BioCro Run Using the Virtual Machine

1. Download the **VirtualBox** application from here and open the application. Additionally, download the extension pack from the Downloads page under the 'VirtualBox 6.0.6 Oracle VM VirtualBox Extension Pack' section. Add the extension pack by opening VirtualBox application, going to "Preferences" -> "Extensions", clicking on the blue box with the green plus sign, and adding the downloaded .vbox-extpack file.

2. Download the PEcAn **virtual machine** .ova file from here and open it from within VirtualBox by going to "File" -> "Import appliance" and selecting the .ova. Start the virtual machine by clicking on the green arrow in VirtualBox. This will bring up a new window, and the virtual machine is ready when it displays `pecan login:`.

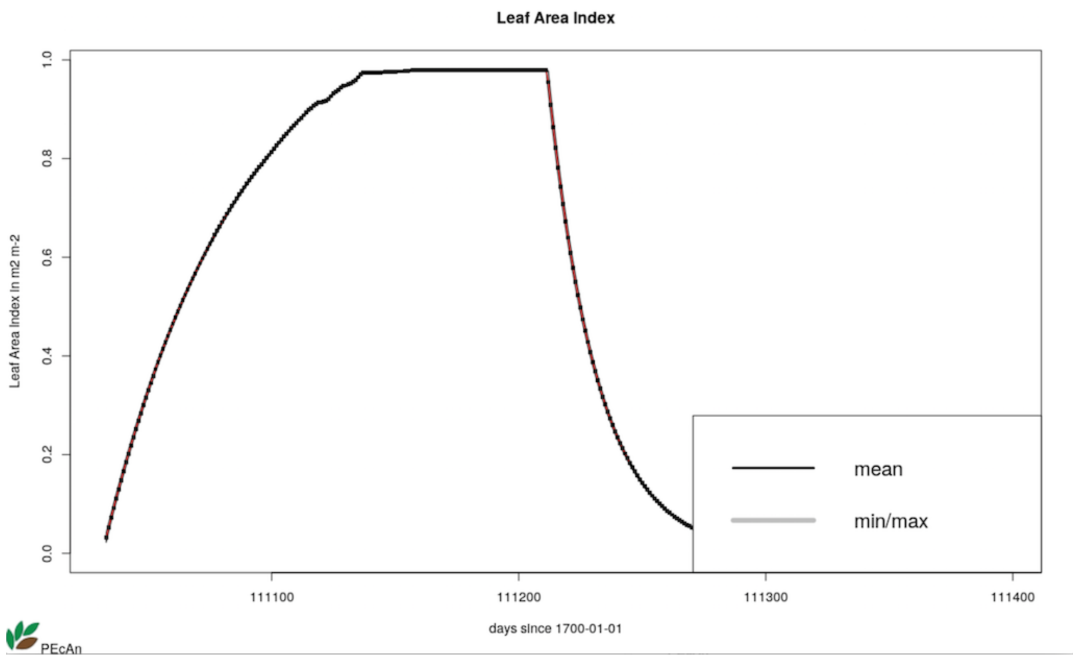
3. Open up the **RStudio** interface to the VM by going to localhost:6480/rstudio/ in your browser. To log in, the username is `carya` and the password is `illinois`.

4. Use ssh to open up the **command line** for the VM. Do this by opening up a command line program on your computer and execute `ssh -p 6422 carya@localhost`. You will be prompted for a password with `carya@localhost's password:`. The password is `illinois`, and the letters will not appear when you type them in.

5. Create a new folder for the run results using `mkdir biocro_results`. Edit the XML to include this file path by doing `nano pecan/tests/pecan64.biocro.xml` and changing the third line from `<outdir>pecan</outdir>` to `<outdir>biocro_results</outdir>` and saving these changes. Additionally, change the 19th line from `<name>Miscanthus_x_giganteus</name>` to `<name>salix<name>` to set the PFT to one that has trait data. Do the **model run** by putting the BioCro XML file through PEcAn by executing `pecan/web/workflow.R --settings pecan/tests/pecan64.biocro.xml`.

