



An integrated system
for dispersion modelling
with CALPUFF



riannone@me.com



[@riannone](https://twitter.com/riannone)



[rich-iannone](https://github.com/rich-iannone)



[rich_i](https://medium.com/@rich_i)



[richard-iannone](https://www.linkedin.com/in/richard-iannone)



[Richard_Iannone](https://www.researchgate.net/profile/Richard_Iannone)

Contents

- | | | | |
|----|----------------------------------|----|--|
| 01 | Dispersion Modelling and CALPUFF | 04 | Installation of R, RStudio, and PuffR |
| 02 | The Goals of the PuffR R Package | 05 | The Current PuffR Workflow and Examples of Use |
| 03 | Proposed Data Sources for PuffR | 06 | The Project Roadmap |

Dispersion Modelling as a Means to Characterize Regional Air Quality

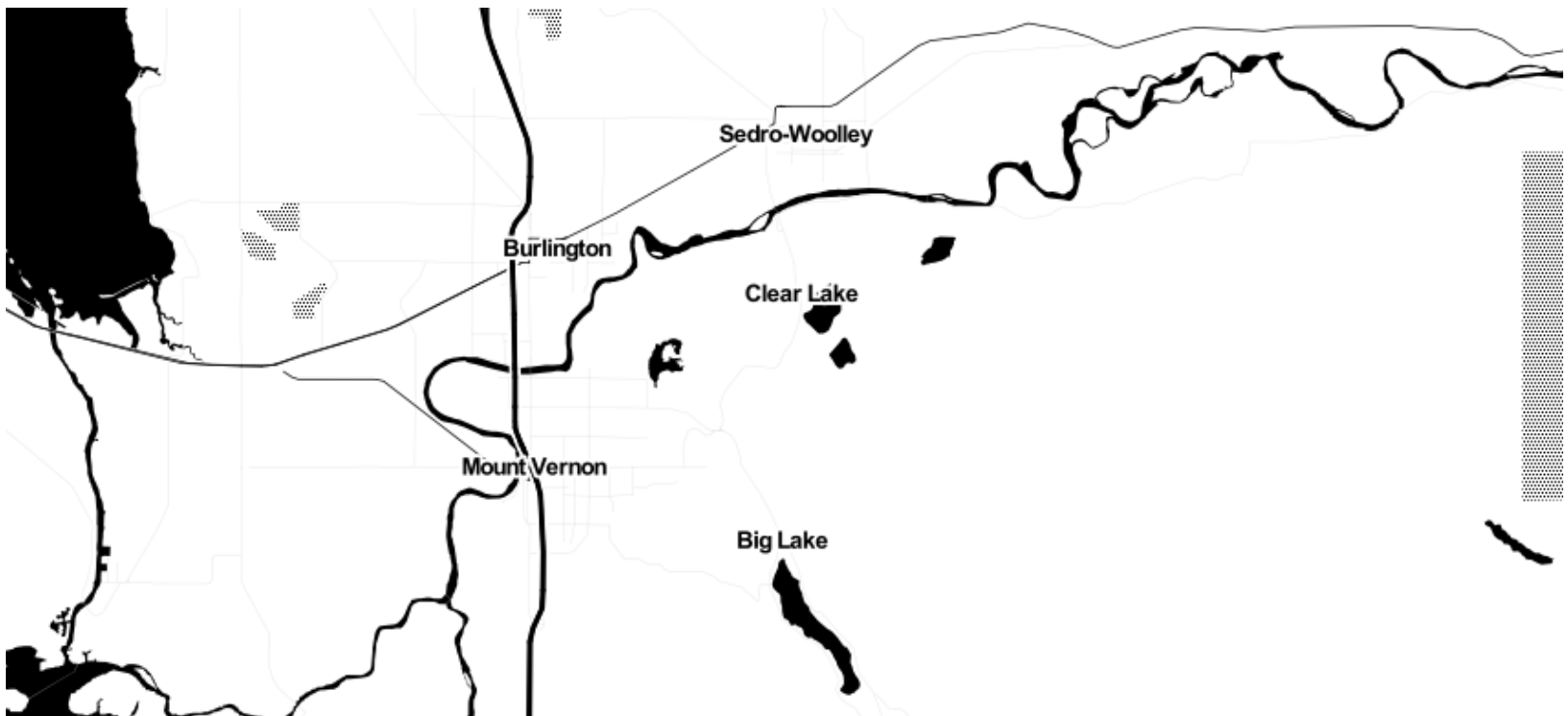
Dispersion modelling is a great tool for understanding how pollutants disperse from sources to receptors.

Air quality modelling is a great tool for describing the causal relationship between emissions, meteorology, atmospheric concentrations, deposition, and other factors. Air pollution measurements give useful quantitative information about ambient concentrations and deposition, however, such measurements can only describe air quality at specific locations and times.

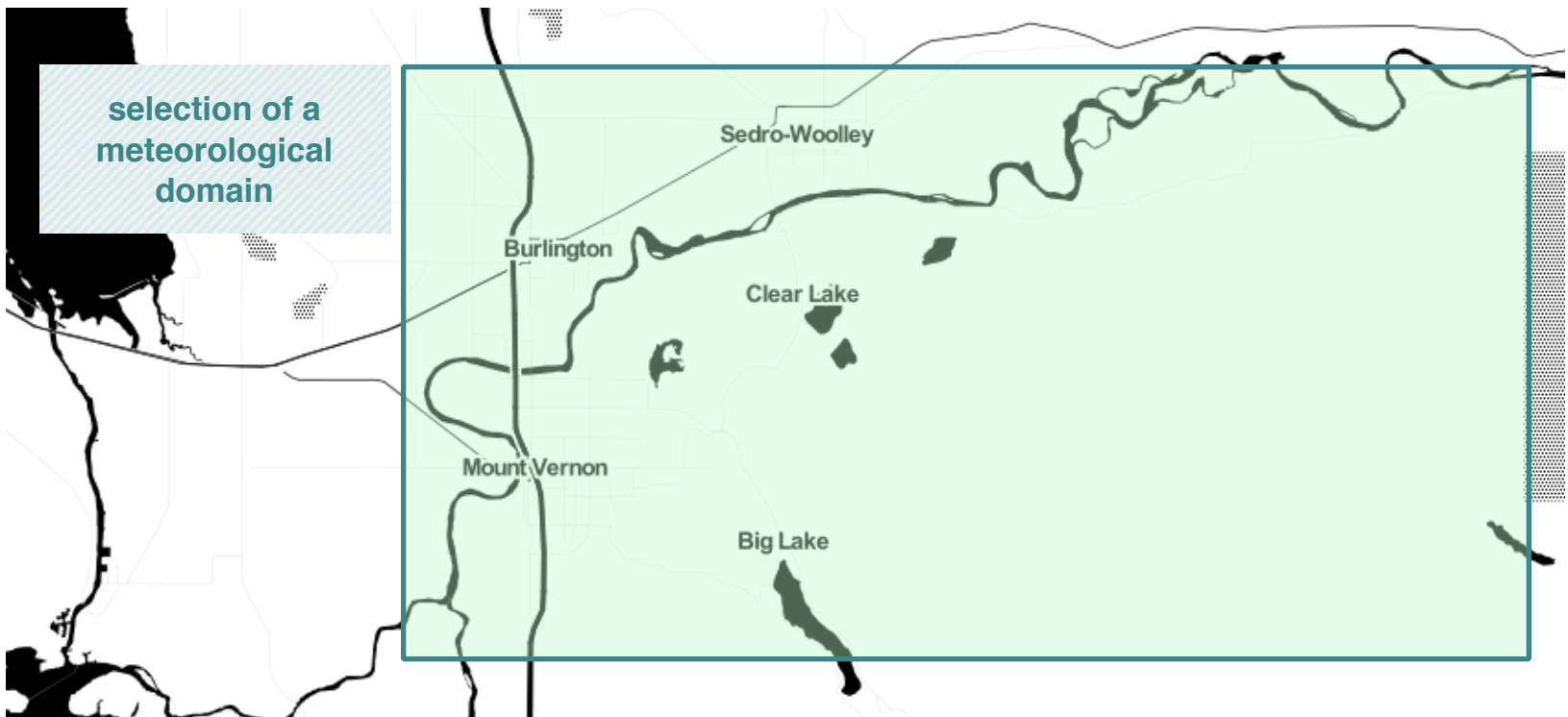
Moreover, monitoring usually doesn't provide very good information concerning the causes of the air quality problem.

Dispersion modelling can instead provide a more complete deterministic description of the air quality problem, including an analysis of factors and causes (e.g., emission sources, meteorological processes, physical changes, and chemistry). Air quality models play an important role in science and in policy, because of their capability to assess the relative importance of the relevant processes and to provide an assessment of potential human exposure.

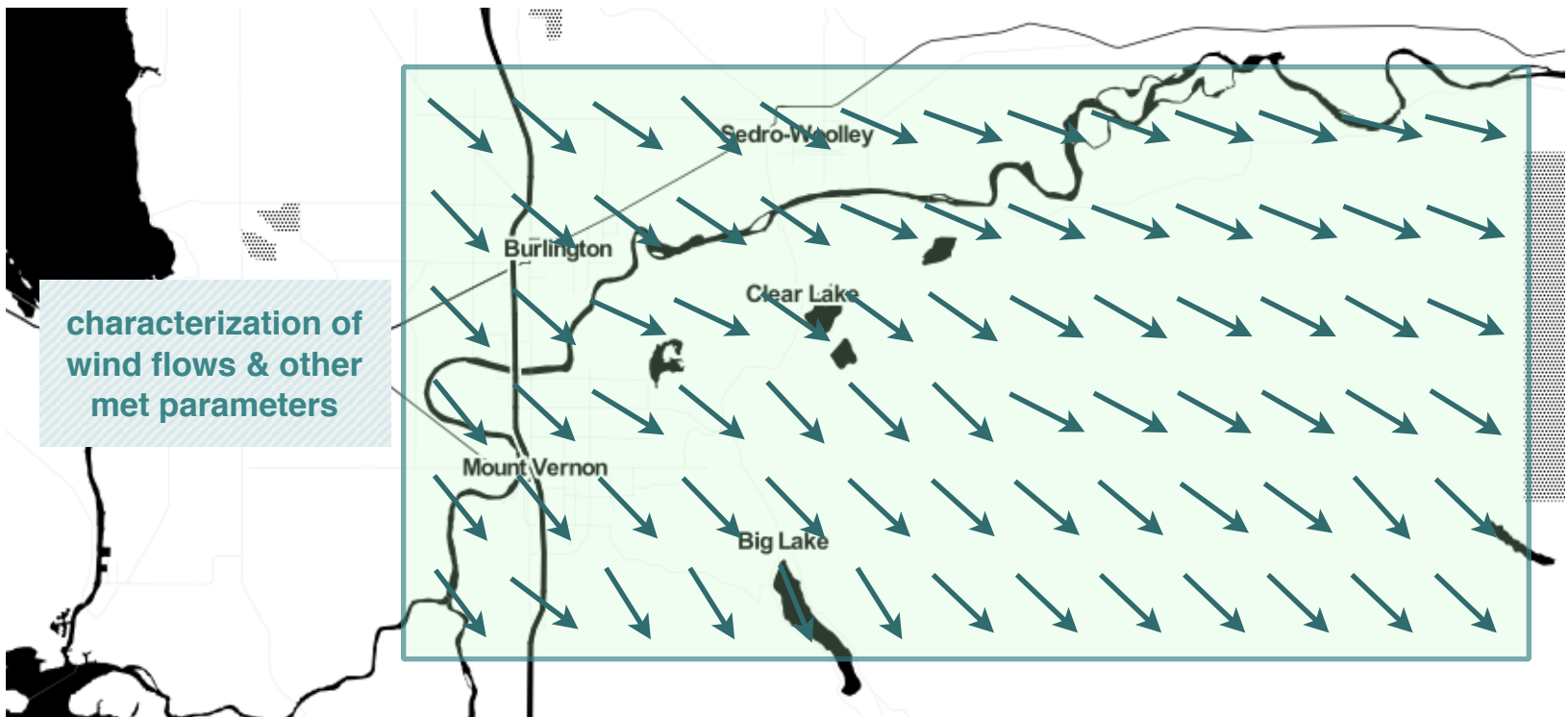
Basic Scheme for How Dispersion Models Work



