Software Documentation

for

ecosystem Production in Space and Time (ePiSaT)

Document Version 1.

Prepared by Bradley Evans

Macquarie University

10 September, 2013

This project is supported by the Australian National Data Service (ANDS). ANDS is supported by the Australian Government through the National Collaborative Research Infrastructure Strategy Program and the Education Investment Fund (EIF) Super Science Initiative.

For more information visit the ANDS website ands.org.au and Research Data Australia services.ands.org.au

Table of Contents

1. Background and Context

The ecosystem Production in Space and Time (ePiSaT) aimed at addressing the national need for information on the spatial and temporal distribution of gross primary production (GPP) across the diverse natural, semi-natural and agricultural ecosystems of the Australian continent. GPP is the daily, monthly or annual flux of carbon from atmospheric carbon dioxide (CO_2) into green plants, generated by photosynthesis. It is the foundation for all vegetation, food, fibre and bioenergy production. Reliable GPP data are a pre-requisite for assessing Australia's present carbon balance and the potential for increased carbon sequestration on land. Because GPP and water use by vegetation are closely coupled, GPP is also necessary information to assess the impact of vegetation and land use changes on water resources.

To model GPP across the Australian content, the software system is designed to analyse and partition the best observed estimate: OzFlux eddy covariance carbon dioxide flux data. Flux towers don't measure GPP, they measure the movement if air and concentrations of carbon dioxide. At present, OzFlux provide these observations and estimates of net ecosystem exchange. For the development of ePiSaT's national maps of GPP, as demonstrated in this document, the OzFlux community provided a subset of their tower data. The ePiSaT too takes these data, performs statistical and ecosystem process based analysis to determine light use efficiency (LUE) so that continental GPP can be determined using a simple model.

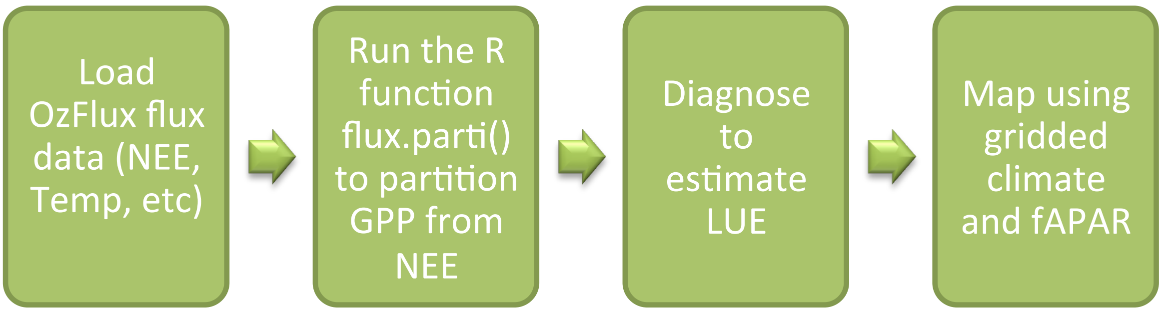
2. Aims and Objectives

The primary aim is to produce a software system capable of generating estimates of GPP and capturing the ecosystem variables necessary to determine light use efficiency (LUE). A key objective behind this aim is to make this an R-Project package.

The second aim is to model continental GPP by using the flux tower derived estimate of LUE, gridded climate and remotely sensed estimates of the fraction of absorbed photosynthetically active radiation (fAPAR).

3. Workflow and Outputs

To achieve the first and second aims, the ePiSaT R package can be used exclusively. The entire project was developed using R. The workflow is as follows; performing each of these steps is explained hereafter.



4. The Software system

The software system is a set of R-Project functions intended to be used as an R-Project package, as-is commonly practiced in the target community: ecosystem scientists. In due course the software system will be published as a stand-alone R-Project package published on CRAN. The current version of the ePiSaT software system contains in-package help available at the R command line. This document can be used as "How To" vignette designed to expose the workflow for using this package. As such, the presently prescribed way to use this software is to work through the examples provided in this document from each relevant topic.

