Supplementary material B: Ages and transit times as important diagnostics of model performance for predicting carbon dynamics in terrestrial vegetation models

Contents

Motivation

One of the major controls on global C sources and sinks with respect to the atmosphere is the terrestrial vegetation (Canadell et al. 2007). Furthermore, the strength of the source/sink capacity of ecosystems is determined in large part by plant phenology and carbon allocation (CA) strategies (Lacointe 2000; Schiestl-Aalto et al. 2015). In fact, we can study the sink strength capacity of carbon in vegetation by quantifying canopy photosynthesis and C residence times (τ) in plant compartments (Luo et al. 2003).

There is uncertainty in how C allocation should be represented in models. Here, we analyze the influence of different allocation strategies on ecosystem level C cycling using the following metrics: 1) the age distribution of C in the system and in each compartment, 2) the transit time distribution of the system, and 3) the dynamics of radiocarbon for individual compartments.

Objectives and research questions

The main objective of this work was to assess the influence of three different CA schemes on the rates of C cycling in vegetation. These strategies are summarized in figure 1.

The research questions were the following:

- What is the effect of model structure for representing CA on overall cycling rates of carbon in vegetation?
- Is there empirical evidence that provides support for any of the proposed model structures?

This workflow shows how these questions can be answered.

Methods and results

First load `R` packages.

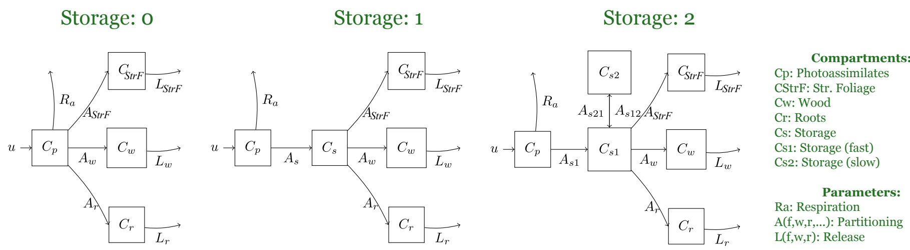

Figure 1: Three carbon allocation strategies in vegetation models. These strategies differ in the number of storage compartments, Storage: 0, Storage: 1, and Storage: 2. Adapted from [**?**]

```
library(SoilR)
library(FME)
library(ggplot2)
library(latex2exp)
library(plyr)
library(Hmisc)
library(RColorBrewer)
library("matrixStats")
set.seed(15)
```

Preparation of the data sets with the environmental drivers and observations

In order to find appropriate parameter values for the implemented models, we use published data sets with field observations and measurements obtained from the following sources:

- Harvard Forest Environmental Measurement Site
- AmeriFlux US-Ha1 Harvard Forest EMS Tower (HFR1)
- US-Ha1: Harvard Forest EMS Tower (HFR1)
- Raw LAI, see also index

This data is processed in the following way:

1) Calculate the amount of carbon in wood, from the aboveground biomass inventory data

```
obs = read.csv("AMF_US-Ha1_BIF_LATEST.csv", stringsAsFactors = FALSE)
obs_w<-subset(obs,VARIABLE_GROUP=="GRP_AG_BIOMASS_TREE")
wood <-obs_w[grep("AG_BIOMASS_DATE|AG_BIOMASS_TREE$", obs_w$VARIABLE),]
value<-subset(wood[grep("AG_BIOMASS_TREE",wood$VARIABLE),])
value<-value[c("DATAVALUE")] # unit: gC m-2
date<-subset(wood[grep("AG_BIOMASS_DATE",wood$VARIABLE),])
date<-date[c("DATAVALUE")]
wood <-obs_w[grep("AG_BIOMASS_DATE|AG_BIOMASS_TREE_SPATIAL_VARIABILITY", obs_w$VARIABLE),]
sd<-subset(wood[grep("AG_BIOMASS_TREE_SPATIAL_VARIABILITY",wood$VARIABLE),])
sd<-sd[c("DATAVALUE")]
d_W <- cbind(date,value,sd)
names(d_W) <- c("time","mean","sd")
d_W<-d_W[order(d_W$time),]
d_W$time <- as.numeric(d_W$time)
d_W$mean <- as.numeric(d_W$mean)
d_W$sd <- as.numeric(d_W$sd)
d_W$max = as.numeric(d_W$mean)+as.numeric(d_W$sd)
d_W$min = as.numeric(d_W$mean)-as.numeric(d_W$sd)
dW <- data.frame("time" = d_W$time,"Cw" = d_W$mean,"sd" = d_W$sd)
```

