column_names <- c("Direction", "10.00", "20.00", "30.00", "40.00",
"50.00", "60.00", "70.00", "80.00", "90.00")
names(annualoutput_blk1_final) <- column_names

annualoutput_blk1_final <- annualoutput_blk1_final %>%
mutate(Direction = str_trim(Direction, side = "right"),
Direction = sub("\\|", "", Direction)) %>%
mutate_all(as.numeric)
```

```
#block 2
annualoutput_blk2 <- output[420:455]
annualoutput_blk2 <- str_trim(annualoutput_blk2, side = c("both"))

annualoutput_blk2_final<-str_split_fixed(annualoutput_blk2," \\s+",10)
annualoutput_blk2_final <- as.data.frame(annualoutput_blk2_final)

column_names <- c("Direction", "100.00", "120.00", "140.00", "160.00",
"180.00", "200.00", "250.00", "300.00", "350.00")
names(annualoutput_blk2_final) <- column_names

annualoutput_blk2_final <- annualoutput_blk2_final %>%
mutate(Direction = str_trim(Direction, side = "right"),
Direction = sub("\\|", "", Direction)) %>%
mutate_all(as.numeric)


#block 3
annualoutput_blk3 <- output[472:507]
annualoutput_blk3 <- str_trim(annualoutput_blk3, side = c("both"))
annualoutput_blk3_final<-str_split_fixed(annualoutput_blk3," \\s+",10)

annualoutput_blk3_final <- as.data.frame(annualoutput_blk3_final)

column_names <- c("Direction", "400.00", "500.00", "600.00", "700.00",
"800.00", "900.00", "1000.00", "1500.00", "2500.00")
names(annualoutput_blk3_final) <- column_names

annualoutput_blk3_final <- annualoutput_blk3_final %>%
mutate(Direction = str_trim(Direction, side = "right"),
Direction = sub("\\|", "", Direction)) %>%
mutate_all(as.numeric)

#block 4
annualoutput_blk4 <- output[524:559]
annualoutput_blk4 <- str_trim(annualoutput_blk4, side = c("both"))

annualoutput_blk4_final<-str_split_fixed(annualoutput_blk4," \\s+",4)

annualoutput_blk4_final <- as.data.frame(annualoutput_blk4_final)
column_names <- c("Direction", "5000.00", "7500.00", "10000.00")
names(annualoutput_blk4_final) <- column_names

annualoutput_blk4_final <- annualoutput_blk4_final %>%
mutate(Direction = str_trim(Direction, side = "right"),
Direction = sub("\\|", "", Direction)) %>%
mutate_all(as.numeric)


#create data frame combining individual blocks and find maximum value at
each distance across all 36 directions

annual_df <- left_join(annualoutput_blk1_final, annualoutput_blk2_final,
by = "Direction") %>%
```

```
  left_join(annualoutput_blk3_final, by = "Direction") %>%
  left_join(annualoutput_blk4_final, by = "Direction") %>%
  pivot_longer(names_to = "distance", values_to = "annual_avg", cols =
c(2:31)) %>%
  group_by(distance) %>%
  summarize(annual_max = max(annual_avg)) %>%
  mutate(distance = as.numeric(distance)) %>%
  arrange(distance) %>%
  pivot_wider(names_from = distance, values_from = annual_max) %>%
  mutate(Aver_Period = "Annual") %>%
  relocate(Aver_Period, .before = "10")


  df[i,] <- annual_df



}

#create final data frame with the stack parameter information

final_df <- cbind(results_table, df)

#write file to csv
write.csv(final_df, "user defined directory to store output dispersion
table/dispersion_table_wor_annual_urban.csv", row.names = FALSE)
```

****

A1.5 MADEP Final Dispersion Tables Generation
Author: Caroline Watson and Ambrish Sharma, Abt Associates Inc,
Purpose: Script that reads in extended dispersion lookup tables, generates short format tables with computed statistics, and generates final consolidated table(s) to be used in risk screening ready tool
Year: 2023

```
library(dplyr)
library(stringr)
library(readr)
library(tidyr)


#read in data
setwd("/user defined working directory/")
dir <- "/directory with extended output dispersion tables/"

files <- list.files(path = paste0(dir, "/Output Tables/"), "*.csv")

dispersion_df <- fs::dir_ls(dir, regexp = "\\.csv$") %>%
map_dfr(read_csv, .id = "filename") %>%
```

```
mutate(file_name = gsub(".*Output Tables/", "", filename),
file_name = str_remove(file_name, ".csv"),
city = substr(file_name, 18, 20)) %>%
select(-filename, -file_name)

#create vectors for For loop
aver_period <- c("1-hr", "24-hr", "Annual average")
city_name <- c("bos", "wor")

#Create short format tables for all 6 met station-averaging period
combinations - each table has the max, min, quantile 25, quantile 50,
quantile 75, quantile 90, and average of dispersion factors calculated for
each stack-height, boundary distance combination

for(i in 1:length(city_name)){
for(j in 1:length(aver_period)){

df <- dispersion_df %>%
filter(city == city_name[i],
Aver_Period == aver_period[j])%>%
pivot_longer(cols = c(8:37), names_to = "distance", values_to = "amount")
%>%
group_by(stack_height, distance, Aver_Period) %>%
summarise(max = max(amount),
min = min(amount),
quant25 = quantile(amount, 0.25),
quant50 = quantile(amount, 0.5),
quant75 = quantile(amount, 0.75),
quant90 = quantile(amount, 0.9),
avg = mean(amount)) %>%
mutate(distance = as.numeric(distance)) %>%
arrange(stack_height, distance) %>%
select(-Aver_Period)

write.csv(df, paste0("/user defined directory for short format tables
output/stats_df_", city_name[i], "_", aver_period[j],".csv"), row.names =
F)

}
}

# Produce consolidated tables using max dispersion values from short
format tables with each consolidated table to have info for all three
averaging periods, generate final dispersion lookup table(s) using average
of two stations data

for(i in 1:length(city_name)){

max_df <- dispersion_df %>%
filter(city == city_name[i]) %>%
pivot_longer(cols = c(8:37), names_to = "distance", values_to = "amount")
%>%
group_by(stack_height, distance, Aver_Period) %>%
summarise(max = max(amount)) %>%
```

```
mutate(distance = as.numeric(distance)) %>%
arrange(stack_height, distance) %>%
pivot_wider(names_from = "distance", values_from = "max") %>%
relocate(Aver_Period, .before = stack_height) %>%
arrange(factor(Aver_Period, levels = c("1-hr", "24-hr", "Annual
average")))

write.csv(max_df, paste0("/user defined directory for storing dispersion
table/output_files/max_df_", city_name[i],".csv"), row.names = F)

quant75_df <- dispersion_df %>%
filter(city == city_name[i]) %>%
pivot_longer(cols = c(8:37), names_to = "distance", values_to = "amount")
%>%
group_by(stack_height, distance, Aver_Period) %>%
summarise(quant75 = quantile(amount, 0.75)) %>%
mutate(distance = as.numeric(distance)) %>%
arrange(stack_height, distance) %>%
pivot_wider(names_from = "distance", values_from = "quant75") %>%
relocate(Aver_Period, .before = stack_height) %>%
arrange(factor(Aver_Period, levels = c("1-hr", "24-hr", "Annual
average")))

write.csv(quant75_df, paste0("/user defined directory for storing
dispersion table/output_files/quant75_df_", city_name[i],".csv"),
row.names = F)

quant90_df <- dispersion_df %>%
filter(city == city_name[i]) %>%
pivot_longer(cols = c(8:37), names_to = "distance", values_to = "amount")
%>%
group_by(stack_height, distance, Aver_Period) %>%
summarise(quant90 = quantile(amount, 0.9)) %>%
mutate(distance = as.numeric(distance)) %>%
arrange(stack_height, distance) %>%
pivot_wider(names_from = "distance", values_from = "quant90") %>%
relocate(Aver_Period, .before = stack_height) %>%
arrange(factor(Aver_Period, levels = c("1-hr", "24-hr", "Annual
average")))

write.csv(quant90_df, paste0("/user defined directory for storing
dispersion table/output_files/quant90_df_", city_name[i],".csv"),
row.names = F)


