Development and Application of a Simple Model Assessment R Tool (SMART)

Daiwen Kang

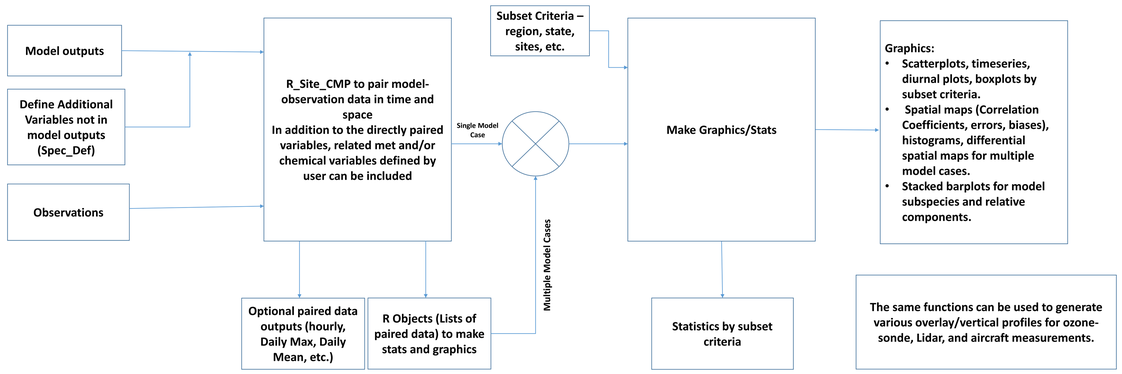
National Exposure Research Laboratory, U.S. Environmental Protection Agency, Research Triangle Park, North Carolina

Daiwen Kang | kang.daiwen@epa.gov | 919-541-4587

Introduction

As an environment for statistical computing and graphics, the programming language R is widely used and freely available. The EPA's Atmospheric Model Evaluation Tool (AMET) is developed using R to produce statistical analysis and generate various graphics. Complementary to AMET, SMART provides an alternative tool to perform model evaluation for simple projects independent of databases and other languages for pre-processing. With the design of unique data structure and common interfaces to process model outputs and observations, SMART can easily pair any model outputs with observations from common observation networks and special field studies. The large memory requirement and slow processing speed associated with the R platform (the language and running environment) when dealing with large datasets are resolved by using SQLDF and its dependent packages and parallel computing packages. Saved R objects serve as pseudo database tables to perform statistical analysis and generate graphics for various model scenarios. With functionality to subset and merge for the designed data structure, it is easy and fast to generate statistics and graphics for any subsets of the model-observation pairs in time and space. Since R is the only language used for pairing model output with observations and performing statistical analysis, it is completely portable to any computing environment with R installed. To run SMART, except for the shell script that is used to define all the environmental variables (such as input and output directories and variable names), users are not required to have any knowledge of R.

Design



Data Structure and Functionality

The primary data structure used in this tool is multiple-level list that saves memory and storage, an example is shown below:

The first element of List of 1990 (1st level)
$:List of 6 (2nd level)
..$ sid : chr "010030010"
..$ vars : chr [1:2] "O3" "NOX"
..$ units : chr [1:2] "ppb" "ppb"
..$ datetime : chr [1:744] "2011-07-01 00:00:00 GMT" ...
..$ var.names: chr [1:6] "O3" "O3.obs" "O3.mod" "NOX" ...
..$ cd :List of 4 (3rd level)
.. ..$ pair.datetime: chr [1:744] "2011-06-30 18:00:00"...
.. ..$ O3.obs : num [1:744] 64 58 52 52 48 46 NA 18 20 ...
.. ..$ O3.mod : num [1:744] 68.1 64.7 62.2 59.5 55.2 ...
.. ..$ NOX.mod : num [1:744] 1.76 2.04 2.17 2.41 2.81 ...

