Simulation and Parameter Estimation for Biomass Crops

Fernando E. Miguez
Department of Agronomy
Iowa State University

December 17, 2014

Abstract

Simulation and parameter estimation of photosynthesis and crop growth.

```
## Loading required package:  lattice
```

1 Introduction

The package BioCro started as a way to continue work on the ideas developed in the WIMOVAC model. WIMOVAC was developed by Stephen Long and Steve Humphries (http://www.life.illinois.edu/plantbio/wimovac/) and there have been several publications using this model. I have used the model for some initial efforts at modeling *Miscanthus* × *giganteus*.

2 Leaf-level Photosynthesis

There is a large range in the complexity of models that have been used to simulate photosynthesis. BioCro offers functions `c4photo` and `c3photo`. Both functions take as minimum input radiation (PAR $\mu mol\ m^{-2}\ s^{-1}$), temperature (Celsius) and relative humidity (0-1).

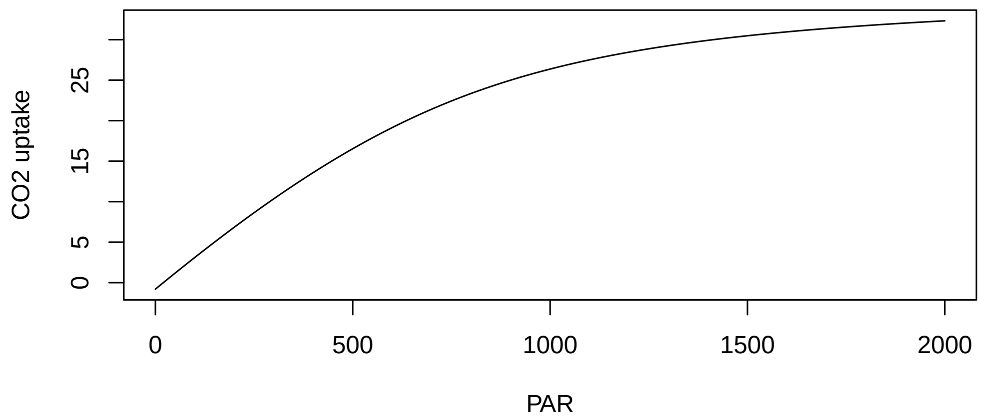
Since WIMOVAC originated as a photosynthesis model we can start with a simple example. For the C_4 photosynthesis model the best reference is Collatz et al. (1992).

```
c4photo(1500,25,0.7)

## $Gs
## [1] 277.9524
##
## $Assim
## [1] 30.48496
##
## $Ci
## [1] 204.517
```

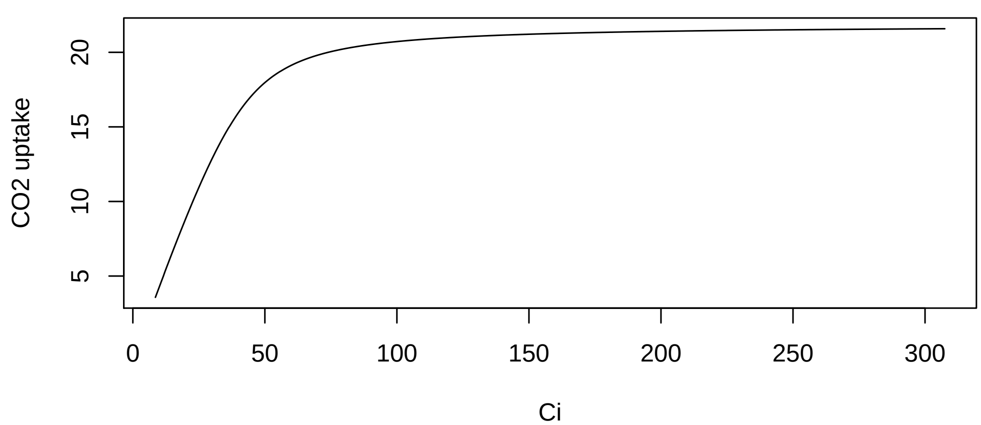
This example shows that with PAR 1500, temperature of 25 and relative humidity of 0.7 (70%) as inputs we get simulation of CO_2 assimilation (`Assim`), stomatal conductance (`Gs`) and the intercellular CO_2 (`Ci`). For units and other details see `?c4photo`. Another model available for simulating C4 photosynthesis is `eC4photo`, see docs for details. This model has not been used much. In `c4photo` the computation is carried out in compiled `C`, but there is a pure R function `c4photoR` which might be useful for understanding the calculations.

There is an equivalent function `c3photo` which is closely based on the c3 photosynthesis model described in WIMOVAC. Notice that the parameters and interpretation are different from the `c4photo` function.

```
pr <- seq(0,2000)
temp <- rep(25, length(pr))
rh <- rep(0.7, length(pr))
res <- c4photo(pr, temp, rh)
plot(pr, res$Assim, ylab="CO2 uptake",xlab="PAR",type='l')
```

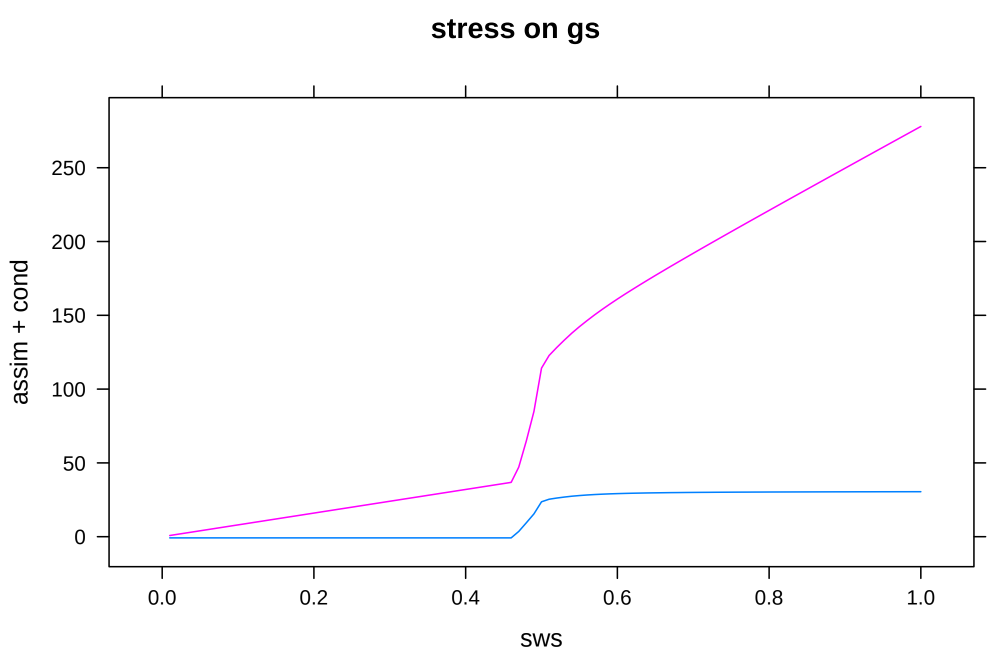



```r
Ca <- seq(15,500)
pr <- rep(1000, length(Ca))
temp <- rep(20, length(Ca))
rh <- rep(0.7, length(Ca))
res <- c4photo(pr,temp,rh,Catm=Ca)
plot(res$Ci, res$Assim, type = 'l', ylab = "CO2 uptake", xlab = "Ci")
```



Besides some of the typical parameters for both functions there is the option of including stress. The argument is `stress`. The stress can be applied to stomatal conductance (default) or V_{max}.

```
sws <- seq(0,1,0.01)

assim <- numeric(length(sws))
cond <- numeric(length(sws))
ws <- 'gs'

for(i in 1:length(sws)){

    assim[i] <- c4photo(1500, 25, 0.7, stress=sws[i],ws=ws)$Assim
    cond[i] <- c4photo(1500, 25, 0.7, stress=sws[i],ws=ws)$Gs

}

xyplot(assim + cond ~ sws, type='l', main="stress on gs")
```



The previous functions are relevant for leaf-level photosynthesis. Scaling up to the canopy level is not trivial since it requires developing a light macro environment which simulates the partitioning between direct and diffuse radiation (see function `lightME`). The function `sunML` is used to predict the proportion of light for each layer of a multiple layered canopy.

TODO

- include an example using `c3photo`
- Discuss meaning and relationship among parameters

3 Canopy Photosynthesis

The function `CanA` integrates the previous functions to simulate canopy CO_2 assimilation for a complete canopy. This function also simulates transpiration based on Penman-Monteith, Penman and Priestly.

The `CanA` function is designed to run at an hourly timestep. The inputs should all be of length 1. As with other canopy models the canopy is discretized in layers and each layer has unique conditions in terms of light levels, leaf temperature, wind and relative humidity. See `?CanA` for details.