}

#Create Boston and Worcester consolidated max dispersion tables

bos_max_df <- read_csv("/user defined directory for storing dispersion
table/output_files/max_df_bos.csv")
wor_max_df <- read_csv("/user defined directory for storing dispersion
table/output_files/max_df_wor.csv")
```

```
#Calculate average of Boston and Worcester consolidated max dispersion
tables

average_df <- rbind(bos_max_df, wor_max_df) %>%
pivot_longer(cols = c(3:32), names_to = "distance", values_to = "amount")
%>%
group_by(Aver_Period, stack_height, distance) %>%
summarize(average = mean(amount)) %>%
mutate(distance = as.numeric(distance)) %>%
arrange(stack_height, distance) %>%
pivot_wider(names_from = "distance", values_from = "average") %>%
relocate(Aver_Period, .before = stack_height) %>%
arrange(factor(Aver_Period, levels = c("1-hr", "24-hr", "Annual
average")))

#Create final table to be used in screening tool
write.csv(average_df, "/user defined directory for storing dispersion
table/output_files/average_df.csv", row.names = F)


# Create other consolidated tables- other options for risk screening tool-
#Boston 75th and 90th percentile tables, Worcester 75th and 90th
#percentile tables, Average bos-wor 75th and 90th percentile tables

bos_quant75_df <- read_csv("/user defined directory for storing dispersion
table/output_files/quant75_df_bos.csv")
wor_quant75_df <- read_csv("/user defined directory for storing dispersion
table/output_files/quant75_df_wor.csv")

average_quant75_df <-  rbind(bos_quant75_df, wor_quant75_df) %>%
pivot_longer(cols = c(3:32), names_to = "distance", values_to = "amount")
%>%
group_by(Aver_Period, stack_height, distance) %>%
summarize(average_quant75 = mean(amount)) %>%
mutate(distance = as.numeric(distance)) %>%
arrange(stack_height, distance) %>%
pivot_wider(names_from = "distance", values_from = "average_quant75") %>%
relocate(Aver_Period, .before = stack_height) %>%
arrange(factor(Aver_Period, levels = c("1-hr", "24-hr", "Annual
average")))

write.csv(average_quant75_df, "/ user defined directory for storing
dispersion table/output_files/average_quant75_df.csv", row.names = F)

bos_quant90_df <- read_csv("/user defined directory for storing dispersion
table/output_files/quant90_df_bos.csv")
wor_quant90_df <- read_csv("/user defined directory for storing dispersion
table /output_files/quant90_df_wor.csv")

average_quant90_df <-  rbind(bos_quant90_df, wor_quant90_df) %>%
pivot_longer(cols = c(3:32), names_to = "distance", values_to = "amount")
%>%
group_by(Aver_Period, stack_height, distance) %>%
```

```
summarize(average_quant90 = mean(amount)) %>%
mutate(distance = as.numeric(distance)) %>%
arrange(stack_height, distance) %>%
pivot_wider(names_from = "distance", values_from = "average_quant90") %>%
relocate(Aver_Period, .before = stack_height) %>%
arrange(factor(Aver_Period, levels = c("1-hr", "24-hr", "Annual
average")))