1) Calculate the amount of carbon in foliage, from LAI and LMA, based on the formula $Cf = lma * lai[g/m^2]$ or $Cf = lai * lma * canopyarea[g]$ (Sitch et al. (2003)). The Leaf Mass per Area (LMA) $[gdrymatter/cm^2]$, which is the inverse of the specific leaf area (SLA), was obtained from Bartlett et al. (2011).

```
obs_l = read.csv("lai_98_15.csv")
LMA = read.csv("LMA.csv", stringsAsFactors = FALSE)
lma <- mean(LMA[,4])
lma <- lma*1e4 # unit: g/m^2
```

```
# Correct for issues with values in Bartlett et. al, 2011, and convert dry mass to C:
lma <- lma/(10*2)
obs_l$mean <- obs_l$LAI * lma
year_l = obs_l$year
year_l = unique(year_l)
d_F <- data.frame()
for(i in year_l){
    sub_obs <- subset(obs_l, year == i)
    df_sub <- data.frame(time = i, mean = mean(sub_obs$mean), sd = sd(sub_obs$mean))
    d_F <- rbind(d_F,df_sub)
}
#see also: http://www.srs.fs.usda.gov/pubs/ja/ja_lockhart015.pdf
d_F$max = as.numeric(d_F$mean)+as.numeric(d_F$sd)
d_F$min = as.numeric(d_F$mean)-as.numeric(d_F$sd)
dF <- data.frame("time" = d_F$time,"Cf" = d_F$mean,"sd" = d_F$sd)
```