This approach is necessary because some of the key inputs necessary for the models operation have not yet been released into the public domain and it is therefore likely these sections will change in the very near future. For example, reading and formatting of the fAPAR used and described in Section 4.8. Addressing this, a comprehensive example function for performing this step is provided and can be easily implemented by the user. If these data were distributed in the public domain, for example on the eMAST RDSI THREDDS Server planed for launch late 2013, then functions to connect, extract and use the data within the package could have been implemented in ePiSaT today.

4.1 Get started: Set up the R-Project framework

The R-Project framework can be downloaded from http://www.r-project.org/

The R-Project website contains additional information on software including information about the foundation, community and how to install the software and packages, such as *zoo, xts* (time series functionality) and *raster*, *ncdf, and sp* (spatial data functionality) and optim and minipak.lm (curve fitting / regression modelling) rather the re-writing the base level data manipulation functionalities these R-Packages provide. The package sirad, was adapted and integrated into this project. These functions are used to estimate radiation.

In addition to the R-Project framework, we recommend the use of an integrated development environment (IDE) such as R-Studio, which can be downloaded from: http://www.rstudio.com/

Once you have downloaded and installed the R-Framework base, running either the default R console or the RStudio version will produce a "command line" welcome message like this one;

```
R version 2.15.1 (2012-06-22) -- "Roasted Marshmallows"
Copyright (C) 2012 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-apple-darwin9.8.0/x86_64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
```

```
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
```

You can now setup the R packages required to work with the ePiSaT software, these generally install automatically but it is prudent to install them prior and understand what they do. This can be done using features of the GUI you are using OR, by entering the following code at the command line.

```
> install.packages("ncdf", repos="http://R-Forge.R-project.org")
> install.packages("minipak.lm", repos="http://R-Forge.R-project.org")
> install.packages("sp", repos="http://R-Forge.R-project.org")
> install.packages("raster", repos="http://R-Forge.R-project.org")
> install.packages("zoo", repos="http://R-Forge.R-project.org")
> install.packages("xts", repos="http://R-Forge.R-project.org")
```

4.2 Downloading and understanding the software system

It is recommended that you take some time to look at the files that make up the software package. The software can be downloaded from the ePiSaT projects github repository.

https://github.com/belgaroo/ePiSaT-ecosystem-Production-in-Space-and-Time

Unpack the software system into a known location and browse the folder using your favourite software. This can also be done with R on the command line as follows;

```
> setwd("~/episat/")
> list.files(getwd())
[1] "const_GlobalVariables.R"   "const_OzFlux_PAR.R"        "const_OzFlux.R"
[4] "flux.gfpar.R"              "flux.gpp.R"                "flux.outfiles.R"
[7] "flux.parti.R"              "flux.recthyp.R"            "flux.rss.R"
[10] "sirad.corrEarthSunDist.R"  "sirad.dayLength.R"         "sirad.daylightTimeFactor.R"
[13] "sirad.dayOfYear.R"         "sirad.degrees.R"           "sirad.exd.R"
[16] "sirad.exh.R"               "sirad.extrat.R"            "sirad.radians.R"
[19] "sirad.solarDecl.R"         "sirad.solarZenithAngle.R"
```

You will notice a number of files of different types and that there is some pattern in their naming. There are files appended *.R, which are R language script files and others appended *.dat that contain example datasets for use with this document. There are several classes of *.R script files, so these have been prefixed. Files prefixed with flux.* are used for processing the flux data. Files prefixed sirad.* are used by other functions in the package. An additional class of files, the constants, are prefixed const_* and usually used in a number of functions or workflow scripts.

