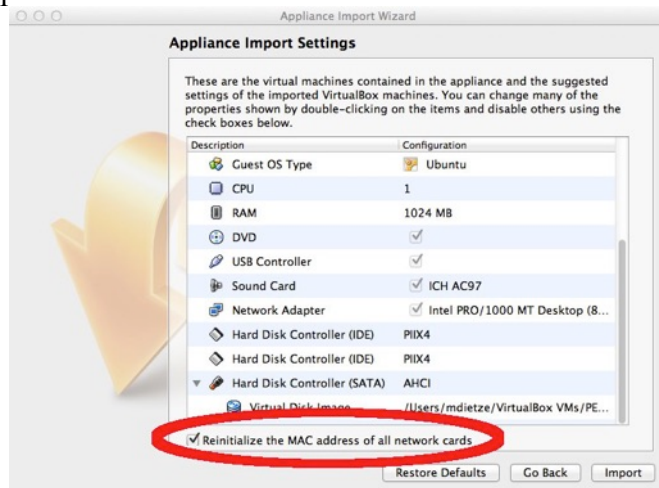


PEcAn v1.4.0 Hands-On Demo

Installing PEcAn:

1. PEcAn consists of a set of scripts and code that is compiled within a Linux operating system and saved in a “virtual machine (VM)”. Virtual machines allow for running consistent set-ups without worrying about differences between operating systems, library dependencies, compiling the code, etc.
2. To run the PEcAn VM you will need to [install VirtualBox](#), the program that runs the virtual machine (<http://www.virtualbox.org>). On Windows you may see a warning about Logo testing, it is okay to ignore the warning.
3. After you have Virtual Box installed you’ll need to [download the PEcAn virtual machine](http://isda.ncsa.illinois.edu/download/index.php?project=PEcAn): <http://isda.ncsa.illinois.edu/download/index.php?project=PEcAn>. The virtual machine is available under the tag vm. Download the 32 bit version and note that the download is ~3 GB so will take from several minutes to hours depending on the connection speed.
4. To open up the virtual machine you’ll first want to [open up VirtualBox](#)
5. The first time you use the VM you’ll want to use [File → Import Appliance](#) in VirtualBox in order to import the VM. This will create a virtual machine from the disk image. When asked about the Appliance Import Settings **make sure you select "Reinitialize the MAC address of all network cards"**. This is not selected by default and can result in networking issues since multiple machines might claim to have the same network MAC Address. That said, users who have experienced network connection difficulties within the VM have sometimes had better luck after reinstalling without reinitializing.
6. Next, [click “Import”](#). You only have to do this Import step once, in the future you can just skip to the next step.



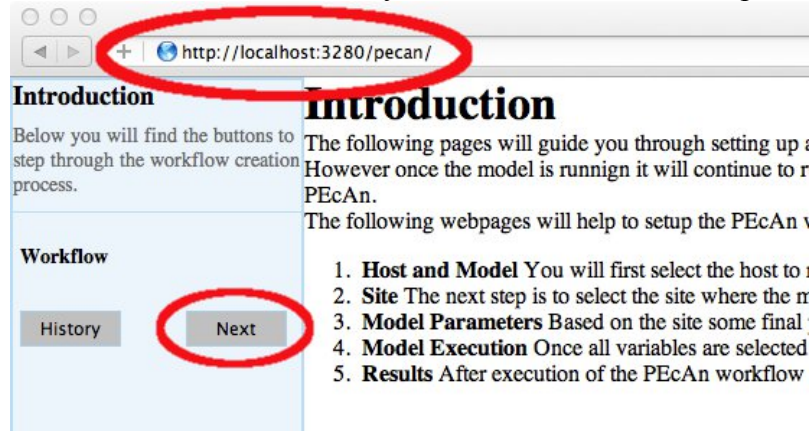
7. [Start Pecan](#) by double clicking on the icon for the VM. A terminal window will pop up showing the machine booting up which may take a minute. It is done booting when you get to the “pecan32 login:” prompt. You do not need to login as the VM behaves like a server that we will be accessing through you web browser. Feel free to [minimize the VM window](#).