Implementation of the three model structures

For the initial simulations we use parameter values based on previous work done at the study site (Fox et al. 2009). Since we will implement the three models using the `SoilR` function `GeneralModel_14`, we need to prepare the arguments of the function:

```
# Duration of model run (years)
t_start=1951
t_end=2009
sequence=seq(t_start, t_end)
Age = seq(0,200,by=0.01)

# Initial parameters
input=1400 #unit: gC m^-2 yr^-1

# Matrices of C cycling parameters for each model
A0 = function(pars){
    A = matrix(
      c((-pars["Ra"]-pars["Af"]-pars["Aw"]-pars["Ar"]), 0,  0,  0,
             pars["Af"],       -pars["Lf"], 0,  0,
             pars["Aw"],        0, -pars["Lw"], 0,
             pars["Ar"],        0,  0, -pars["Lr"]),
      byrow=TRUE,
      nrow=4,
      ncol=4)
    return(A)
}

At = function(pars){
  A=A0(pars)
  At=new(Class="BoundLinDecompOp",
    t_start,
    t_end,
    function(t0){A}
  )
  return(At)
}

Ab0 = function(pars){
    Ab = matrix(
      c(-pars["Ra"]-pars["As"],       0,         0,      0,      0,
         pars["As"], (-pars["Af"]-pars["Aw"]-pars["Ar"]), 0,      0,      0,
         0,       pars["Af"],   (-pars["Lf"]), 0,      0,
         0,       pars["Aw"],        0, (-pars["Lw"]), 0,
         0,       pars["Ar"],        0,      0, (-pars["Lr"])),
      byrow=TRUE,
      nrow=5,
      ncol=5)
    return(Ab)
}
```

```
Atb = function(pars){
  Ab=Ab0(pars)
  Atb=new(Class="BoundLinDecompOp",
    t_start,
    t_end,
    function(t0){Ab}
  )
  return(Atb)
}

Ac0 = function(pars){
    Ac = matrix(
      c(-pars["Ra"]-pars["As"],         0,             0,       0,        0,        0,
         pars["As"], (-pars["As12"]-pars["Af"]-pars["Aw"]-pars["Ar"]), pars["As21"], 0, 0, 0,
              0,         pars["As12"],      (-pars["As21"]), 0,        0,        0,
              0,         pars["Af"],            0, (-pars["Lf"]),  0,        0,
              0,         pars["Aw"],            0,       0, (-pars["Lw"]),   0,
              0,         pars["Ar"],            0,       0,        0,  (-pars["Lr"])),
      byrow=TRUE,
      nrow=6,
      ncol=6)
    return(Ac)
}

Atc = function(pars){
  Ac=Ac0(pars)
  Atc=new(Class="BoundLinDecompOp",
    t_start,
    t_end,
    function(t0){Ac}
  )
  return(Atc)
}

# Input vectors (input * partitioning coefficients)
s = function(pars){
  s = matrix(nrow=4,ncol=1,input*c(1,0,0,0))
  return(s)
}

inputs = function(pars){
  s=s(pars)
  inputs=inputFluxes=new(
    "TimeMap",
    t_start,
    t_end,
    function(t0){s}
  )
  return(inputs)
}

sb = function(pars){
  sb = matrix(nrow=5,ncol=1,input*c(1, 0, 0, 0, 0))
  return(sb)
}

inputsb = function(pars){
  sb=sb(pars)
  inputsb=inputFluxes=new(
    "TimeMap",
    t_start,
    t_end,
    function(t0){sb}
  )
  return(inputsb)
```

```
}

sc = function(pars){
  sc = matrix(nrow=6,ncol=1,input*c(1, 0, 0, 0, 0, 0))
  return(sc)
}

inputsc = function(pars){
  sc=sc(pars)
  inputsc=inputFluxes=new(
    "TimeMap",
    t_start,
    t_end,
    function(t0){sc}
  )
  return(inputsc)
}

# Initial C stocks
c0=c(100, 100, 5000, 1000) # Cp, Cf, Cw, Cr
c0b=c(100, 560, 100, 5000, 1000) # Cp, Cs, Cf, Cw, Cr
c0c=c(100, 280, 280, 100, 5000, 1000) # Cp, Cs1, Cs2, Cf, Cw, Cr

# Set conditions to predict the radiocarbon, based on the bomb-spike
F0=ConstFc(c(0, 0, 0, 0),"Delta14C")
F0b=ConstFc(c(0,0, 0, 0, 0),"Delta14C")
F0c=ConstFc(c(0,0,0, 0, 0, 0),"Delta14C")
AtmFc=BoundFc(Hua2013$NHZone2[,1:2], format="Delta14C")
th=5730
k=log(0.5)/th #note that k is negative and has the unit y^-1
```

Then, the models can run via the following functions:

```
modelA = function(pars){
     model=GeneralModel_14(t=sequence, A=At(pars), ivList=c0, initialValF=F0,
     inputFluxes=inputs(pars), inputFc=AtmFc, di=k, pass = TRUE)
     return(model)
}

modelB = function(pars){
     model=GeneralModel_14(t=sequence, A=Atb(pars), ivList=c0b, initialValF=F0b,
     inputFluxes=inputsb(pars), inputFc=AtmFc, di=k, pass = TRUE)
     return(model)
}

modelC = function(pars){
     model=GeneralModel_14(t=sequence, A=Atc(pars), ivList=c0c, initialValF=F0c,
     inputFluxes=inputsc(pars), inputFc=AtmFc, di=k, pass = TRUE)
     return(model)
}
```