4.3 Installing the software

Once downloaded, you can install ePiSaT using the following terminal command (blue indicates the use of a linux/ unix terminal example, in this case, run on Mac OS 10x.

```
$ R CMD INSTALL episat_1.0.tar.gz
* installing to library ‘/Library/Frameworks/R.framework/Versions/3.0/Resources/library’
* installing *source* package ‘episat’ ...
** R
** preparing package for lazy loading
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded
* DONE (episat)
```

Alternatively, all users and Windows users can install from an R GUI go the package manager and follow the prompts to install a package from source. Once successfully installed, the package can be loaded as follows;

```
> library(episat)
Loading required package: zoo

Attaching package: ‘zoo’

The following object is masked from ‘package:base’:

    as.Date, as.Date.numeric

Loading required package: xts
Loading required package: raster
Loading required package: sp
Loading required package: ncdf
```

Once installed. Users can look at the code of each function at the R command line by simply typing its name. Here is an example of the flux.recthyp function:

```
> flux.recthyp
function (X, B)
{
    B[1] + B[2] * (X/(B[3] + X))
}
<environment: namespace:episat>
```

4.4 Working with OzFlux Flux tower data

To use the software you will need tower data from a 3rd Party such as OzFlux or Fluxnet. Provided with this package is a sample from the Howard Springs fluxtower near Darwin in Australia’s Northern Territory. This sample dataset is a subset of the complete dataset which can be obtained from the OzFlux website (http://ozflux.org.au/).

4.4.1 Read the Howard Springs data

The Howard Springs flux tower data can be downloaded separate from the github repository described in section 4.2. The data is stored as an R data.frame() and running the following command will load the data into the workspace.

```
> data=load("~/Dropbox/GePiSaT/ePiSaT-code/package/HowardSprings.Rdata")
```

Once the HowardSprings data.frame has been loaded, it can be used to run the ePiSaT system. First, take a minute to look at the columns of data provided in the sample dataset.

```
> names(HowardSprings)
 [1] "TimeStamp" "JDAY"      "UST"       "TA"        "FC"        "FC2"       "RH"        "SWin"
 [9] "Day"       "Month"     "Year"      "Hour"      "Minute"    "PAR"
```

NOTE: R syntax convention substitutes <- for = We use the = to make the code more easily understandable.

Should you wish to run ePiSaT on a subset of the data, this data can be easily subsetted. The following code demonstrates how this is done in at various stages in the ePiSaT software.

```
> howard.2010 = subset(HowardSprings, Year==2010)
```

4.5 Running the flux partitioning system, flux.parti.R

The ePiSaT flux partitioning system can be ran with a single line of code. To execute the flux.parti() function, simply run the following at the R command line;

```
> flux.parti("HowardSprings",  "~/ePiSaT-out", "test")
```

The flux.parti() function takes three parameters, the name of the flux data.frame to be partitioned e.g. “HowardSprings”. This data frame has to be loaded in the workspace, as per section 4.4. Should you wish to run ePiSaT with a different set of data, for example, the howard.2010 data.frame created in section 4.4, simply change the first parameter. The second key input is the folder that captures the output of the flux.parti() function. The third input creates an output folder within this directory. When ran, flux.parti() creates a sub directory structure. To view this structure, use the following command in the R command line.

```
> list.files("~/ePiSaT-out/test/")
[1] "dd"  "hh"  "mm"  "pdf"
```

The folder “dd” contains the daily output of the flux.parti(). This folder will contain a comma separated file (CSV) containing daily estimates of GPP and the ecosystem variables flux.parti() calculated to make this estimation. The folder “hh” contains the half hourly version of these data. The folder “mm” contains the monthly aggregation of these data. Finally, the “pdf” folder contains an image of each of the monthly rectangular hyperbola model regressions, a visualisation of flux-parti(). Figure 1. Below shows an example of this.