6. Go to the RStudio instance that is open in your browser. In the file directory, there should be a new `biocro_results` folder, and you can see all the files by clicking on it. The .nc data files that are in the `out` subfolder can be **plot** in an R script, as is shown for leaf area index.

```
PEcAn.visualization::plot.netcdf("biocro_results/out/SA-salix-chi_leaf-0.159/2004.nc", "LAI")
```



BioCro Run Using University of Arizona's RStudio Instance

1. Access RStudio in a browser by navigating to welsch.cyverse.org:8787/. Log in using your username and password, which can be requested here.
2. Create a new folder that will hold the necessary files for the run in your home directory by going to the Terminal tab and typing in `mkdir biocro_files`. This new folder should appear in your file structure under the Files tab.
3. The first needed file is an XML that contains the configuration information required to get the data and run the model. This example run is for the willow tree PFT for a site just outside of Champaign, Illinois using weather data from the entire year of 2004. We are also running ensemble and sensitivity analyses, and a meta-analysis.

   Create the file by clicking the new file button with the green plus sign in the upper left hand corner and selecting text file. Copy and paste the content below into this file, click the save button, and save this file as `pecan64.biocro.xml` in the `biocro_files` folder.

```
<pecan>
  <outdir>biocro_results</outdir>

  <database>
    <bety>
      <driver>PostgreSQL</driver>
      <user>bety</user>
      <password>bety</password>
      <host>postgres</host>
      <dbname>bety</dbname>
      <write>FALSE</write>
    </bety>
    <dbfiles>biocro_results/dbfiles</dbfiles>
  </database>

  <pfts>
    <pft>
      <name>salix</name>
    </pft>
  </pfts>

  <ensemble>
    <variable>AbvGrndWood</variable>
  </ensemble>

  <meta.analysis>
    <iter>3000</iter>
    <random.effects>FALSE</random.effects>
    <threshold>1.2</threshold>
    <update>AUTO</update>
  </meta.analysis>

  <sensitivity.analysis>
    <quantiles>
      <sigma>-1</sigma>
      <sigma>1</sigma>
    </quantiles>
    <variable>AbvGrndWood</variable>
  </sensitivity.analysis>

  <model>
    <type>BIOCRO</type>
    <binary>/home/kristinariemer/pecan/models/biocro/inst/biocro.Rscript</binary>
    <revision>1.0</revision>
  </model>

  <run>
    <site>
      <id>753</id>
    </site>
    <inputs>
      <met>
        <output>BIOCRO</output>
        <path>/home/kristinariemer/pecan/models/biocro/tests/testthat/data/US-Bo1</path>
      </met>
    </inputs>
      <start.date>2004/01/01</start.date>
      <end.date>2004/12/30</end.date>
    <host>
      <name>localhost</name>
    </host>
  </run>
</pecan>
```

4. The second file that needs to be created is an R script that runs through all of PEcAn's functions to run the model and produce the output. Create this file by clicking on the new file button and selecting R script. Copy and paste the below text into the new script and save in the `biocro_files` folder as `workflow.R`.

```r
#!/usr/bin/env Rscript
#-------------------------------------------------------------------------------
# Copyright (c) 2012 University of Illinois, NCSA.
# All rights reserved. This program and the accompanying materials
# are made available under the terms of the
# University of Illinois/NCSA Open Source License
# which accompanies this distribution, and is available at
# http://opensource.ncsa.illinois.edu/license.html
#-------------------------------------------------------------------------------

# ----------------------------------------------------------------------
# Load required libraries
# ----------------------------------------------------------------------
library(PEcAn.all)
library(PEcAn.utils)
library(RCurl)

# make sure always to call status.end
options(warn=1)
options(error=quote({
  PEcAn.utils::status.end("ERROR")
  PEcAn.remote::kill.tunnel(settings)
  if (!interactive()) {
    q(status = 1)
  }
}))

#options(warning.expression=status.end("ERROR"))


# ----------------------------------------------------------------------
# PEcAn Workflow
# ----------------------------------------------------------------------
# Open and read in settings file for PEcAn run.
args <- commandArgs(trailingOnly = TRUE)
if (is.na(args[1])){
  settings <- PEcAn.settings::read.settings("pecan.xml")
} else {
  settings.file <- args[1]
  settings <- PEcAn.settings::read.settings(settings.file)
}

# Check for additional modules that will require adding settings
if("benchmarking" %in% names(settings)){
  library(PEcAn.benchmark)
  settings <- papply(settings, read_settings_BRR)
}

if("sitegroup" %in% names(settings)){
  if(is.null(settings$sitegroup$nSite)){
    settings <- PEcAn.settings::createSitegroupMultiSettings(settings,
                                                              sitegroupId = settings$sitegroup$id)
  } else {
    settings <- PEcAn.settings::createSitegroupMultiSettings(settings,
                                                              sitegroupId = settings$sitegroup$id,
                                                              nSite = settings$sitegroup$nSite)
  }
  settings$sitegroup <- NULL ## zero out so don't expand a second time if re-reading
}

# Update/fix/check settings. Will only run the first time it's called, unless force=TRUE
settings <- PEcAn.settings::prepare.settings(settings, force = FALSE)

# Write pecan.CHECKED.xml
PEcAn.settings::write.settings(settings, outputfile = "pecan.CHECKED.xml")

# start from scratch if no continue is passed in
statusFile <- file.path(settings$outdir, "STATUS")
if (length(which(commandArgs() == "--continue")) == 0 && file.exists(statusFile)) {
  file.remove(statusFile)
}

# Do conversions
settings <- PEcAn.workflow::do_conversions(settings)

# Query the trait database for data and priors
```