write.csv(average_quant90_df, "/user defined directory for storing
dispersion table /output_files/average_quant90_df.csv", row.names = F)
```

****

Appendix 2

Supplemental Figures and
Statistical Analysis Plots

Appendix 2 contains graphs comparing the final dispersion values used in the Massachusetts Air Toxics Risk Screening Tool (MATRiST) with those used by MPCA in its air emissions Risk Assessment Screening Spreadsheet (RASS) tool. Additionally, statistical plots showing results from the analysis of the MassDEP AERMOD batch runs are included. It should be noted that the underlying parameterization in the MPCA and MassDEP modeling contains differences. Therefore, the plots included here are a comparison of final dispersion values used in the risk screening tools. For comparison, the dispersion values from the final dispersion lookup tables from MassDEP and MPCA are grouped by averaging period, stack height, and receptor distance.

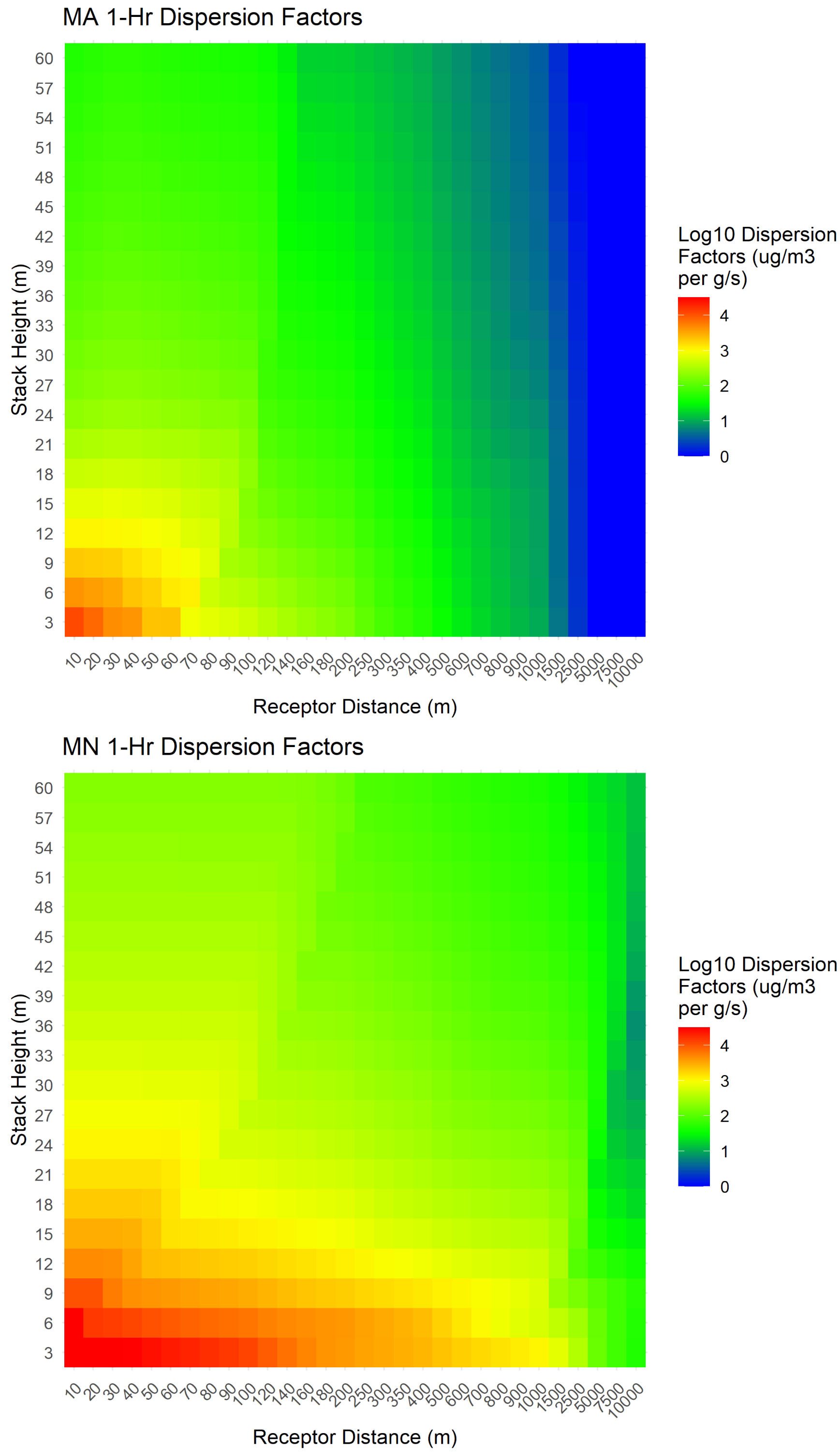


Figure 1. Variation of 1-hour dispersion factors (rural dispersion), used by MassDEP (top) and MPCA (below) in their air emissions risk screening tools, with select modeled stack heights and receptor distances. Dispersion factors are log10 transformed to capture the wide range of values.