Run models with estimated parameter sets and prepare data frames for plotting C stocks and releases

```
fmC <- readRDS(file = "fmC.Rda")
freqC <- plyr::count(fmC$pars)
mfpC <- freqC[freqC$freq == max(freqC$freq),]
finpars = as.double(mfpC[,1:10])
names(finpars) = c("Ra","Af","Ar","Aw","Lf","Lr","Lw","As","As12","As21")

fVEGa = modelA(finpars)
fCta=getC(fVEGa)
fRta=getReleaseFlux(fVEGa)
fC14ta=getF14(fVEGa)
```

```r
Rp <- data.frame("time"=sequence,"stock"=fCta[,1],"rel"=fRta[,1])
Rp$col <- "Photoassimilates"
Rf <- data.frame("time"=sequence,"stock"=fCta[,2],"rel"=fRta[,2])
Rf$col <- "Str. Foliage"
Rw <- data.frame("time"=sequence,"stock"=fCta[,3],"rel"=fRta[,3])
Rw$col <- "Wood"
Rr <- data.frame("time"=sequence,"stock"=fCta[,4],"rel"=fRta[,4])
Rr$col <- "Roots"
df_a <- do.call(rbind, list(Rp,Rf,Rw,Rr))

fVEGb = modelB(finpars)
fCtb=getC(fVEGb)
fRtb=getReleaseFlux(fVEGb)
fC14tb=getF14(fVEGb)

Rp <- data.frame("time"=sequence,"stock"=fCtb[,1],"rel"=fRtb[,1])
Rp$col <- "Photoassimilates"
Rs <- data.frame("time"=sequence,"stock"=fCtb[,2],"rel"=fRtb[,2])
Rs$col <- "Storage"
Rf <- data.frame("time"=sequence,"stock"=fCtb[,3],"rel"=fRtb[,3])
Rf$col <- "Str. Foliage"
Rw <- data.frame("time"=sequence,"stock"=fCtb[,4],"rel"=fRtb[,4])
Rw$col <- "Wood"
Rr <- data.frame("time"=sequence,"stock"=fCtb[,5],"rel"=fRtb[,5])
Rr$col <- "Roots"
df_b <- do.call(rbind, list(Rp,Rs,Rf,Rw,Rr))

fVEGc = modelC(finpars)
fCtc=getC(fVEGc)
fRtc=getReleaseFlux(fVEGc)
fC14tc=getF14(fVEGc)

Rp <- data.frame("time"=sequence,"stock"=fCtc[,1],"rel"=fRtc[,1])
Rp$col <- "Photoassimilates"
Rs1 <- data.frame("time"=sequence,"stock"=fCtc[,2],"rel"=fRtc[,2])
Rs1$col <- "Storage (fast)"
Rs2 <- data.frame("time"=sequence,"stock"=fCtc[,3],"rel"=fRtc[,3])
Rs2$col <- "Storage (slow)"
Rf <- data.frame("time"=sequence,"stock"=fCtc[,4],"rel"=fRtc[,4])
Rf$col <- "Str. Foliage"
Rw <- data.frame("time"=sequence,"stock"=fCtc[,5],"rel"=fRtc[,5])
Rw$col <- "Wood"
Rr <- data.frame("time"=sequence,"stock"=fCtc[,6],"rel"=fRtc[,6])
Rr$col <- "Roots"
df_c <- do.call(rbind, list(Rp,Rs1,Rs2,Rf,Rw,Rr))
```

