Editors: Cláudio T. Silva, csilva@nyu.edu
Daniel Weiskopf, weiskopf@visus.uni-stuttgart.de

UV-CDAT: Analyzing Climate Datasets from a User's Perspective

By Emanuele Santos, Jorge Poco, Yaxing Wei, Shishi Liu, Bob Cook, Dean N. Williams, and Cláudio T. Silva

The Ultra-scale Visualization Climate Data Analysis Tools (UV-CDAT) is a new tool for analyzing and visualizing climate data. Here we provide some pointers, background information, and examples to show how the system works.

Climate scientists have made substantial progress in understanding the Earth's climate system, particularly at global and continental scales. Climate research is now focused on understanding climate changes over wider ranges of time and finer-space scales (for example, 2- to 100-km resolution), which generates ultra-scale datasets. At such scales, a single snapshot of data will result in a terabyte or more of data, and modest time scales will result in petabytes of data.

An insightful analysis in climate science depends on using software tools to discover, access, manipulate, and visualize the datasets of interest. These data exploration tasks can be complex and time-consuming, and they frequently involve many resources from both the modeling and observational climate communities. Because of the complexity of the explorations, number of tools, and amount of data involved, provenance is critical. Provenance allows scientists to ensure reproducibility, revisit existing computational pipelines, and more easily share analyses and results. Also, because the results of this work can impact policy, having provenance available is important in demonstrating the scientific basis for decision making.

With this in mind, Ultra-scale Visualization Climate Data Analysis Tools (UV-CDAT) was developed to address the needs of scientists for access to, analysis of, and visualizations of computer model output resulting from high-resolution, long-term, climate change projections performed as part of the US Department of Energy's Office of Biological and Environmental Research (BER) Program. This program funds a multiagency effort toward modeling and simulating long-term climate change, and for the past several years, this effort has been an extremely important resource for the research community. To date, BER has been a critical contributor to climate science research on a national and international scale—helping to support the Community Earth System Model development and contributing to clouds and aerosols impact studies. Data supported through BER's work are used by thousands of scientists worldwide.

Impact of UV-CDAT

As an example of the research progress enabled under this effort, the DOE BER-funded Program for Climate Model Diagnosis and Intercomparison (PCMDI) has collected and disseminated a Model Intercomparison Project (MIP) simulation output from most of the world's premier climate modeling centers. The Coupled Model Intercomparison Project, phases 3 and 5 (CMIP3 and CMIP5, respectively), were performed in support of the Intergovernmental Panel on Climate Change's (IPCC's) Fourth and Fifth Assessment Reports (AR4 and AR5, respectively). Currently, the two federated archives hold more than 2 petabytes of data, and more than 2 petabytes of data have been distributed to thousands of researchers worldwide, resulting in more than 1,000 peer-reviewed climate publications.

Recent advances in climate models have included the incorporation of explicit land biosphere components to simulate feedbacks associated with CO_2. A preliminary assessment of these models was conducted during the North American Carbon Programs Regional Synthesis, whose goals were to compare observations, biosphere models, and inverse models.[1,2] A companion to the CMIP Model Intercomparison Project—called Multi-Scale Synthesis and Terrestrial Biosphere Model Intercomparison Project (MsTMIP; see http://nacp.ornl.gov/MsTMIP.shtml) has engaged leading terrestrial biosphere models in an intercomparison and benchmarking project. The overall goal of the MsTMIP, funded by NASA, is to provide feedback to the terrestrial biospheric modeling community to improve the diagnosis and attribution of carbon sources and sinks across regional and global scales. The approach taken by MsTMIP is to facilitate comparison with model results and observations through an integrated evaluation framework.

 Copublished by the IEEE CS and the AIP 1521-9615/13/$31.00 © 2013 IEEE

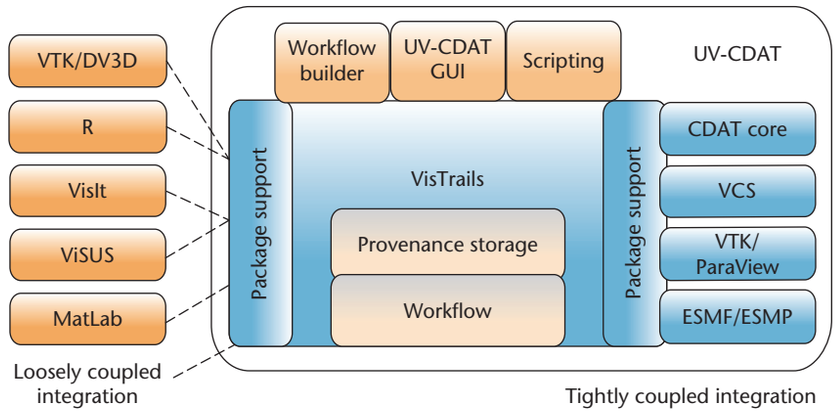

Figure 1. Ultra-scale Visualization Climate Data Analysis Tools (UV-CDAT) system architecture. (VTK = Visualization Toolkit, DV3D = Data Viewer 3D, ESMF = Earth System Modeling Framework, and ESMP = Environmental and Social Management Plans.)

UV-CDAT Overview

UV-CDAT is a workflow-based, provenance-enabled system that integrates climate data analysis libraries and visualization tools in an end-to-end application. UV-CDAT lets users build complex data analysis and visualization workflows that use predefined components for data transformation and analysis, data collection from disparate data sources outside the Earth System Grid Federation (ESGF), and visualization as well as user-defined processing steps.

The design of UV-CDAT has considered the following desiderata:

- workflow analysis and provenance management;
- parallel visualization and analysis tools (exploiting parallel I/O);
- local and distance visualization and data access;
- comparative visualization and statistical analyses;
- robust tools for regridding, reprojection, and aggregation; and
- support for unstructured grids and nongridded observational data, including geospatial formats often used for observational datasets.

