

# IMPROVEMENTS TO THE PYTHON-BASED PERFORMANCE ANALYSIS SUPPORT SYSTEM (PYPASS): NEW TOOLS TO SUPPORT AIR QUALITY MODEL EVALUATIONS

Leiran Biton, Byeong-Uk Kim, William Vizquete\* and Harvey E. Jeffries

Department of Environmental Sciences & Engineering, University of North Carolina, Chapel Hill, NC, USA

New 8-hour regulations and guidelines for ozone from the Environmental Protection Agency pose challenges to model performance evaluators. This paper introduces improvements to the Python-based Performance Analysis Support System (PYPASS), which can assist in the evaluation process.

## 1. INTRODUCTION

The Python-based Performance Analysis Support System (PYPASS) was recently developed at The University of North Carolina at Chapel Hill as a modeling diagnostic tool to assist in the implementation of the Protocol for Regulatory Ozone Modeling Performance Tests (PROMPT). In contrast to traditional statistical measures of model performance, PROMPT provides regulatory modelers the framework to achieve a more holistic and systematic approach to assess model reliability. At the inception of PROMPT, however, model diagnostic tools did not possess the sophistication needed to implement its guidelines. This need led to the creation of the PYPASS diagnostic tool, which has significant computational and interfacial advantages over traditional support tools, such as Package for Analysis and Visualization for Environmental (PAVE) Data.

Since its introduction last year, notable improvements have been incorporated into the PYPASS system. These changes enhance the flexibility of PYPASS to evaluate a myriad of modeling episodes and to accommodate broader user specifications. Specific modifications include: expanded mathematical manipulation of species concentrations, aircraft VOC visualization, 8-hour ozone metrics, expanded monitoring data visualizations (including design value tracking) and gathering capabilities, improved labeling schemes, integration with Python 2.4, and porting to Mac OS X platforms. New PYPASS functionalities allow for

more detailed, precise, and scientific model evaluations. These improvements in conjunction with the guidance provided by PROMPT, enables a more comprehensive model performance evaluation for photochemical air quality models.

## 2. PYPASS OVERVIEW

PYPASS was developed to aid in the analysis and evaluation of model simulations, as required by the Clean Air Act Amendments of 1990 (CAAA). The system was designed to meet the following design constraints: it must (1) generate information rapidly; (2) provide information not readily available through the use of existing tools, such as PAVE; (3) be publicly available and cheap or free to use; and (4) be flexible and extensible to accommodate future changes to data formats and analytical needs.

To accomplish these goals, the system was implemented to: (1) take advantage of command-line batch operations, which optimize efficiency for large-scale data processing and information generation; (2) create pre-defined graphics as defined by the user; (3) rely on Python (Python Software Foundation 2006), a free-source script-based object-oriented programming language, with supporting scientific computing and visualization libraries.

Since its unveiling (Kim and Jeffries 2005), PYPASS has undergone significant improvement and refactoring. Most changes have reflected specific needs for analysis or library updates, but the system has also undergone streamlining and optimization.

### 2.1 Library and Platform Upgrades

To take advantage of several supporting library upgrades, and an upgrade for Python itself, we have performed significant refactoring of code within PYPASS. Additionally, PYPASS has been ported to Mac OS X. [Though it has not been tested on other systems to date, PYPASS should in theory operate without problems on other UNIX-based systems.] See Table 1 for the current list of supporting libraries and platforms. Table 1 also

---

\*Corresponding author: William Vizquete, Department of Environmental Sciences and Engineering, UNC-Chapel Hill, 113 Rosenau Hall, CB#7431, Chapel Hill, NC 27599-7431; e-mail: [vizquete@unc.edu](mailto:vizquete@unc.edu)

includes applications that are supported by the current PyPASS system.

Table 1: Applications and libraries required for PyPASS.