Plots of the results for C stocks, comparing the observed data and the fitted models

```r
knitr::opts_chunk$set(dev = 'pdf')
ggplot(df_a, aes(x=time, y=stock))+
  geom_point(data = dR, aes(time, Cr), colour="red") +
  scale_shape_manual(values = 15) +
  geom_errorbar(
      data = dR,
      aes(time, Cr, ymin = Cr-sd, ymax = Cr+sd),
      width = 0.05,alpha=0.5) +
  geom_point(data = d_W, aes(time, mean), colour="brown") +
  geom_errorbar(
    data = d_W,
    aes(time, mean, ymin = min, ymax = max),
    width = 0.05,alpha=0.5) +
  geom_point(data = d_F, aes(time, mean), colour="green") +
  geom_errorbar(
    data = d_F,
    aes(time, mean, ymin = min, ymax = max),
```

```r
    width = 0.05,alpha=0.5) +
  geom_line(aes(colour=col)) +
  theme_bw(23) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = c(1,0.5), legend.justification = c(1,1),
        legend.background = element_rect(fill = 'transparent'),
        legend.title = element_blank()) +
  xlim(1995, 2009) +
  xlab("Year") +
  ylab(bquote('C stocks ['*gC~m^-2*']')) +
  scale_color_manual(values=c("green","red","orange","brown")) +
  guides(color=guide_legend(
    keywidth=0.3,
    keyheight=0.3,
    default.unit="inch"))
```

```
## Error in fortify(data): object 'dR' not found
```

```r
knitr::opts_chunk$set(dev = 'pdf')
ggplot(df_b, aes(x=time, y=stock))+
  geom_point(data = dS, aes(time, Cs), colour="Purple") +
  geom_errorbar(
      data = dS,
      aes(time, Cs, ymin = Cs-sd, ymax = Cs-sd),
      width = 0.05,alpha=0.5) +
  geom_point(data = dR, aes(time, Cr), colour="red") +
  scale_shape_manual(values = 15) +
  geom_errorbar(
      data = dR,
      aes(time, Cr, ymin = Cr-sd, ymax = Cr+sd),
      width = 0.05,alpha=0.5) +
  geom_point(data = d_W, aes(time, mean), colour="brown") +
  geom_errorbar(
    data = d_W,
    aes(time, mean, ymin = min, ymax = max),
    width = 0.05,alpha=0.5) +
  geom_point(data = d_F, aes(time, mean), colour="green") +
  geom_errorbar(
    data = d_F,
    aes(time, mean, ymin = min, ymax = max),
    width = 0.05,alpha=0.5) +
  geom_line(aes(colour=col)) +
  theme_bw(23) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = c(1,0.5), legend.justification = c(1,1),
        legend.background = element_rect(fill = 'transparent'),
        legend.title = element_blank()) +
  xlim(1995, 2009) +
  xlab("Year") +
  ylab(bquote('C stocks ['*gC~m^-2*']')) +
  scale_color_manual(values=c("green","red","purple","orange","brown")) +
  guides(color=guide_legend(
    keywidth=0.3,
    keyheight=0.3,
    default.unit="inch"))
```

```
## Error in fortify(data): object 'dS' not found
```

```r
knitr::opts_chunk$set(dev = 'pdf')
ggplot(df_c, aes(x=time, y=stock))+
  geom_point(data = dS, aes(time, Cs), colour="Purple") +
  geom_errorbar(
      data = dS,
      aes(time, Cs, ymin = Cs-sd, ymax = Cs-sd),
      width = 0.05,alpha=0.5) +
```

```r
  geom_point(data = dR, aes(time, Cr), colour="red") +
  scale_shape_manual(values = 15) +
  geom_errorbar(
      data = dR,
      aes(time, Cr, ymin = Cr-sd, ymax = Cr+sd),
      width = 0.05,alpha=0.5) +
  geom_point(data = d_W, aes(time, mean), colour="brown") +
  geom_errorbar(
    data = d_W,
    aes(time, mean, ymin = min, ymax = max),
    width = 0.05,alpha=0.5) +
  geom_point(data = d_F, aes(time, mean), colour="green") +
  geom_errorbar(
    data = d_F,
    aes(time, mean, ymin = min, ymax = max),
    width = 0.05,alpha=0.5) +
  geom_line(aes(colour=col)) +
  theme_bw(23) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = c(1,0.5), legend.justification = c(1,1),
        legend.background = element_rect(fill = 'transparent'),
        legend.title = element_blank()) +
  xlim(1995, 2009) +
  xlab("Year") +
  ylab(bquote('C stocks ['*gC~m^-2*']')) +
  scale_color_manual(values=c("green","red","purple","blue","orange","brown")) +
  guides(color=guide_legend(
    keywidth=0.3,
    keyheight=0.3,
    default.unit="inch"))
```

```
## Error in fortify(data): object 'dS' not found
```