Basic Scheme for How Dispersion Models Work



Basic Scheme for How Dispersion Models Work



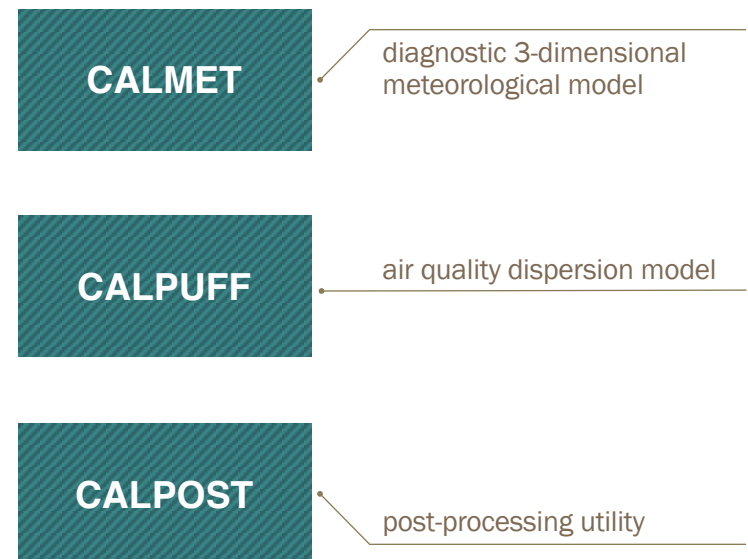
Basic Scheme for How Dispersion Models Work



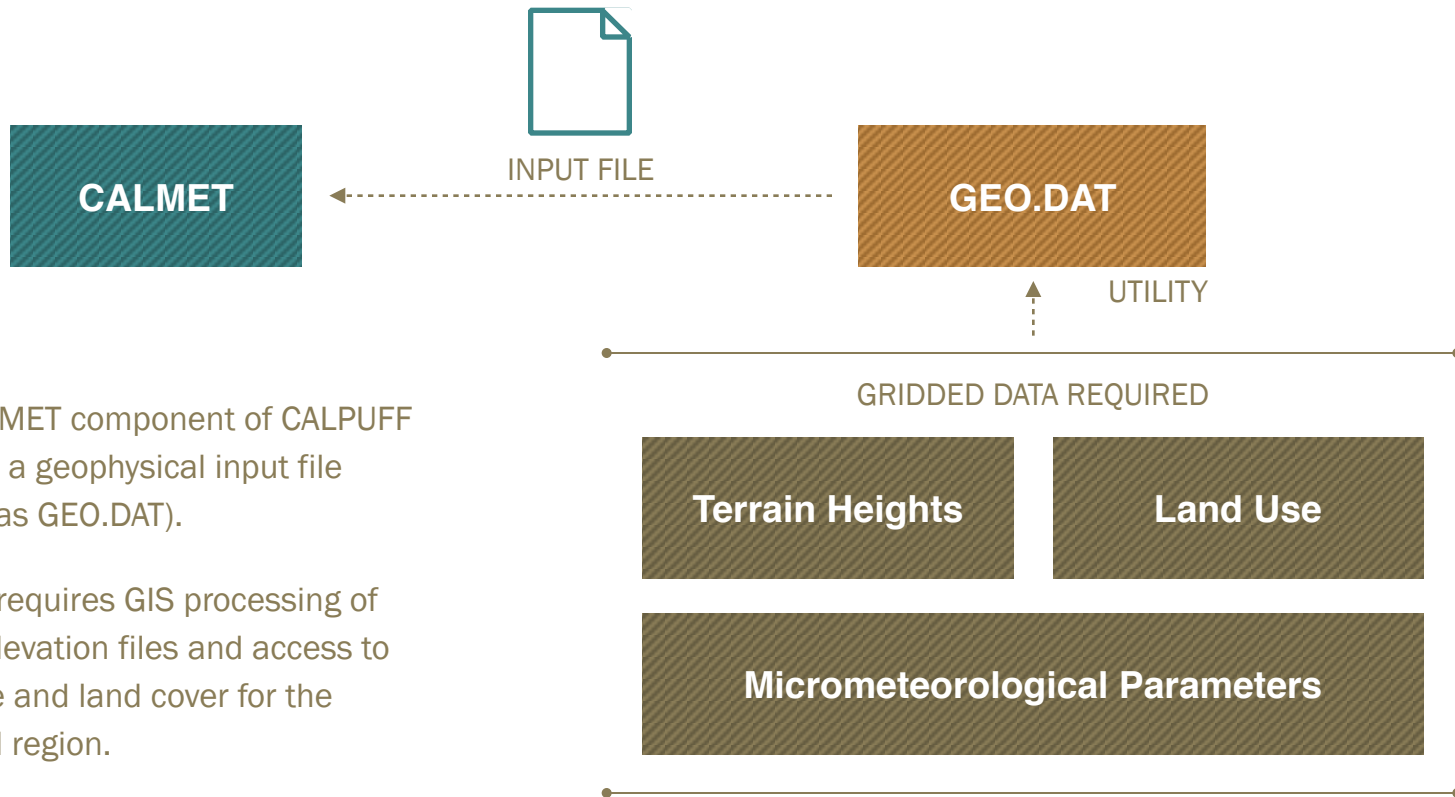
The CALPUFF Modelling System

The CALPUFF modelling system consists of a collection of formatted input files and binary executables.

The CALPUFF modelling system consists of three main components and a set of preprocessing and postprocessing programs. The main components of the modelling system are CALMET, CALPUFF, and CALPOST (typically used in that order). A number of utility programs aid in the production of the necessary input files.



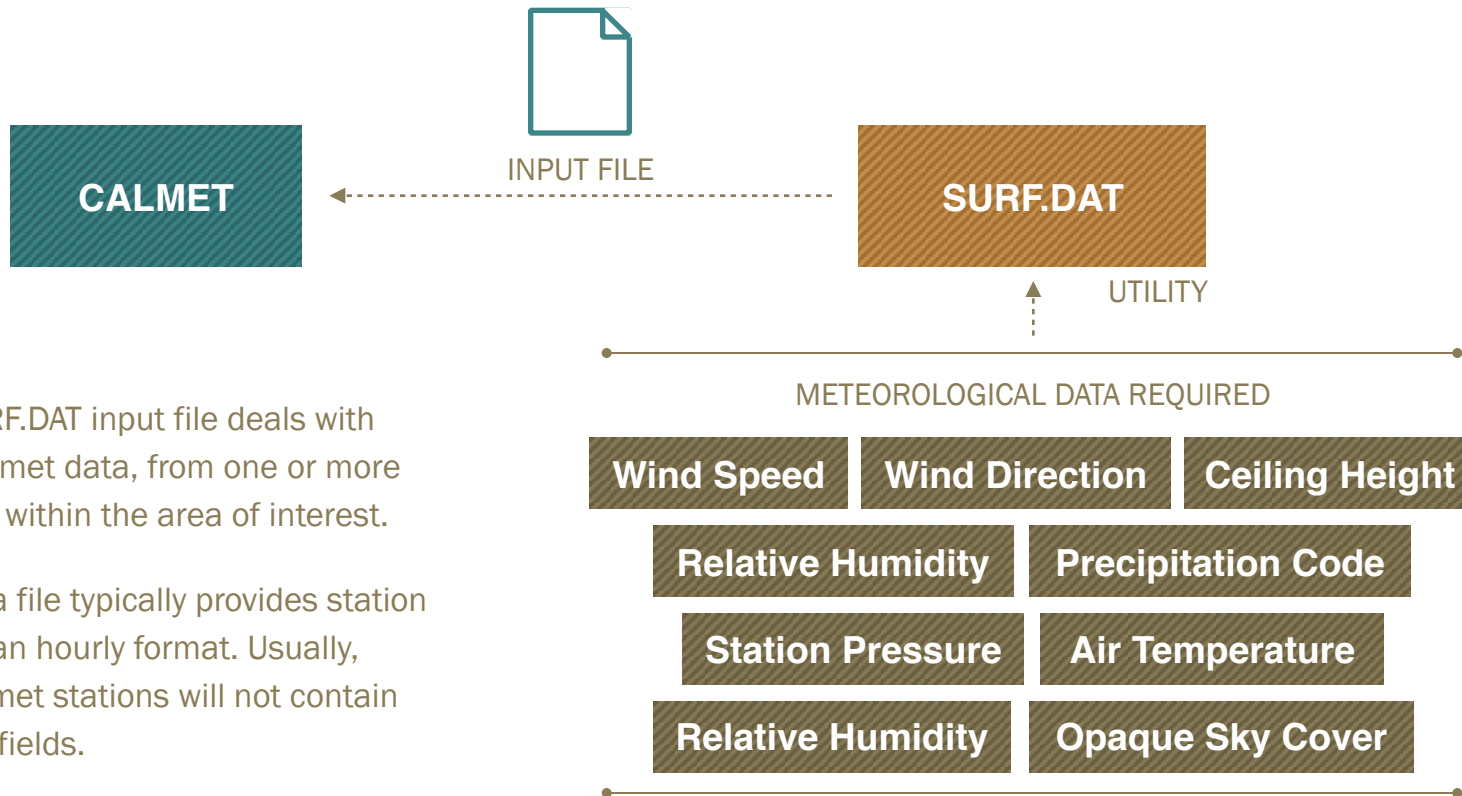
CALMET Primer: Geophysical Data



The CALMET component of CALPUFF requires a geophysical input file (known as GEO.DAT).

The file requires GIS processing of digital elevation files and access to land use and land cover for the selected region.