<b>Application/Library (Version), Source</b>
CAMxSubset/CMAQExtract (8.0), <a href="http://ftpozone.sph.unc.edu">http://ftpozone.sph.unc.edu</a>
Python (2.4.3), <a href="http://www.python.org/">http://www.python.org/</a>
ChartDirector (4.0), <a href="http://www.advsofteng.com/">http://www.advsofteng.com/</a>
NetCDF (3.6.1), <a href="http://www.unidata.ucar.edu/software/netcdf/">http://www.unidata.ucar.edu/software/netcdf/</a>
gdal (1.3.2), <a href="http://www.gdal.org/index.html">http://www.gdal.org/index.html</a>
HDF5 (1.6.5), <a href="http://hdf.ncsa.uiuc.edu/HDF5/">http://hdf.ncsa.uiuc.edu/HDF5/</a>
mxDateTime (2.0.6), <a href="http://www.egenix.com/files/python/mxDateTime.html">http://www.egenix.com/files/python/mxDateTime.html</a>
numarray (1.5.1), <a href="http://www.stsci.edu/resources/software_hardware/numarray">http://www.stsci.edu/resources/software_hardware/numarray</a>
PROJ.4 (4.4.9), <a href="http://proj.maptools.org/">http://proj.maptools.org/</a>
pyarsing (1.4.3), <a href="http://pyarsing.wikispaces.com/">http://pyarsing.wikispaces.com/</a>
PyRXP (1.09), <a href="http://www.reportlab.org/">http://www.reportlab.org/</a>
PyTables (1.3.2), <a href="http://pytables.sourceforge.net/">http://pytables.sourceforge.net/</a>
ReportLab (2.0), <a href="http://www.reportlab.org/">http://www.reportlab.org/</a>

The biggest functional changes to PyPASS have been the port to Mac OS X, as new runscripts have been developed for the UNIX environment, and the upgrade to Python 2.4, as the code has been updated to exclude deprecated functions.

## 2.2 Upgraded Capabilities

The newly expanded visualization scheme available in PyPASS includes mathematical manipulation, as is typically used for model intercomparison in PAVE. We have also enabled functions appropriate for use in model episode selection, such as observation-only plotting and design value tracking. In addition, volatile organic compound (VOC) speciation schemes can be fed to a new subroutine to compare monitored versus model predicted VOC concentrations. These new utilities offer opportunities to evaluate the episode itself, not just the model. Formaldehyde has been integrated into the existing aircraft plots, and there are plans to make these plots even more extensible. The expanded mathematical capability