DEMO 1: Single Run

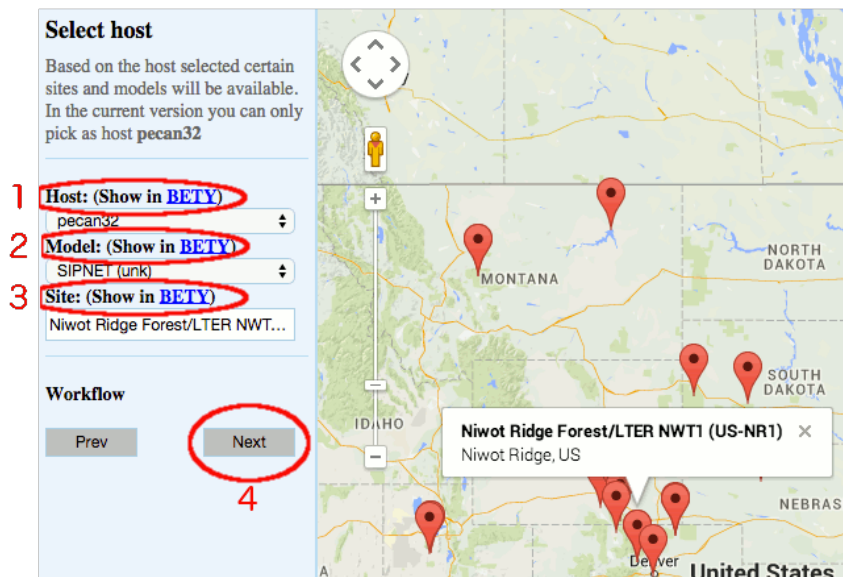
We will begin by exploring a set of web-based tools that are designed to run single-site model runs.

1. To access the web pages you can use a web browser on your own computer's desktop and [connect to the following address http://localhost:3280/pecan/](http://localhost:3280/pecan/)

Note: if you downloaded the 64-bit version you will want to connect to port 6480



1. From the Introduction page, [click the 'Next' button](#) to move to the “Select Host” page.
2. There are three steps to running PeCAN: selecting a site, specifying drivers, and running the model, as detailed below.
3. **Site Selection**



1. **Select Host:** For this exercise you will want to [select the local machine](#) “**pecan32**” to see the sites that are stored locally on the VM and have driver data. PEcAn is designed to conduct runs both on the local VM and on remote machines. By selecting different servers from the “HOST” pull down menu you will be able to see what sites exist within the PEcAn databases on different servers. As the database also records sites where data was collected,

not all sites will have the necessary driver files to run the models.

2. **Select Model:** Next, you'll want to [select the ecosystem model](#) you want to run. We'll begin by using **SIPNET** because it is considerably faster than ED2.
3. **Select Site:** Finally, you'll want to [select the site](#) where you want to run the model by [clicking on a pin icon on the Google Map](#). Feel free to zoom out to examine sites outside the initial frame, or to zoom in to examine clusters of sites or to see exactly where a site is located. When you click on a site's flag on the map, it will give you the name and location of the site. It will also put that site in the "Site:" box on the left hand site, which indicated its your current site selection. For this tutorial, choose the Niwot Ridge (US-NR1) site as the site for your run.
4. Once you have selected a Host, Site, and Model, [click "Next"](#)
5. Throughout PEcAn the **Show in BETY** links will redirect you to the web front end for the underlying database where you can see the detailed entries for various selections (e.g. Host server, Model, Site, PFT, etc.)

4. Run Specification

Selected Site
Set parameters for the run.

- 1 **PFT** [Show in BETY](#)
populus
temperate.coniferous
temperate.deciduous
- 2 **Start Date***
2006/01/01
End Date*
2006/12/31
- 3 **Sipnet.climna***
Use Ameriflux
- 4 **Email**

Edit pecan.xml

Edit model config

* are required fields.

Prev **Next**

5

Niwot Ridge Forest/LTER NWT1 (US-NR1)
Niwot Ridge, CO, US

Next we will select the input files required to run the model and the vegetation type. Be aware that the inputs required for any particular model may vary somewhat so there may be additional input selections available for other models.

1. **PFT:** You'll want to [select the Plant Functional Type](#) that you want to run from the PFT menu. While ecosystem models do have broad similarities, there is still a degree of model-specificity in model parameters, thus the list of available PFTs will vary depending upon which model you selected. Some models, such as ED2, allow for the competition between PFTs within a site and thus selecting multiple PFTs is allowed (and in fact encouraged). SIPNET currently only runs one PFT at a time so you will want to only select a single PFT
2. **Start/End Date:** Select the dates you want to run the model. At the moment PEcAn does not automatically enforce that selected dates correspond to the date ranges of available

inputs (e.g. weather data) so be aware that this may lead to errors later in the workflow.