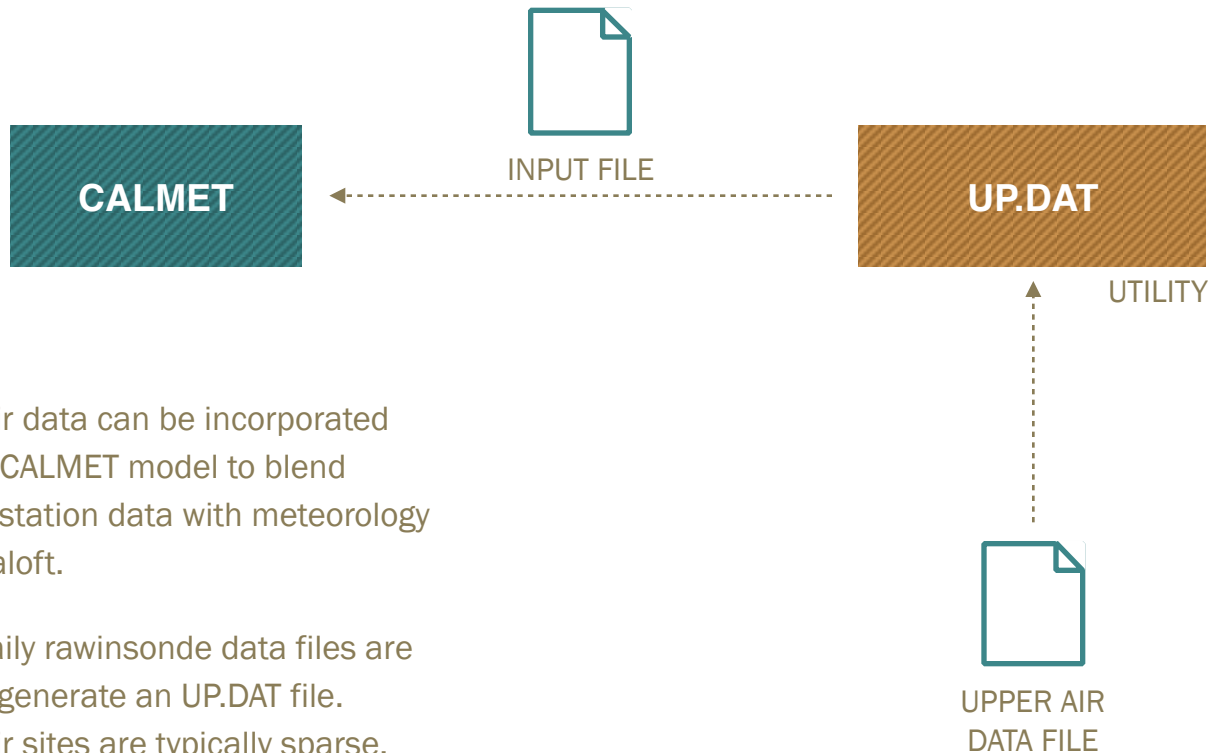
CALMET Primer: Surface Met Data



The SURF.DAT input file deals with surface met data, from one or more stations within the area of interest.

The data file typically provides station data in an hourly format. Usually, nearby met stations will not contain all data fields.

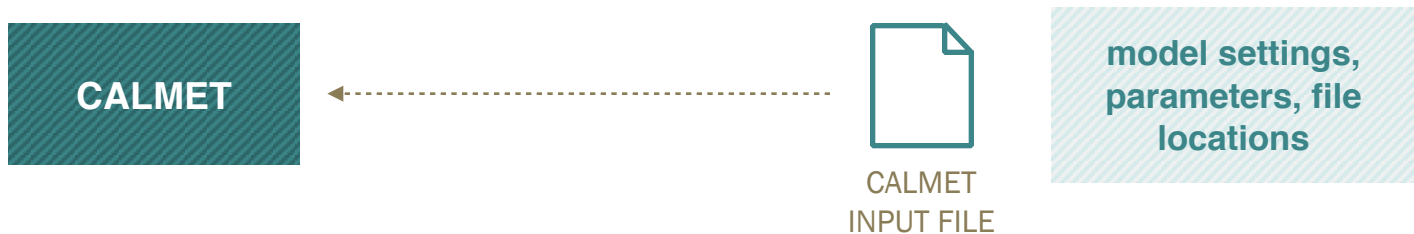
CALMET Primer: Upper Air Data



Upper air data can be incorporated into the CALMET model to blend surface station data with meteorology further aloft.

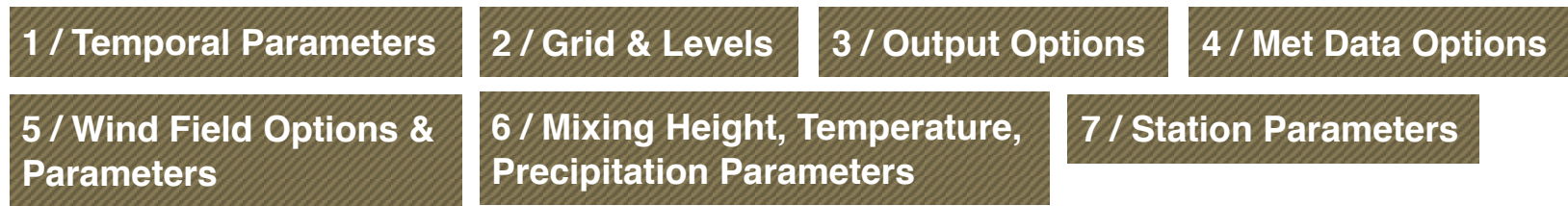
Twice-daily rawinsonde data files are used to generate an UP.DAT file.
Upper air sites are typically sparse.

CALMET Primer: Control File

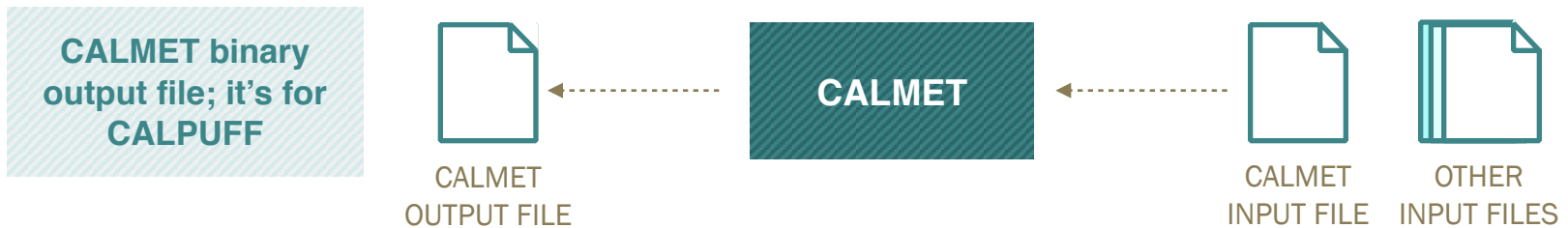


The CALMET input file contains a large variety of model option, parameters, and I/O settings.

These controls are split across several input groups:



CALMET Primer: Model Execution



The CALMET model is to be run with the input file and the associated input data files.

There are several other types of input data files that could be used, depending on model settings:

Precipitation Data File

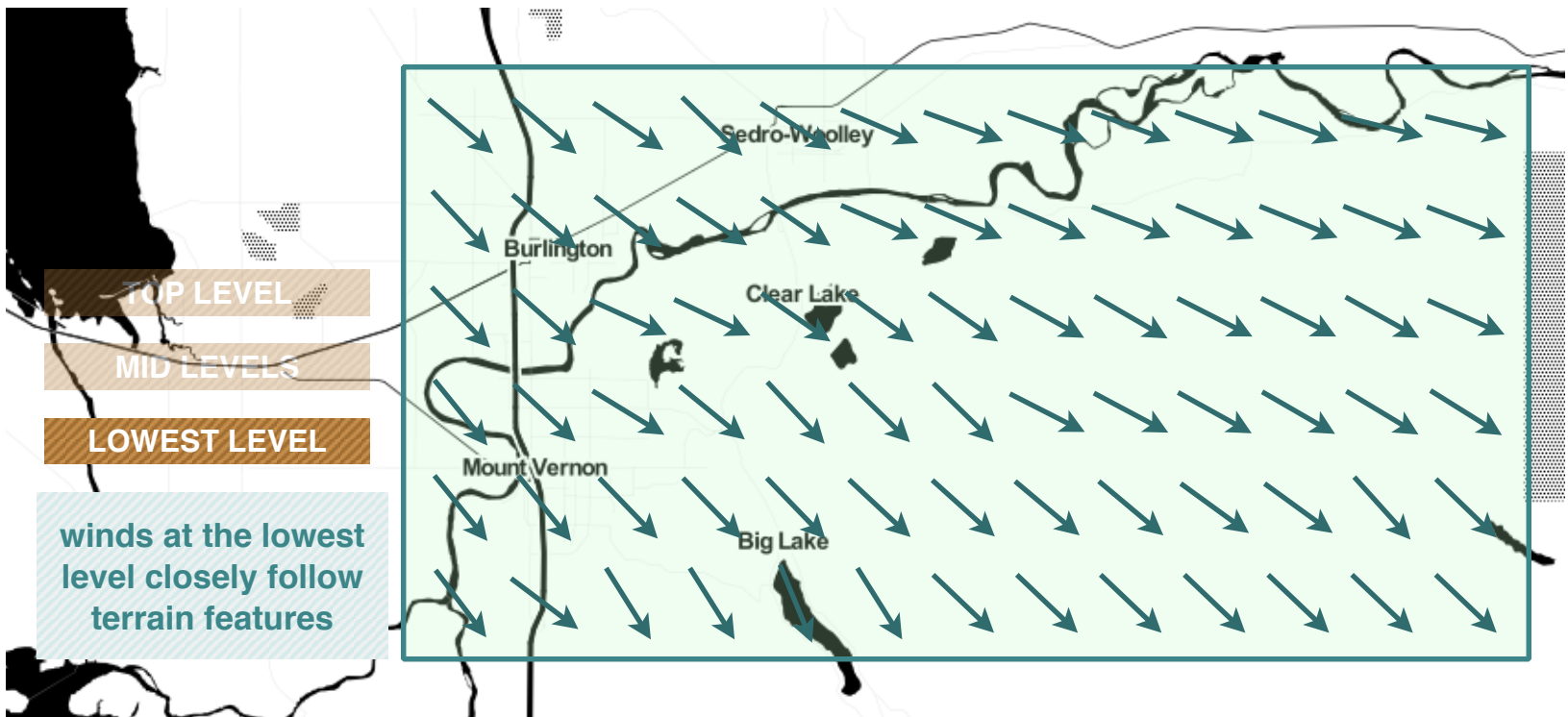
Overwater Station Files

Gridded Cloud Field File

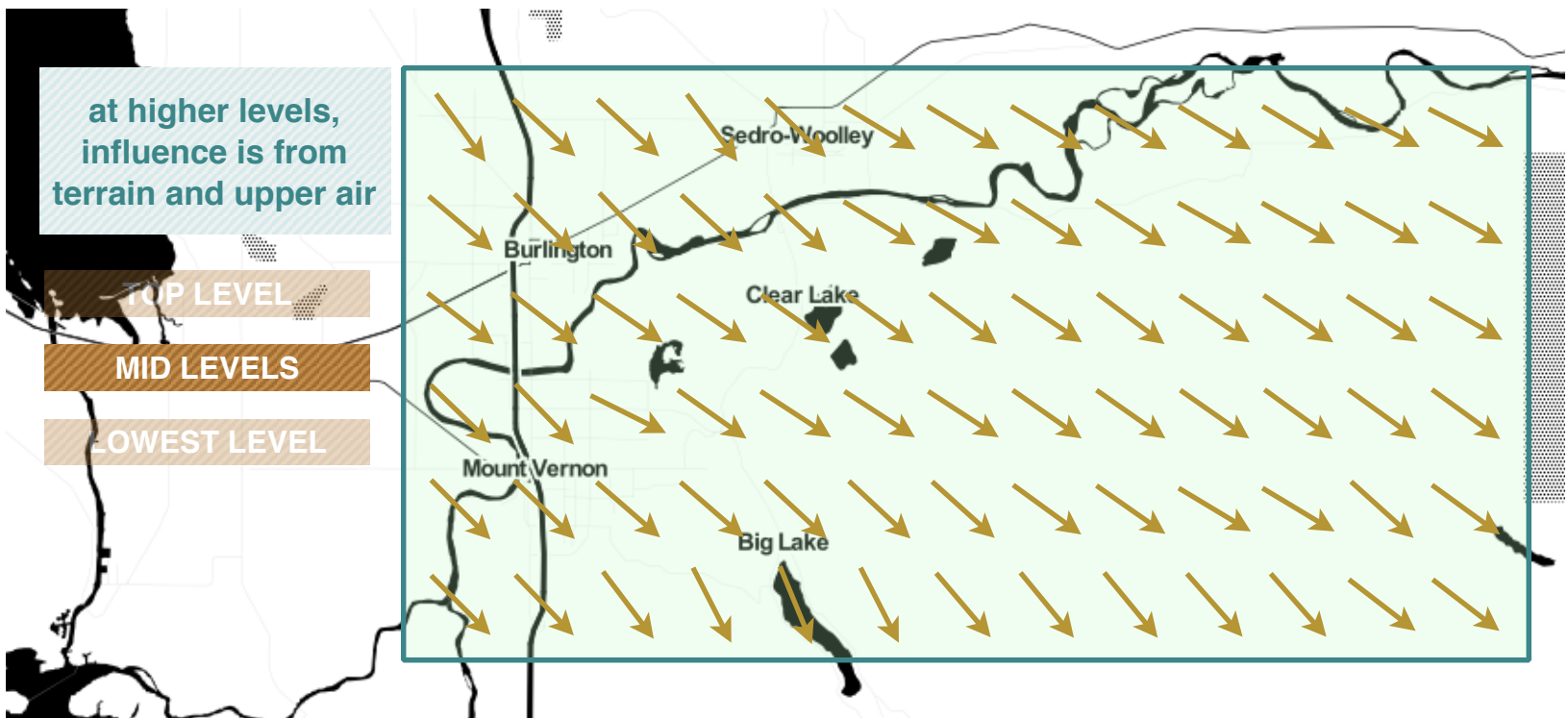
Preprocessed Met Data for Diagnostic Wind Module

Hourly Gridded Wind Fields

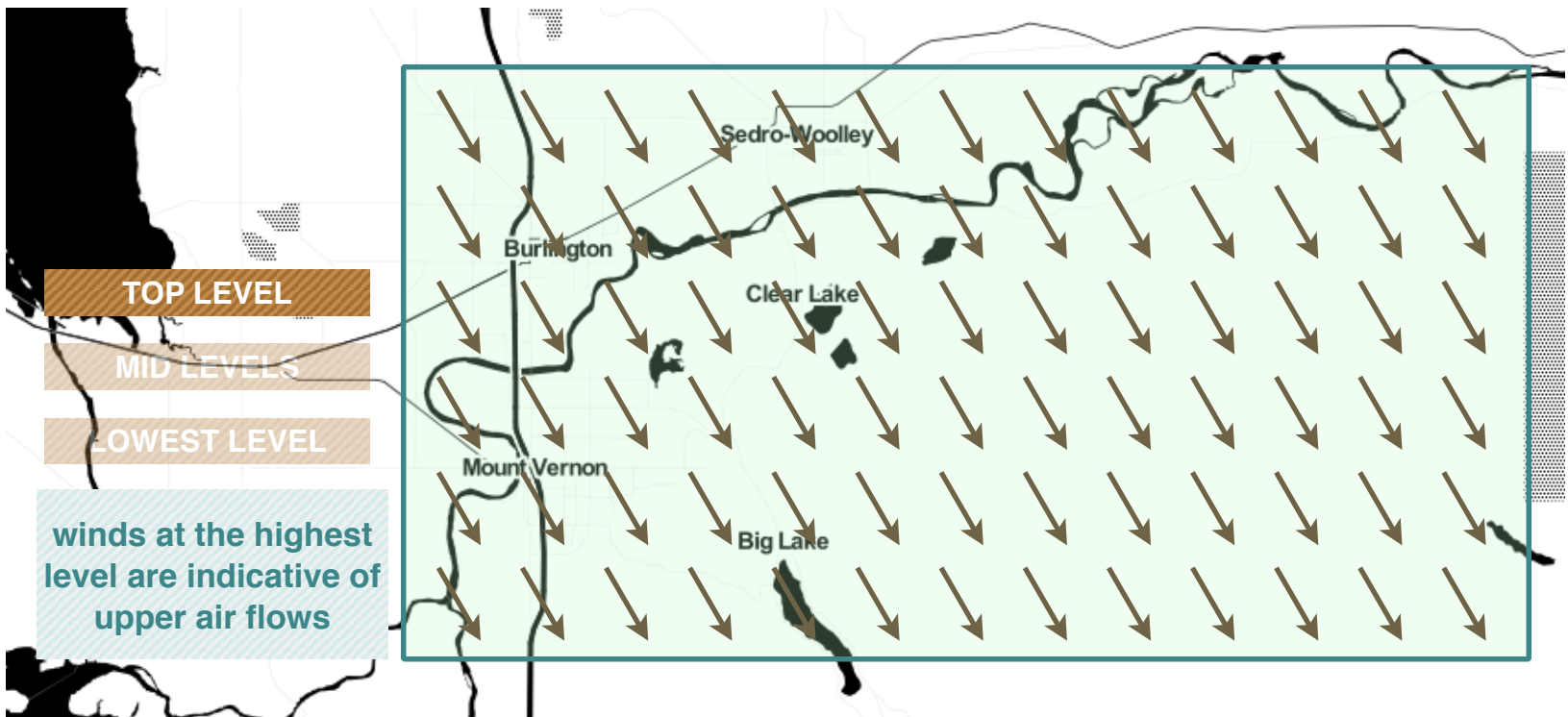
CALMET Primer: Model Wind Fields



CALMET Primer: Model Wind Fields



CALMET Primer: Model Wind Fields

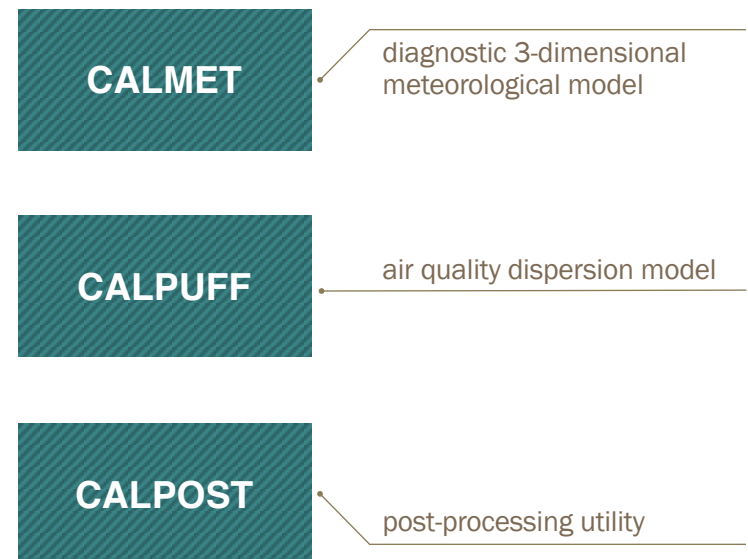


The CALPUFF Modelling System

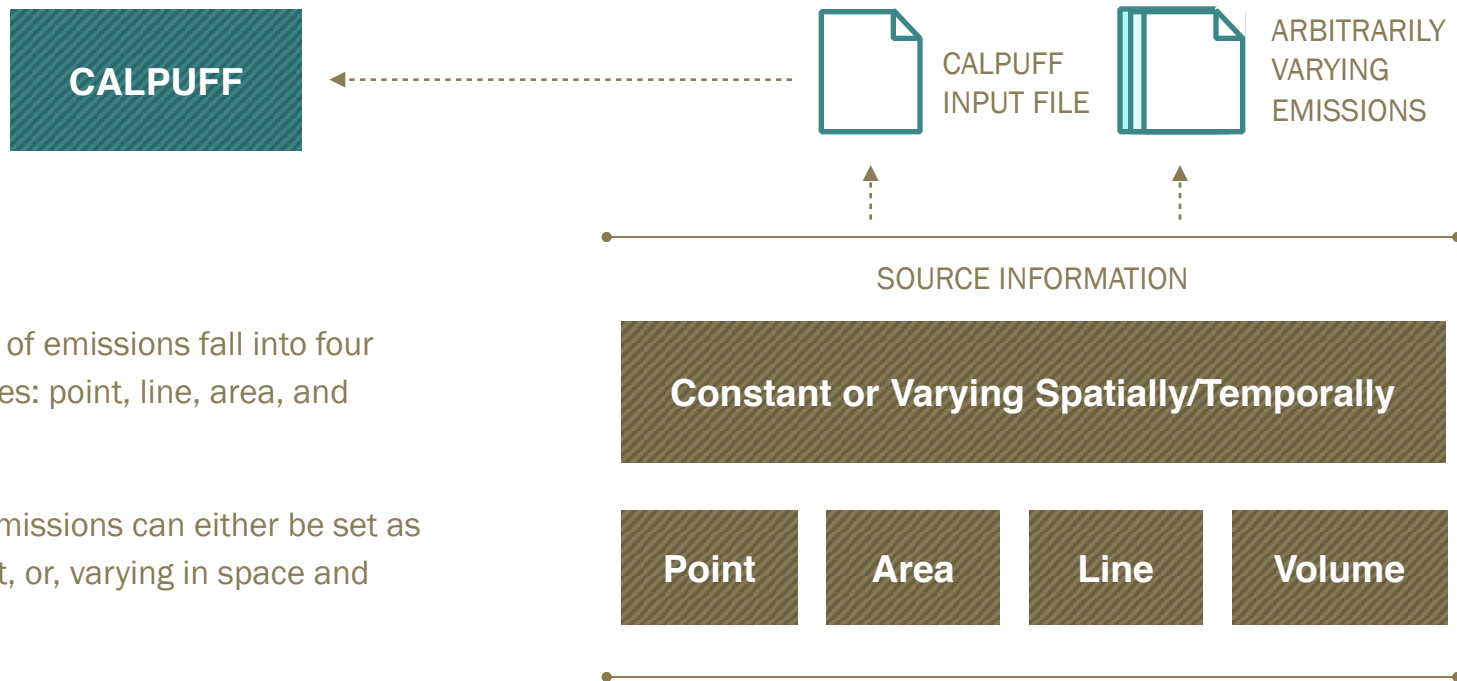
After running the CALMET model, inputs to CALPUFF must be prepared.