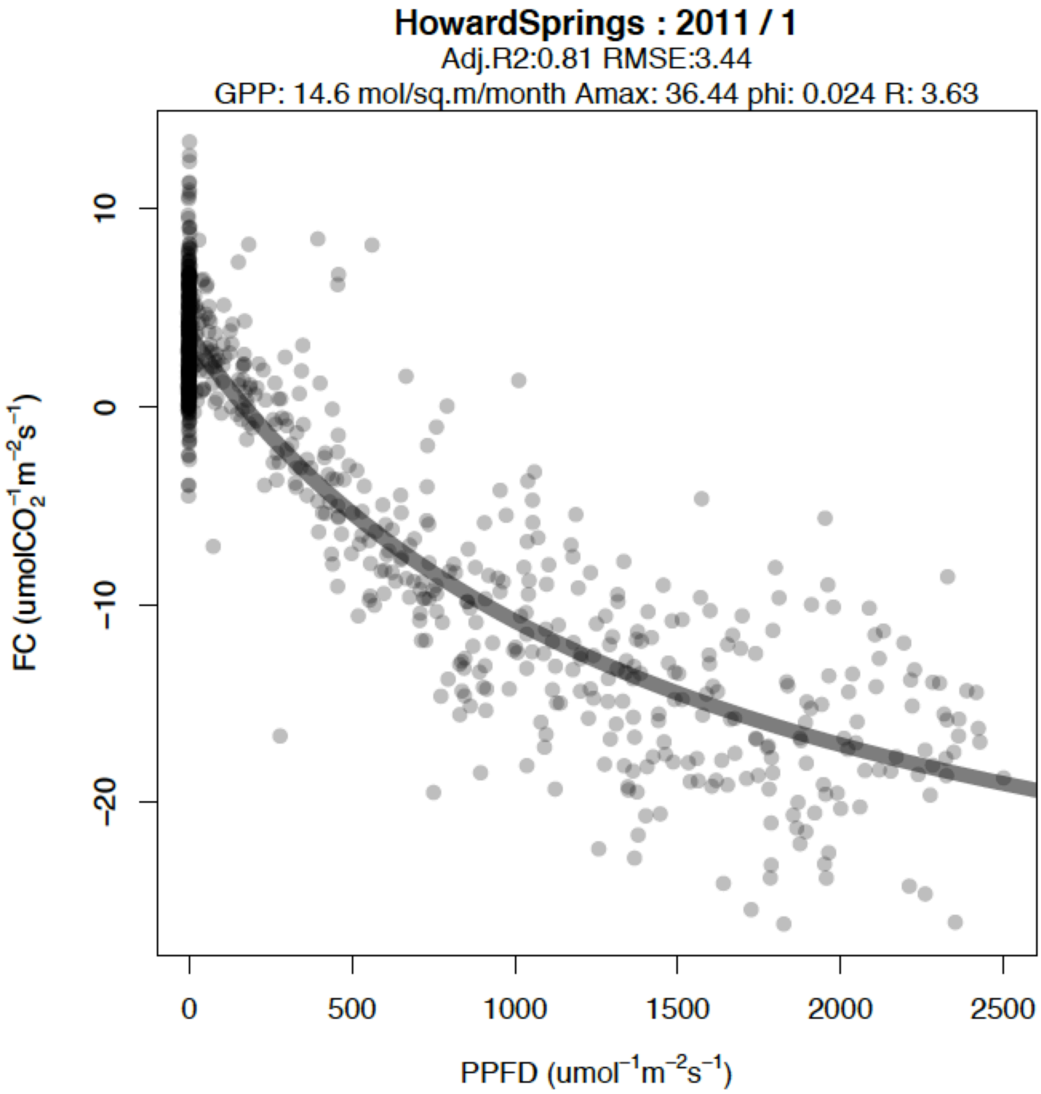


Figure 1. An example of the flux.parti() functions rectangular hyperbola curve fit of eddy flux (FC) and PAR (PPFD)

4.6 Gap filling of PAR in the function flux.gfpar()

Because observations are often incomplete, filling of gaps is necessary to produce a complete estimate of the PAR exposure for any given month. The gap filled PAR is used to generate GPP. Users do not need to run flux.gfpar() directly, the function is executed from within flux.parti(), nonetheless, independent execution is possible. Because gaps vary in frequency and duration, the function flux.gfpar() first filters the gaps, then fills them using linear interpolation if the gaps are small enough. If the gaps are larger, a backup 'PAR Climatology' has been provided for each of the OzFlux sites. These climatologies were generated at each OzFlux site from all available flux data and summarized as a monthly average total PAR. Larger gaps are filled at the half hourly time step. This is achieved by estimating daily extra-terrestrial solar radiation using a set of functions which estimate extra-terrestrial radiation. These functions are prefixed with "sirad" and found in the ePiSaT code. The synthetic, idealized solar radiation creates a diurnal bell curve. This curve is then scaled to local conditions using the PAR climate and the gaps are filled. The result is a quasi-observational idealised solar radiation gap filling.