To achieve this design, UV-CDAT uses several components, including CDAT (used in the climate community since 1995), VisTrails, DV3D, ParaView, and the Visualization Toolkit (VTK) to provide high-performance parallel-streaming data analysis and visualization of massive climate datasets (see Figure 1). In addition, UV-CDAT is a highly flexible and extensible system. Using VisTrails' package mechanism, it's straightforward to integrate other tools, including VisIt, R, Visualization Streams for Ultimate Security (ViSUS), and Matlab for data analysis and visualization without modifications to the system core. This package mechanism enables developers to expose their libraries (written in any language) to the system using a thin Python interface through a set of VisTrails modules.[3] Users are able to interact with the system in different ways, by using the UV-CDAT GUI, the VisTrails' workflow builder, or through Python scripts.

We believe that providing these different ways of interaction, including a simple-but-powerful GUI, reduces the barriers for adoption. In addition, relatively simple interfaces are needed for other target audiences (for example, adaptation and mitigation researchers and decision makers).

Figure 2 shows the main window that acts as the UV-CDAT GUI. It's based around the notion of a *spreadsheet* (middle), a resizable grid where each cell contains a visualization. Using intuitive drag-and-drop operations, visualizations can be created, modified, copied, rearranged, and compared. Spreadsheets maintain their provenance and can be saved and reloaded. These visualizations can be used for data exploration and decision making, while at the same time being completely customizable and reproducible.

Around the spreadsheet are the tools for building visualizations. The project view (top left) lets you group spreadsheets into projects, and name visualizations and spreadsheets. The plot view (bottom left) lets you use and customize your available plot types. The variable view (top right) lets you use and edit data variables. The bottom right contains a variable editor widget, where you can modify a variable similar to how you would use a pocket calculator. Using the UV-CDAT GUI, creating a visualization can be as simple as dragging a variable to a spreadsheet cell, then dragging a plot type to the same cell.

Visualizations, Workflows, and Provenance

UV-CDAT was designed to provide visualization techniques as part of the data exploration process.

Data exploration through visualization requires scientists to go through several steps. To successfully analyze and validate various hypotheses, we must pose several queries, correlate disparate data, and often use different tools to create insightful visualizations of both the simulated processes and observed phenomena. As Figure 3 shows, a scientist must assemble and execute complex workflows that consist of dataset selection, a specification of a series of operations that must be applied to the data, and the creation of appropriate visual representations, before finally viewing and analyzing the results. Often, insight comes from comparing the results of multiple visualizations created during the exploration process, for example, by applying

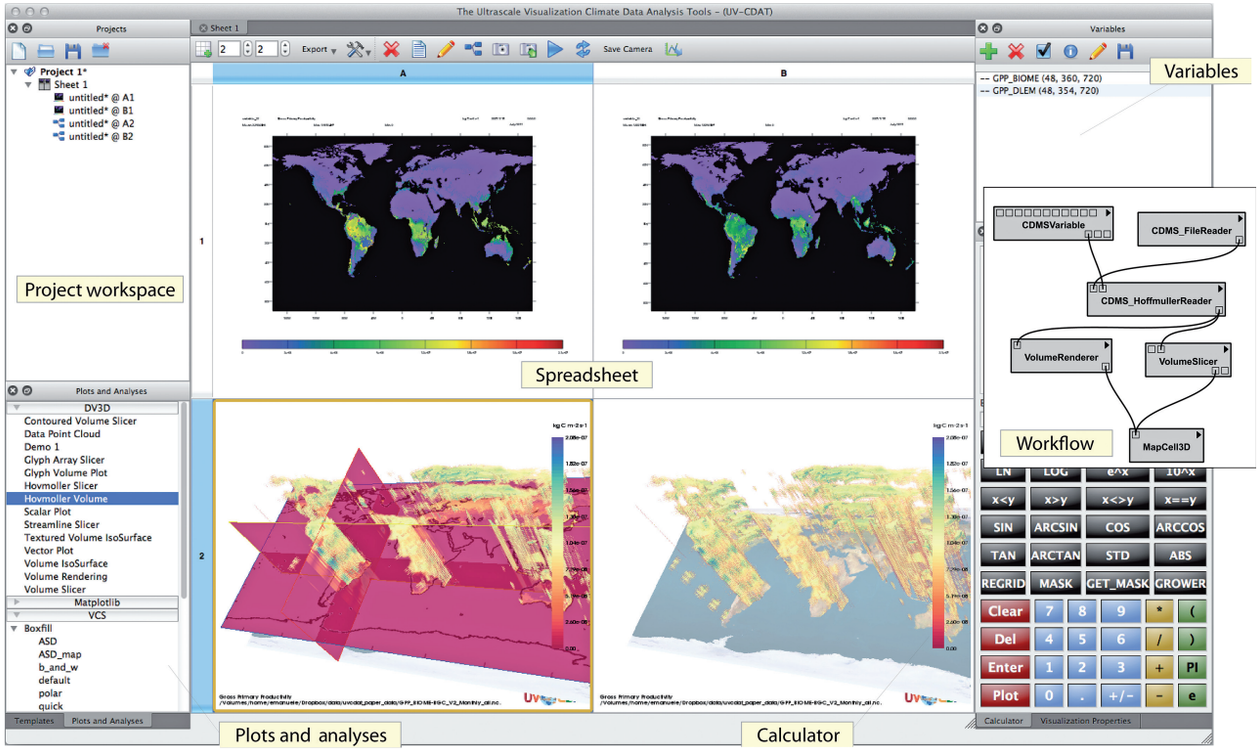

Figure 2. UV-CDAT's main window, which acts as the interface. Spreadsheet (middle), project view (top left), plot view (bottom left), variable view (top right), and calculator (bottom right).