The CALPUFF air quality dispersion model requires CALMET model output and a properly formatted CALPUFF input control file.

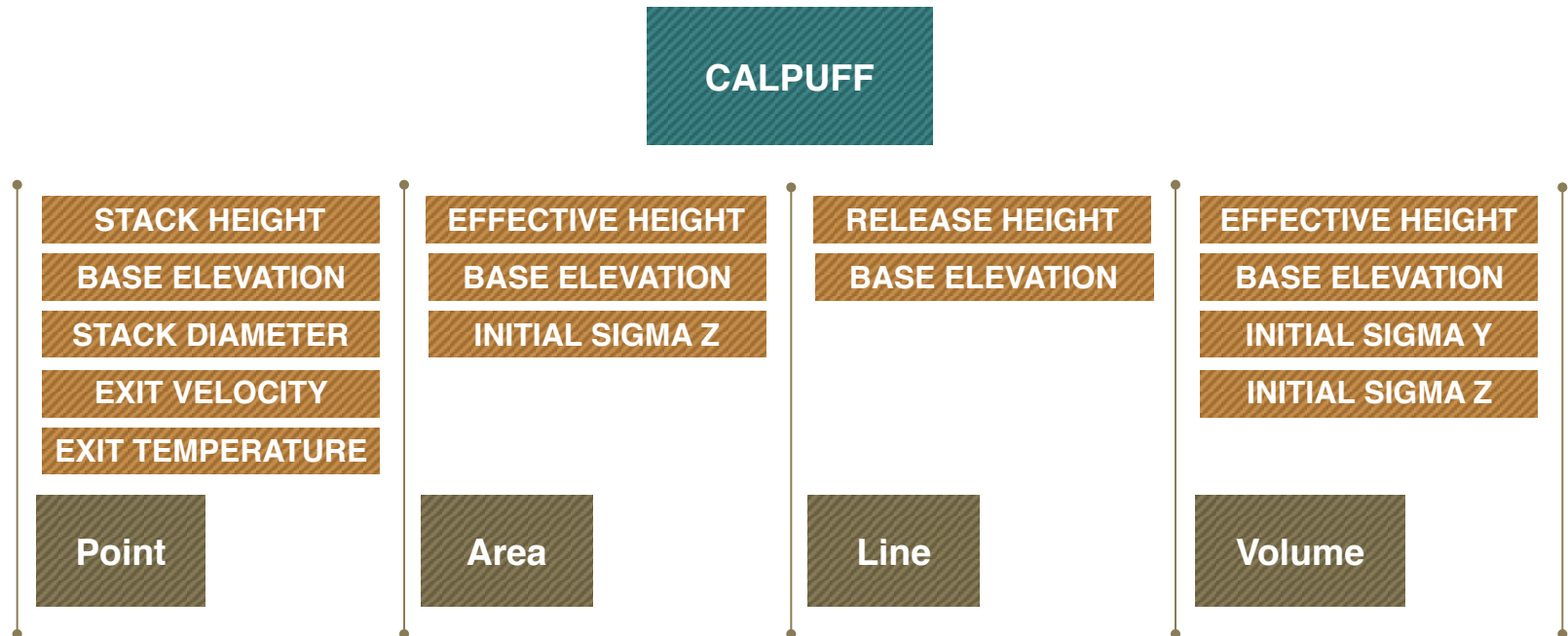
The CALPUFF input control file contains a vast array of options for defining a dispersion model.



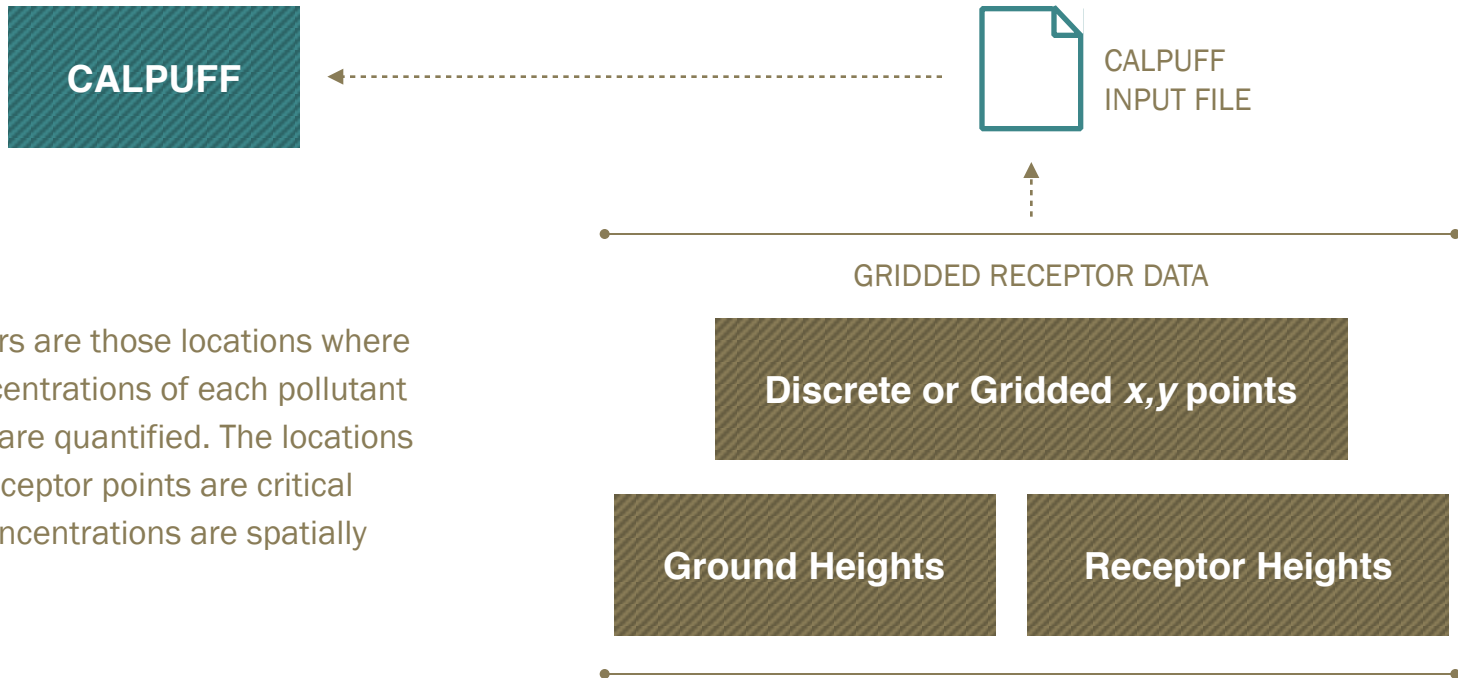
CALPUFF Primer: Sources



CALPUFF Primer: Source Parameters



CALPUFF Primer: Receptors



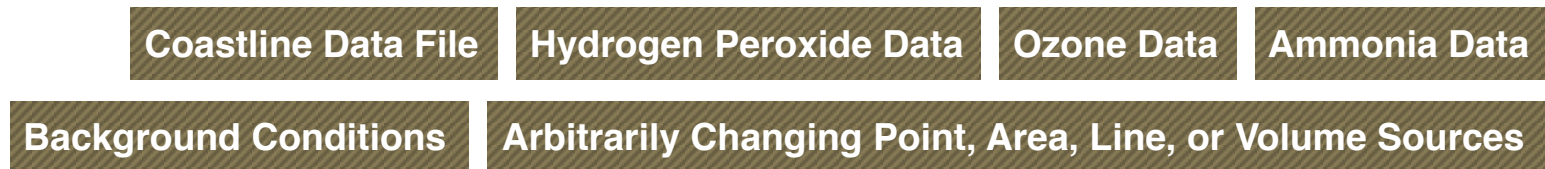
Receptors are those locations where the concentrations of each pollutant species are quantified. The locations of the receptor points are critical since concentrations are spatially varied.

CALPUFF Primer: Model Execution



The CALPUFF model is to be run with the input file and the associated input data files.

There are several other types of input data files that could be used, depending on model settings:



Contents

- 01 Dispersion Modelling and CALPUFF
- 02 The Goals of the PuffR R Package Project
- 03 Proposed Data Sources for PuffR
- 04 Installation of R, RStudio, and PuffR
- 05 The Current PuffR Workflow and Examples of Use
- 06 The Project Roadmap

The Goals of the PuffR R Package Project

Main Goals

The workflow for atmospheric dispersion modelling with CALPUFF needs to be reconsidered—both in the interest of saving time and ensuring that the quality of the input data is high

There are many goals for the PuffR package. Here are some of the ways PuffR can provide some value.

**data visualizations
generated at each
step of the process**

**rely on sensible
defaults and DRY
principles to
automate mundane
data collection tasks**

**useful help system
and example library
to aid in the
understanding of
every model setting**

**automatically collect
and process the best
data available**

Data Visualization

CALPUFF requires (and generates!) a lot of data. It would be nice to visualize that.

If it's not visualized, how can we make sense of the data? Visuals confirm what we might know and they allow us to explore the data and more easily find connections and patterns.

The ocular nerve should be used for more than looking at pages of plaintext data tables.

**data visualizations
generated at each
step of the process**

TERRAIN MAPS

WIND VECTORS

STABILITY CLASSES

SOURCES

RECEPTORS

CONCENTRATIONS

VERTICAL PROFILES

ANIMATIONS

Sensible Defaults

Much of the time, atmospheric dispersion modelling follows a predictable workflow.

Reducing the need to repeat information across different tasks can reduce the possibility for user error.

Reducing the need to specify commonly-used settings and parameters is also great. If you need to drill down and modify things, that's possible (but won't be often required).

rely on sensible defaults and DRY principles to automate mundane data collection tasks

DON'T

REPEAT

YOURSELF

DON'T

REPEAT

YOURSELF

DON'T

REPEAT

YOURSELF

JUST DON'T

Documentation and Examples

It cannot be understated how important a good documentation system is for complex tasks.

To aid in understanding of what is actually going on when you conduct model runs, a help system is vital. You'll learn more.

I'm a big believer in using examples to learn about why certain things are done. To that end, full-featured examples will be provided as part of the PuffR package.

**useful help system
and example library
to aid in the
understanding of
every model setting**

THEORY

SETTINGS

PARAMETERS

CASE STUDIES

EXAMPLES

DEFINITIONS

SEARCHABLE INDEX

Automation of Data Collection

There is an staggering quantity of publicly-available data. PuffR will collect some of that on your behalf.

A model is nothing without good input data. Fortunately there is plenty of no-cost source data, and, PuffR will obtain the best data and prepare the appropriate input data files.

The aim is to allow for the automated preparation of CALMET input data for the widest range possible of global locations.