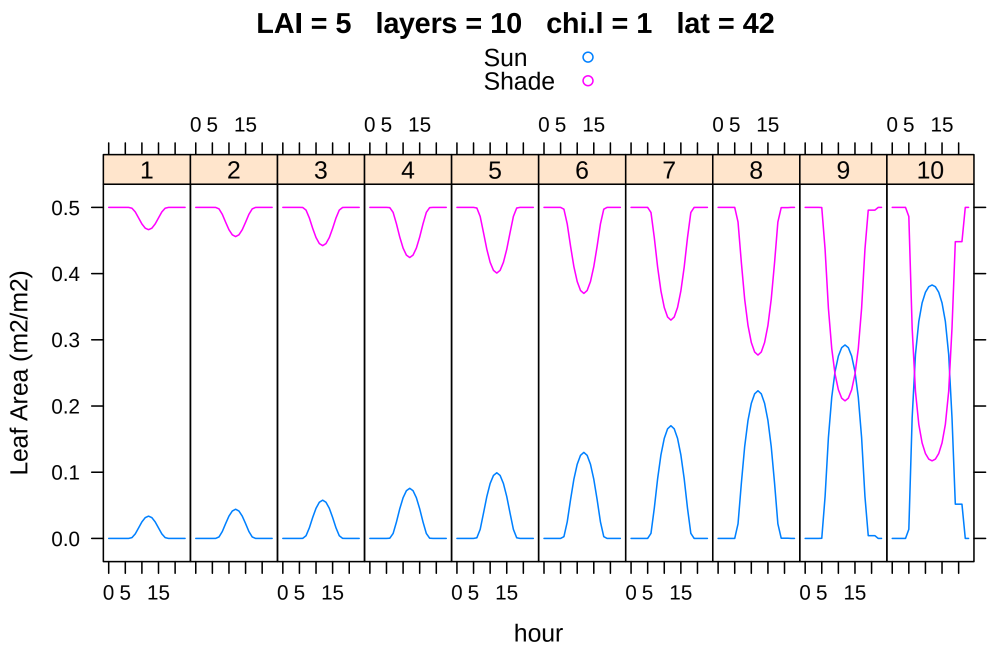
```
nlay <- 8
res <- CanA(lai=3, doy=200, hr=12, solar=1500, temp=25, rh=0.7,
            windspeed=2, nlayers=nlay)
```

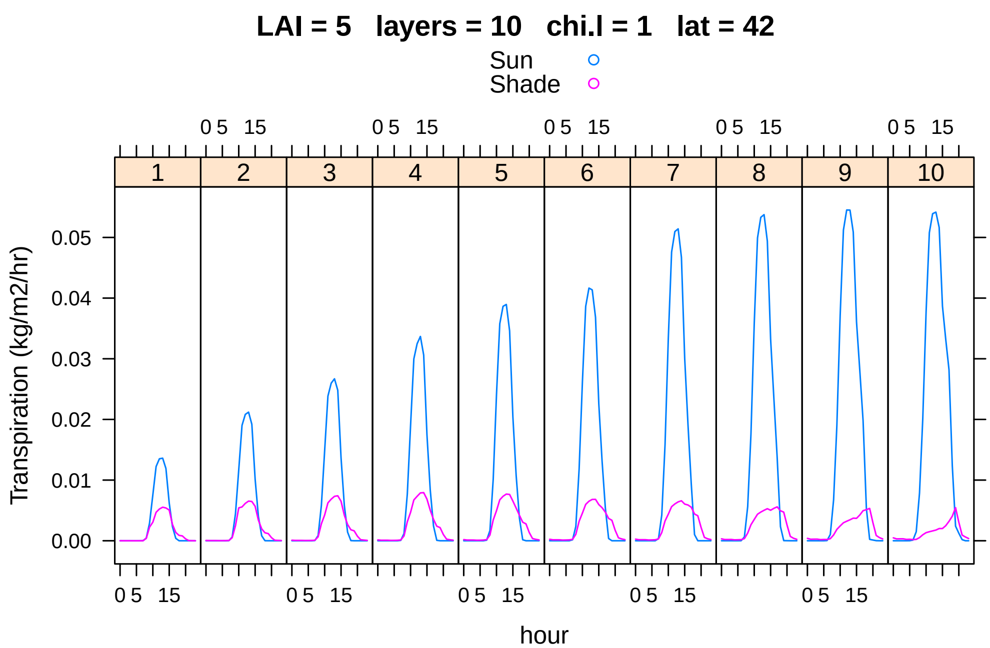
The distribution of leaves in the sun and in the shade is an important characteristic of a canopy and the architecture can be modified mainly by changing the `chi.l` parameter which represents the ratio of horizontal leaf projections to vertical leaf projections.

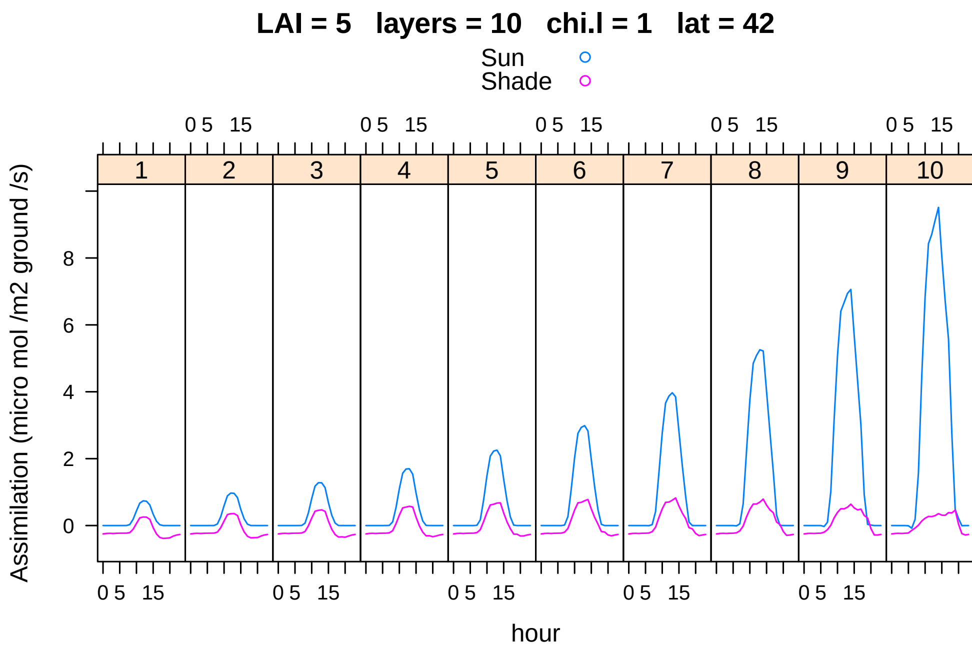
```
apply(res$LayMat[,3:4], 2, sum)

##   Leafsun Leafshade
##  1.354043  1.645957
```

In this example, 1.35 m^2 of leaf are in the sun and 1.65 of m^2 are in the shade for a total of 3 LAI.

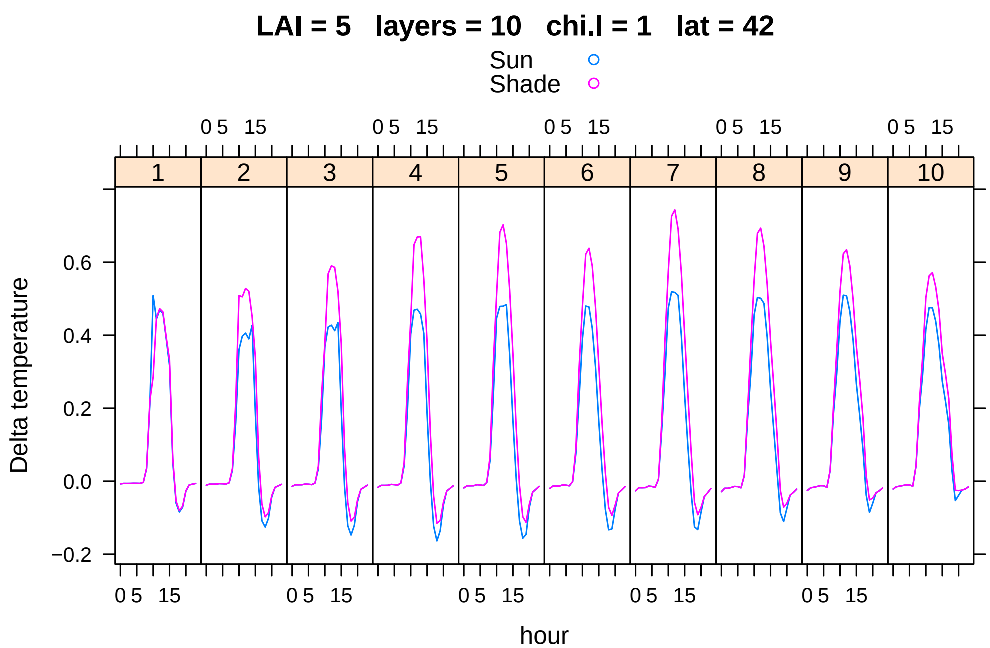
Next we can look in detail about the properties of the canopy by layer


LAI = 5 layers = 10 chi.l = 1 lat = 42
Sun
Shade
1
2
3
4
5
6
7
8
9
10
Leaf Area (m2/m2)
hour