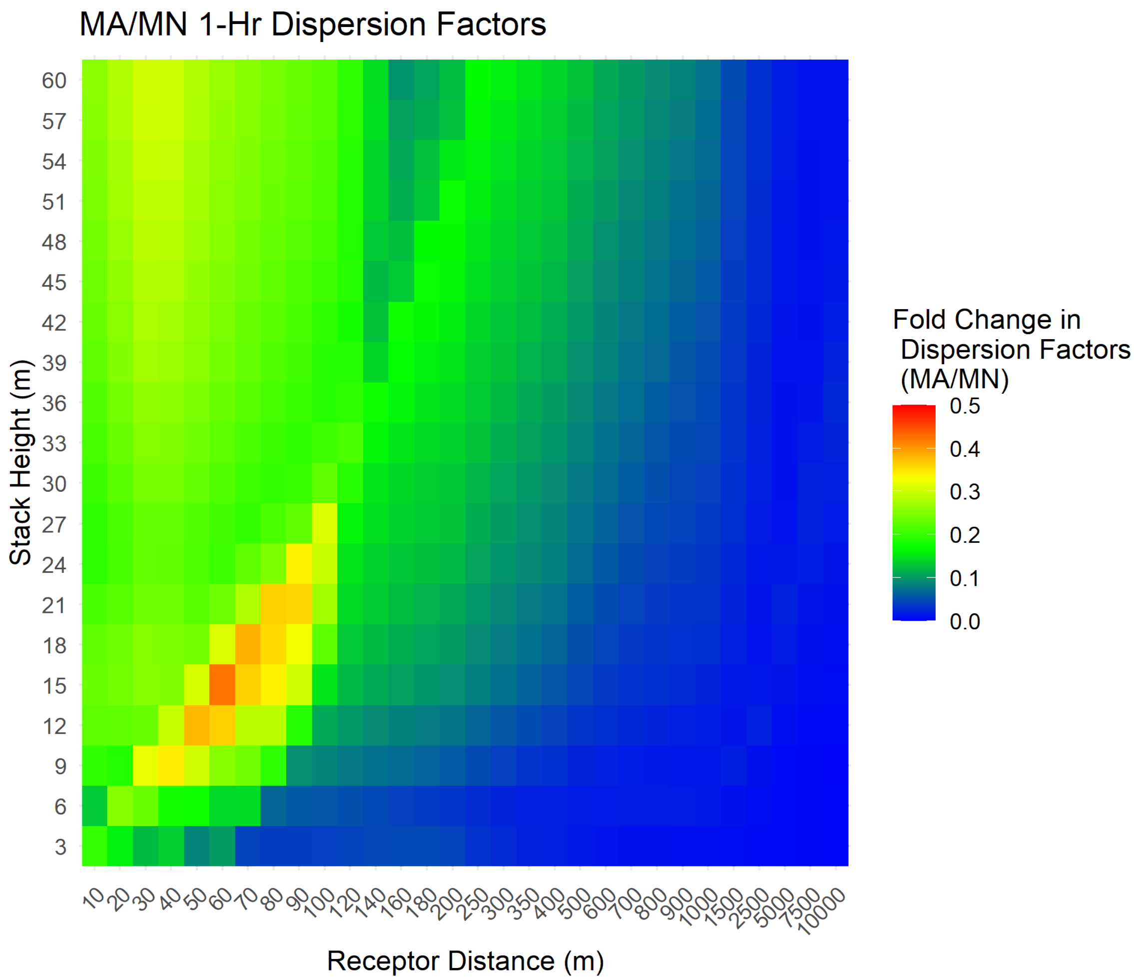


Figure 2. Fold change in 1-hr dispersion factors. Fold change is calculated by dividing MassDEP dispersion factors (rural dispersion) by MPCA dispersion factors for each stack height-receptor distance combination. Red areas indicate higher dispersion factor values for MassDEP than MPCA and blue areas indicate lower dispersion factor values for MassDEP than MPCA for a particular stack height-receptor distance combination.

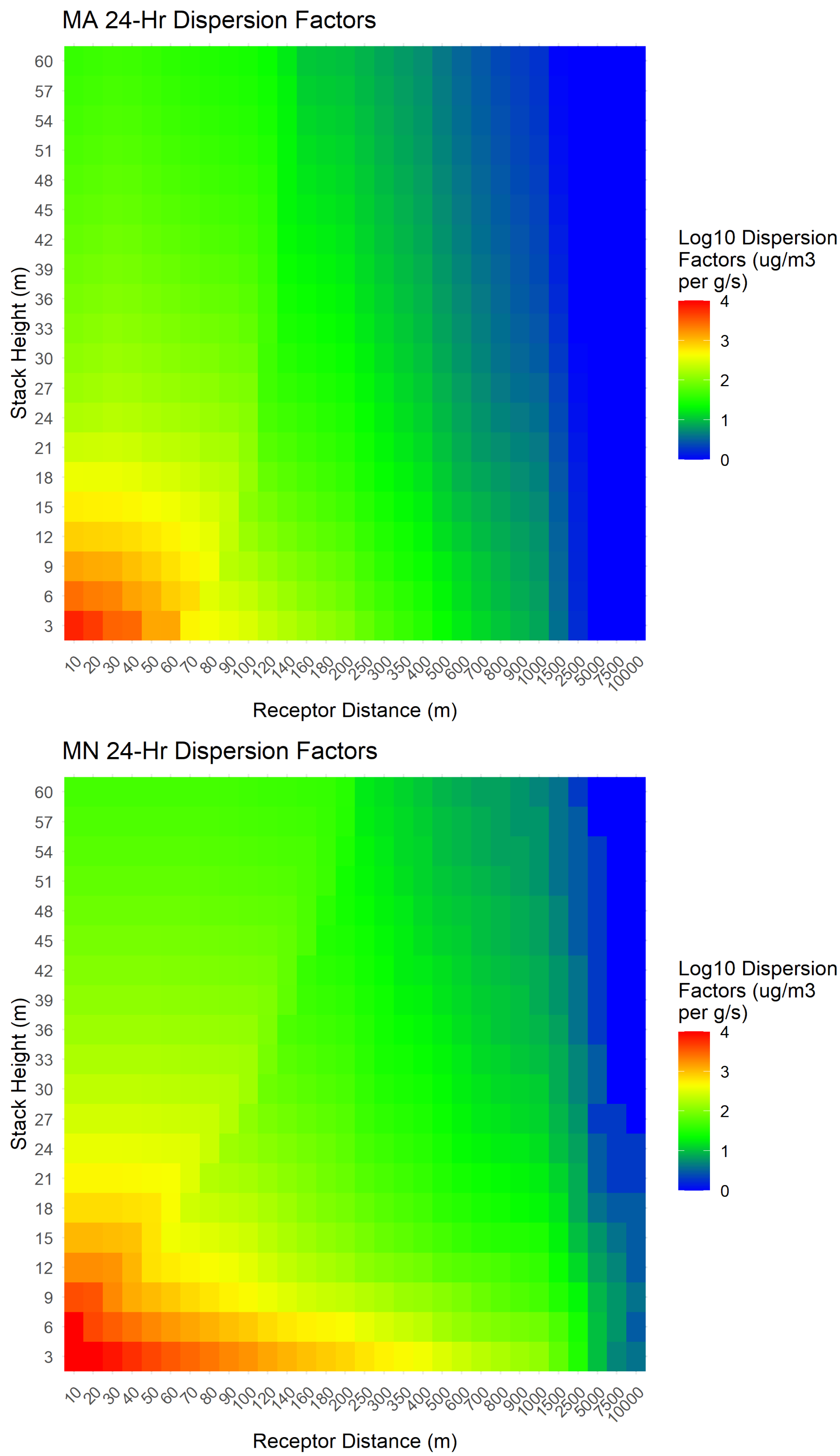


Figure 3. Variation of 24-hour dispersion factors (rural dispersion), used by MassDEP (top) and MPCA (below) in their air emissions risk screening tools, with select modeled stack heights and receptor distances. Dispersion factors are log10 transformed to capture the wide range of values.

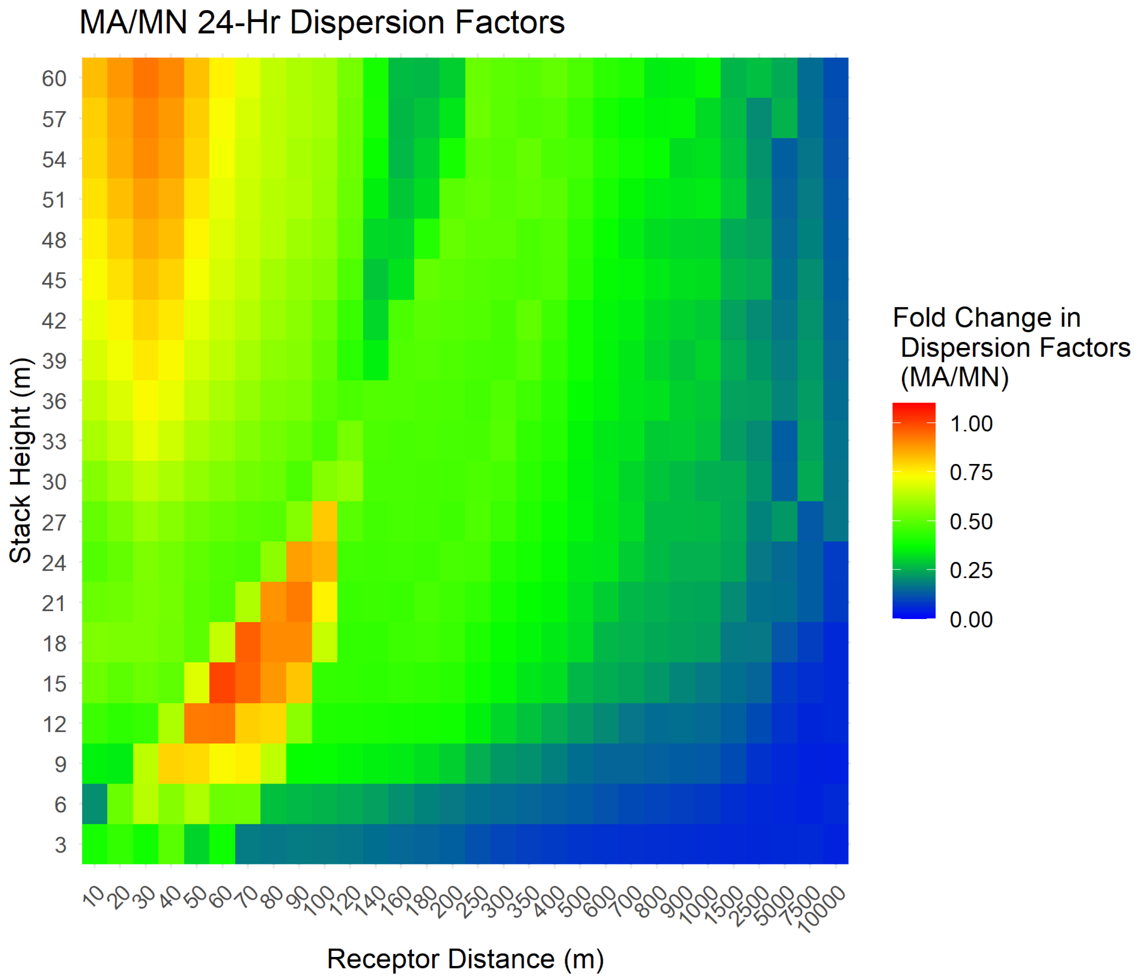


Figure 4. Fold change in 24-hr dispersion factors. Fold change is calculated by dividing MassDEP dispersion factors (rural dispersion) by MPCA dispersion factors for each stack height-receptor distance combination. Red