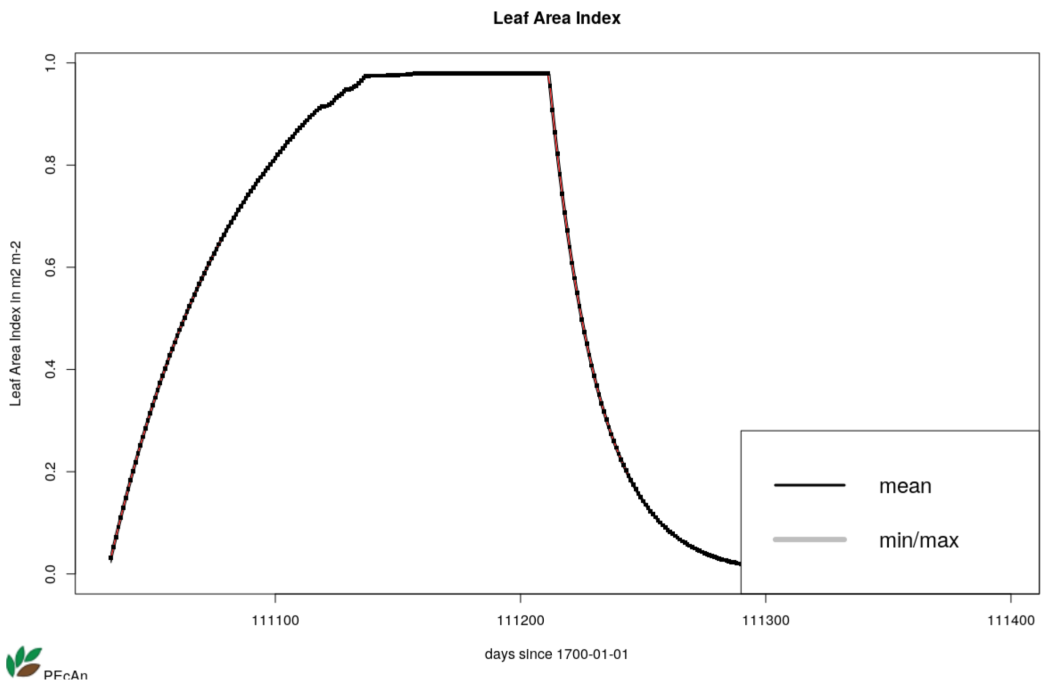
```r
if (PEcAn.utils::status.check("TRAIT") == 0){
  PEcAn.utils::status.start("TRAIT")
  settings <- PEcAn.workflow::runModule.get.trait.data(settings)
  PEcAn.settings::write.settings(settings, outputfile='pecan.TRAIT.xml')
  PEcAn.utils::status.end()
} else if (file.exists(file.path(settings$outdir, 'pecan.TRAIT.xml'))) {
  settings <- PEcAn.settings::read.settings(file.path(settings$outdir, 'pecan.TRAIT.xml'))
}


# Run the PEcAn meta.analysis
if(!is.null(settings$meta.analysis)) {
  if (PEcAn.utils::status.check("META") == 0){
    PEcAn.utils::status.start("META")
    PEcAn.MA::runModule.run.meta.analysis(settings)
    PEcAn.utils::status.end()
  }
}

# Write model specific configs
if (PEcAn.utils::status.check("CONFIG") == 0){
  PEcAn.utils::status.start("CONFIG")
  settings <- PEcAn.workflow::runModule.run.write.configs(settings)
  PEcAn.settings::write.settings(settings, outputfile='pecan.CONFIGS.xml')
  PEcAn.utils::status.end()
} else if (file.exists(file.path(settings$outdir, 'pecan.CONFIGS.xml'))) {
  settings <- PEcAn.settings::read.settings(file.path(settings$outdir, 'pecan.CONFIGS.xml'))
}

if ((length(which(commandArgs() == "--advanced")) != 0) && (PEcAn.utils::status.check("ADVANCED") == 0)
  PEcAn.utils::status.start("ADVANCED")
  q();
}

# Start ecosystem model runs
if (PEcAn.utils::status.check("MODEL") == 0) {
  PEcAn.utils::status.start("MODEL")
  PEcAn.remote::runModule.start.model.runs(settings, stop.on.error = FALSE)
  PEcAn.utils::status.end()
}

# Get results of model runs
if (PEcAn.utils::status.check("OUTPUT") == 0) {
  PEcAn.utils::status.start("OUTPUT")
  runModule.get.results(settings)
  PEcAn.utils::status.end()
}

# Run ensemble analysis on model output.
if ('ensemble' %in% names(settings) & PEcAn.utils::status.check("ENSEMBLE") == 0) {
  PEcAn.utils::status.start("ENSEMBLE")
  runModule.run.ensemble.analysis(settings, TRUE)
  PEcAn.utils::status.end()
}

# Run sensitivity analysis and variance decomposition on model output
if ('sensitivity.analysis' %in% names(settings) & PEcAn.utils::status.check("SENSITIVITY") == 0) {
  PEcAn.utils::status.start("SENSITIVITY")
  runModule.run.sensitivity.analysis(settings)
  PEcAn.utils::status.end()
}

# Run parameter data assimilation
if ('assim.batch' %in% names(settings)) {