3. **Weather Data:** This pull-down (Sipnet.climna for SIPNET) will allow you to [select the meteorological driver](#) dataset that you want to use by name and time span. Data sets that are already stored locally will have a name and data range. Data sets that can be automatically downloaded and processed will say “Use” (e.g. Use Ameriflux).
4. **Optional Settings:** [leave all blank for demo run](#)
 1. **Email** will send a message to the specified address when the run is complete. This is handy for slower model or for more complex analyses that involve many model runs, but is not required for this demo.
 2. **Edit pecan.xml** will allow you to modify the PEcAn settings file before the PEcAn workflow is executed. We will use this in Demo 2 in order to turn on additional analyses. This feature can also be used to configure host information for doing model runs on remote servers. Full details on the PEcAn.xml can be found at (<https://github.com/PecanProject/pecan/wiki/PEcAn-Configuration>).
 3. **Edit model config** will pause the workflow after PEcAn has written all model specific settings but before the model runs are called. This allows the user to configure any additional settings internal to the model. Please check with individual modeling teams for information about the details and documentation of such files. Finally, this would also allow users to modify model parameters by hand, however as we’ll discuss below this is not encouraged since PEcAn is estimating model parameters on demand based on it’s internal trait databases. Adding empirical data to this trait database is the preferred methods of informing model parameters.
5. Finally, [click Next](#) to start the model run.

5. Model Run Workflow

Job is running

Job is currently executing, please wait.

Start Over
Results

Execution Status

Stage Name	Start Time	End Time	Status
query.trait	2014-12-12 06:00:40	2014-12-12 06:00:41	DONE
meta.analysis	2014-12-12 06:00:41	2014-12-12 06:00:41	DONE
conversions	2014-12-12 06:00:41		Running
write.config			
advanced.edit			
model			
output.conversion			
ensemble.analysis			
sensitivity.analysis			
finished			

SIPNET will execute the model run fast enough that you are liable to miss some of the following steps (it's more apparent with ED2), but the PEcAn workflow will automatically run the following, indicating model progress at each step

1. **query.trait / meta.analysis:** PEcAn estimates model parameters by performing a meta-analysis of the available trait data for a PFT. Query.trait will extract relevant trait data from the database. Meta.analysis performs a hierarchical Bayes meta-analysis of available trait

data (1). The output of this analysis is a probability distribution for each model parameter. For default single runs PEcAn will select the median value of this parameter, but in Demo 2 we will see how PEcAn can use this parameter uncertainty to make probabilistic forecasts and assess model sensitivity and uncertainty. Errors at this stage usually indicate errors in the trait database or incorrectly specified PFTs (e.g. defining a variable twice).

2. **Conversions:** PEcAn next downloads and processes and model input data that was selected but not available locally. In v1.4.0 this is currently just the “Use Ameriflux” meteorology workflow, which will download L2 data off the Ameriflux website, convert this to a standard format (netCDF files in CF metadata standard for variable names, and units), gap fill missing data, and then convert this data to a model-specific format. This workflow can take a while the first time a site is run, but it caches previous conversions so that it can skip steps if a run is repeated. For example, if we reran this analysis using the ED2 model, the workflow would pick up after the gapfilling step and only have to perform the model-specific conversion. This feature allow different models to easily be run for multiple inputs. Performing all operations on a standard format (netCDF CF) allows us to build reusable tools that work with all models and all data.
3. **write.config:** writes model-specific settings and parameter files, as well as the script (jobs.sh) that PEcAn uses to initiate a model run.
4. **model:** run model. If an analysis involves multiple runs this will also show you which run is currently executing.
5. **output.conversion:** Like how we treat inputs, all model outputs are converted from their model-specific format to a general format (netCDF files, MsTMIP variables & units)
6. **ensemble.analysis & sensitivity analysis:** If these analyses are enabled in the pecan.xml, PEcAn will post-process the output across numerous model runs in order to generate a model projection mean & confidence interval, and a sensitivity & uncertainty analysis, respectively.

FYI, **adding a new model to PEcAn** does not require modification of the model’s code, just the implementation of two modules, write.configs and output.conversion. See the PEcAn wiki (<https://github.com/PecanProject/pecan/wiki/Adding-an-Ecosystem-Model>) for details.

If, by chance, you get an error in your run today it is probably simplest to go back and try changing options and re-running (e.g. with a different site or PFT), as time does not permit detailed debugging.

6. Congratulations! If you got this far, you have managed to run an ecosystem model without ever touching a line of code! Now it’s time to look at the results.

7. Output and Visualization

1. Graphs

1. Select a **Year and Y-axis Variable**, and then click '**Plot run/year/variable**'. Initially leave the X-axis as time.
2. Within this figure the **points indicate the daily mean** for the variable while the **envelope encompasses the diurnal variability** (max and min).
3. Units, variable names, and descriptions are all drawn automatically from the meta-data in a standardized, model-independent netCDF output format (derived from NACP, LBA, PaleON, and MsTMIP model-data inter-comparison projects). See

http://nacp.ornl.gov/MsTMIP_variables.shtml for variable names and units.