**automatically collect
and process the best
data available**

THEORY

SETTINGS

PARAMETERS

CASE STUDIES

EXAMPLES

DEFINITIONS

SEARCHABLE INDEX

Contents

- 01 Dispersion Modelling and CALPUFF
- 02 The Goals of the PuffR R Package
- 03 Proposed Data Sources for PuffR

- 04 Installation of R, RStudio, and PuffR
- 05 The Current PuffR Workflow and Examples of Use
- 06 The Project Roadmap

Summary of Data Sources

Type of Data	Name and Description	Provider
Surface Station Meteorology	1-Hourly Datasets for Global Meteorological Stations	National Climatic Data Center (NCDC)
Upper Air Data	RAOB Radiosonde Database	Earth System Research Laboratory (ESRL) and National Oceanographic and Atmospheric Administration (NOAA)
Surface Elevation	1 // U.S. National Elevation Data (NED) 2 // Canadian Digital Elevation Data (CDED) 3 // Global SRTM V4 GeoTIFF Archive	1 & 3 // U.S. Geological Survey (USGS) 2 // GeoBase.ca
Land Use and Land Cover	1 // U.S. National Land Cover Data (NLCD) 2011 2 // GeoBase Land Cover Product (Canada) 3 // GlobCover 2009 (Global Land Cover Map) 4 // MODIS Gridded Land Cover Data	1 // U.S. Geological Survey (USGS) / Multi-Resolution Land Characteristics Consortium (MRLC) 2 // GeoBase.ca 3 // European Space Agency (ESA) data user element (due) 4 // MODIS

Hourly Surface Met Station Data

The National Climatic Data Center (NCDC) has an archive of hourly (and sub-hourly) surface station data. It's coverage is global.

Surface station data is essential if you're not running CALPUFF solely with prognostic meteorological fields (e.g., WRF, RUCS, etc.).

Even if you do have modelled met fields, the surface station data can be blended together with that (i.e., running a hybrid model).

```
027071785099999200601010004+49017-122783SY-SA+0013CWWK
0070717850999992006010101004+49017-122783SAO +0013CWWK
0070717850999992006010102004+49017-122783SAO +0013CWWK
0070717850999992006010103004+49017-122783SAO +0013CWWK
0070717850999992006010104004+49017-122783SAO +0013CWWK
0070717850999992006010105004+49017-122783SAO +0013CWWK
0270717850999992006010106004+49017-122783SY-SA+0013CWWK
0070717850999992006010107004+49017-122783SAO +0013CWWK
0070717850999992006010108004+49017-122783SAO +0013CWWK
0070717850999992006010109004+49017-122783SAO +0013CWWK
0070717850999992006010110004+49017-122783SAO +0013CWWK
0070717850999992006010111004+49017-122783SAO +0013CWWK
0260717850999992006010112004+49017-122783SY-SA+0013CWWK
0070717850999992006010113004+49017-122783SAO +0013CWWK
0070717850999992006010114004+49017-122783SAO +0013CWWK
0070717850999992006010115004+49017-122783SAO +0013CWWK
0070717850999992006010116004+49017-122783SAO +0013CWWK
0070717850999992006010117004+49017-122783SAO +0013CWWK
0070717850999992006010119004+49017-122783SAO +0013CWWK
0070717850999992006010120004+49017-122783SAO +0013CWWK
```

Upper Air Rawinsonde Data

Upper air data: because the met observations are not the same at the surface as they are further aloft.

CALMET generates meteorological data fields within the x,y grid of cells for every z level specified. Upper air soundings provide vital data for wind, temperatures, and pressure values well above the surface.

The RAOBS archive contains twice-daily global soundings. UP.DAT files can be made from them.

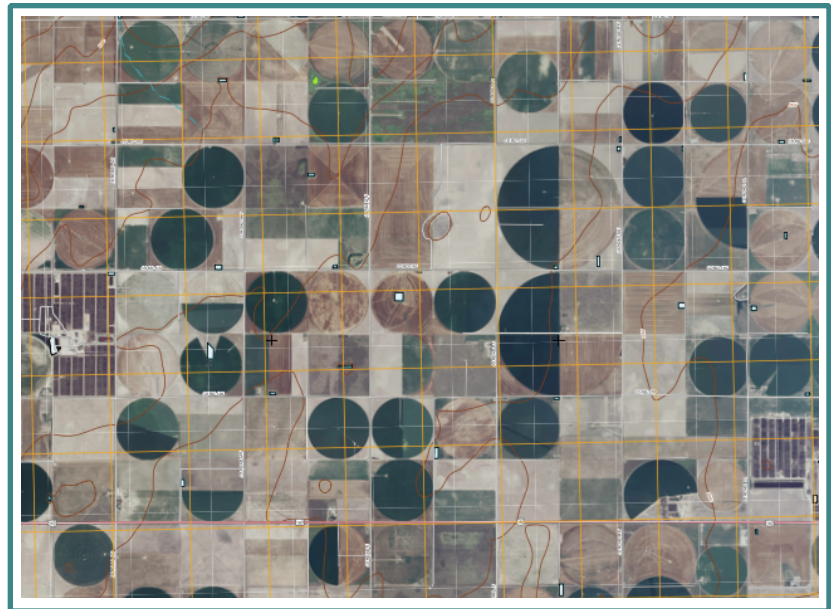
254	0	7	NOV	2014		
1	3990	72249	32.80N	97.30W	196	2307
2	70	122	1350	151	99999	3
3		FWD			99999	kt
9	10020	196	202	32	20	7
4	10000	212	200	30	20	8
6	9545	609	99999	99999	20	14
4	9250	877	146	-54	20	18
6	9209	914	99999	99999	25	20
5	9120	996	138	-82	99999	99999
5	9030	1079	132	-118	99999	99999
6	8880	1219	99999	99999	25	22
5	8800	1294	120	-60	99999	99999
5	8510	1572	102	-78	99999	99999
4	8500	1585	102	-68	10	21
6	8253	1828	99999	99999	20	30
5	8190	1891	82	-48	99999	99999
5	8010	2073	72	-68	99999	99999
6	7952	2133	99999	99999	25	23
5	7860	2228	74	-116	99999	99999

Surface Elevation Data

The knowledge of surface elevation and presence of coastlines has a great effect on near-surface wind flow.

There are many high-resolution terrain datasets that PuffR can use to generate gridded surface elevation fields for the CALMET GEO.DAT file.

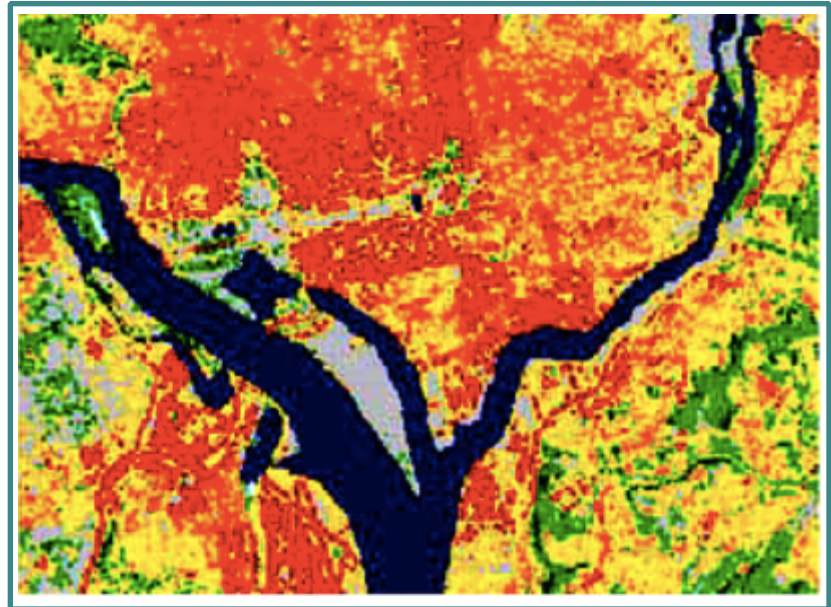
In the U.S., the National Elevation Data (NED) product is best. In Canada, it's Canadian Digital Elevation Data (CDED). Globally, the SRTM V4 GeoTIFF Archive is suitable.



Land Use and Land Cover

The characteristics of the land have a marked effect on the wind flow and a host of micrometeorological parameters.

Different world jurisdictions have their own land use and land cover databases. The GlobCover Global Land Cover Map provides a standardized dataset for land cover. While the data sources are diverse, a goal off the PuffR project is to incorporate as many of these datasets as possible. This data is essential.



Contents

- 01 Dispersion Modelling and CALPUFF
- 02 The Goals of the PuffR R Package
- 03 Proposed Data Sources for PuffR

- 04 Installation of R, RStudio, and PuffR
- 05 The Current PuffR Workflow and Examples of Use
- 06 The Project Roadmap

Where to Get the Installers

There are two places where you must go to get software installed. It'll be worth it.



r-project.org

What's R? It's a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Mac OS X, and Windows.



rstudio.com

RStudio is an IDE that provides a great set of tools for working in R. You get a code editor, a console, and a customizable panes for package management, navigating local files, source control, etc.

Installation of R, RStudio, and PuffR

Some Information on RStudio: that Excellent IDE for R

Basic overview on where stuff is.

source

console

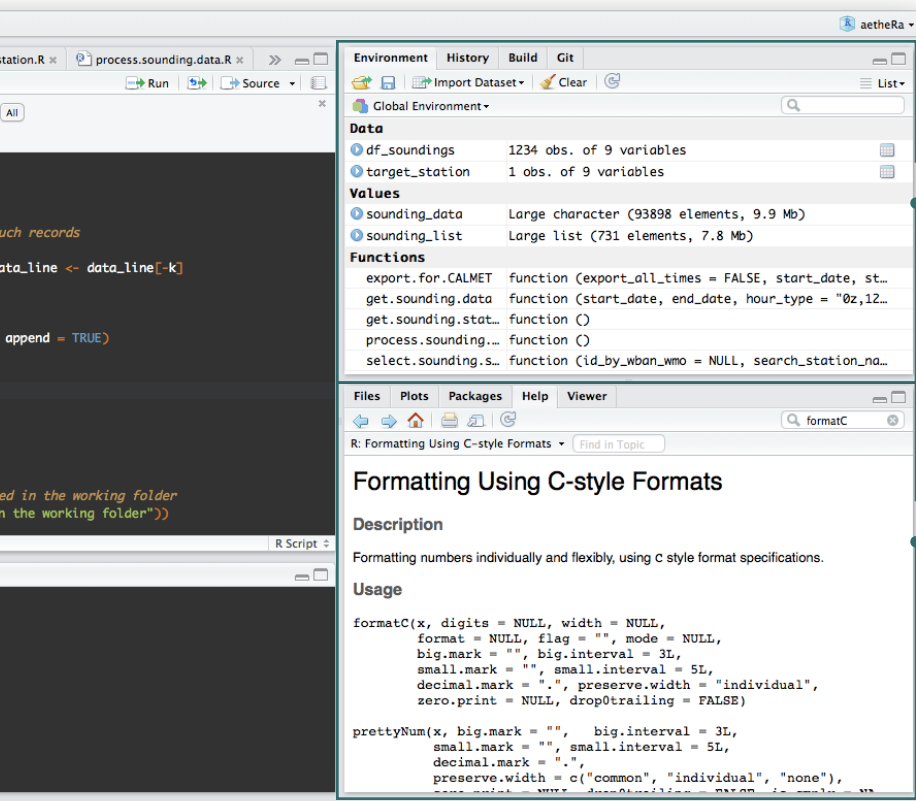
The screenshot displays the RStudio interface. The top pane shows the source editor with an R script file named 'export.for.CALMET.R'. The script contains comments and code for processing sounding data, including a loop to scan for negative heights and a function to write data to a file. The bottom pane shows the console with the output of the 'tail(sounding_data, 140)' command, displaying a table of data rows. The right sidebar shows the Environment pane with a list of objects and the Files pane showing the project structure.

```
209  
210     # Close loop for data lines  
211   }  
212  
213   # Scan for entries with negative height and delete such records  
214   for (k in 1:length(data_line)) {  
215     if (trimmed_sounding_list[[i]][[2]][[k, 3]] < 0) data_line <- data_line[-k]  
216   }  
217  
218   # Write data lines to file  
219   cat(data_line, sep = "\n", file = "test_output.txt", append = TRUE)  
220  
221   # Close loop for header line  
222 }  
223  
224 # Remove objects from global environment  
225 rm(header_line_constant, header_line, data_line,  
226     export_all_times, top_pressure_level, i, j, k)  
227  
228 # Add notification that UP.DAT file generated and placed in the working folder  
229 print(paste("An UP.DAT file was generated and placed in the working folder"))  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```