Comparing the C stocks of all the models

```r
knitr::opts_chunk$set(dev = 'pdf')
S0w <- data.frame("time"=sequence,"stock"=fCta[,3])
S0w$col <- "Storage: 0"
S1w <- data.frame("time"=sequence,"stock"=fCtb[,4])
S1w$col <- "Storage: 1"
S2w <- data.frame("time"=sequence,"stock"=fCtc[,5])
S2w$col <- "Storage: 2"
DF_w <- do.call(rbind, list(S0w,S1w,S2w))

ggplot(DF_w, aes(x=time, y=stock))+
  geom_point(data = d_W, aes(time, mean)) +
  xlim(1951, 2009) +
  geom_errorbar(
    data = d_W,
    aes(time, mean, ymin = min, ymax = max),
    width = 0.05,alpha=0.5) +
  geom_line(aes(colour=col)) +
  theme_bw(23) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = c(0.5,1), legend.justification = c(1,1),
        legend.background = element_rect(fill = 'transparent'),
        legend.text = element_text(size = 22),
        legend.title = element_blank()) +
  xlab("Year") +
  ylab(bquote('C stock in Wood ['*gC~m^-2*']')) +
  guides(color=guide_legend(
    keywidth=0.3,
    keyheight=0.3,
```

```
    default.unit="inch"))
```

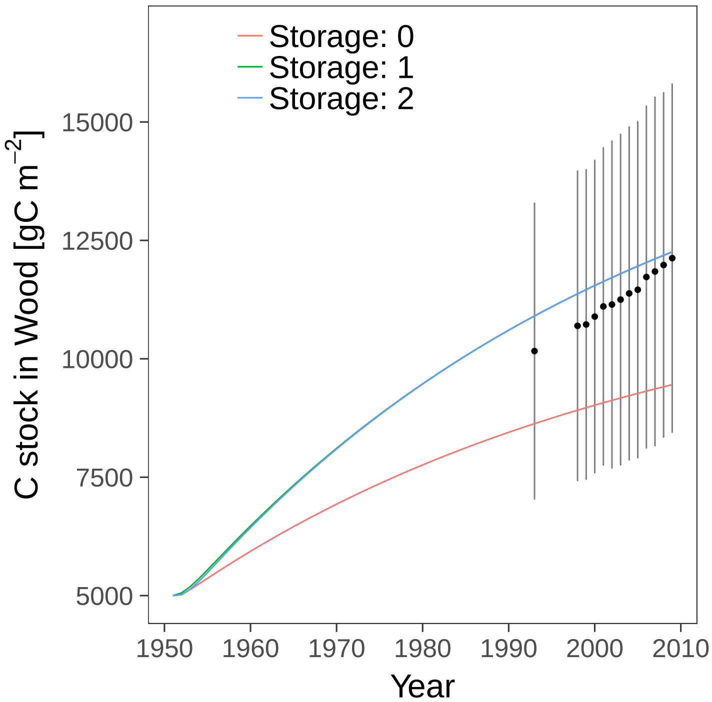

Figure 2: plot of chunk sp_C_stocks_all

```
S0f <- data.frame("time"=sequence,"stock"=fCta[,2]+fCta[,1])
S0f$col <- "Storage: 0"
S1f <- data.frame("time"=sequence,"stock"=fCtb[,3]+fCtb[,1])
S1f$col <- "Storage: 1"
S2f <- data.frame("time"=sequence,"stock"=fCtc[,4]+fCtc[,1])
S2f$col <- "Storage: 2"
DF_f <- do.call(rbind, list(S0f,S1f,S2f))

ggplot(DF_f, aes(x=time, y=stock))+
  xlim(1951, 2009) +
  geom_point(data = d_F, aes(time, mean)) +
  scale_shape_manual(values = 15) +
  geom_errorbar(
    data = d_F,
    aes(time, mean, ymin = min, ymax = max),
    width = 0.05,alpha=0.5) +
  geom_line(aes(colour=col),lty=1) +
  theme_bw(23) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = c(0.55,0.35), legend.justification = c(1,1),
        legend.background = element_rect(fill = 'transparent'),
        legend.text = element_text(size = 22),
        legend.title = element_blank()) +
  xlab("Year") +
  ylab(bquote('C stocks in Foliage ['*gC~m^-2*']')) +
  guides(color=guide_legend(
    keywidth=0.3,
    keyheight=0.3,
    default.unit="inch"))
```