4.7 Calculation of Gross Primary Production

The flux.parti() function executes the flux.gpp() function which performs the half hourly estimation of GPP and makes the daily and monthly aggregates and writes them to file. Because flux.gpp() is integrated with the flux.parti() inputs and outputs, and requires the regressed estimates of ecosystem variables as input parameters, it is not possible to execute flux.gpp() independently. Nonetheless, understanding the flux.gpp() function is a core part of the ePiSaT system. The line of code that estimates the GPP has the general form;

GPP = A_{max}* PPFD/ (PPFD + C)

Where A_{max} (maximum rate of carbon assimilation) and C (a rectangular hyperbola function) are estimated by the flux.parti() regression functions and available in the output files and PPFD is gap filled PAR outputted by the flux.gfpar() function.

4.8 Diagnosis of Light Use Efficiency

In order to estimate light use efficiency from flux tower data one can look at the slope of the regression between the estimate of GPP and the product of PAR and fAPAR. Exactly, how this is done depends on the users GPP model. The code to do this has not been distributed with the R package because the ePiSaT analysis determined that a light use efficiency of 0.02 was found consistently across the OzFlux network. Nonetheless, the following function can be used to replicate this step in the process. It requires the extraction and preparation of a file containing monthly estimates of fAPAR. The script reads in ecosystem variables (generated by flux.parti()), then fuses them with fAPAR, aggregates as required, the calculates and plots the GPP~LUE relationship.

```
plot.gpp=function(site.id,tower.path,tower.file,rs.path,rs.file,a.path, a.file, site,outp,type){
  # Read in the model output
  data=read.csv(paste(tower.path,tower.file, sep=""), header=T, sep=delim, dec=dec,
stringsAsFactors=FALSE)
  dat.site=data.frame(cbind(year=as.numeric(data$year),
month=as.numeric(data$month),GPP=as.numeric(data$GPP.mol),
PPFD=as.numeric(data$gf.PPFD.mol),

Tmean=as.numeric(data$Tmean),Tmin=as.numeric(data$Tmin),Tmax=as.numeric(data$Tmax),
                        Trange=as.numeric(data$Trange)))
  dat.sm=data.frame(cbind(year=dat.site$year, month=dat.site$month,GPP=dat.site$GPP,
PPFD=dat.site$PPFD,
                       Tmean=dat.site$Tmean,Tmin=dat.site$Tmin,Tmax=dat.site$Tmax,
Trange=dat.site$Trange))
  # Read in the fAPAR
  fpar=read.table(paste(rs.path,rs.file, sep=""),
                header=T, sep=",")
  fpar.site=subset(fpar, site==OzFlux$site[site.id])
  fpar.mean=aggregate(. ~ year+month, data = fpar.site, FUN=mean)
  fpar.dat=merge(dat.sm, fpar.mean, by= intersect(names(dat.sm), names(fpar.mean)))
  dat=data.frame(cbind(year=dat.sm$year, month=dat.sm$month,GPP=dat.sm$GPP,
PPFD=dat.sm$PPFD, Tmean=dat.sm$Tmean,Tmin=dat.sm$Tmin,Tmax=dat.sm$Tmax,
Trange=dat.sm$Trange, fapar=fpar.dat$value))
  #
  x=dat$fapar.ppfd
  y=dat$GPP
```

```
  lmod= lm( I(y-0) ~ I(x-0) + 0)
  if (type=="jpeg"){
    jpeg(filename = paste(outp,"lue_monthly.jpg", sep="" ),
         width = 480, height = 600, units = "px", pointsize = 12,
         quality = 100,
         bg = "white")
  }
  if (type=="pdf"){
    pdf(file=paste(outp,"lue_monthly.pdf", sep="" ), paper="a4r")
  }
  par(xaxs="i", yaxs="i")
  main.text=paste(site, ": GPP vs. fAPAR.PPFD ") # Slope=",round(coef(lmod)[[1]],4), " : RSE : ",
round(coef(summary(lmod))[, "Std. Error"],4))
  with(dat, plot(x,y, xlim=c(0,max(x)), ylim=c(0, max(y, na.rm=T)),
               xlab=expression(paste("fAPAR.PPFD (","mol m"^{-2}," month"^{-1},")",sep="")),
               ylab=expression(paste("GPP (mol m"^{-2}," month"^{-1},")",sep="")) , #main=main.text,
               pch=c('j','f','m','a','M','J','u','A','s','o','n','d')[month]))
  #
  mtext("j=jan f=feb m=mar a=apr M=may J=jun u=jul A=aug s=sep o=oct n=nov d=dec ", side=3,
line=0)
  mtext(paste("LUE=",round(coef(lmod)[[1]],4), "RSE : ", round(coef(summary(lmod))[, "Std. Error"],4)),
side=3, line=1)
  mtext(main.text, side=3, line = 3, font=2)
  abline(lmod, lwd=3)
  yhat <- as.data.frame(predict(lmod, interval = "confidence", level=0.95))
  lines(x[order(x)], yhat$lwr[order(x)], lty=3)
  lines(x[order(x)], yhat$upr[order(x)], lty=3)
  legend("topleft",c("slope","95% conf. int. "), lty=c(1,3), lwd=c(3,1))
  dev.off()
}
```