4. Figures are *.png files that can be saved by right-clicking and selecting “Save Image As...”. However, as will be described below it is easy to go back to past runs in PEcAn so you don't have to download figures or files in order to keep a record of your results.
5. Try looking at a number of different output variables over different years.
6. Try [changing the X-axis](#) to look at bivariate plots of how different output variables are related to one another

2. **Inputs**

This menu lets you look at and download:

1. A summary file (README.txt) describing the run (location, run ID, model, dates, etc)
2. The model-specific input files fed into the model
3. The jobs.sh file used to submit the model run

3. **Outputs:**

A number of files generated by the underlying ecosystem model are archived and available for download. These include:

1. Output files in the standardized netCDF ([year].nc)
2. Raw model output in model-specific format (e.g. sipnet.out).

For example, one could easily download model output and plot model versus data in Matlab or R, or visualize netCDF output in ncvew or panoply.

4. **PFTs:**

There is a wide array of outputs available that are related to the process of estimating the model parameters for a specific Plant Functional Type. We will revisit many of these files in Demo 2 when we look at the meta-analysis in more detail, but the following is the general idea of what is contained in different files:

1. *.bug files provide the statistical code used to estimate each parameter in JAGS.
2. The ma.summaryplots*.pdf are collections of diagnostic plots produced in R after the above JAGS code is run that are useful in assessing the statistical model.
3. The Rdata files trait.data.Rdata and madata.Rdata are, respectively, the available trait data extracted from the database that was used to estimate the model parameters and that same data cleaned and formatted for the statistical code.
4. The file traits.mcmc.Rdata contains the raw output from the statistical code. This includes samples from all of the parameters in the meta-analysis model, not just those that feed forward to the ecosystem, but also the variances, fixed effects, and random effects.
5. The files prior.distns.Rdata and post.distns.Rdata store simple tables of the prior and posterior distributions for all model parameters in terms of the name of the distribution and its parameters.
6. Finally, posteriors.pdf provides graphics showing, for each model parameter, the prior distribution, the data, the smoothed histogram of the posterior distribution (labeled *post*), and the best-fit analytical approximation to that smoothed histogram (labeled *approx*)

5. **PEcAn Files:**

Likewise, a number log files from the PEcAn workflow are available from the execution of the workflow. These include:

1. The STATUS file giving a summary of the steps of the workflow, the time they took, and whether they were successful

2. The PEcAn settings file (pecan.xml),
3. The workflow script (workflow.R),
4. The corresponding log file (workflow.Rout),
5. The Monte Carlo samples used in the runs (samples.Rdata),
6. Various ensemble analysis files, which are not terribly interesting in the current context since by default the web interface only does one run (ensemble size, n=1). We'll come back to this in Demo 2

These files are primarily useful if the workflow generates an error, and as metadata & provenance (a detailed record of how data was generated), but also includes useful diagnostics from the meta-analysis.

8. Model Run Archive

1. [Click on the HISTORY button](#) at the start or end of the workflow to go to a table recording past model runs. [Click on any previous run in the "ID" column](#) to go to the current state of that run's execution. This is most commonly the output and visualization page, but if you are doing a long run you can always return to that run in-progress this way. The run you just did should be the more recent entry in the table. We encourage you in your own time other models for your site for comparison and because some models (e.g. ED2) also have a more diverse range of output variables to explore.

DEMO 2: Ensemble & Sensitivity Analyses

In Demo 2 we will be looking at how PEcAn can use information about parameter uncertainty to perform three automated analyses:

- **Ensemble Analysis:** Repeat numerous model runs, each sampling from the parameter uncertainty, in order to generate a probability distribution of model projections. [Allows us to put a confidence interval on the model](#)
- **Sensitivity Analysis:** Repeats numerous model runs to assess how changes in model parameters will affect model outputs. [Allows us to identify which parameters the model is most sensitive to.](#)
- **Uncertainty Analysis:** Combines information about model sensitivity with information about parameter uncertainty in order to determine the contribution of each model parameter to the uncertainty in model outputs seen in the Ensemble Analysis. [Allow us to identify which parameters are driving model uncertainty.](#)

1. Return to the main menu for PEcAn (e.g. click Start Over)
2. Repeat the steps for site selection and run specification, however this time **Click to turn on “Edit pecan.xml”**
3. Modify settings to run a meta-analysis, sensitivity analysis, and ensemble analysis. When the pecan.xml comes up in the Advanced Edit model you will need to make the following changes:

The screenshot shows the 'Advanced Edit' window in PEcAn. On the left, there is a 'File Selection' dropdown menu with 'pecan.xml' selected, and two buttons: 'Start Over' and 'Continue'. A red number '4' is placed below the 'Continue' button. The main area on the right displays the XML configuration for the model. A red box highlights the following XML tags: `<ensemble>`, `<size>25</size>`, `<variable>NPP</variable>`, `</ensemble>`, `<sensitivity.analysis>`, `<quantiles>`, `<sigma>-2</sigma>`, `<sigma>-1</sigma>`, `<sigma>1</sigma>`, `<sigma>2</sigma>`, `</quantiles>`, `<start.year>2006</start.year>`, `<end.year>2006</end.year>`, `<variable>NPP</variable>`, and `</sensitivity.analysis>`. A red number '1' is placed to the right of the ensemble tag, and a red number '2' is placed to the right of the sensitivity analysis tag. At the bottom of the XML editor, there is a 'Save' button with a red number '3' placed below it.