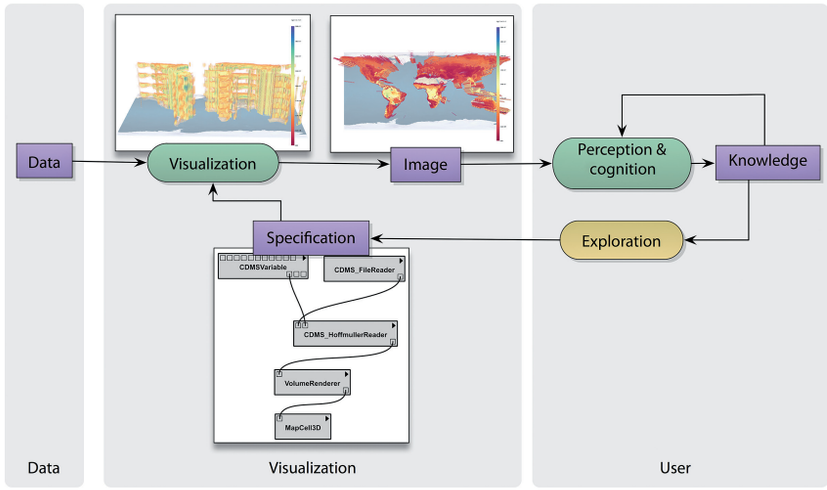

Figure 3. The visualization discovery process. Users first select the data to visualize, and then specify the algorithms and visualization techniques to visualize the data. This specification is adjusted in an iterative process, as the users generate, explore, and evaluate hypotheses about the information under study. This figure is adapted from other work.[4]

a given visualization process to multiple datasets generated in different simulations, by varying the values of certain visualization parameters, or by applying different variations of a given process (for example, using different visualization algorithms) to a dataset.

UV-CDAT simplifies the data exploration process by using VisTrails' comprehensive provenance management infrastructure that can be easily combined with existing tools and libraries. The provenance (also referred to as the audit trail, lineage, or pedigree) of a data product contains information about the process and data used to derive the data product.[5]

Similar to VisTrails and other visualization systems, UV-CDAT is based on the notion of workflows. A workflow is a graph where nodes represent computations and edges represent data streams; each node or module corresponds to a procedure that's applied on the input data and generates some output as a result. The flow of data in the graph determines the order in which the processing nodes are executed. UV-CDAT uses workflows (or pipelines) to uniformly capture changes to both parameter values and workflow definitions by unobtrusively tracking all changes that users make to workflows in an exploration task. This enables a series of operations that simplify exploratory processes and foster

reflective reasoning. For example, scientists can easily navigate through the space of workflows created for a given exploration task, visually compare workflows and their results, and explore large parameter spaces.

UV-CDAT facilitates the data exploration process even further by recording and managing workflow provenance under an intuitive GUI. We'll concentrate on three different aspects of this process: loading data, plot and data manipulation, and workflow generation and scripting.

Datasets

In all of our examples, we used two datasets. The first one is provided by MsTMIP. These data consist of multiple variables, but for our purposes we're using only two variables: gross primary production (GPP) and net primary production (NPP). Both variables have a monthly temporal resolution from the years 1901–2010 and 0.5 degrees of spatial resolution, but we're using only four years. Thus, the dataset size is 360 × 720 × 48 (latitude × longitude × time). The second dataset is provided by the North American Carbon Program (NACP).[1,2] Here, we're using the NPP variable for North America with one degree of spatial resolution and a monthly temporal resolution from 2000 to 2005. The final data size is 74 × 120 × 72 (latitude × longitude × time).

More details about the MsTMIP dataset can be found at http://nacp.ornl.gov/MsTMIP.shtml. For the NACP dataset, a complete set of variables and description can be found at ftp://nacp.ornl.gov/synthesis/2008/firenze/continental/1_continental_data_model_inventory.html. Additionally, the pipelines and the data involved for each figure in this article are available at http://bigdata.poly.edu/projects/uvcdat/cise2012data.

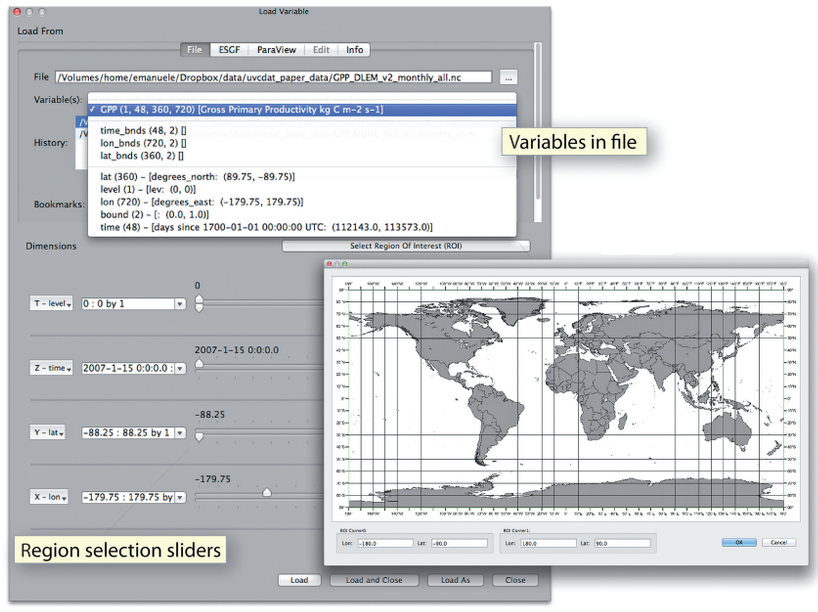

Figure 4. The Load Variable tool. (a) The main panel contains list of variables and controllers to manipulate dataset dimensions. (b) The region-of-interest (ROI) selection tool lets users select an ROI by directly interacting with a world map.