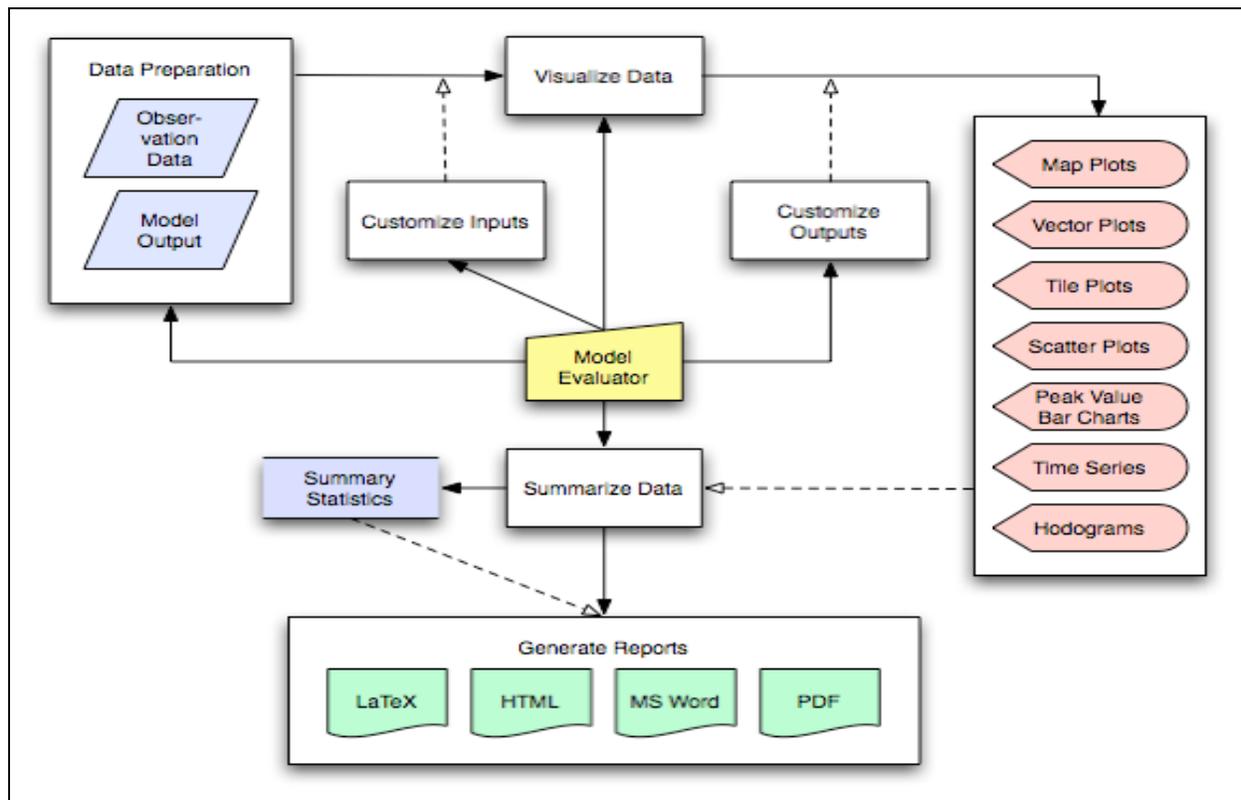


Figure 1: PyPASS Flow-Chart. The Model Evaluator (center) represents human input and decision making, closed rectangles represent user-initiated processes, pink shapes represent PyPASS generated plots, green shapes represent documents, and blue shapes represent data inputs (rhombus) and outputs.

proves most useful in use with tile plots. For example, model sensitivity run output can be differenced to show concentration variation resulting from changes to inputs. Sample charts will be shown in the Illustrative Examples section, below.

### 2.3 Model Evaluation with PyPASS

The PyPASS tool is designed to assist in model evaluations, as detailed in the PROMPT meta-protocol (Kim and Jeffries). PROMPT requires detailed analysis using a holistic site-by-site, day-by-day approach offered by PyPASS. Used in conjunction with other analytical tools, such as process analysis, PyPASS can help models perform systematic model evaluation, as suggested by the EPA 8-hour modeling guidelines (EPA 2005).

### 2.4 Future Expansion

Planned development for PyPASS includes emission visualization tools, such as those already available in PAVE, and comprehensive tile plots (with statistical analysis).

## 3. ILLUSTRATIVE EXAMPLES

For the analysis of modeling episodes spanning tens of days, PyPASS can generate hundreds of plots relatively quickly (depending on processor speed). This section highlights components of a hypothetical model evaluation. Operating PyPASS, in essence, follows the framework detailed in Figure 1. Users prepare data, customize input and output, and assemble plots and associated data into automatically

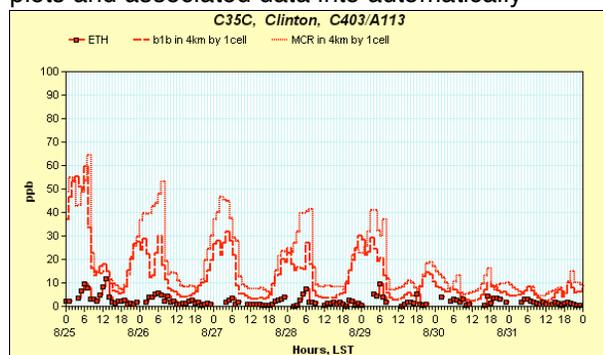


Figure 2: Model time-series comparison with observations for ethylene. Two simulations (dashed and dotted lines) are vastly overpredict from what is monitored (discrete points).

generated reports.

### 3.1 Data Preparation

New functionality in PyPASS enabled the conversion of explicit automatic gas-chromatograph (autoGC) observed to model output Carbon Bond IV (CBIV) species according to William Carter's speciation profile database (2006) Modeled CBIV concentrations are plotted against speciated monitoring site information in existing time series and scatter plots (Figure 2).

### 3.2 Data Visualization

After additional data is prepared, further data visualization can be performed. The difference in ambient concentrations between two simulations can be plotted on tile plots. Figure 3 shows two tile plots of ozone concentrations from different simulations (compared to observations marked by diamonds), and the difference below them.