```
> tail(sounding_data, 140)  
[1] " 6 132 28956 99999 99999 125 123"  
[2] " 5 124 29318 -649 99999 99999 99999"  
[3] " 6 119 29565 99999 99999 95 159"  
[4] " 5 116 29726 -629 99999 99999 99999"  
[5] " 6 108 30176 99999 99999 105 113"  
[6] " 6 103 30480 99999 99999 140 118"  
[7] " 4 100 30640 -657 99999 130 118"  
[8] " 6 98 30784 99999 99999 110 134"  
[9] " 5 94 31013 -681 99999 99999 99999"  
[10] " 6 93 31089 99999 99999 110 175"  
[11] " 6 88 31384 99999 99999 120 129"
```

Installation of R, RStudio, and PuffR

Some Information on RStudio: that Excellent IDE for R



environment
history
build (optional)
version control (optional)

files
plots
packages
help
viewer

How to get the development version of PuffR

Getting the development version of PuffR is not very complicated. Just follow these steps.

You have R and possibly RStudio? Great! The next thing to do is make certain that you have the ‘devtools’ package installed. It’s a package that makes it easy to install R packages that are not on CRAN (the Comprehensive R Archive Network). Just type this into the R console:

```
install.packages(devtools)
```

The PuffR package source code is hosted on GitHub. However, you don’t need to visit the repository to get the package installed. Simply use the following command in the R console:

```
install_github('PuffR', 'rich-iannone')
```

Then you should have it. Between sessions, you may have to use the following command to load the package:

```
require(PuffR)
```

Otherwise, the PuffR functions will not be available for use.

General Warnings about Packages in Active Development

code on the
repository changes *a
lot*, without warning

the changing code
may break certain
functions, or, may
cause unintended
errors in values

the code needs to
change for the better;
but bugs do happen
and they will always
just happen

blah, blah, blah...
yeah, this is all very
scary stuff

Installation of R, RStudio, and PuffR

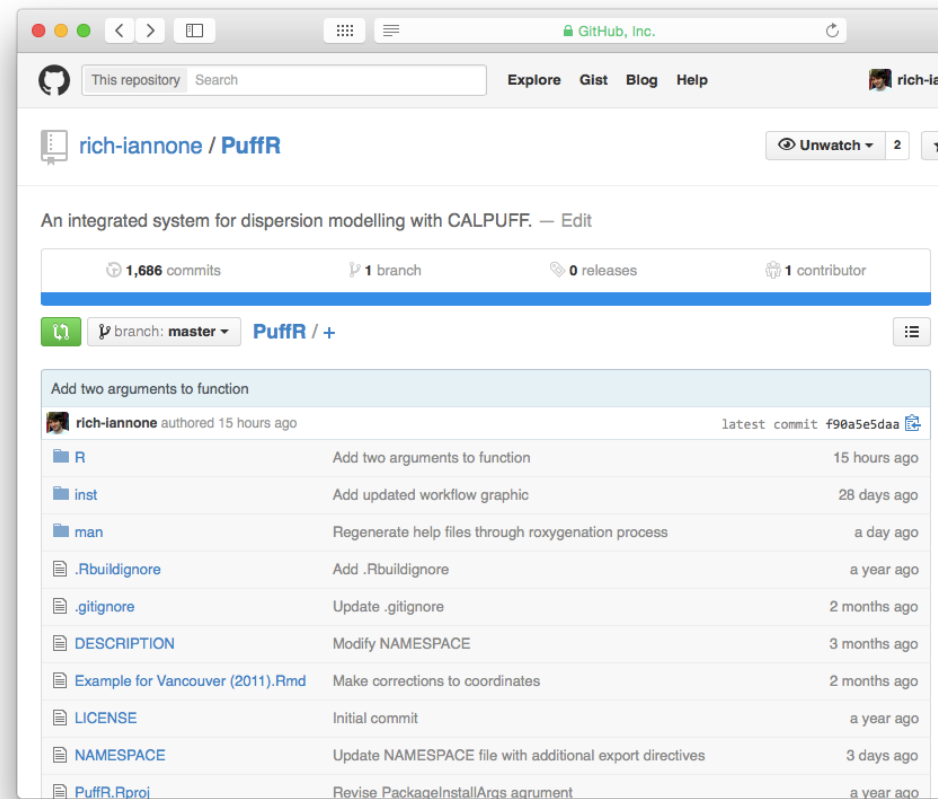
The PuffR Source Repository on GitHub

The source for the PuffR package lives on GitHub. Have a peek at the code or the README, if you're so inclined.

Point your browser to this  address:

<https://github.com/rich-iannone/PuffR>

If you're a member of GitHub and you like what you see. Give me a ★— I really appreciate my stargazers.



Contents

- | | | | |
|----|----------------------------------|----|--|
| 01 | Dispersion Modelling and CALPUFF | 04 | Installation of R, RStudio, and PuffR |
| 02 | The Goals of the PuffR R Package | 05 | The Current PuffR Workflow and Examples of Use |
| 03 | Proposed Data Sources for PuffR | 06 | The Project Roadmap |

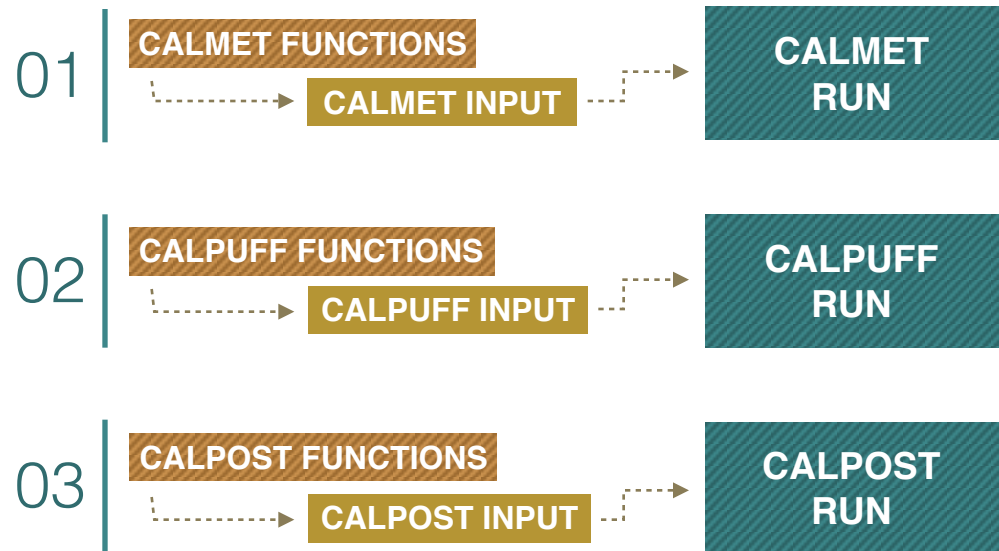
The Current PuffR Workflow and Examples of Use

From CALMET to CALPOST— It's Functions All the Way Down

The PuffR workflow proceeds in the expected order.

The basic method for use of PuffR is to call its functions for building the model inputs. You are to supply the basic info and get specific only when you need to.

There is a specified order, since later function calls will depend on output generated from earlier-called functions.



Using the PuffR Function to Define Receptor Locations for CALPUFF

In this example, the geophysical input file (GEO.DAT) is produced with a single function call.

First, some preliminaries! Create a folder with the name of the location. Create an R script and place it in that folder. In that new script, add a require statement for PuffR and set the working directory:

```
require(PuffR)
setwd('~/.PuffR/vancouver')
```

Use the following function call:

```
calmet_define_geophys(  
  location_name = 'vancouver',  
  lat_dec_deg = 49.196116,  
  lon_dec_deg = -122.505866,  
  domain_width_m = 117000,  
  domain_height_m = 43250,  
  cell_resolution_m = 500,  
  download_SRTM = TRUE)
```

This will consist of a grid centered on 49.196116°N and 122.505866°W. The width (E–W distance) of the grid will be 117000 m, and the height (N–S distance) will be 43250 m. We will download SRTM terrain height data and store it locally.

Using PuffR to Generate Surface Met and Upper Air Data Files

The surface met input file—commonly known as SURF.DAT—can be easily made.

Here is the function call:

```
calmet_surface_met(  
  location_name = 'vancouver',  
  year = 2011,  
  lat_dec_deg = 49.196116,  
  lon_dec_deg = -122.505866,  
  domain_width_m = 117000,  
  domain_height_m = 43250,  
  cell_resolution_m = 500,  
  time_offset = -8)
```

It relies on the location's domain to obtain hourly met data for 2011 from the NCDC archive.

Same goes for the upper air data file: UP.DAT.

This will look familiar:

```
calmet_upper_air(  
  location_name = 'vancouver',  
  year = 2011,  
  lat_dec_deg = 49.196116,  
  lon_dec_deg = -122.505866,  
  domain_width_m = 117000,  
  domain_height_m = 43250,  
  cell_resolution_m = 500,  
  time_offset = -8)
```

This collects upper air data from two nearby sounding sites and generates the required files.

Putting Together the CALMET Input Control File

CALMET input control files (CALMET.INP) are created right before executing the CALMET model.

Start with a template:

```
calmet_inp_generate_template()
```

This creates an effectively empty CALMET input file in the working directory. While this file is readable plaintext, it really shouldn't be modified by hand.

We will use a group of functions to populate input files with parameter values. In this way, validation of inputs can be performed at every step.

The CALMET input file can be built up using a series of functions that address each of the input file's main sections.