4.9 National Gridded Gross Primary Production

The national grids of GPP are based on fAPAR data provided by ePiSaT team (Huete, Restro-Coupe and Davies) from the University of Technology Sydney and climate data (Radiation, Temperature, Vapour Pressure Deficit) from the CSIRO led Australian Water Availability Project. The light use efficiency model has the general form;

GPP = fAPAR.PAR.LUE

Where fAPAR and PAR is gridded and LUE is derived using the ePiSaT system against the OzFlux network. As per the diagnosis of LUE, given the form of this depends heavily on the users desired model, a function performing this step was not included. The following function illustrates how simple this step can be performed in R.

```
calc.gpp<-function(fapar, par, lue){
        gpp=fapar*par*lue
        return(gpp)
}
```

Thus, calculation of the national grids becomes a simple matrix calculation performed with a single line of code. This assumes the gridded data is (a) on the same grid and the (b) there is a constant estimate of LUE. The ePiSaT team assessed all available OzFlux data and for the HowardSprings data provided here, the LUE can be fixed at 0.02 (unitless). Thus the code to generate the national estimate of GPP is simply as per the general equation. At 0.05 degree resolution and on a late model machine this calculation can be made in seconds. Once generate, the national maps of GPP can be visualised as shown in Figure 2.

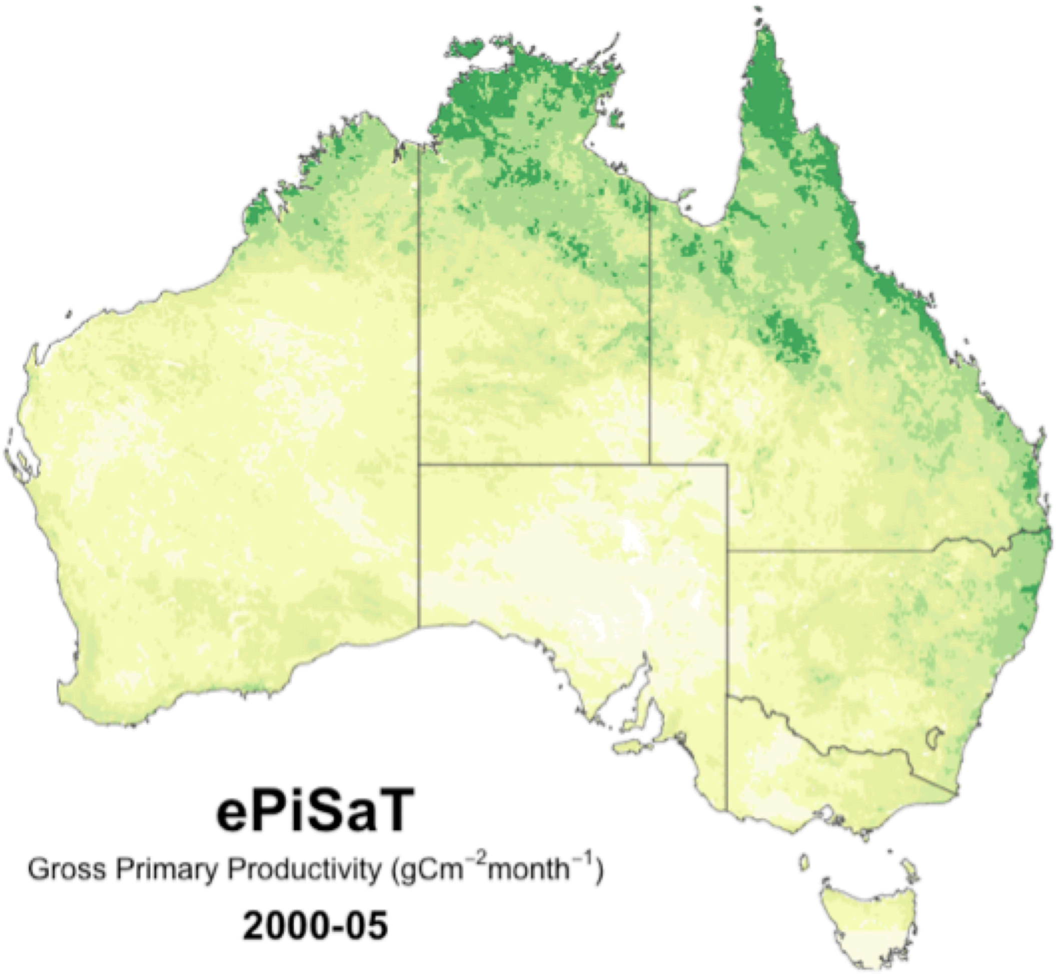


Figure 2. The ePiSaT national grid of Gross Primary Productivity

5. Software Distribution and Maintenance

The ANDS funded ePiSaT software will be distributed via an online service github. This service was selected because it has strong penetration within the Terrestrial Ecosystem Research Network (TERN) ecosystem Modelling and Scaling infrastrucTure (eMAST) community. Presently, the ePiSaT code has been published as described in section 4.2.

The software continues to be developed, maintained and distributed by TERN eMAST under the leadership of Professor Colin Prentice and Dr Bradley Evans. The software will eventually be registered with CRAN.

5.1 Revisions and Version control