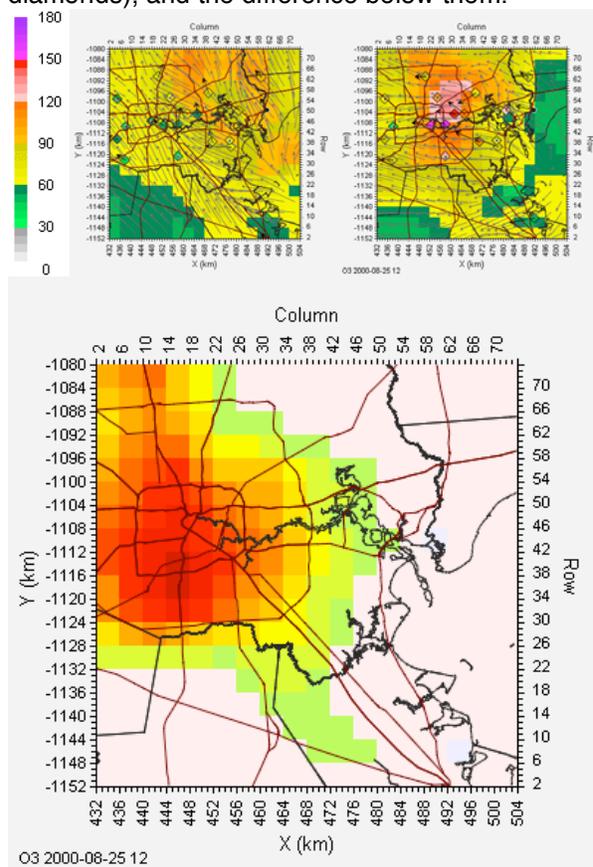


Figure 3: Differenced tile plots in sensitivity simulations for ozone. Small plots are original (left) and sensitivity (right) simulation concentrations for ozone, while large plot is the difference of concentrations from these simulations.

Aircraft plots have been extremely valuable to model evaluators, and *PyPASS* now supports graphical display of formaldehyde (or any other non-hourly sampled species) in aircraft plots. The plot in Figure 4 shows moderate model agreement with ozone and formaldehyde (with some significant underprediction), but model concentrations of carbon monoxide of twice that observed.

Furthermore, new eight-hour regulations for ozone require new visualization techniques, as implemented in *PyPASS*. These new visualizations, which incorporate components of traditional one-hour evaluations, assist evaluators in determining exceedance levels and average model performance over the photochemical day. Figure 5 illustrates one such use of the 8-hour metrics, in simply determining which monitors are out of attainment.

Another feature of the 8-hour regulations is the design-value for monitors. *PyPASS* integrates an innovative visualization which quantifies not only the 8-hour average input values that go into the design value for a year, but also the peak 1-hour average value during that span and the resultant 8 hour wind vector during that period. As shown in Figure 6, this information provides a fuller picture for evaluators than simply looking at the average value.

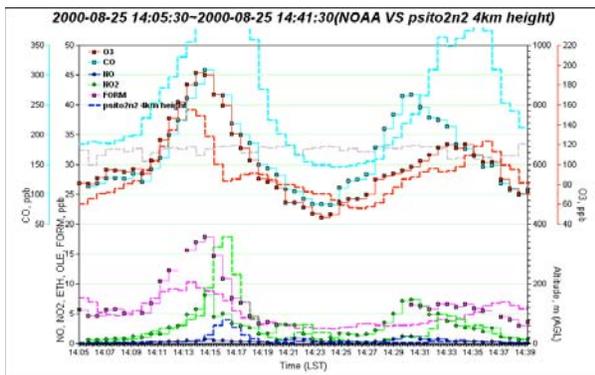


Figure 4: Aircraft plot including formaldehyde data.

#### 4. DISCUSSION

Model evaluators face significant challenges, especially with the dissemination of 8-hour ozone modeling guidelines. *PyPASS* is a tool that can help in the model performance evaluation process, and while it is flexible enough to have utility across a variety of modeling scenarios, the outputs are

designed carefully to maximize information provided in plots.