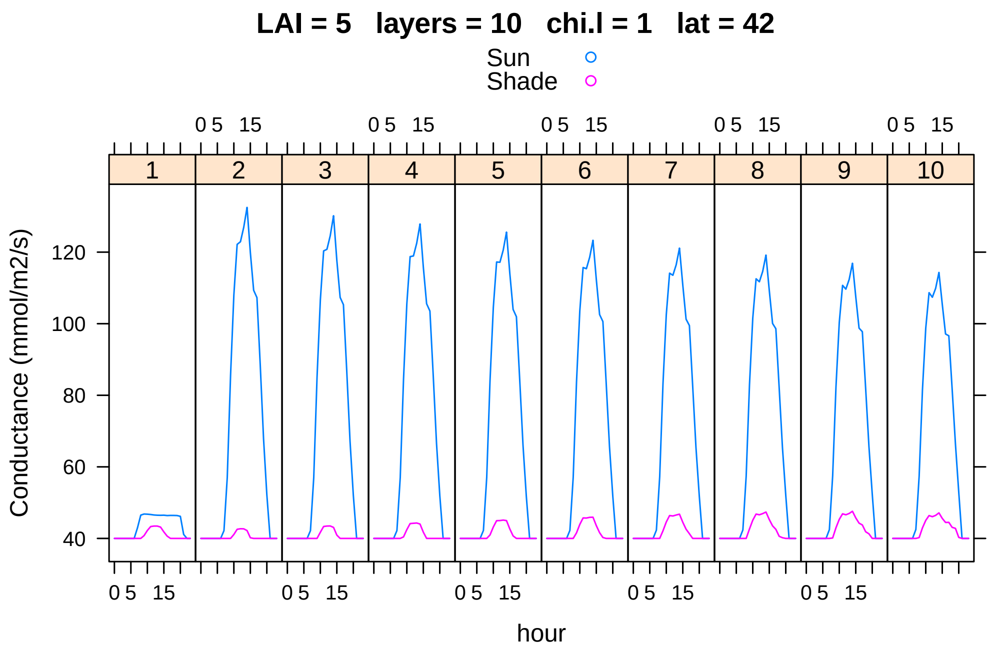

LAI = 5 layers = 10 chi.l = 1 lat = 42
Sun
Shade
1
2
3
4
5
6
7
8
9
10
Transpiration (kg/m2/hr)
hour

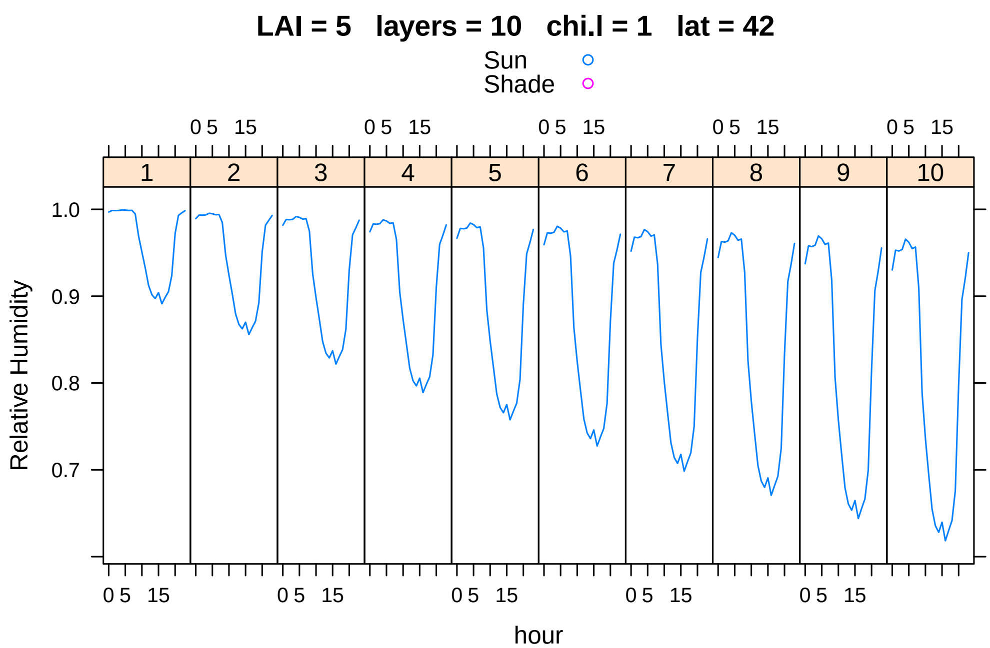


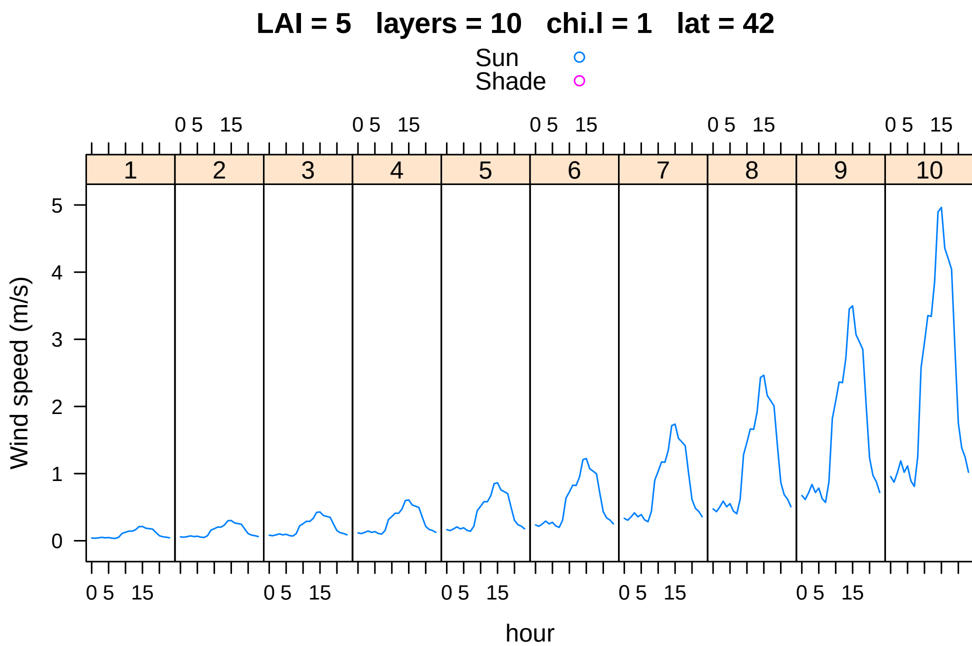
LAI = 5 layers = 10 chi.l = 1 lat = 42
Sun
Shade
Assimilation (micro mol /m2 ground /s)
hour
1
2
3
4
5
6
7
8
9
10

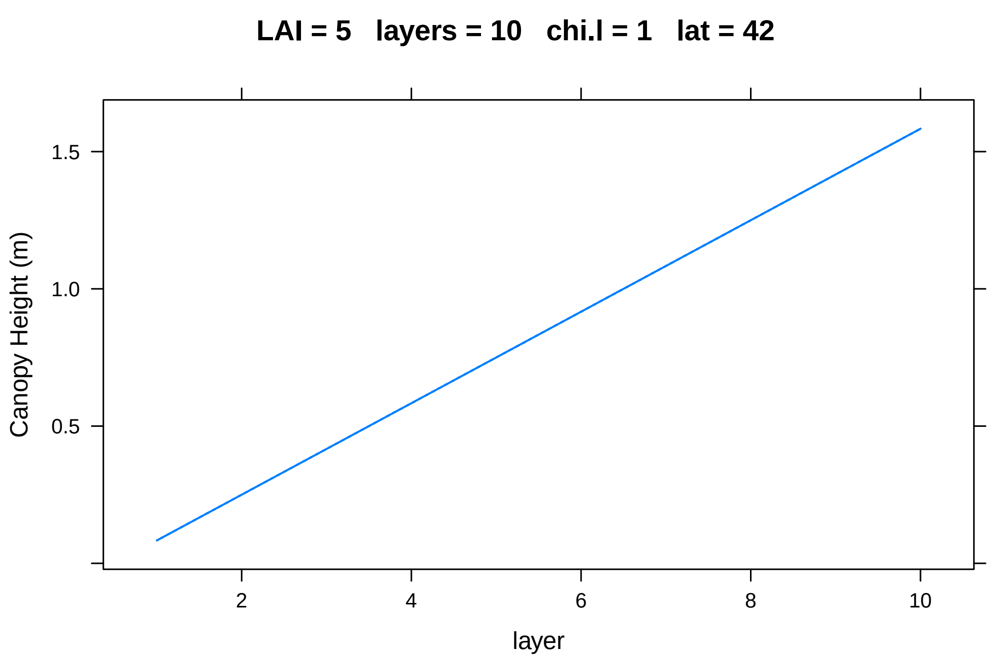


LAI = 5 layers = 10 chi.l = 1 lat = 42
Sun
Shade
Delta temperature
hour
1
2
3
4
5
6
7
8
9
10


LAI = 5 layers = 10 chi.l = 1 lat = 42
Sun
Shade
Conductance (mmol/m2/s)
hour
1
2
3
4
5
6
7
8
9
10


LAI = 5 layers = 10 chi.l = 1 lat = 42
Sun
Shade
Relative Humidity
hour
1
2
3
4
5
6
7
8
9
10





Some important assumption of the multi-layer canopy model are

- it distributes the LAI equally among layers, this is not necessarily a realistic assumption

- relative humidity increases with canopy depth which causes stomatal conductance to increase as well (I don't know what is going on in layer 1).