Loading Data

The process of loading data is one of the most important steps in climate data analysis. Although there are predefined structures to store the model output (for example, storing each time step in a separate file or storing all time steps for a single variable in a single file), usually these rules aren't followed consistently by all modeling teams. Consequently, we need an intuitive tool in the loading stage. UV-CDAT has many intuitive features. When first starting UV-CDAT, the user will see the UV-CDAT main window (see Figure 2). To load a variable, the user clicks on the Plus button in the toolbar of the variable view. This will launch another window where users can select files (UV-CDAT supports many different file formats), easily inspect the list of variables stored in the file (see Figure 4a), and if a variable is selected, then all the metadata related to that variable is shown to the user.

Once the variable is selected, the next step is to decide if we want to work with the complete spatial–temporal extension, or if we need to select a subregion. UV-CDAT provides two mechanisms to extract subregions. First, we can use the sliders shown in Figure 4a to specify the range for each axis (usually latitude, longitude, altitude, and time). The second mechanism is to define a region of interest (ROI) by selecting a rectangular region directly on a world map (Figure 4b), and UV-CDAT automatically will determine which coordinates correspond to this region.

After defining the variable, the user clicks on the Load button (or the Load As button, if the variable should be loaded using a different name), and the variable is ready to be used. In our example, we load the variable *GPP* from the `GPP_BIOME-BGC_V2_Monthly_all.nc` file, loading the entire extent of the variable and renaming it as *GPP_BIOME* by clicking on the Load As button and typing in the name.

Visualizing Data

Now that we've loaded the data, let's look at some of the tools available in UV-CDAT to visualize the data.

Basic plot and data manipulation

UV-CDAT already provides different types of visualization tools (and more

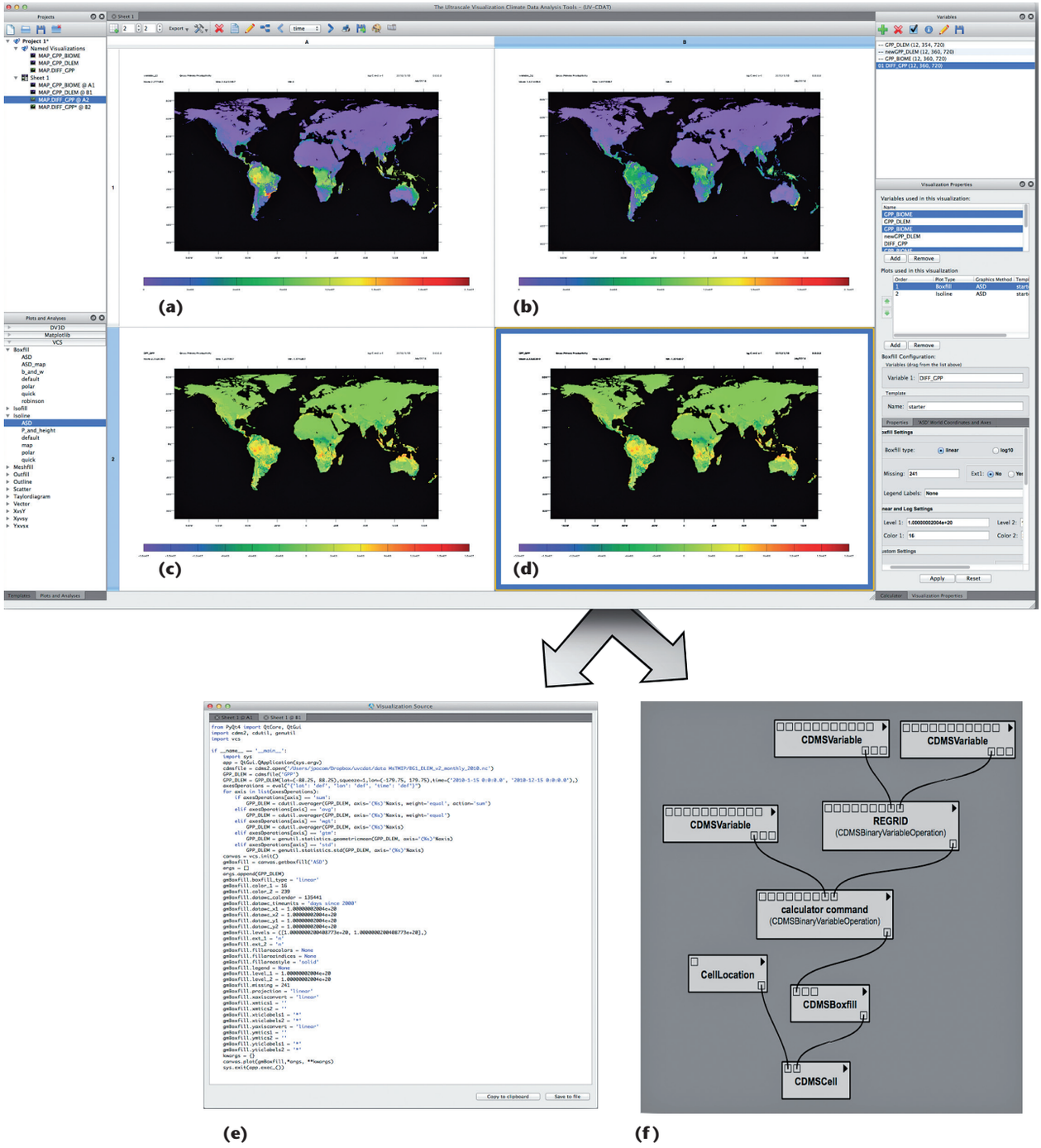

Figure 5. Visualizations of the Gross Primary Production (*GPP*) variable, from (a) BIOME-BGC and (b) DLEM models. (c) Plot of the difference between the *GPP* variable from both models converted to the same grid. (d) Same plot as in (c), using an overlay of contour lines. (e) Python script for the plot in (d). (f) The workflow automatically generated for the plot in (d).

can be easily incorporated). For example, Figure 5 displays a set of visualizations of the *GPP* variable from the BioGeochemical Cycles model (BIOME-BGC) and the Dynamic Land Ecosystem Model (DLEM), respectively. To create them, we just need to drag the variable name *GPP_BIOME* from the variable panel into one cell of the spreadsheet. Then on the plot panel (bottom left), go to the version control system (VCS), expand the Boxfill plot type, and drag the ASD preset type into the same cell where the variable was dragged.