areas indicate higher dispersion factor values for MassDEP than MPCA and blue areas indicate lower dispersion factor values for MassDEP than MPCA for a particular stack height-receptor distance combination.

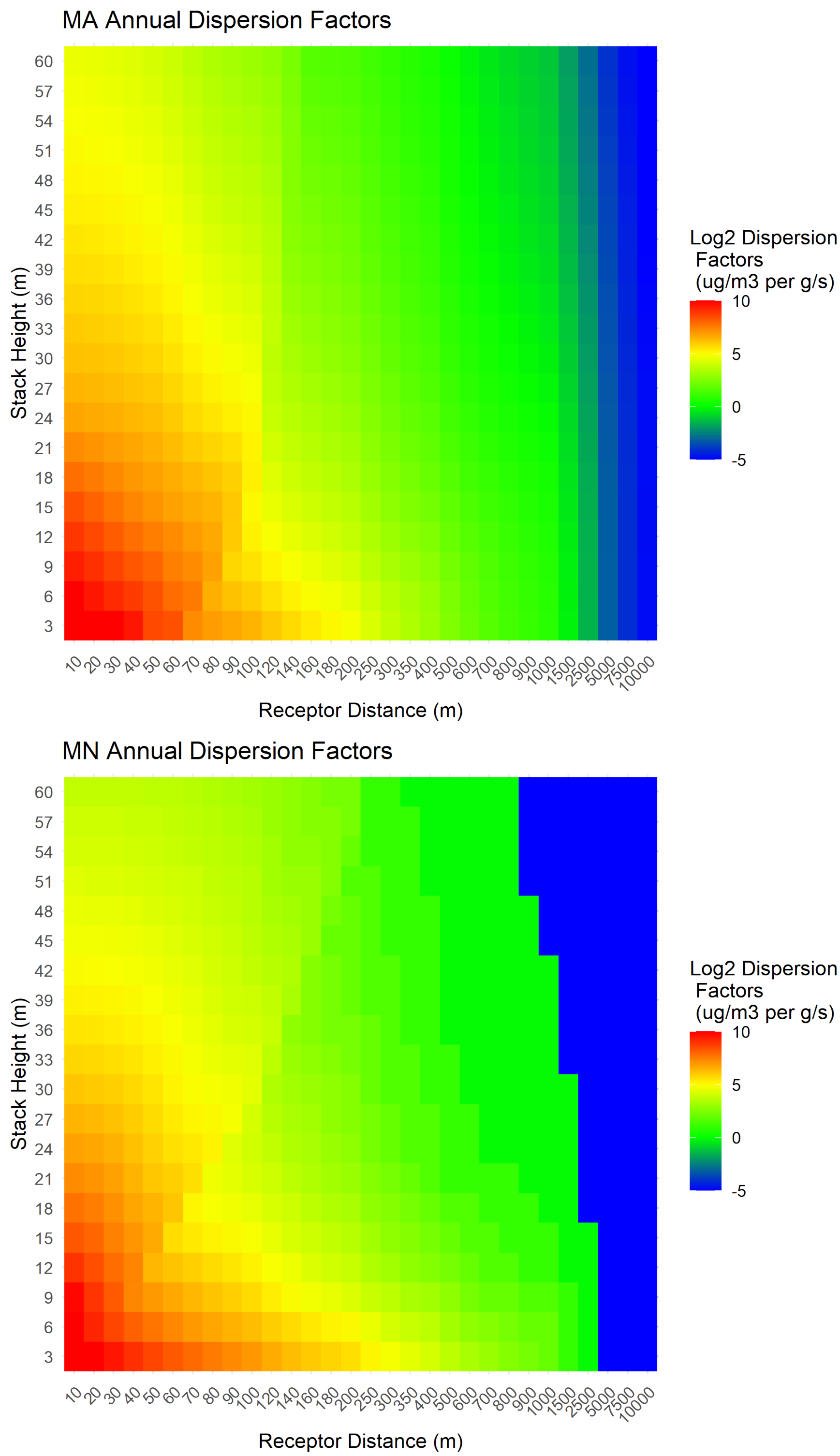


Figure 5. Variation of annual dispersion factors (rural dispersion), used by MassDEP (top) and MPCA (below) in their air emissions risk screening tools, with select modeled stack heights and receptor distances. Dispersion factors are log2 transformed to capture the wide range of values. Areas in dark blue depict 0 or near 0 values. Fold change was not plotted for annual dispersion factors due to the large number of 0 values in the data.

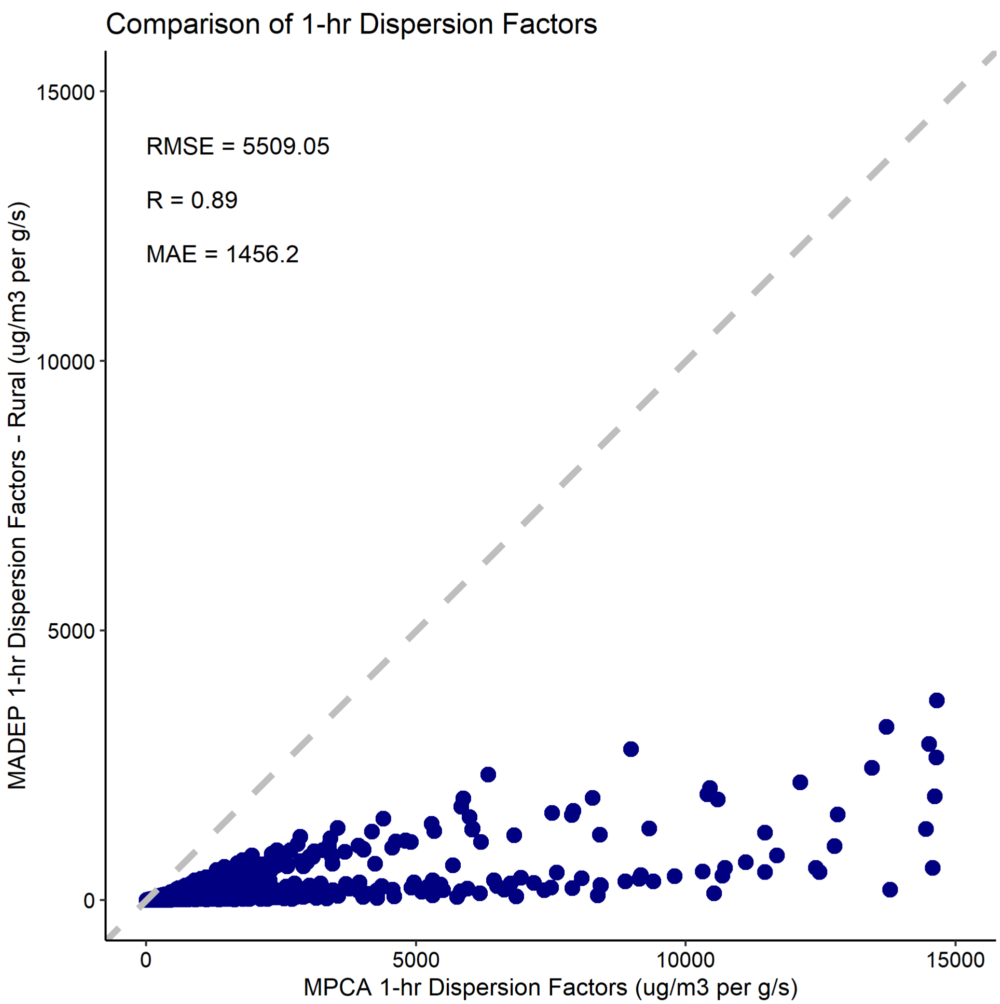


Figure 6. Scatterplot showing the relationship between 1-hr dispersion factors used by MassDEP (rural dispersion) and by MPCA in their respective air emissions risk screening tools. Root mean square error (RMSE), correlation coefficient (R) and mean absolute error (MAE) are shown.

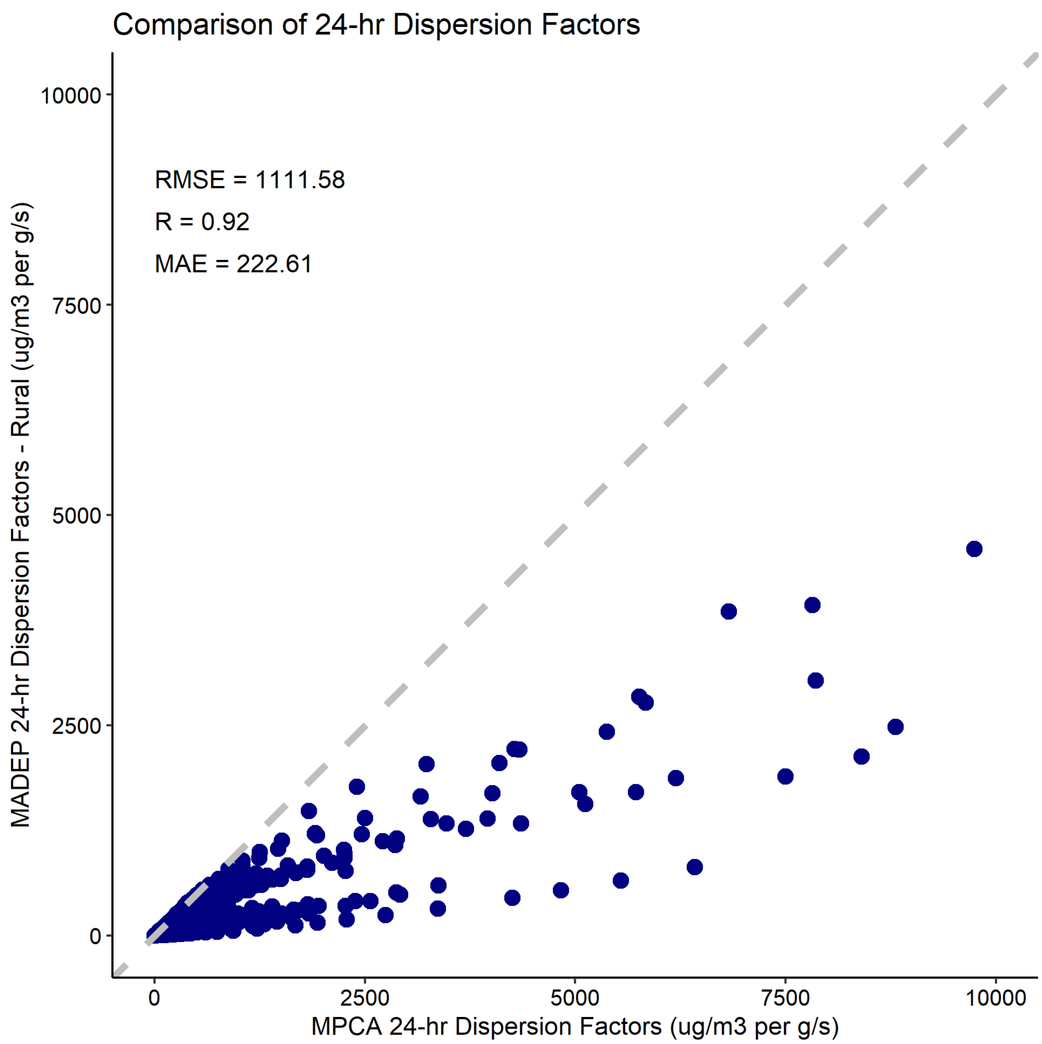