1. In order to run an ensemble analysis, we need to tell how many runs we want in our ensemble. We do this by changing the value set in the ensemble tag from 1 run to 25 runs so that it looks like.:


```
<ensemble>
  <size>25</size>
</ensemble>
```

As a reminder, the ensemble analysis will sample all model parameters randomly from their posterior distributions in order to propagate this uncertainty into the model output. In practice you would want to use a larger ensemble size (100-5000) than we are using in the demo.

2. In order to run the sensitivity analysis, we likewise need to turn on this module, and tell PEcAn how we want the parameters varied. PEcAn's sensitivity analysis holds all parameters at their median value and then varies each parameter one-at-a-time based on the *quantiles* of the posterior distribution. For example, to sample at the edge of a 95% CI you would use the `<quantile>` tag to set the quantiles to 0.025 and 0.975. PEcAn also includes a handy shortcut, the `<sigma>` tag, that converts a specified standard deviation into its Normal quantile equivalent. For example, 1 and -1 are converted to 0.157 and 0.841 (a 68.2% CI). Finally, we need to specify what output variable we are calculating sensitivity for and what time period we are averaging this variable over. To specify time you use the `<start.year>` and `<end.year>` tags. The `<variable>` tag is a character string that follows the output standards used by recent inter-comparison projects (MsTMIP, NACP, LBA, PaleON), see http://nacp.ornl.gov/MsTMIP_variables.shtml for variable names and units.

Below is an example for NPP that samples from -2 to 2 sigma using data from just 2006. Copy and paste it into your demo.xml file right after the `</ensemble>` tag:

```
<sensitivity.analysis>
  <quantiles>
    <sigma>-2</sigma>
    <sigma>-1</sigma>
    <sigma>1</sigma>
    <sigma>2</sigma>
  </quantiles>
  <start.year>2006</start.year>
  <end.year>2006</end.year>
  <variable>NPP</variable>
</sensitivity.analysis>
```

Check that the variable, start.year, and end.year for the ensemble and sensitivity.analysis are the same. If one set of these tags is missing, the system will assume that they are the same and use the information from one to fill in the other.

3. **“Save” you changes**
4. **Click “Continue”**
5. Within the pecan.xml file you will also see other settings for different modules. Hopefully the naming of these is intuitive. The latest version of the pecan.xml documentation, and further documentation of the PEcAn system in general, can be found in the wiki section of our GitHub website (<https://github.com/PecanProject/pecan/wiki/PEcAn-Configuration>). The wiki also included information on how to connect PEcAn to a remote cluster, which is a common and often necessary alternative to doing hundreds of model runs on your laptop (<https://github.com/PecanProject/pecan/wiki/Connecting-pecan-to-a-remote-server>).

4. Additional Outputs:

The PEcAn workflow will take considerably longer to complete since we have just asked for over a hundred model runs. Once the runs are complete you will return to the output visualization page where there will be a few new outputs to explore, as well as a few outputs that were present earlier that we'll explore in greater details:

1. **Run ID:**

While a number of the PEcAn outputs synthesize across each of the runs, you can also select individual runs from the Run ID menu. For an individual run you can then use the Graphs menu to visualize each individual run. The README.txt file can be used to get a quick view of whether each run was part of the sensitivity or ensemble analysis, as well as what variables are being modified in the sensitivity analysis.