Data from different models usually have different resolutions and/or spatial locations. To compare data from different models, one of the most common tasks performed by climate scientists is *regridding*. UV-CDAT integrates several regridding algorithms

from well-known libraries such as the Earth System Modeling Framework (ESMF), Climate and Forecast Library (LibCF), and Regrid2. Regridding our variables is simple: First, load the two variables using the procedure previously described in the "Loading Data" section. For this example, we'll regrid two GPP variables from different models. We already loaded the variable *GPP* from `GPP_BIOME-BGC_V2_Monthly_all.nc` and renamed it *GPP_BIOME*, and we'll also load variable *GPP* from `GPP_DLEM_v2_monthly_all.nc` and rename it *GPP_DLEM*. After loading them, we just need to select the variable we want to regrid (*GPP_DLEM*), then the variable with the desired grid (*GPP_BIOME*), and finally click the "regrid" button from the calculator. This will create another variable called *regrid_GPP_DLEM_GPP_BIOME* corresponding to the regridded variable; for simplicity, we're going to rename this variable *regridded_GPP_DLEM*. To do that, we select the variable and click on the "Edit" button (indicated by a pencil on the variable panel toolbar). Other regridding algorithms are available in the menu `PCMDITools->Regridding`.

Once we obtain our two GPP variables from different models in the same grid, we can then manipulate our data using the calculator in the bottom-right panel (see Figure 2). The calculator provides common operations that we can apply directly to our variables. In addition, if we need to apply a more complex operation that isn't available in the calculator, we can use the command line above the calculator to type our command, using Python syntax.

For example, to see the difference between the GPP variables from both models, we select the two variables from the variable panel (we first select *regridded_GPP_DLEM* and then *GPP_BIOME* to compute "`regridded_GPP_DLEM - GPP_BIOME`") and click the "-" button on the calculator. The newly created variable *sub_regridded_GPP_DLEM_GPP_BIOME* can be visualized in a new cell using any plot. Figure 5c shows a plot of *sub_regridded_GPP_DLEM_GPP_BIOME* using the Isofill-ASD type from the VCS panel.

Each cell in the spreadsheet can display a visualization type from any of the packages available in UV-CDAT, so you can combine 2D and 3D plots in the same sheet. In addition, each visualization can be edited using the properties panel, which is customized for each plot type. Using the properties panel, we can change the color map, line width, font size, and so on.

UV-CDAT also supports overlaying *multiple plots*. For instance, we can visualize the difference computed before using the isofill and isolines plots. To add the isoline plot, we just drag Isoline-ASD from the plot panel into the cell displaying the isofill plot. To change the order of the plots or to adjust other properties, we can also use the properties panel.

Extended Plotting Capabilities

A key package available in UV-CDAT is DV3D (see http://portal.nccs.nasa.gov/DV3D), developed by Thomas Maxwell from NASA. This package provides advanced visualization and analysis of climate data, such as orthogonal slicers, volume rendering, isosurfaces, and so on, through simple GUI elements. The application builds on VTK, an open source, object-oriented library for visualization and analysis. DV3D provides the high-level interfaces, tools, and application integration required to make the analysis and visualization power of VTK readily accessible to users without exposing details such as actors, cameras, renderers, and transfer functions. We can use the same drag-and-drop interactions previously described to create new visualizations using DV3D.

Interactivity is important in the exploration process. For example, if we want to see the time series in our *GPP* variable, we can use the plots provided by the DV3D package. Figure 6a shows the orthogonal slicers (latitude, longitude, and altitude) of our data. Because our data don't include altitude, it was omitted in the visualization, but it shows an interesting tool to visualize the time series. When the user clicks on any position on the map, a time-series plot appears, which is updated when the user moves the mouse over different coordinates on the map. If we find something interesting on a specific coordinate, we can export that time series to another cell and save it for later analysis. Additionally, if we create another orthogonal slicer for the *GPP* variable from another model, and we select both cells, then the time-series plot will appear on both cells and we can compare them.

Another set of tools provided by the DV3D package includes the Hovmoller slicer and Hovmoller volume. These two types of plots use time as the third dimension in the volume. This way we're able to see the time series in the vertical slicers or play with the transfer function in the Hovmoller volume. Figures 6b and 6c show these two plots using the *GPP* variable from the BIOME-BGC model.

Pipeline Generation and Scripting

Each plot created in UV-CDAT and displayed on the spreadsheet has a corresponding workflow and its provenance. This information is stored through the VisTrails software development kit.