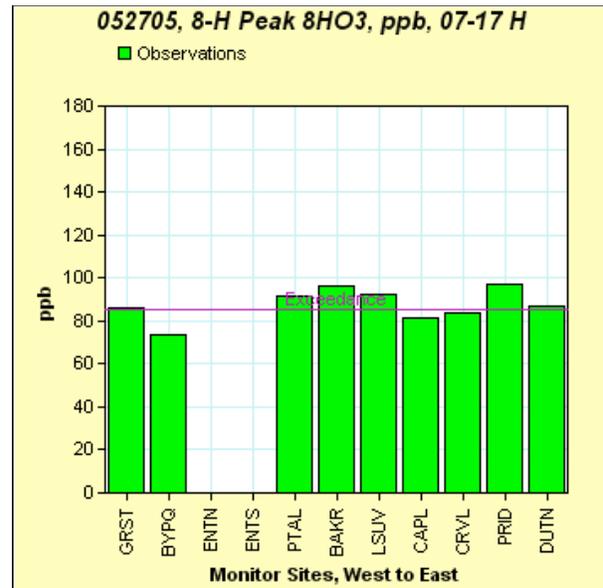


Figure 5: Observation-only peak-bar plot for eight-hour averaged ozone.

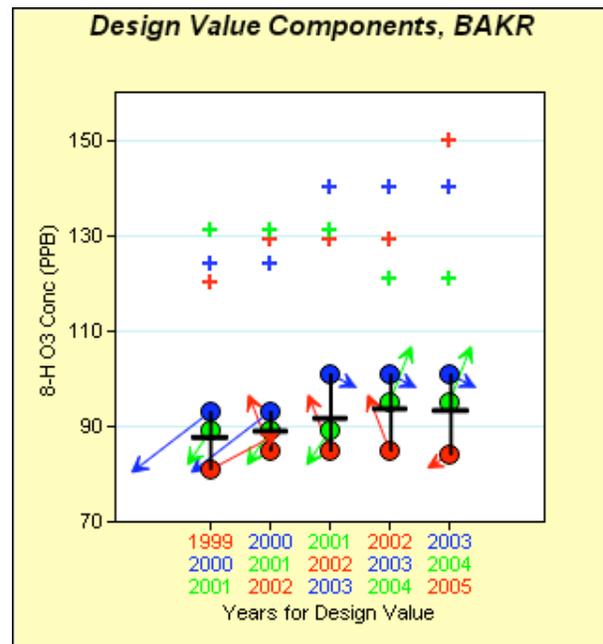


Figure 6: Design value plot for 5-year period. 8-hour wind vectors are represented by arrows stemming from circles, which represent the component 8-hour average value. 1-hour peak values are crosses and correspond with 8-hour average.

## 5. REFERENCES

- Advanced Software Engineering Limited, 2006:  
ChartDirector Version 4.0,  
<http://www.advsofteng.com/>
- CAMxSubset/CMAQExtract 8.0,  
<http://ftpozzone.sph.unc.edu>
- Carter, W.P.L., 2006: Development of an Improved  
Chemical Speciation Database for Processing  
Emissions of Volatile Organic Compounds for  
Air Quality Models,  
<http://pah.cert.ucr.edu/~carter/emitdb/>
- Cartographic Projections Library, 2006: PROJ.4  
Version 4.4.9, <http://proj.maptools.org/>
- eGenix, 2006: eGenix mx.DateTime Version 2.0.6  
<http://www.egenix.com/files/python/>
- EPA 8-hour modeling guidelines, 40 CFR 81.
- Geospatial Data Abstraction Library, 2006: GDAL  
Version 1.3.2, <http://www.gdal.org/index.html>
- Hierarchical Data Format 5, 2006: HDF5 Version  
1.6.5, <http://hdf.ncsa.uiuc.edu/HDF5/>
- Kim, B.-U. and H.E. Jeffries, 2005: Python-based  
Performance Analysis Supporting System  
(PyPASS): A software tool set for the  
performance analysis of regulatory  
photochemical air quality modeling, Extended  
Abstracts, *4<sup>th</sup> Annual CMAS Models-3 Users'  
Conference*, Chapel Hill, NC, CMAS Center.
- Network Common Data Format, 2006: NetCDF  
Version 3.6.1,  
<http://www.unidata.ucar.edu/software/netcdf/>
- Python Software Foundation, 2006, Python  
Version 2.4, <http://www.python.org/>
- PyRXP (1.09), <http://www.reportlab.org/>
- pyparsing Version, 1.4.3,  
<http://pyparsing.wikispaces.com/>
- PyTables (1.3.2), <http://pytables.sourceforge.net/>
- ReportLab (2.0), <http://www.reportlab.org/>
- Space Telescope Science Institute, 2006:  
numarray Version 1.5.1,  
[http://www.stsci.edu/resources/software\\_hardware/numarray](http://www.stsci.edu/resources/software_hardware/numarray)