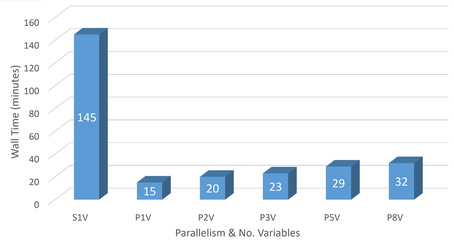
Functions to convert list to data frame (list2df), subset a list with criteria (list.select), merge and combine lists (cdata.merge) are implemented in addition to the functions associated with the "rlist" package.

Common interfaces (polymorphism) to read in model data (WRF, CMAQ, MPAS, etc) and observational data distinguished by network names/formats (AQS, AIRNOW, CASTNET, CSN, ICARTT, IMPROVE,NADP, SEARCH, Sonde, TOLNet, etc.):
getCelldata <- function(site.list, mod.file, mod.vars, in_units=NULL, llay=1, ulay=1, tz="GMT", isParallel=F, cfactor=1, …)
read.OBS <- function(network="AQS", ...)

Memory and Speed

The major drawbacks associated with the base R packages and the computing environment are the limitation of memory and computing speed when dealing with large datasets. For example, the annual AQS hourly dataset can easily exceeds 2 GB that is beyond R's capacity to handle. But with the adoption of SQLDF package, that transparently sets up a database and imports the data frames into that database, performs SQL-like queries to the database using a heuristic method, thus bridges the dataset with the R environment avoiding loading the entire file into memory, R can access any large files efficiently.

The slow processing speed when dealing with large dataset and complex computations is overcome by using the parallel computing packages in R. As shown in the right figure, during the site compare process to process the entire monthly data for 12 km US CMAQ simulation and AQS observations, when the parallel processing is implemented, dramatic speedup is achieved even with multiple variables.

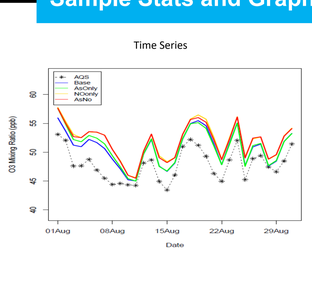


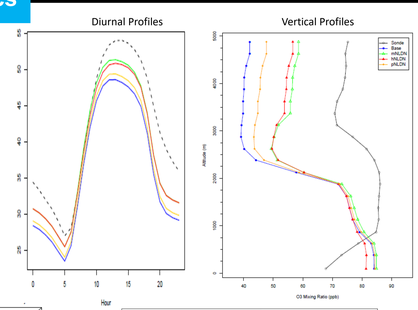
The running wall time using serialized and parallel processing (16 processors) for different number of variables. S1V: single processor with 1 variable, P1V: multiple processors (16) with 1 variable, P2V: multiple processors with 2 variables, etc.

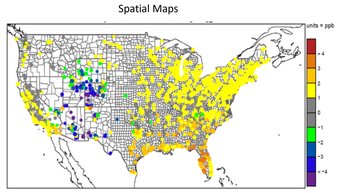
It takes about 15 to 30 minutes to generate the statistics and the generic graphics for one variable with 2-4 model cases. The time to make the vertical profiles varies with model cases and graphic types, but it ranges from a few minutes to about 1 hour.