Figure 7. Scatterplot showing relationship between 24-hr dispersion factors used by MassDEP (rural dispersion) and by MPCA in their respective air emissions risk screening tools. Root mean square error (RMSE), correlation coefficient (R) and mean absolute error (MAE) are shown.

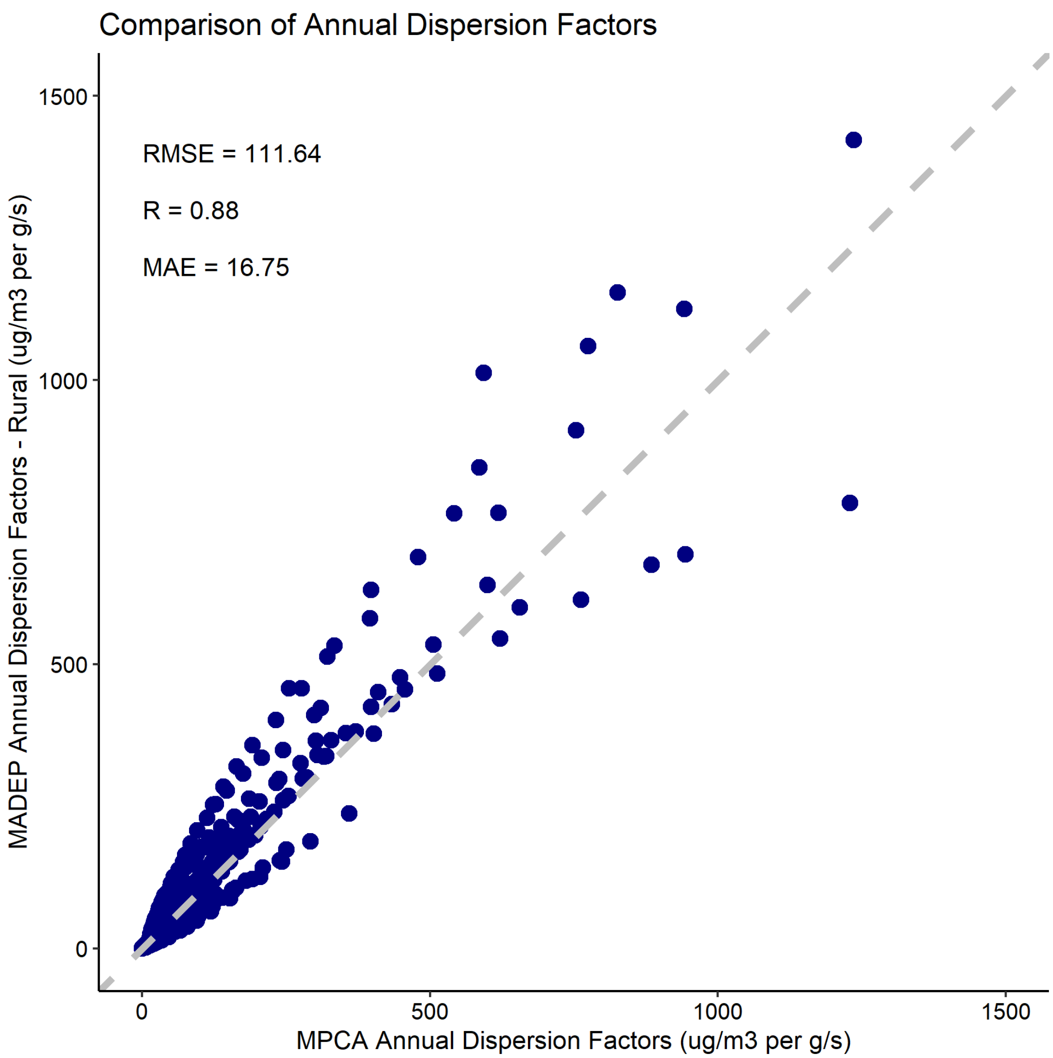
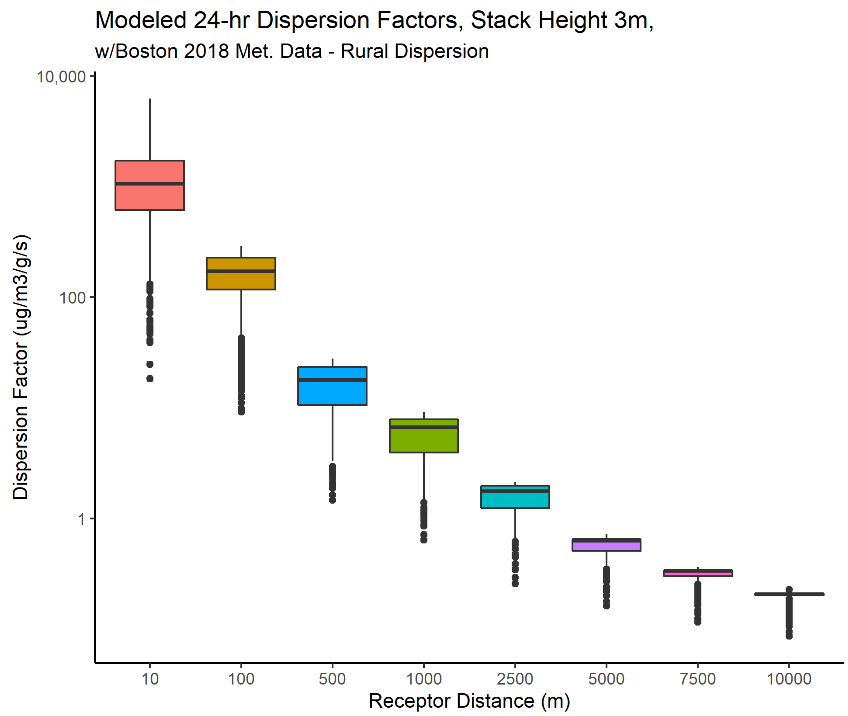


Figure 8. Scatterplot showing relationship between annual dispersion factors used by MassDEP (rural dispersion) and by MPCA in their respective air emissions risk screening tools. Root mean square error (RMSE), correlation coefficient (R) and mean absolute error (MAE) are shown.



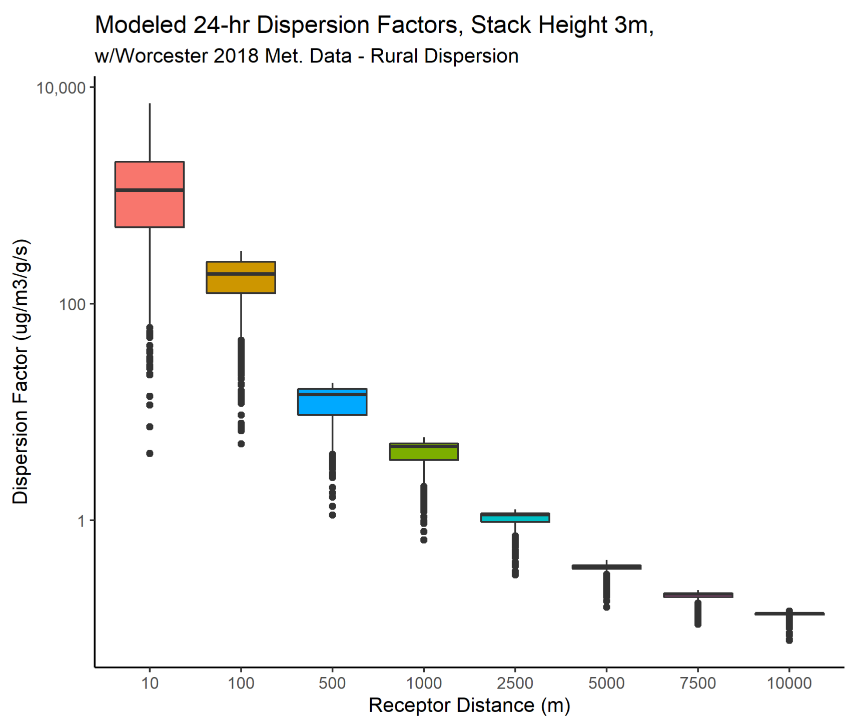
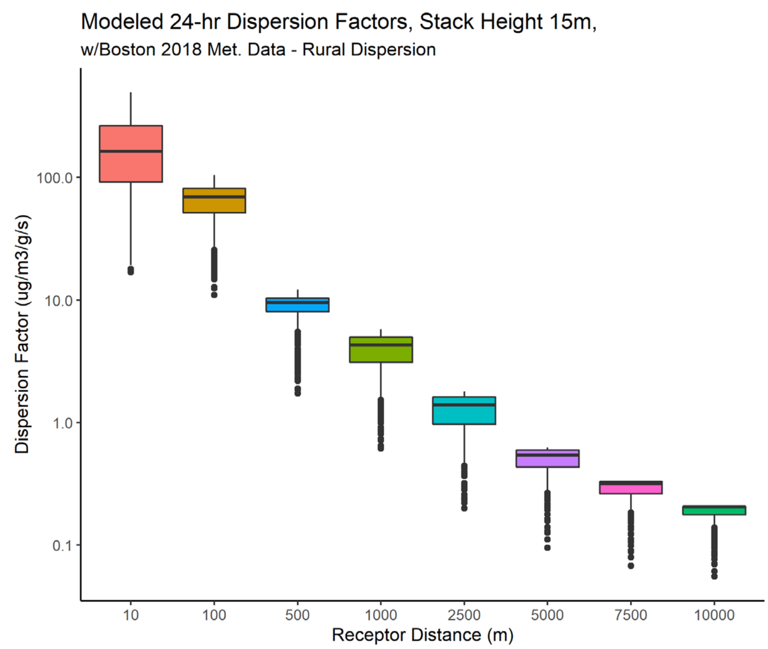


Figure 9. Boxplots showing minimum, maximum, median, 25th and 75th percentile values of 24-hour dispersion factors (rural dispersion) for a 3-meter stack height at select receptor distances. The dispersion factors represent the MassDEP risk screening dispersion modeling output using Boston meteorology (above) and Worcester meteorology (below). The averages of the maximum Boston and Worcester values shown here at each distance are used in the MassDEP Air Toxics Risk Screening Tool rural dispersion factor table.



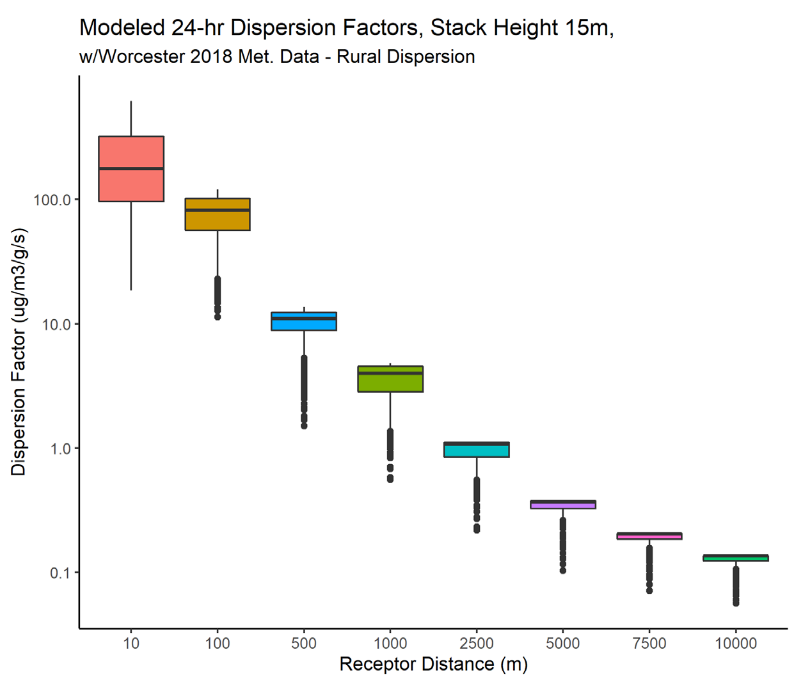