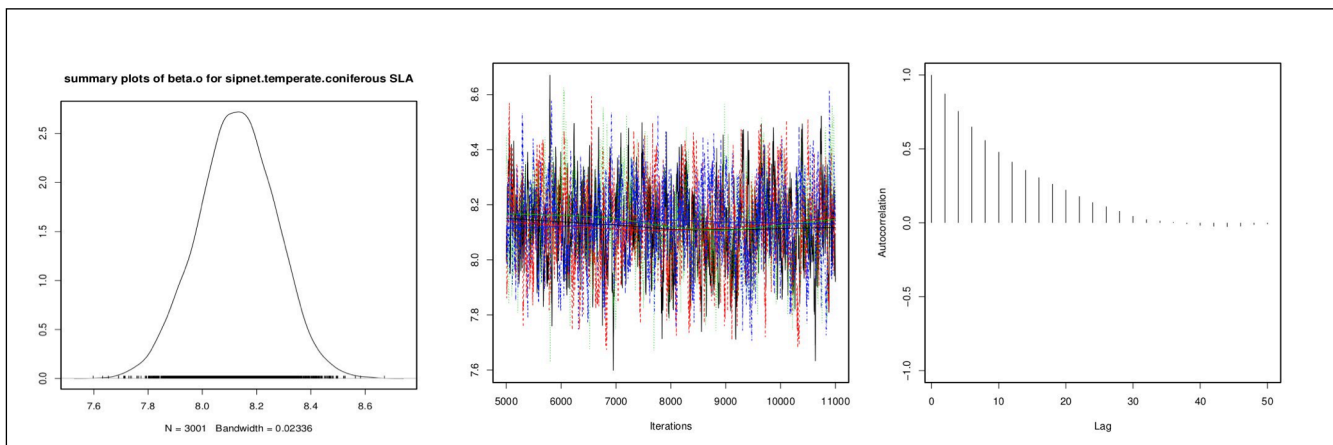
- ##### 2. **PFTs & Trait Data:**
- The list of variables that are queried is determined by what variables have priors associated with them in the definition of the PFTs. Likewise, the list of species that are associated with a PFT determines what subset of data is extracted out of all data matching a given variable name. Demo 3 will demonstrate how a PFT can be created or modified. The trait output is written to two Rdata files, **trait.data.Rdata** and **prior.distns.Rdata**. To look at these in R you could use the menu to download the file and then within R [click on these files to load them into your workspace](#). You can further examine them in the **Environment** window or accessing them at the command line. For example, try typing `names(trait.data)` as this will tell you what variables were extracted, `names(trait.data$Amax)` will tell you the names of the columns in the Amax table, and `summary(trait.data$Amax)` will give you summary data about the Amax values.

3. **Meta-analysis**

1. The meta-analysis code begins by distilling the trait.data to just the values needed for the meta-analysis statistical model, with this being stored in **madata.Rdata**. This reduced form includes the conversion of all error statistics into precision (1/variance), and the indexing of sites, treatments, and greenhouse. In reality, the core meta-analysis code can be run independent of the trait database as long as input data is correctly formatted into the form shown in madata.
2. The evaluation of the meta-analysis is done using a Bayesian statistical software package called JAGS that is called by the R code. For each trait, the R code will generate a `[trait].model.bug` file that is the JAGS code for the meta-analysis itself. This code is generated on the fly, with PEcAn adding or subtracting the site, treatment, and greenhouse terms depending upon the presence of these effects in the data itself. If the `<random.effects>` tag is set to FALSE then all random effects will be turned off even if there are multiple sites and treatments.
3. The MCMC outputs a number of diagnostics, including the summary statistics of the model, an assessment of whether the posterior is consistent with the prior, and the status of the Gelman-Brooks-Rubin convergence statistic (which is ideally 1.0 but should be less than 1.1). All of this output can be found in the **meta-analysis.log**

file.

- The code generates a number of diagnostic figures in a `ma.summaryplots.[pft].[trait].pdf` plot for each pft and trait. [Open up one of these pdfs](#) to evaluate the shape of the posterior distributions (they should generally be unimodal), the convergence of the MCMC chains (all chains should be mixing well from the same distribution), and the autocorrelation of the samples (should be low).
- Finally, the code attempts to fit an analytical approximation to the MCMC posterior. The outputs of this are `post.distns.Rdata`, which is structured identically to the `prior.distns` but updated for the meta-analysis, and `posteriors.pdf`, which is a series of graphs comparing the MCMC posterior, the analytical approximation, the data, and the priors. [Open posteriors.pdf and compare the posteriors to the priors and data.](#)



- Model Parameters** for the various runs are stored in `samples.Rdata`. This file contains two data objects, `sa.samples` and `ensemble.samples`, that are the parameter values for the sensitivity analysis and ensemble runs respectively.
- Ensemble Analysis:**
 - Raw results are stored in `ensemble.Rdata`, which contains the object `ensemble.output`, which is the model predictions at the parameter values given in `ensemble.samples`.
 - [Open ensemble.analysis.pdf](#) contains the ensemble prediction as both a histogram and a boxplot.
 - [Open ensemble.ts.pdf](#) contains a time-series plot of the ensemble mean, median, and 95% CI
- Sensitivity Analysis:**
 - Raw results are stored in `sensitivity.Rdata` which contains the object `sensitivity.output` which is the model outputs corresponding to the parameter values in `sa.samples`.
 - Sensitivity plots are in `sensitivityanalysis.pdf`, which shows the raw data points from univariate one-at-a-time analyses and spline fits through the points

7. **Uncertainty Analysis:**

The variance decomposition plot, **variancedecomposition.pdf**, contains three columns, the coefficient of variation (normalized posterior variance), the elasticity (normalized sensitivity), and the partial standard deviation of each model parameter.