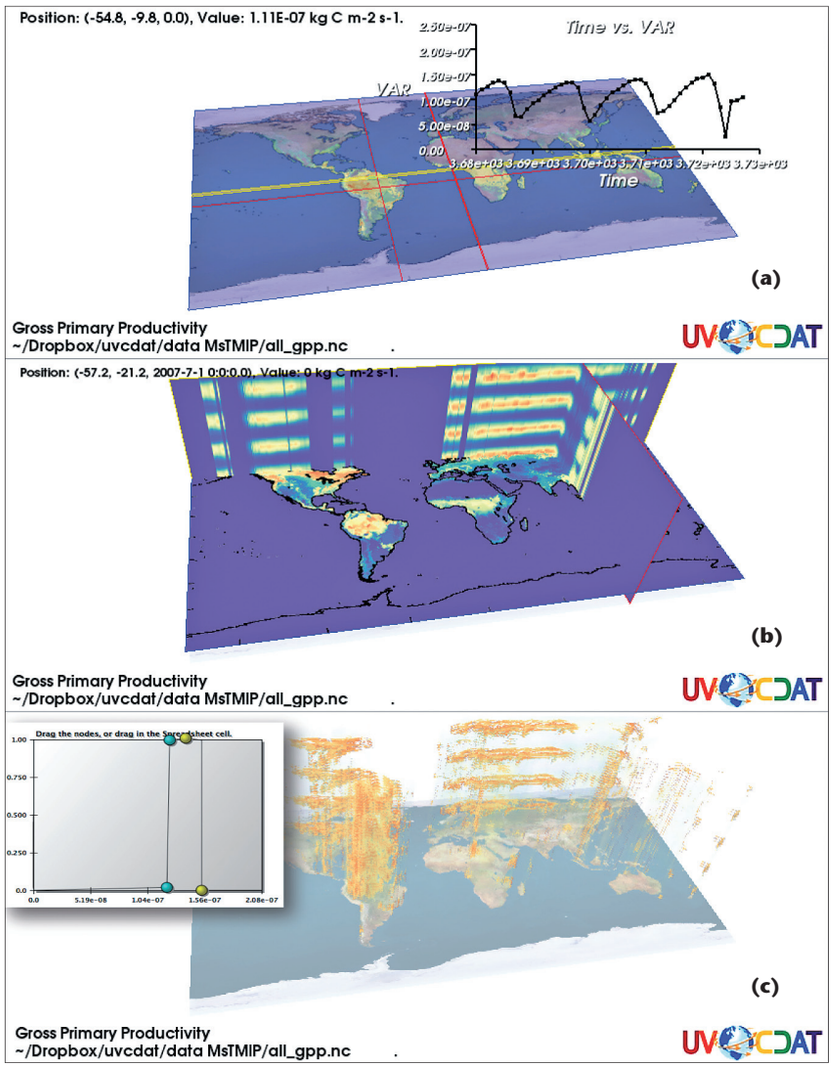

Figure 6. Three different ways to see time series. (a) Using an interactive plot to visualize the time series for a specific spatial location, (b) using orthogonal planes, and (c) using a volume-rendering technique to select a portion of the GPP output.

To access the workflow of a selected plot, we click the spreadsheet toolbar button that resembles a workflow. Figure 5f shows the workflow for the plot in Figure 5d. This pipeline can be divided in two subpipelines. In the first group, we have all the components related to data manipulation—for example, read variables *GPP_BIOME* and *GPP_DLEM*, regrid *GPP_BIOME* into *regridded_GPP_DLEM*, and compute the difference between *GPP_BIOME* and *regridded_GPP_DLEM*. The second subpipeline corresponds to the visualization components such as cell location and plot type.

Another useful feature in UV-CDAT is the ability to generate Python scripts. These scripts can be executed in batch mode using the VisTrails functionality to run a script in the background. Depending on the plot types used to generate a figure, the scripts can be coarse or fine detailed. For example, VCS plots generate a very detailed script for each plot. On the other hand, DV3D generates a simple script that executes the workflow that created that plot. VisTrails provides an API that exports a workflow as a python function, exposing the input parameters of the workflow as input parameters for the function. Figure 5e shows an example of a detailed script using the VCS package.

Direct Workflow Editing

Now that we have a grasp on some of the features offered, let's look at how we can manipulate and reuse the pipelines generated by UV-CDAT. This text was written to be mostly self-contained, but given the nature of the material that uses VisTrails workflows, we also advise the reader to check out the VisTrails user's guide (see www.vistrails.org/usersguide/dev/html/creating.html and www.vistrails.org/usersguide/dev/html/controlflowdev.html).

We have some examples about this functionality, using conceptual workflows created by a climate scientist. For example, let's consider a monthly long-term mean series plot and Taylor diagrams for a set of models.[6] Figure 7 shows one conceptual workflow to process the *GPP* variable of both model outputs and benchmark data and to display the monthly long-term-mean plots.

Monthly spatial GPP data is first subset with a reference ecoregion mask map to retrieve data in the North American (NA) Boreal ecoregion. Then a spatial aggregation is made to calculate the monthly total GPP in the NA Boreal ecoregion. After that, a time-average process is made to calculate the monthly long-term-mean total GPP in the NA Boreal ecoregion. Eventually, each model/benchmark dataset has 12 values, with each value representing the total GPP in the NA Boreal ecoregion for each month. Finally, these values are visualized using time-series plots.

Now we'd like to implement the conceptual workflow explained before using UV-CDAT. As you can see, this conceptual workflow consists of simple tasks that are quite common for