- by default photosynthetic parameters are constant in the profile but it is possible to make them depend on a profile of N concentration (see argument `lnControl`)

3.1 Parameters to adjust

Which parameters are relevant at the canopy level? Of course photosythetic parameters are important but these were discussed before so they are assume to be reasonable here. LAI is a very important input to this function so it is not really an adjustable parameter.

- `nlayers` The number of layers has an effect on many of the results. This can be modified by the user if there is a good rationale for doing it. It is possible that taller canopies benefit more from having multiple layers and shorter canopies benefit less.

- `kd` extinction coefficient for diffuse light. Although this can be calculated it is not at this point.

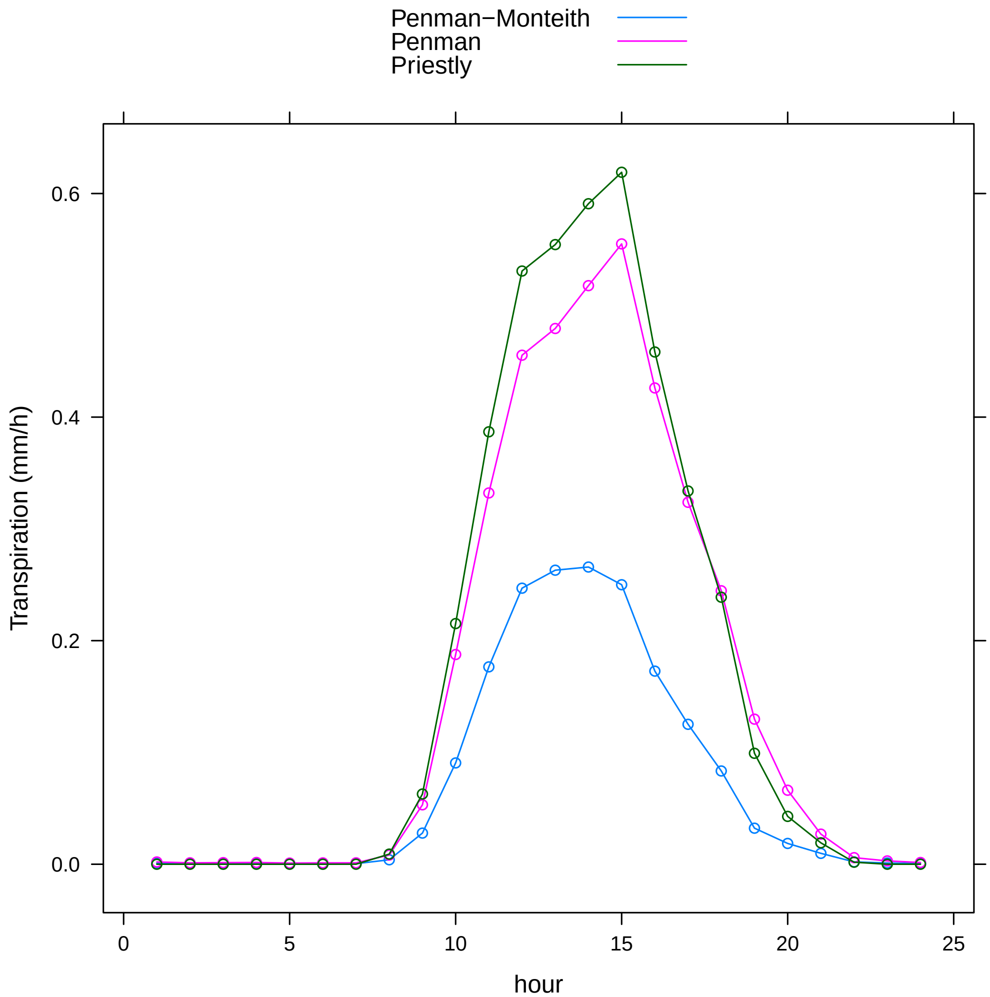
- `chi.l` is the ratio of horizontal to vertical projection of leaf area. Lower than 1 values for more erect canopies and less than 1 for canopies with higher proportion of flat leaves.

- `leafwidth` average leaf width. Today(12-12-2014) it does not affect any results. Fixme

- `heightFactor` factor relating LAI to height. Adjust it to match reasonable height for a crop.

3.2 Calculation of Transpiration

`CanA` simulates transpiration using Priestly (driven by solar radiation and temperature), Penman (adjusted for the aerodynamic component) and Penman-Monteith (adjusted for the aerodynamic plus stomatal component).

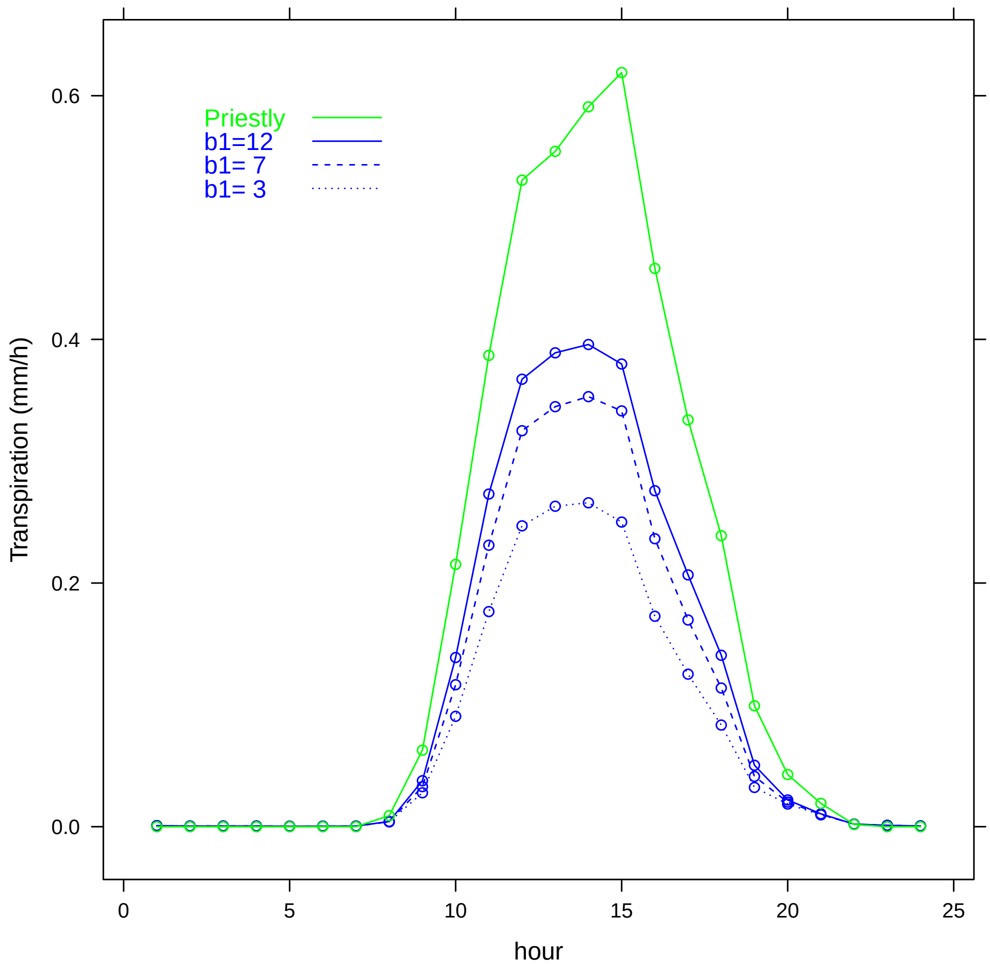



```
## CanopyTrans     TranPen  TranEpries
##    1.774365    3.825788    4.163035
```

The total transpiration for the day is estimated to be highest for the Priestly method, lowest for the Penman-Monteith and intermediate for Penman.

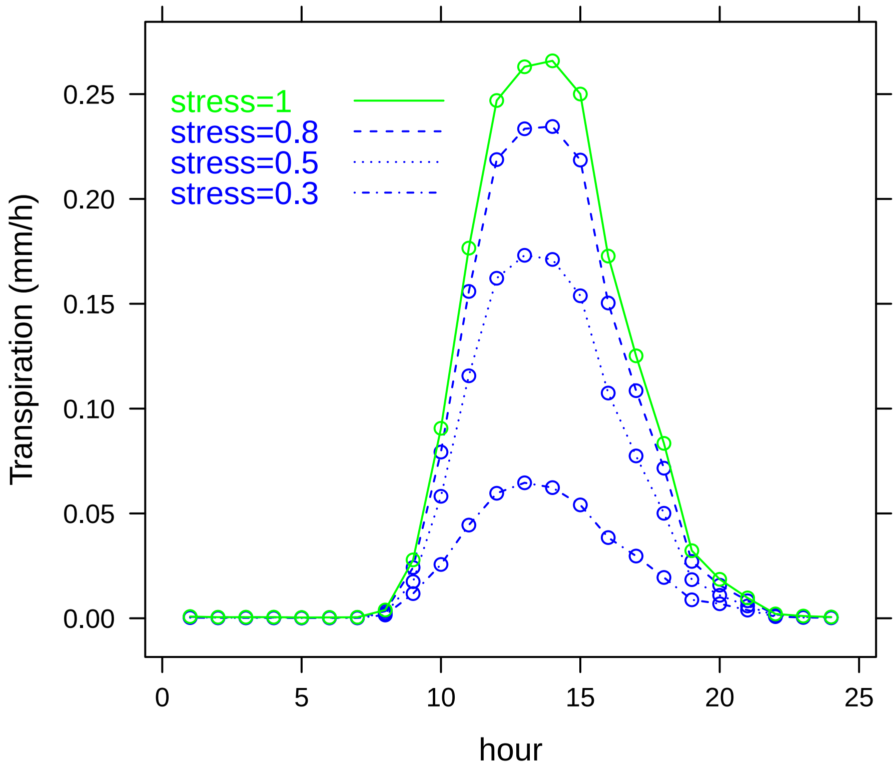
3.3 Effect of Ball-Berry slope parameter

The slope of the Ball-Berry model can have a significant effect on the results, but only for the Penman-Monteith method. The parameters for Ball-Berry however should be set from previous literature data or from analysis of gas exchange measurements. The purpose of this is to show that it has a large effect. It is supplied by the `photoParms` function. This increases transpiration in the Penman-Monteith model but only up to a point. Priestly and Penman are almost always higher than Penman-Monteith.