QUESTIONS:

8. The Variance Decomposition graph is sorted by the variable explaining the largest amount of variability in the model output (right hand column). [From this graph identify the top-tier parameters that you would target for future constraint.](#)
9. A parameter can be important because it is highly sensitive, because it is highly uncertain, or both. [Identify parameters in your output that meet each of these criteria. Additionally, identify parameters that are highly uncertain but unimportant \(due to low sensitivity\) and those that are highly sensitive but unimportant \(due to low uncertainty\).](#)
10. Parameter constraints could come from further literature synthesis, from direct measurement of the trait, or from data assimilation. [Choose the parameter that you think provides the most efficient means of reducing model uncertainty and propose how you might best reduce uncertainty in this process.](#) In making this choice remember that not all processes in models can be directly observed, and that the cost-per-sample for different measurements can vary tremendously (and thus the parameter you measure next is not always the one contributing the most to model variability). Also consider the role of parameter uncertainty versus model sensitivity in justifying your choice of what parameters to constrain.

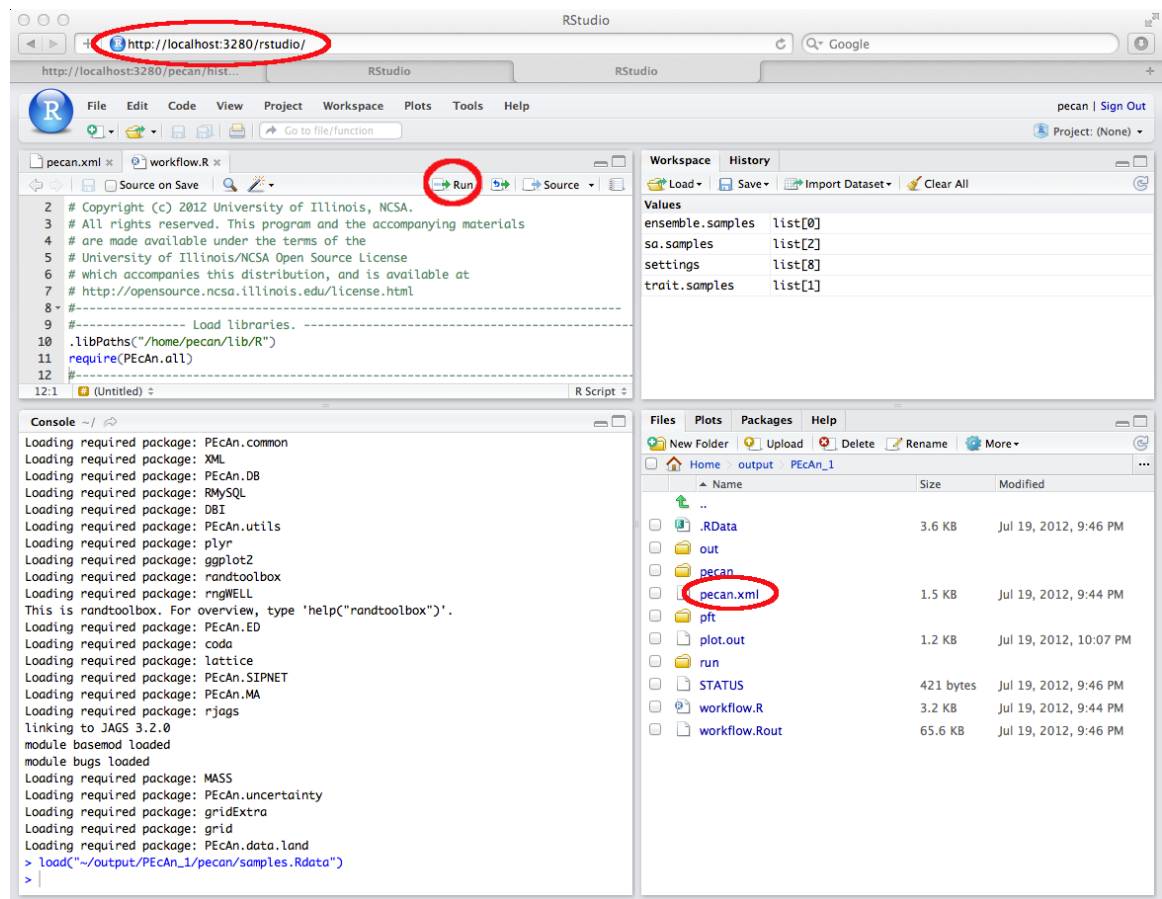
Additional material

In addition to the materials below you are encouraged to take a look at the **Site-Level Runs** section on the PEcAN wiki (<https://github.com/PecanProject/pecan/wiki/Getting-started#site-level-runs>) which describes the process of setting up a new site, new meteorology, new PFT, etc.