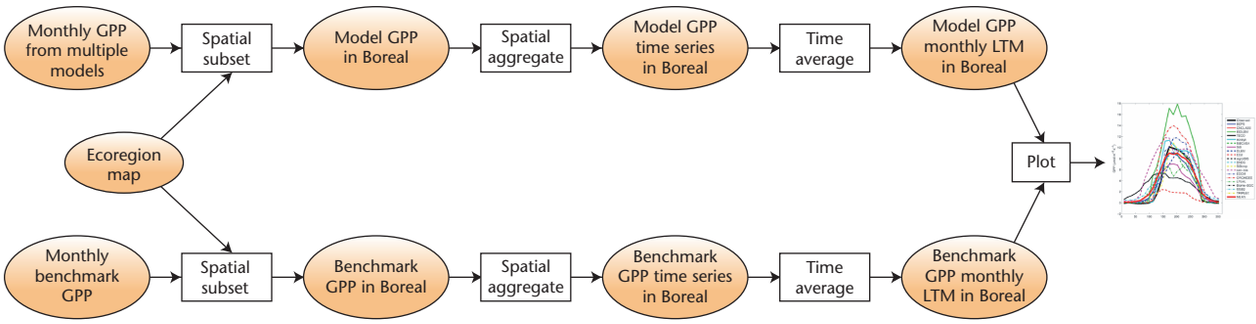

Figure 7. Conceptual workflow to process the *GPP* variable and display the long-term mean plots. After selecting the GPP data of the North American Boreal ecoregion from multiple model outputs and benchmark data, we aggregate them to compute the monthly total GPP and then apply a time-average process to compute and display the monthly long-term-mean total GPP in that ecoregion.

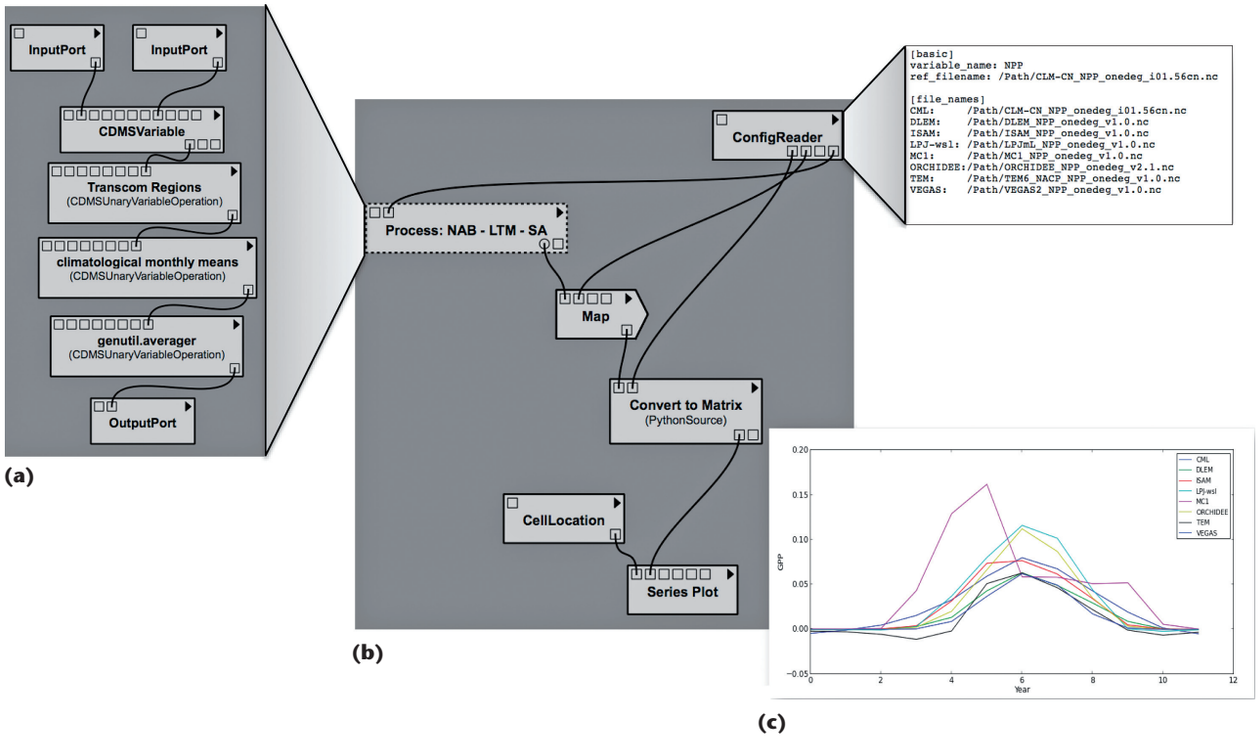

Figure 8. Implementation of the conceptual workflow shown in Figure 4. The subpipeline in (a) is created automatically by UV-CDAT. (b) Using VisTrails components to execute the subpipeline multiple times.

climate scientists. Most of these tasks are available in the CDAT package, which is a core part of UV-CDAT. Using the graphics interface and all the features exposed in the main window, we can easily reproduce all the steps for one model. Figure 8a shows the generated pipeline.

In the current release of UV-CDAT, to apply this process to different variables, we must rely on VisTrails' direct workflow-editing features. As we mentioned before, UV-CDAT's workflow functionality is built on top of VisTrails' software development kit, which means that all the tools available on VisTrails are also available on UV-CDAT. To finish implementing the conceptual workflow, we edit the generated pipeline and use the "Control Flow" package provided by VisTrails. This package provides the module `Map` to create loops in workflows. `Map` applies a function to a list of inputs and returns a sequence of results, so the first task here is to create a single module representing the function that will be used by the `Map` module. This new module is obtained by grouping the pipeline generated by UV-CDAT in a group-module named `Process: NAB-LTM-SA`; in this new group, we just need to specify which ones are going to be the new inputs and outputs. The next step is to define our input list—in our case, it's a sequence of filenames storing the remaining data variables. Now we can connect this module to the `Map` module

and the output will be a sequence of time series. Each time series corresponds to 12 values for each model.

To complete the conceptual workflow, we just need to plot all of the time series. This can be easily performed by selecting the output of the `Map` module and connecting it to the module `Series Plot` provided by VisTrails.

At first, these steps can seem a little daunting and complicated for new users, but we think that UV-CDAT is a powerful tool that can be exploited to implement complex conceptual workflows. To make it easier to understand, we can divide the steps into three phases. First, use the UV-CDAT interface to create a processing pipeline. Second, group the generated pipeline and specify the list of inputs that are going to repeated in the `Map` module. Third, select an appropriate plot to visualize the output from the `Map` module.