ePiSaT software revisions are documented by the github system "commits". Details of the revisions of the ePiSaT software can be found here: https://github.com/belgaroo/ePiSaT-ecosystem-Production-in-Space-and-Time/commits/master

5.2 Software Testing Approach and Strategy

The ePiSaT software-testing framework aims to satisfy the following criteria;

1. Test examples should be easy to use and evaluate performance.
2. The test protocol should be simple and easy to understand.
3. The code should be modular and capable of being tested in a modular way.
4. The system should recognize a balance between usability, adaptability and robustness.

During the software system development phase a detailed review of the software was established through Professor Colin Prentice through his establishment of the Global ecosystem Production in Space and Time (GePiSaT). Evans provided the spin-off GePiSaT team Dr Tyler Davis and Xavier Gilbert (PhD candidate), with a training and workshop on a beta version of the ePiSaT code. This collaborative test and review resulted in a comprehensive re-write of the ePiSaT code. The new code was written to be R Coding Conventions (RCC) compliant.

Finally, the first version of the ePiSaT project (as found on github) was generalized and abstracted to meet the four criteria described above. Test examples were made as easy to use as possible and integrated both in this document and the help provided with the package. Complex functions and workflows were simplified such that the flux.parti() function (the core function of the ePiSaT software) can be ran with a single line of code. The complex functions are called by flux.parti(), namely flux.gfpar() and flux.gpp() are made available as separate functions for advanced users. This approach is key to addressing points 2-4 of the framework. This modular approach makes it capable to test these functions independently yet keeps the system usable to as wide a range of possible users. Accordingly, the provision of everything the user needs to replicate the fitting of the OzFlux network of sites (data available on request) makes the system highly adaptable and scientifically robust.