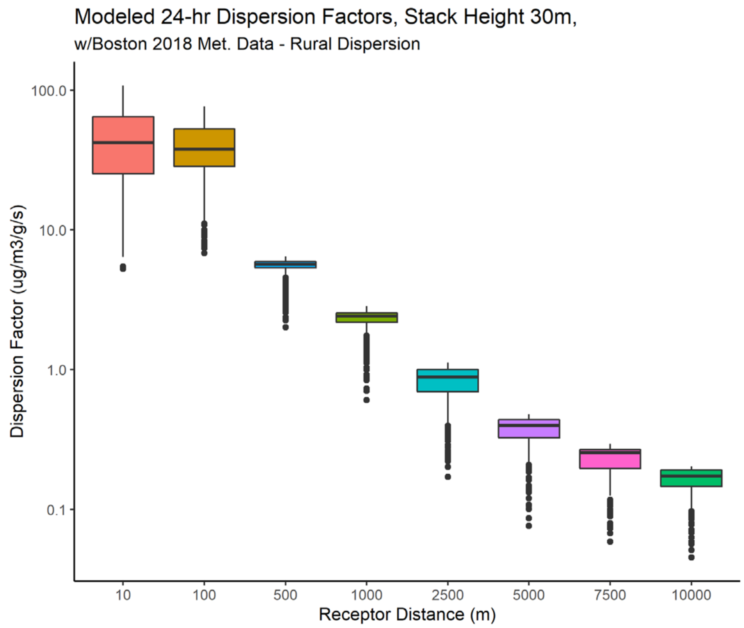


Figure 10. Boxplots showing minimum, maximum, median, 25th and 75th percentile values of 24-hour dispersion factors (rural dispersion) for a 15-meter stack height at select receptor distances. The dispersion factors represent the MassDEP risk screening dispersion modeling output using Boston meteorology (above) and Worcester meteorology (below). The averages of the maximum Boston and Worcester values shown here at each distance are used in the MassDEP Air Toxics Risk Screening Tool rural dispersion factor table.



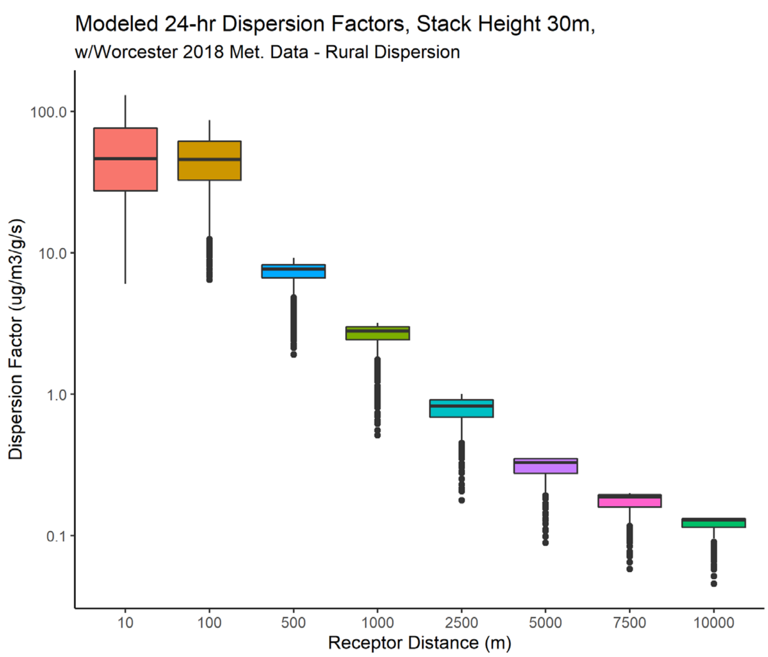
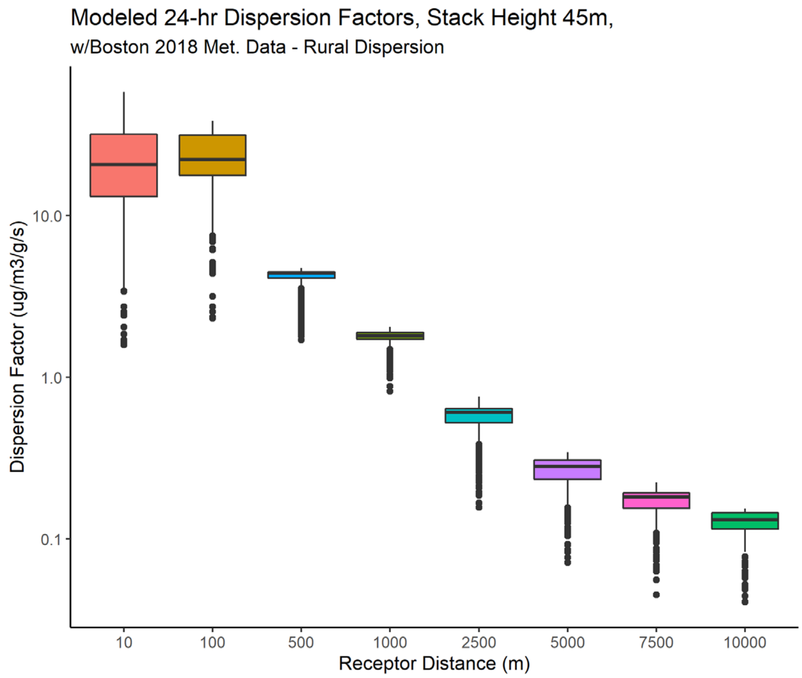


Figure 11. Boxplots showing minimum, maximum, median, 25^{th} and 75^{th} percentile values of 24-hour dispersion factors (rural dispersion) for a 30-meter stack height at select receptor distances. The dispersion factors represent the MassDEP risk screening dispersion modeling output using Boston meteorology (above) and Worcester meteorology (below). The averages of the maximum Boston and Worcester values shown here at each distance are used in the MassDEP Air Toxics Risk Screening Tool rural dispersion factor table.