Additional useful information on the wiki includes: how to report bugs, how to sync your PEcAN database with others, how to grab the latest “development” branch of the PEcAN code, how to install PEcAN on a server, how to add new models and data processing modules, and how to add new code to PEcAN.

Currently, the web interface does not have access to all of PEcAN’s functionality. For additional tasks we will need to work in the statistical programming language R. To make this easier we have installed a browser-based R interface, RStudio Server (<http://rstudio.org>), on the VM.

1. Open RStudio Server:
 1. Open up a second window on your web browser to <http://localhost:3280/rstudio>
If you closed your previous PEcAN session, make sure the first window is open and points to <http://localhost:3280/pecan>
 2. The username for the VM is *carya* and the password is *illinois*.
 3. You do not need to do this now, but FYI it is also possible to ssh into the virtual machine using the following command in terminal:
`ssh -l pecan -p 3222 localhost`



2. Open PEcAn settings file

1. Go back to <http://localhost:3280/pecan> and note the Run ID for your model run. In the visualization window this is listed in the top left pull-down menu

3. PEcAn Workflow

1. In the Files window, go to the **Home > pecan > scripts** directory
 2. Open the file **workflow.R**. This script provides an example of some of the high-level modules within PEcAn and we will use it to run the meta-analysis, sensitivity analysis, ensemble analysis, and variance decomposition.
1. Re-run the sensitivity analysis and variance decomposition for a different output variable, time period, site, or PFT and compare results with first site.
 1. If you are only changing the output variable or time period, you don't need to re-run the whole workflow since all the output variables are in the model output files. You can modify the `pecan.xml` settings object within R and run the workflow from the extraction of the output from the model runs onward (from `get.model.output` on)
 2. If you are running a new site, it is generally a good idea to do a single test run first to make sure the model behaves sensibly before conducting hundreds of (potentially uninformative) runs.

2. Use the web database interface to change priors, define a new site, or define a new PFT
 1. Open up a new browser window to <http://localhost:3280/bety> . This page is the web-interface to the database underlying PEcAn – it started as an independent project as part of our biofuel work, hence the name BETY-db, but is now much more generic (apologies for the fact that web design has not kept up). For those that are interested the full version of the database is at betydb.org.
 2. To modify or add a new Plant Functional Type (PFT), or to change a PFT's priors, navigate on the grey menu bar to Data > PFTs
 1. To add a new pft, click “new PFT” at the top and enter a name and description.
 2. To add new species to a PFT click on [+] View Related Species and type the species, genus, or family you are looking for into the Search box. Click on the + to add.
 3. To remove a species from a PFT, click on [+] View Related Species and click on the X of the species you want to remove from the PFT.
 4. To remove a prior, click [-] View Related Prior and click on the X of the variable who's prior you want to remove. This will cause the parameter to be excluded from all analyses (meta-analysis, sensitivity analysis, etc) and revert to its default value.
 5. To add a prior, choose one from the white box of priors on the right to choose.
 6. To view the specification of a prior, or to add a new prior, click BETY-DB > Priors and enter the information on the variable, distribution name, distribution parameters, etc. N is the sample size underlying the prior specification (0 is ok for uninformative priors).
 7. You can also go to Data > Variables in order to use the search function to find an existing variable (or create a new one). Please try not to create new variables unnecessarily (e.g. changes of variable name or units to what your model uses is handled internally, so you want to find the trait with the correct MEANING).
 3. To add a new site
 1. The database currently requires that a site be associated with a citation. Navigate to > CITATIONS and select a reference for the study site by clicking the check mark, or create a new citation (creating a 'pers. communication' or 'unpublished data' citation IS allowed)
 2. Click on the **New Site** button. Fill out at least a name and lat/lon (which can be done by hand or by clicking on the map).
 3. To upload model driver data [or any other input data] for the site (currently required in order to run the model at a site)
 1. Select RUNS > INPUTS
 2. At the bottom of the page click on New Input
 3. Choose your site from the pulldown menu
 4. Set the date range, file format, and other meta-data
 5. Click “Create”
 6. To associate the Input record in the database, click “Edit” and then “New File”
 7. Select the machine that the file is on (pecan32 is the VM itself) and fill in the directory path where the file is located and the file name
 8. Click Update

1. LeBauer D, Wang D, Richter KT, Davidson CC, Dietze MC (2013) Facilitating feedbacks between field measurements and ecosystem models. *Ecol Monogr* 83:133–154.