  if (PEcAn.utils::status.check("PDA") == 0) {
    PEcAn.utils::status.start("PDA")
    settings <- PEcAn.assim.batch::runModule.assim.batch(settings)
    PEcAn.utils::status.end()
  }
}

# Run state data assimilation
if ('state.data.assimilation' %in% names(settings)) {
  if (PEcAn.utils::status.check("SDA") == 0) {
    PEcAn.utils::status.start("SDA")
    settings <- sda.enfk(settings)
    PEcAn.utils::status.end()
  }
}

# Run benchmarking
if("benchmarking" %in% names(settings) & "benchmark" %in% names(settings$benchmarking)){
  PEcAn.utils::status.start("BENCHMARKING")
  results <- papply(settings, function(x) calc_benchmark(x, bety))
  PEcAn.utils::status.end()
}

# Pecan workflow complete
if (PEcAn.utils::status.check("FINISHED") == 0) {
  PEcAn.utils::status.start("FINISHED")
  PEcAn.remote::kill.tunnel(settings)
  db.query(paste("UPDATE workflows SET finished_at=NOW() WHERE id=",
                 settings$workflow$id, "AND finished_at IS NULL"),
           params = settings$database$bety)

  # Send email if configured
  if (!is.null(settings$email) && !is.null(settings$email$to) && (settings$email$to != "")) {
    sendmail(settings$email$from, settings$email$to,
             paste0("Workflow has finished executing at ", base::date()),
             paste0("You can find the results on ", settings$email$url))
  }
  PEcAn.utils::status.end()
}

db.print.connections()
print("---------- PEcAn Workflow Complete ----------")
```

5. We will then run the newly created XML file through the R script. First create a new folder in which to store the results by executing `mkdir biocro_results` in the Terminal. Then in the Terminal run `biocro_files/workflow.R --settings biocro_files/pecan64.biocro.xml`. This will take a few minutes to finish. The last line in the console should appear as follows.

```
[1] "---------- PEcAn Workflow Complete ----------"
```

6. To test if this works, plot the output data for leaf area index across time. In the console or a new R script, type the following R command and the plot shown should be generated.

```
PEcAn.visualization::plot.netcdf("biocro_results/out/SA-salix-chi_leaf-0.159/2004.nc", "LAI")
```


Leaf Area Index
Leaf Area Index in m2 m-2
days since 1700-01-01
mean
min/max
PEcAn