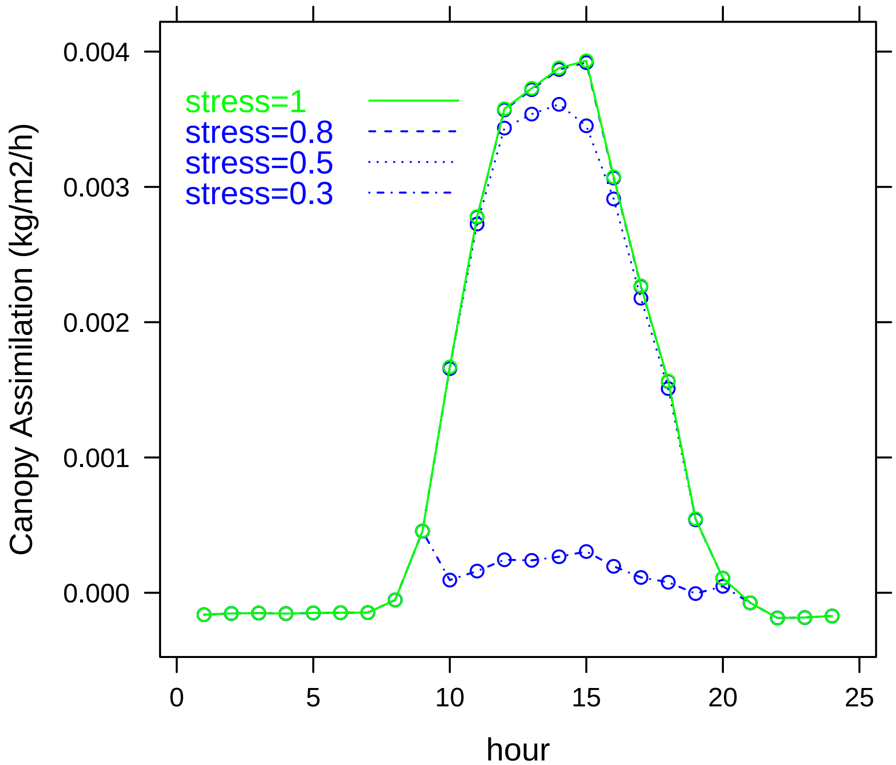


3.4 Effect of stress on diurnal transpiration

Another significant component that will affect transpiration during a day is the level of water stress the plant is experiencing.



stress=1
stress=0.8
stress=0.5
stress=0.3
Transpiration (mm/h)
0.00
0.05
0.10
0.15
0.20
0.25
0
5
10
15
20
25
hour



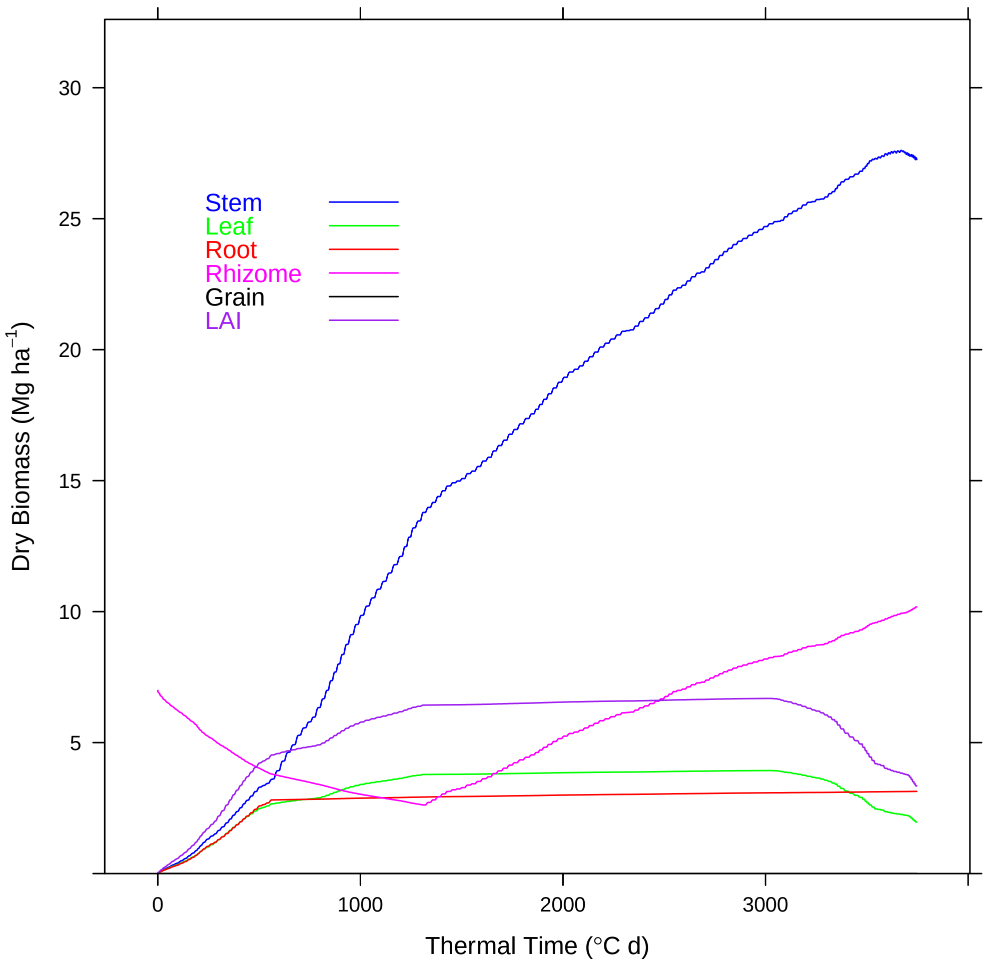
TODO

- Include an example in which I see the effect of canopy height on diurnal transpiration.
- Include an example in which I see the effect of changing chi.l

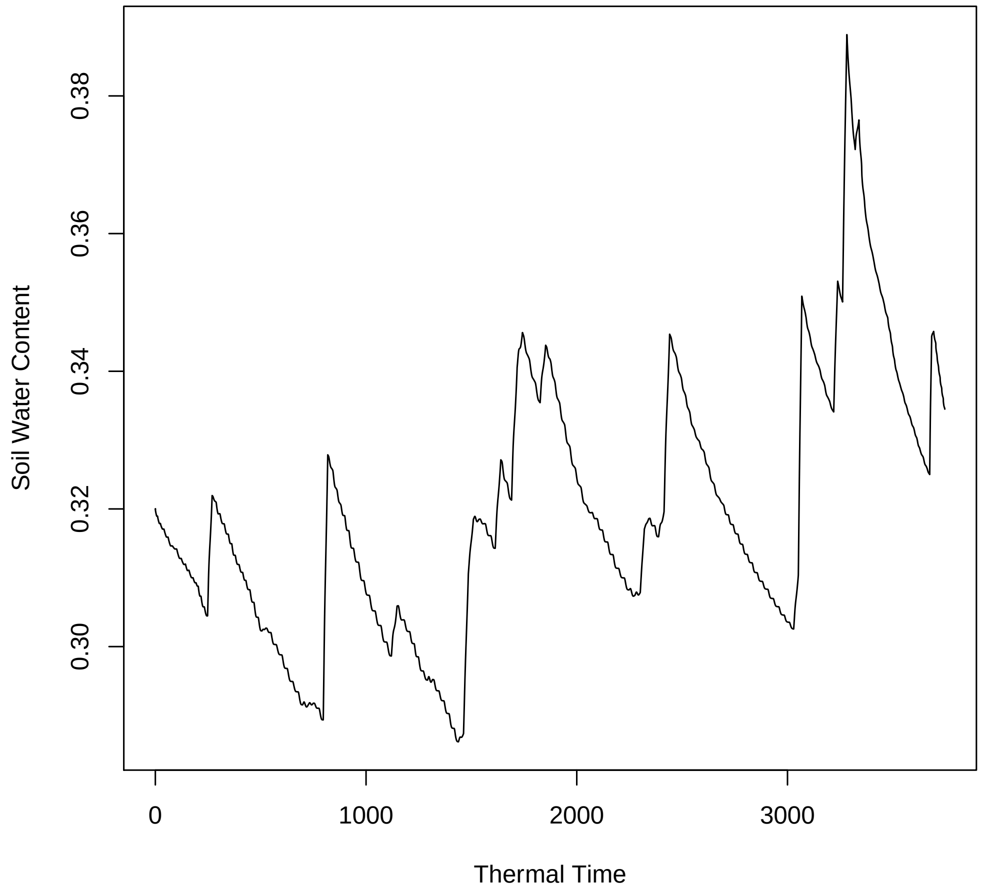
4 Biomass Crop Simulation

When the interest is to perform a simulation for a whole growing season, the function `BioGro` can be used. This function has as minimum input weather data for the whole year (365 days) at hourly time steps. The data can be generated using the `weach` function from daily data.