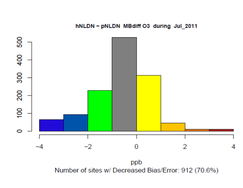
Sample Stats and Graphics

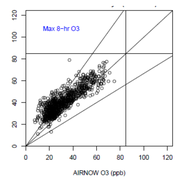
	region	case	type	record	obs	mod	RMSE	NME	MB	NMB	CR
1	all	Bidi_base	hourly	750885	35.054	37.283	12.636	27.284	2.229	6.358	0.770
2	all	Bidi_base	dm8hr	36246	48.202	53.038	12.653	19.338	4.836	10.033	0.703
3	all	Bidi_cec	hourly	750885	35.054	37.214	12.609	27.231	2.160	6.162	0.770
4	all	Bidi_cec	dm8hr	36246	48.202	52.957	12.606	19.265	4.755	9.864	0.704
5	PC	Bidi_base	hourly	112894	32.813	34.190	11.454	26.828	1.377	4.196	0.809
6	PC	Bidi_base	dm8hr	5575	44.718	47.933	11.646	19.990	3.215	7.189	0.801
7	PC	Bidi_cec	hourly	112894	32.813	34.166	11.448	26.808	1.352	4.121	0.809
8	PC	Bidi_cec	dm8hr	5575	44.718	47.904	11.633	19.969	3.186	7.125	0.801
9	RM	Bidi_base	hourly	133070	41.573	38.488	11.775	22.136	−3.085	−7.420	0.702
10	RM	Bidi_base	dm8hr	6259	52.509	50.707	10.188	14.762	−1.802	−3.431	0.559
11	RM	Bidi_cec	hourly	133070	41.573	38.398	11.792	22.181	−3.174	−7.636	0.702
12	RM	Bidi_cec	dm8hr	6259	52.509	50.603	10.197	14.804	−1.906	−3.631	0.559
13	UM	Bidi_base	hourly	171845	36.841	40.277	13.199	26.608	3.436	9.327	0.771
14	UM	Bidi_base	dm8hr	8068	51.615	59.531	14.014	19.558	7.916	15.337	0.628
15	UM	Bidi_cec	hourly	171845	36.841	40.185	13.143	26.490	3.344	9.077	0.772
16	UM	Bidi_cec	dm8hr	8068	51.615	59.422	13.916	19.396	7.808	15.127	0.630
17	NE	Bidi_base	hourly	115132	36.628	38.780	13.499	27.509	2.152	5.875	0.769
18	NE	Bidi_base	dm8hr	5515	50.963	55.173	12.801	17.781	4.211	8.262	0.742

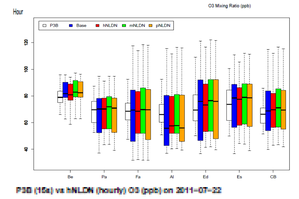


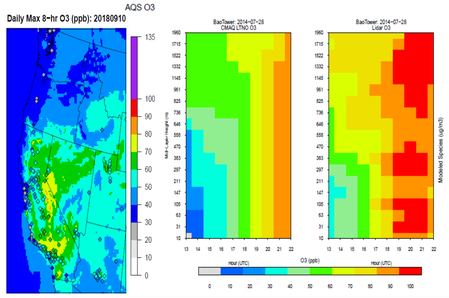


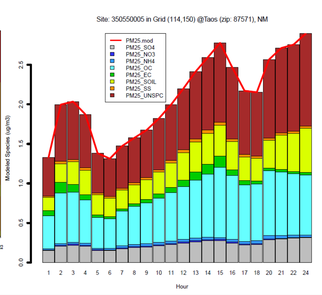


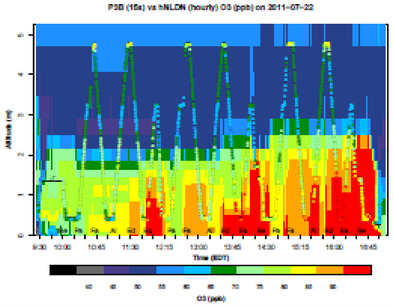












Summary

A Simple Model Assessment R-Tool (SMART) is developed to facilitate model evaluations with the following features:

- Simple: one single programming language (R)
- Portable: can be used on any platforms where R is available.
- Efficient: direct from model outputs and observations to results, additional species are defined during runtime with the same species definition file comes with model chemical mechanism.
- Easy to use: only c-shell scripts are needed to invoke the tool
- Able to deal with large dataset
- Scalable with parallelization
- Suitable for small modeling project with a few sensitivity studies
- Good for daily operational evaluations

U.S. Environmental Protection Agency
Office of Research and Development

Acknowledgements: Thank Christian Hogrefe for his constructive review of this poster and thank Golam Sarwar, Robert Gilliam, Brian Eder, and others for experimenting this preliminary evaluation tool and offering comments and suggestions for improvements.