```
calmet_01_temporal_params()  
calmet_02_grid_levels()  
calmet_03_output_opts()  
calmet_04_met_data_opts()  
calmet_05_wind_field_opts_params()  
calmet_06_mixhgt_temp_precip_params()  
calmet_07_station_params()  
calmet_inp_finalize()
```

These call build up and complete the creation of the CALMET.INP file(s), checking the working directory for created input files. The presence of those files signals inclusion in CALMET model execution.

The Current PuffR Workflow and Examples of Use

The Generated CALMET Input Files

'GEO.DAT' INPUT FILES



geo--vancouver-234x86x500--1-winter.txt
geo--vancouver-234x86x500--2-spring.txt
geo--vancouver-234x86x500--3-summer.txt
geo--vancouver-234x86x500--4-fall.txt
geo--vancouver-234x86x500--5-winter.txt

'UP.DAT' INPUT FILE



up--vancouver-234x86x500--2011.txt

'SURF.DAT' INPUT FILE



surf--vancouver-234x86x500--2011.txt

'CALMET.INP' INPUT FILES

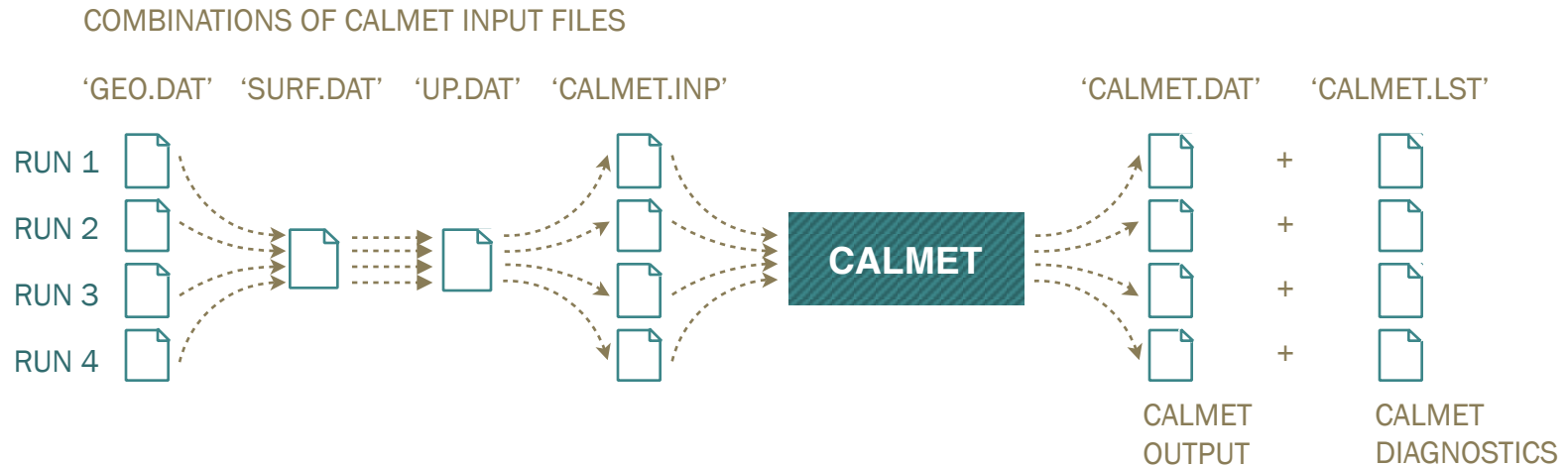


calmet_in--vancouver-234x86x500--2011-1-winter.txt
calmet_in--vancouver-234x86x500--2011-2-spring.txt
calmet_in--vancouver-234x86x500--2011-3-summer.txt
calmet_in--vancouver-234x86x500--2011-4-fall.txt
calmet_in--vancouver-234x86x500--2011-5-winter.txt

Those function calls created quite a few input files!

Those functions used sensible defaults that assumed quite a lot. For instance, there were seasonal splits (making separate geophysical input files in different seasons). Also, the functions automatically chose which public datasets to use. Of course, you can have granular control of this but, oftentimes, you don't need to.

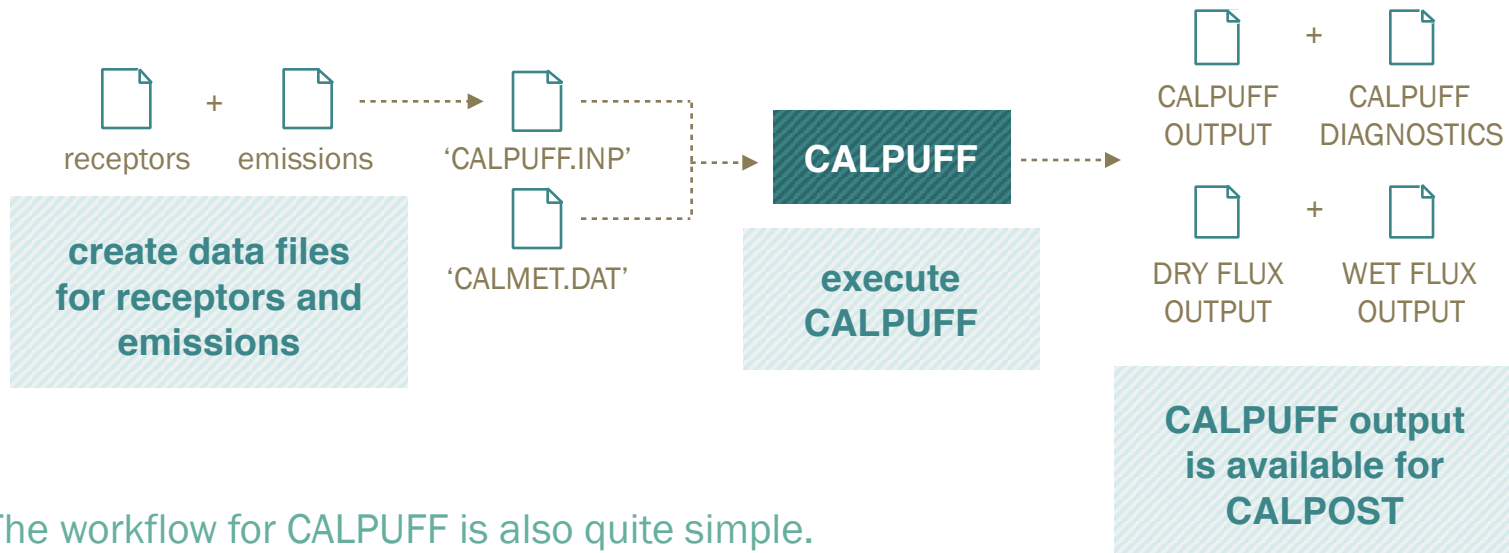
The Generated CALMET Output Files



Combinations of CALMET input files generate the appropriate output files.

A function call executes the CALMET runs from the input files. Once each CALMET model run is executed, binary CALMET output files (and associated diagnostics files) are created with a strict naming convention.

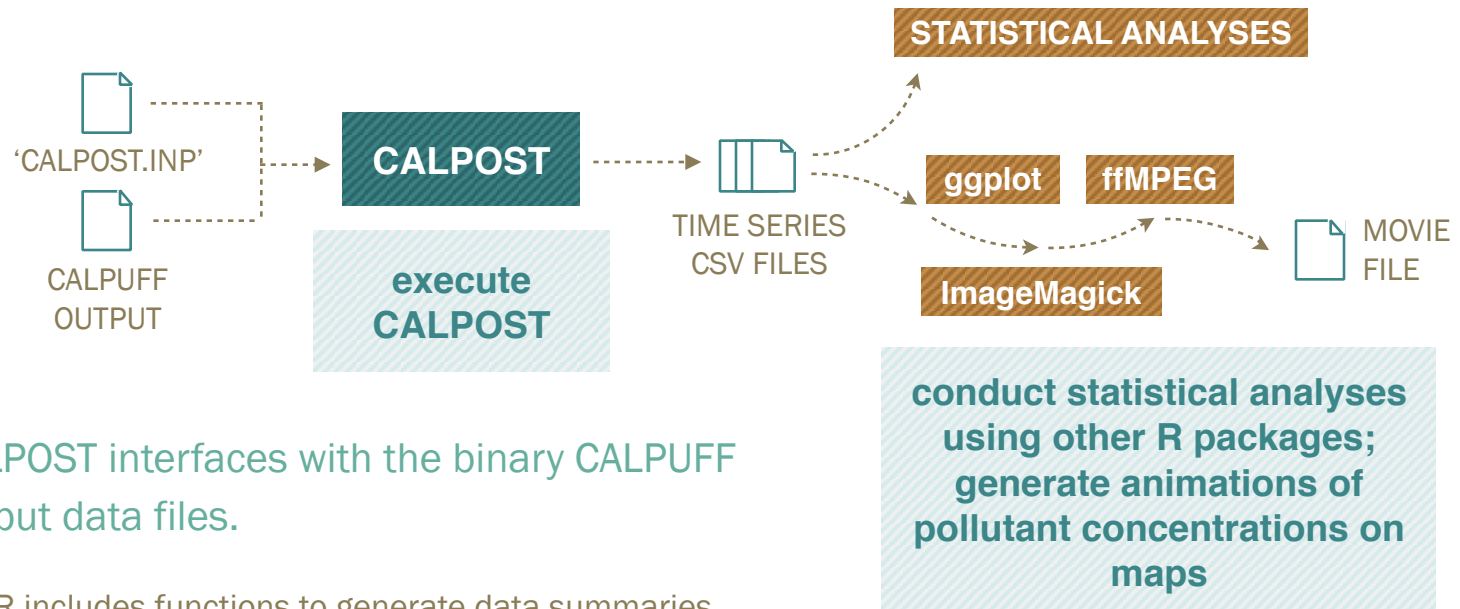
The Analogous CALPUFF Process



The workflow for CALPUFF is also quite simple.

Functions are available to generate receptor grids, define emission sources, build the CALPUFF input control file, and initiate the CALPUFF runs. Self-describing filenames are given to all CALPUFF output.

Analyzing the CALPUFF Output with CALPOST



CALPOST interfaces with the binary CALPUFF output data files.

PuffR includes functions to generate data summaries from binary CALPUFF output data by using the CALPOST utility. For example, time series data for all receptors can be extracted and used in statistical analyses of pollutant exposure.

Contents

- 01 Dispersion Modelling and CALPUFF
- 02 The Goals of the PuffR R Package
- 03 Proposed Data Sources for PuffR
- 04 Installation of R, RStudio, and PuffR
- 05 The Current PuffR Workflow and Examples of Use
- 06 The Project Roadmap

There is a lot planned for this

**greater selection
of data sources**

**method to import and
validate MM4/MM5 files**

**output/visualization of
concentrations at receptors**

**data imputation
methods for upper air
soundings**

**method to create a
precipitation input file
for CALMET**

**method for computing building
downwash for point sources
near large structures**

The intent of PuffR is to greatly simplify the process of running an advanced air quality dispersion model. I think it's important to understand how emissions of pollutants affect populations' exposure to those pollutants.

Aside from simplicity, there will be power. We can take advantage of many high-quality R packages to conduct statistical analyses and gain additional insights from the modelling data. This is all quite exciting and I can't wait to take this further.

**method to create an overwater
meteorological data file for CALMET**

**documentation library for
understanding model options and
parameters**

**methods for specifying emissions
sources and creating time-and-
space-varying emissions**



An integrated system
for dispersion modelling
with CALPUFF



riannone@me.com



[@riannone](https://twitter.com/riannone)



[rich-iannone](https://github.com/rich-iannone)



[rich_i](https://medium.com/@rich_i)



[richard-iannone](https://www.linkedin.com/in/richard-iannone)



[Richard_Iannone](https://www.researchgate.net/profile/Richard_Iannone)