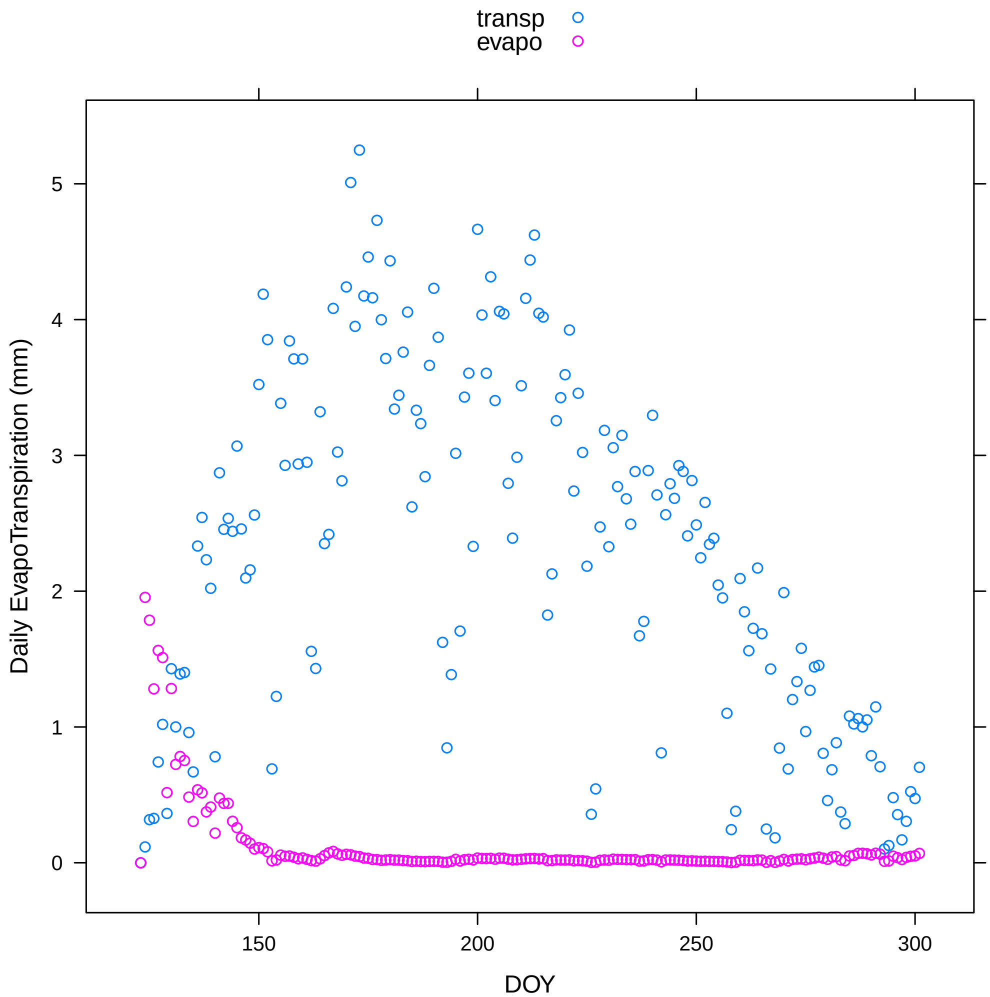
```r
data(cmi05)
summary(cmi05)

##       year           doy          hour          SolarR
##  Min.   :2005   Min.   :  1   Min.   : 0.00   Min.   :   0.0000
##  1st Qu.:2005   1st Qu.: 92   1st Qu.: 5.75   1st Qu.:   0.0000
##  Median :2005   Median :183   Median :11.50   Median :   0.1956
##  Mean   :2005   Mean   :183   Mean   :11.50   Mean   : 447.2008
##  3rd Qu.:2005   3rd Qu.:274   3rd Qu.:17.25   3rd Qu.: 851.8704
##  Max.   :2005   Max.   :365   Max.   :23.00   Max.   :2133.4800
##       Temp              RH               WS
##  Min.   :-18.333   Min.   :0.1450   Min.   :0.2013
##  1st Qu.:  2.431   1st Qu.:0.5673   1st Qu.:1.1069
##  Median : 12.362   Median :0.7516   Median :1.7521
##  Mean   : 11.963   Mean   :0.7116   Mean   :2.1208
##  3rd Qu.: 21.379   3rd Qu.:0.8820   3rd Qu.:2.7644
##  Max.   : 36.278   Max.   :0.9960   Max.   :9.9283
##      precip
##  Min.   :0.00000
##  1st Qu.:0.00000
##  Median :0.00000
##  Mean   :0.09119
##  3rd Qu.:0.01058
##  Max.   :1.94733

soilP <- soilParms(wsFun='linear')
res <- BioGro(cmi05, soilControl=soilP)
plot(res)
```