Using the same approach, we can create a Taylor diagram for multiple modules.[6] Notice that we can also save the edited pipelines to be shared with other projects. The saved pipeline created from one activity can be reused with a different processing module. For example, to create the Taylor diagram, we just change the module `Process: NAB-LTM-SA` for another module named `Compute Statistics`. Again, this new module was created using the UV-CDAT interface.

In this article, we presented UV-CDAT from a user's perspective—a new tool for the analysis and visualization of climate data. For future work, we hope to refine UV-CDAT provenance and workflow capabilities to make it easier for users to create and reuse the more complex workflows not initially available in the system, and to make the integration with other packages work as smoothly as possible. We also plan to add a more intuitive and powerful provenance browser, and make it easier for scientists to publish their analysis, workflows, and data products on the Web. UV-CDAT is available for downloading at http://uv-cdat.llnl.gov.

Acknowledgments

This project has been funded by the US Department of Energy (DOE) Office of Biological and Environmental Research (BER). Additional support was provided by the National Science Foundation (cooperative ageement 0830944 for DataONE) and NASA (projects 09-TE09-07 and 09-TE09-26). This is a large project involving many institutions, including the Lawrence Livermore National Laboratory, Lawrence Berkeley National Laboratory, Los Alamos, Oak Ridge National Laboratory, Kitware, the Polytechnic Institute of New York University, Scientific Computing and Imaging Institute of Utah, and NASA.

References

1. D.J. Hayes et al., "Reconciling Estimates of the Contemporary North American Carbon Balance among Terrestrial Biosphere Models, Atmospheric Inversions and a New Approach for Estimating Net Ecosystem Exchange from Inventory-Based Data," *Global Change Biology*, vol. 18, no. 4, 2012; doi:10.1111/j.1365-2486.2011.02627.x.
2. D. Huntzinger et al., "North American Carbon Project (NACP) Regional Interim Synthesis: Terrestrial Biospheric Model Intercomparison," *Ecological Modeling*, vol. 232, 2012, pp. 144–157; doi:10.1016/j.ecolmodel.2012.02.004.
3. J. Freire et al., "VisTrails," *The Architecture of Open Source Applications: Elegance, Evolution, and a Few Fearless Hacks*, A. Brown and G. Wilson, eds., Lulu.com, 2011, pp. 377–394.
4. J. Van Wijk, "The Value of Visualization," *Proc. IEEE Visualization*, IEEE, 2005, pp. 79–86.
5. J. Freire et al., "Provenance for Computational Tasks: A Survey," *Computing in Science & Eng.*, vol. 10, no. 3, 2008, pp. 11–21.
6. K.E. Taylor, "Summarizing Multiple Aspects of Model Performance in a Single Diagram," *J. Geophys. Research*, vol. 106, no. D7, 2001, pp. 7183–7192; www.agu.org/pubs/crossref/2001/2000JD900719.shtml.

Emanuele Santos is a professor at the Federal University of Ceara, in Brazil. Her research interests include scientific data management and visualization. Santos has a PhD in computing from the University of Utah. Contact her at emanuele@lia.ufc.br.

Jorge Poco is a PhD student in computer science at the Polytechnic Institute of New York University. His research interests include scientific visualization with a focus on climate datasets. Poco has an MS in computer science from the University of São Paulo, Brazil. Contact him at jpocom@nyu.edu.

Yaxing Wei is a geospatial information scientist for the Climate Change Science Institute at Oak Ridge National Laboratory. His research interests include increasing geospatial data usage by promoting information standardization and interoperability, workflow-based advanced geospatial data visualization and analysis, and Web-based GIS applications. He conducted joint research at George Mason University and received his PhD in safety engineering from the University of Science and Technology of China. Contact him at weiy@ornl.gov.

Shishi Liu is a postdoctoral research associate in the Climate Change Science Institute at Oak Ridge National Laboratory. Her research

interests include carbon fluxes in the terrestrial ecosystem and remote sensing of vegetation. Liu has a PhD in physical geography and remote sensing from the University of California-Santa Barbara. Contact her at lius2@ornl.gov.

Bob Cook is a distinguished research staff member in the Climate Change Science Institute at Oak Ridge National Laboratory. He's also the chief scientist of the NASA ORNL Distributed Active Archive Center for Biogeochemistry and principal investigator of the Modeling and Synthesis Thematic Data Center for the North American Carbon Program. His research interests include biogeochemistry, global change, integrated assessments, aqueous geochemistry, transport and fate of contaminants in aquatic ecosystems, and water resources management. Cook has a PhD in biogeochemistry from Columbia University. He's a member of the American Geophysical Union and Cook is an associate editor for *Biogeochemistry*. Contact him at cookrb@ornl.gov.

Dean Williams is a computer scientist at the Lawrence Livermore National Laboratory, where he's also the program director for the Climate Analytics, Informatics, and Management Systems Division and the group leader of the Global Security Computing Application Division. His research interests are in computer science research, focusing on high-performance and distributed computing, as well as climate-analysis tools. Williams has an MS in computer science from the California State University at Chico. He's a member of the American Geophysical Union. Contact him at williams13@llnl.gov.

Cláudio T. Silva is the head of disciplines at the New York University Center for Urban Sciences and Progress and a professor of computer science and engineering at the Polytechnic Institute of New York University. His research interests include big data and urban systems, visualization and data analysis, and geometry processing. Silva has a PhD in computer science from the State University of New York at Stony Brook. He's a member of IEEE, the ACM, Eurographics, and Sociedade Brasileira de Matemática. Contact him at csilva@nyu.edu.

cn *Selected articles and columns from IEEE Computer Society publications are also available for free at http://ComputingNow.computer.org.*