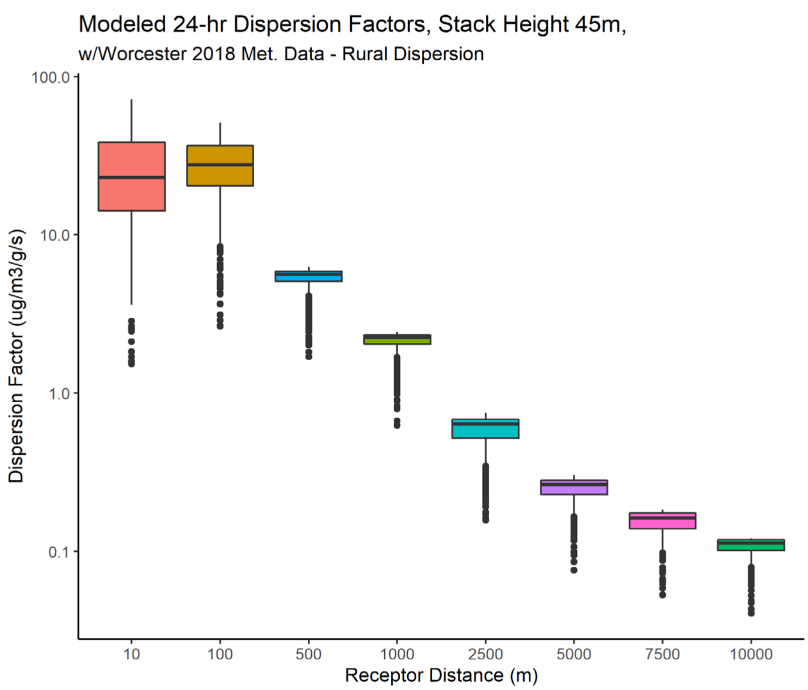
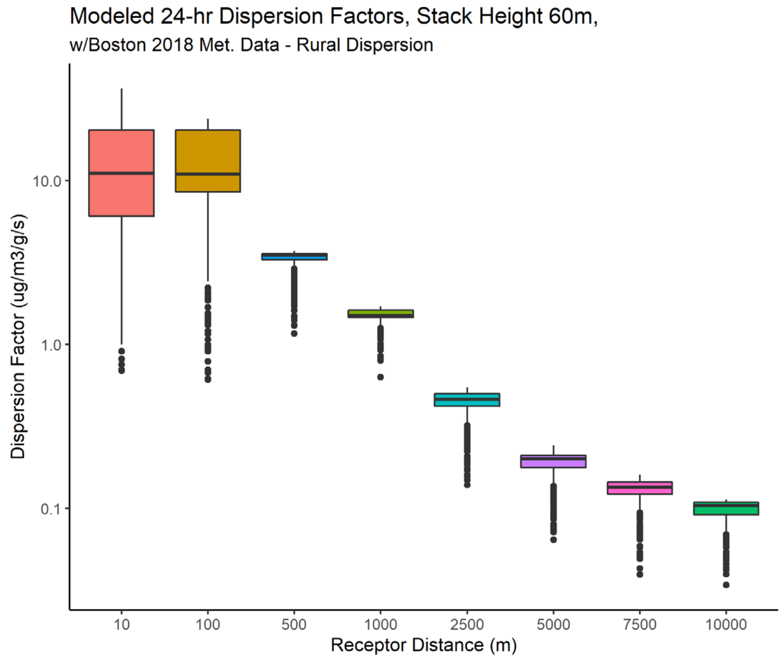


Figure 12. Boxplots showing minimum, maximum, median, 25th and 75th percentile values of 24-hour dispersion factors (rural dispersion) for a 45-meter stack height at select receptor distances. The dispersion factors represent the MassDEP risk screening dispersion modeling output using Boston meteorology (above) and Worcester meteorology (below). The averages of the maximum Boston and Worcester values shown here at each distance are used in the MassDEP Air Toxics Risk Screening Tool rural dispersion factor table.



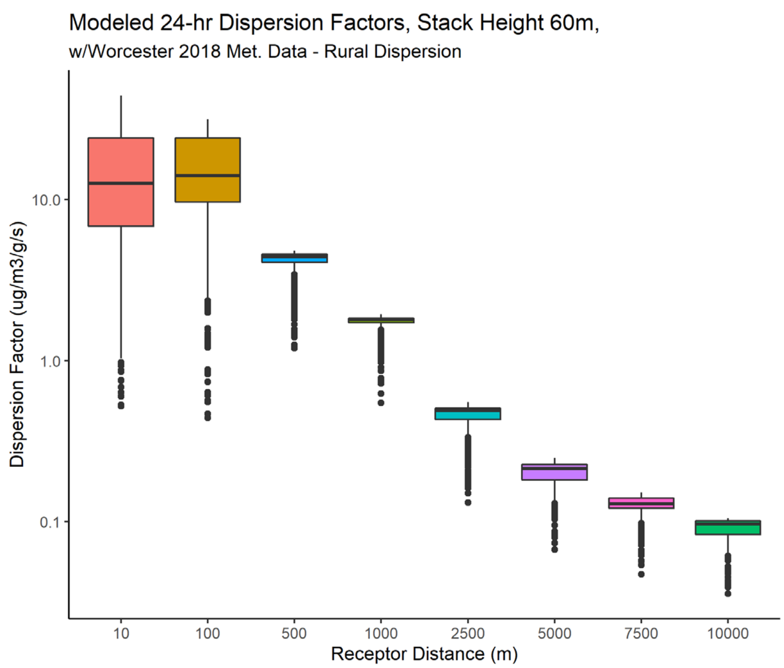


Figure 13 Boxplots showing minimum, maximum, median, 25th and 75th percentile values of 24-hour (rural dispersion) dispersion factors for a 60-meter stack height at selected receptor distances. The dispersion factors represent the MassDEP risk screening dispersion modeling output using Boston meteorology (above) and Worcester meteorology (below). The averages of the maximum Boston and Worcester values shown here at each distance are used in the MassDEP Air Toxics Risk Screening Tool rural dispersion factor table.