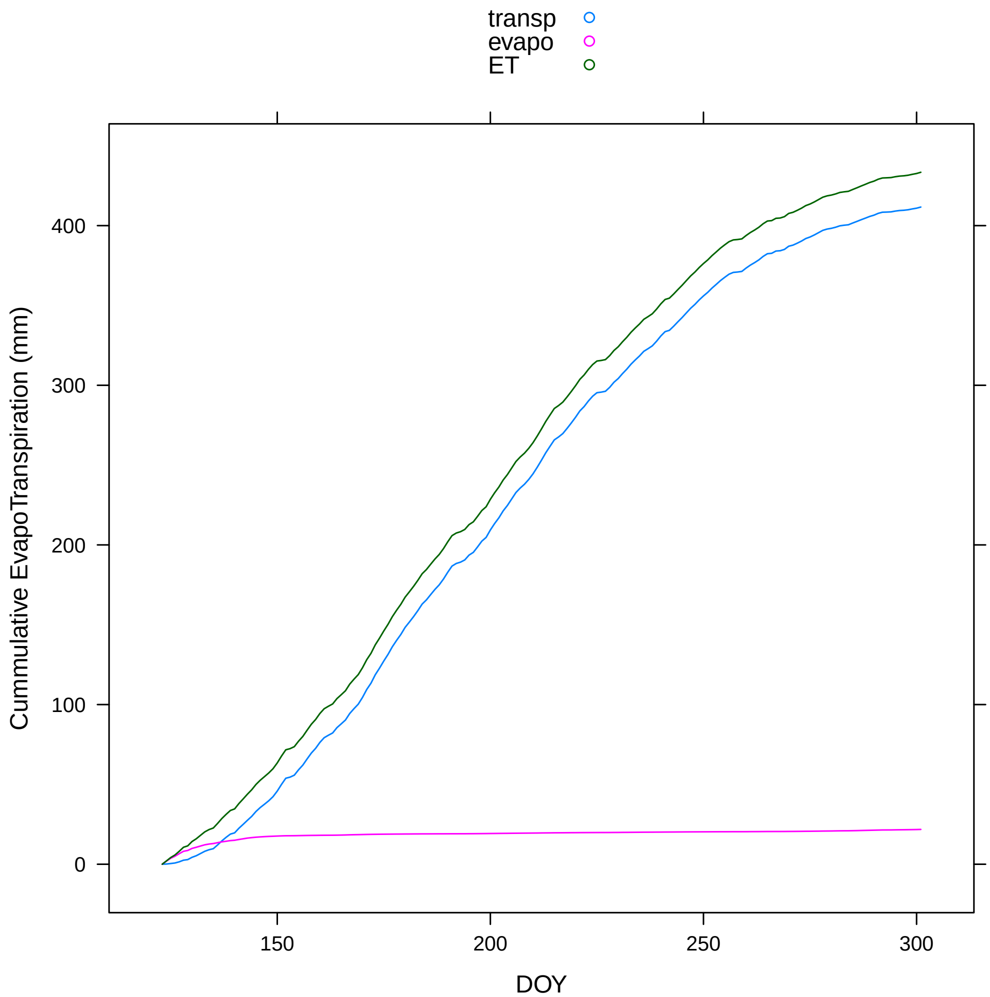

```
plot(res, plot.kind="SW")
```



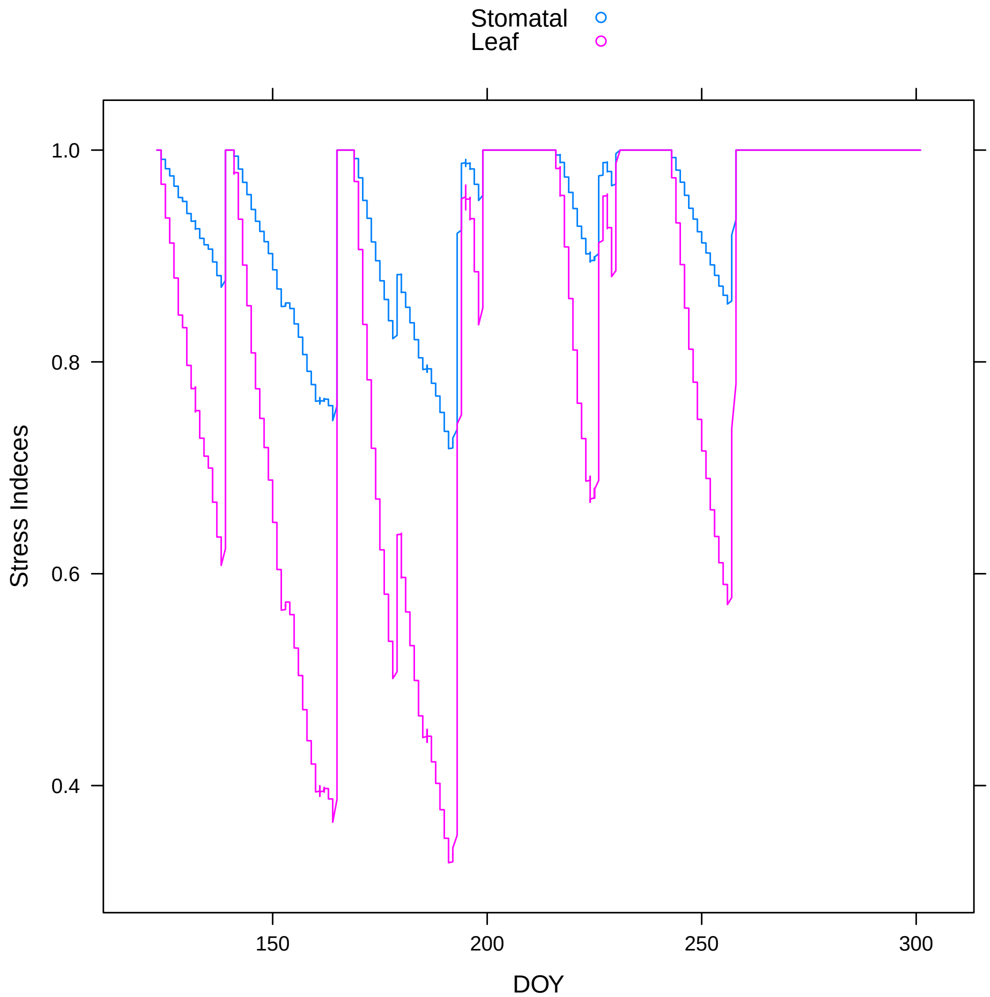

```
plot(res, plot.kind="ET")
```




```r
plot(res, plot.kind="cumET")
```




```
plot(res, plot.kind="stress")
```




```
names(res)

##  [1] "DayofYear"            "Hour"                 "CanopyAssim"
##  [4] "CanopyTrans"          "Leaf"                 "Stem"
##  [7] "Root"                 "Rhizome"              "Grain"
## [10] "LAI"                  "ThermalT"             "SoilWatCont"
## [13] "StomatalCondCoefs"    "LeafReductionCoefs"   "LeafNitrogen"
## [16] "AboveLitter"          "BelowLitter"          "VmaxVec"
## [19] "AlphaVec"             "SpVec"                "MinNitroVec"
## [22] "RespVec"              "SoilEvaporation"      "cwsMat"
```

```
## [25] "psimMat"                "rdMat"                 "SCpools"
## [28] "SNpools"                "LeafPsimVec"           "Drainage"
```

The last command `names(res)` shows the list of objects available for further manipulation. This code works fine on my desktop and two laptops but it fails in the R-forge build. Trying to find out why.

4.1 Soil information and parameters

Given that the model has been adequately described at the leaf and canopy level, when doing a simulation for the whole growing season the soil information becomes highly relevant. The basic information is supplied through the `soilParms` function.

```
soilP <- soilParms()
names(soilP)

##  [1] "FieldC"     "WiltP"      "phi1"       "phi2"
##  [5] "soilDepth"  "iWatCont"   "soilType"   "soilLayers"
##  [9] "soilDepths" "wsFun"      "scsf"       "transpRes"
## [13] "leafPotTh"  "hydrDist"   "rfl"        "rsec"
## [17] "rsdf"       "smthresh"   "lrt"        "lrf"
```

Some of the details are available in `?BioGro`. The first two are important as they are the field capacity `FieldC` and wilting point `WiltP` if they are not supplied they are obtained from a default soil given by `soilType`. To look at the standard soils see

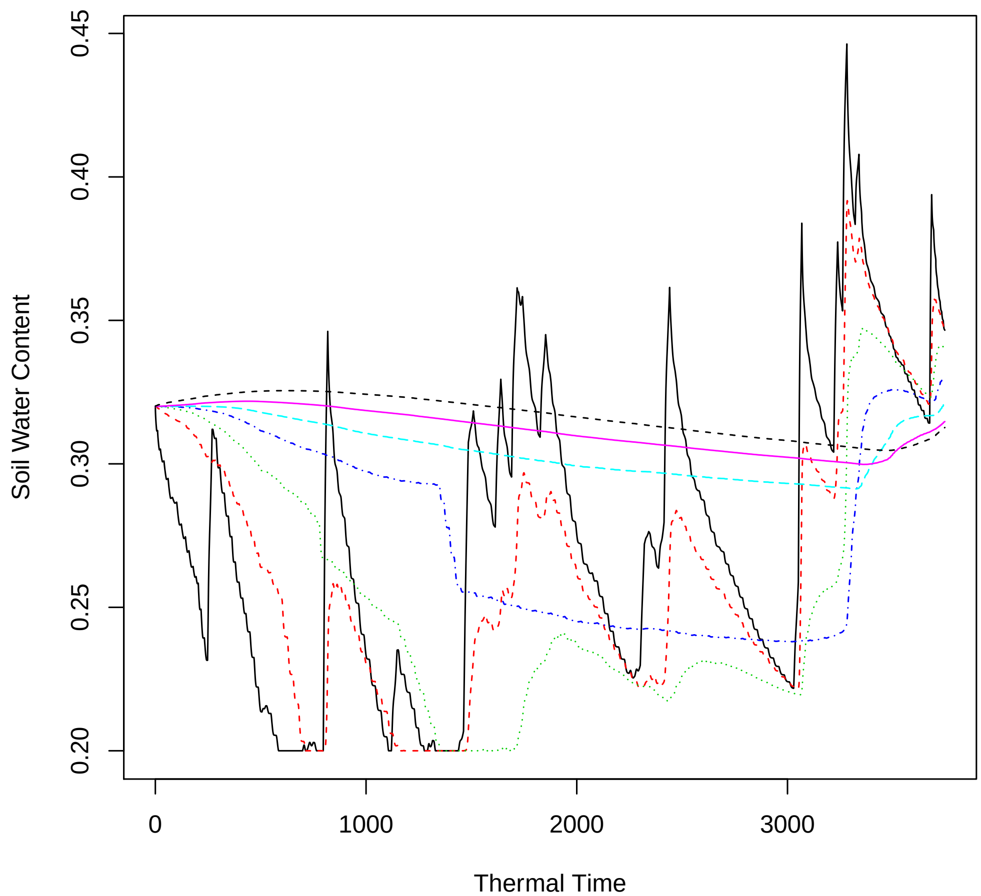
```
showSoilType(0)

## sand soil
## silt =  0.05
## clay =  0.03
## sand =  0.92
## air entry =  -0.7
## b =  1.7
## Ks =  0.0058
## satur =  0.87
```

```
## fieldc =  0.09
## wiltp =  0.03

showSoilType(5)

##  sandy clay loam
## silt =  0.13
## clay =  0.27
## sand =  0.6
## air entry =  -2.8
## b =  4
## Ks =  0.00012
## satur =  0.48
## fieldc =  0.26
## wiltp =  0.15

showSoilType(10)

##  clay
## silt =  0.2
## clay =  0.6
## sand =  0.2
## air entry =  -3.7
## b =  7.6
## Ks =  1.7e-05
## satur =  0.53
## fieldc =  0.4
## wiltp =  0.27
```

This shows a range of soils from clay (10) to sand (0) and an intermediate sandy clay loam.