6. Glossary of Terms

Term	Definition
ACEAS	Australian Centre for Ecological Analysis and Synthesis: a TERN facility dedicated to encouraging joint data synthesis and analysis activities by different communities.
Autotrophic respiration	The release of CO_2 from plants, as compounds produced by photosynthesis are broken down again to yield energy. About half of GPP is converted back to CO_2 by this process.
AVHRR	NASA's Advanced Very High Resolution Radiometer, the first satellite-based instrument to record land-surface "greenness".
Carbon balance	The difference between GPP and CO_2 losses due to autotrophic and heterotrophic respiration, with a small additional loss due to fires. If a region has a positive carbon balance it means that it is taking up more atmospheric CO_2 than it is releasing, i.e. it is acting as a carbon sink.
Carbon Cycle Data Assimilation System (CCDAS)	A published software infrastructure that uses automatic

	differentiation and Bayesian inversion to optimize the predictive power of a couple ecosystem process and atmospheric transport model, using CO_2 concentration data as constraints.
Ecosystem Modelling And Scaling infrasTructure (eMAST)	A TERN facility led by Colin Prentice. eMAST is developing pathways for the flow of information between ecosystem models and different sources of ecosystem data with a view to efficiently and reliably "scaling up" from site measurements to the region or continent, and "scaling down" from large-scale information (e.g. climate model results) to provide data of relevance to natural resource managers.
Eddy covariance	A technique relying on fast-response micrometeorological instruments mounted on a tower or crane above the vegetation canopy, to measure fluxes of CO_2, latent and sensible heat between the atmosphere and the land over a ~ 1 km^2 area.
Evapotranspiration	The total loss of water from the land surface by transpiration (transport of water through plants to the atmosphere), interception (evaporation from wet plant surfaces) and evaporation from bare ground.
Fraction of absorbed photosynthetically active radiation (fAPAR)	The fraction of PAR that is absorbed by green plants. fAPAR is measured from space based on the relative reflectance of the land surface in different wavebands. Absorbed PAR (APAR) is the product of fAPAR and the total PAR incident on the land surface.
Gross primary production (GPP)	The daily. monthly or annual uptake of CO_2 from the atmosphere by photosynthesis and its conversion to organic compounds.
Heterotrophic respiration	The release of CO_2 from soils by the action of microbes and fungi on dead organic matter, originally produced by photosynthesis.
Hydrograph	A gauge that measures the rate of mass flow of water in a stream or river.
International Land Atmosphere Benchmarking (iLAMB)	An international project to improve the reliability of large-scale terrestrial ecosystem models by promoting systematic comparisons of model results with observations.
Light use efficiency model	A proportionality between GPP and absorbed PAR. The constant of proportionality may vary e.g. with temperature, water availability and CO_2 concentration. This type of model has its origin in the classic work on crop growth by John Monteith in the 1970s. Although the relationship between photosynthesis measured over a short period and PAR intensity is asymptotic, there is overwhelming evidence that the long-term relationship is linear; and there is now a good theoretical understanding of why that should be so.
MODIS	Moderate Resolution Imaging Spectroradiometer: an instrument mounted on NASA satellites which records the reflectance of the Earth at different wavebands.
NASA	The United States National Aeronautic and Space Administration.
NOAA	The United States National Oceanic and Atmospheric Administration.
Photosynthetically active radiation (PAR)	The part of the Sun's radiation that can be used by plants for photosynthesis (about 50%). The rest is infrared radiation.
Runoff	The flow of water from the land surface into streams and

	rivers. Over periods long enough for soil moisture changes to be averaged out (a year or longer), runoff is approximately equal to the balance of precipitation and evapotranspiration.
SIO	The Scripps Institute of Oceanography, La Jolla, California, which pioneered precise atmospheric CO_2 measurements and continues to operate a number of measurements stations including two in Australasia.
Terrestrial Ecosystem Research Network (TERN)	TERN is an initiative to unify ecosystem data from diverse sources, set up and implement common protocols for the collection of new data at various scales, and encourage a change in the culture of ecosystem science towards data sharing and synthesis.