Another important information is the soil depth. Typically crops have access to anywhere from 1 to 2.5 m of soil through their soil exploration. If the number of layers of soil is equal to 1 then the soil is treated as a simple bucket and the crop roots have access to the entire profile. If the number of layers is larger than one the roots will only have access to the layers in which they have grown into. An example of a simulation using 7 layers and a soil depth of 1.5m.

```r
soilP <- soilParms(soilLayers = 7, soilDepth = 1.5)
res <- BioGro(cmi05, soilControl = soilP)
plot(res, plot.kind="SW")
```



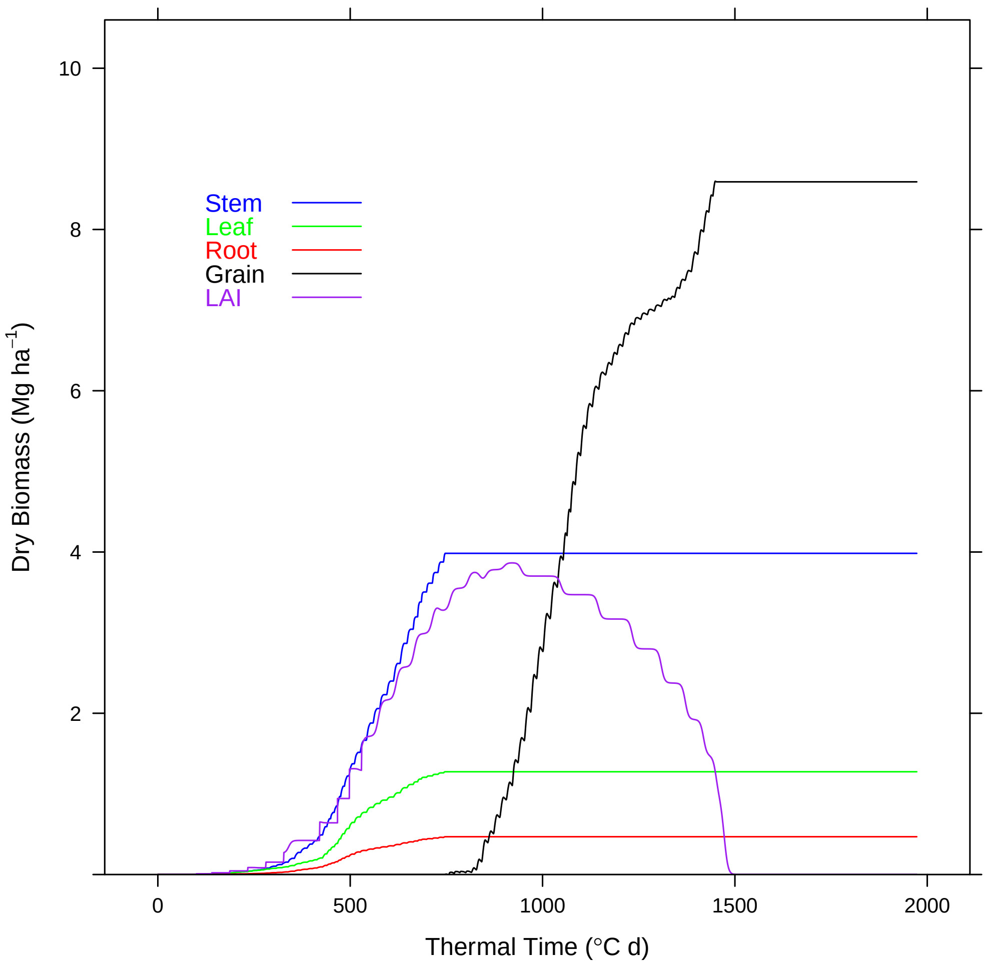
The shallower layers are depleted while the deeper layers still have water in them. This also shows that as the top layer is depleted the crop takes up water from the layers just beneath it and then the next layer down and so on. All the layers started at the same level on day 1. This is the default behavior but it can be modified by changing the argument `iWatCont`. By default water moves from one layer to the next driven by soil water potential `hydrDist =`

`TRUE` this can be turned off but it will likely not produce reasonable results.

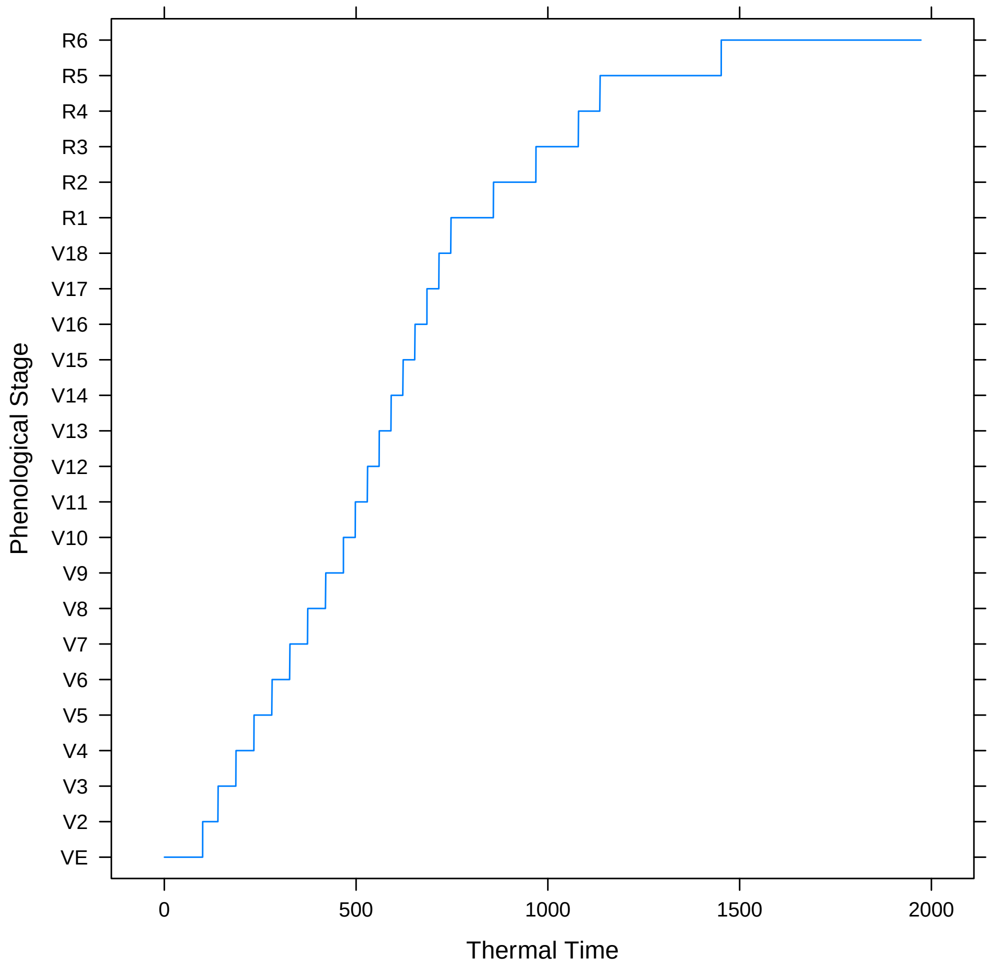
4.2 Maize

There are is also a maize model

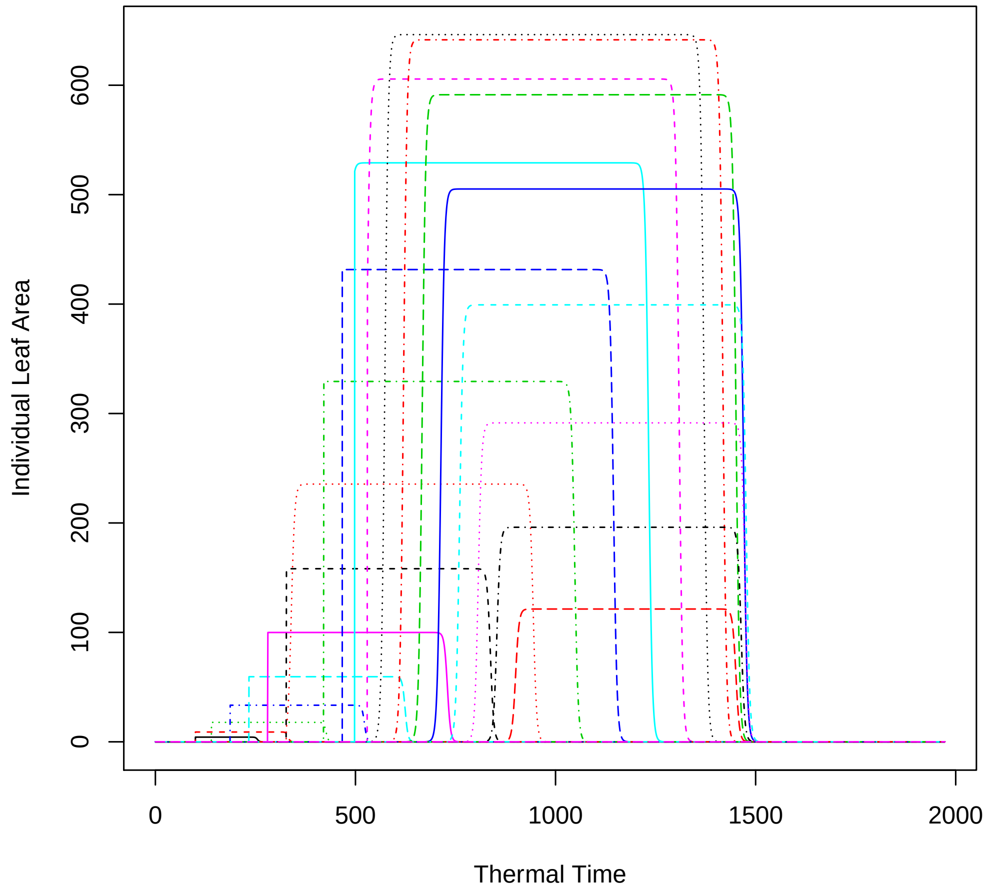
```
data(cmi05)
res <- MaizeGro(cmi05, plant.day = 110, emerge.day = 117, harvest.day = 280)
plot(res)
```



```r
plot(res, plot.kind = "pheno")
```




```r
plot(res, plot.kind = "LAI")
```


Individual Leaf Area
0
100
200
300
400
500
600
0
500
1000
1500
2000
Thermal Time