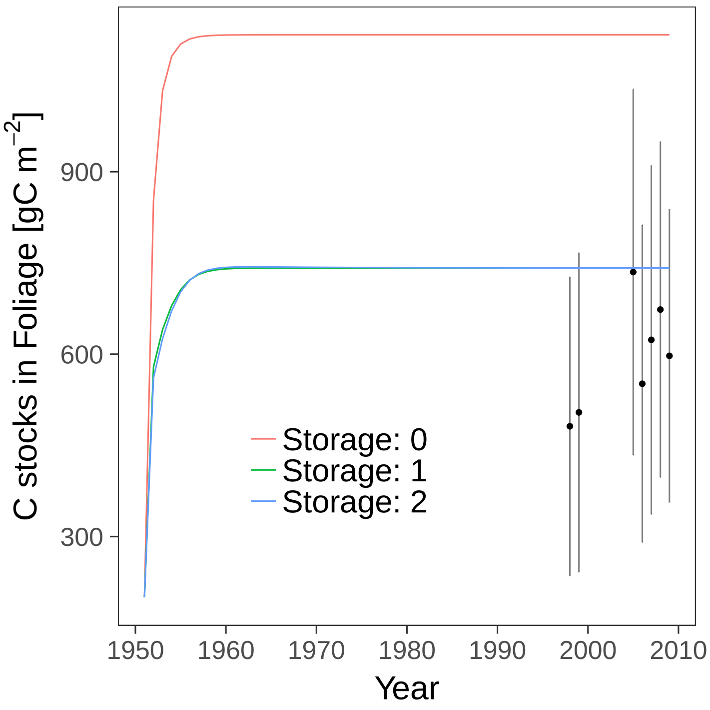

Figure 3: plot of chunk sp_C_stocks_all

Fluxes of C released from the compartments

```
knitr::opts_chunk$set(dev = 'pdf')
ggplot(df_a, aes(x=time, y=rel))+
  geom_line(aes(colour=col)) +
  theme_bw(23) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = c(1,1), legend.justification = c(1,1),
        #plot.margin = unit(c(0.3, 6, 0.3, 0.5), "cm"), # top, right, bottom, left
        legend.background = element_rect(fill = 'transparent'),
        legend.title = element_blank()) +
  xlab("Year") +
  ylab(bquote('Released C ['*gC~m^-2*']')) +
  scale_color_manual(values=c("green","orange","red","brown")) +
  guides(color=guide_legend(
    keywidth=0.3,
    keyheight=0.3,
    default.unit="inch"))

knitr::opts_chunk$set(dev = 'pdf')
ggplot(df_b, aes(x=time, y=rel))+
  geom_line(aes(colour=col)) +
  theme_bw(23) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = c(1,1), legend.justification = c(1,1),
        #plot.margin = unit(c(0.1, 6, 0.3, 0.5), "cm"), # top, right, bottom, left
        legend.background = element_rect(fill = 'transparent'),
        legend.title = element_blank()) +
  xlab("Year") +
  ylab(bquote('Released C ['*gC~m^-2*']')) +
  scale_color_manual(values=c("green","orange","red","purple","brown")) +
  guides(color=guide_legend(
    keywidth=0.3,
    keyheight=0.3,
    default.unit="inch"))
```

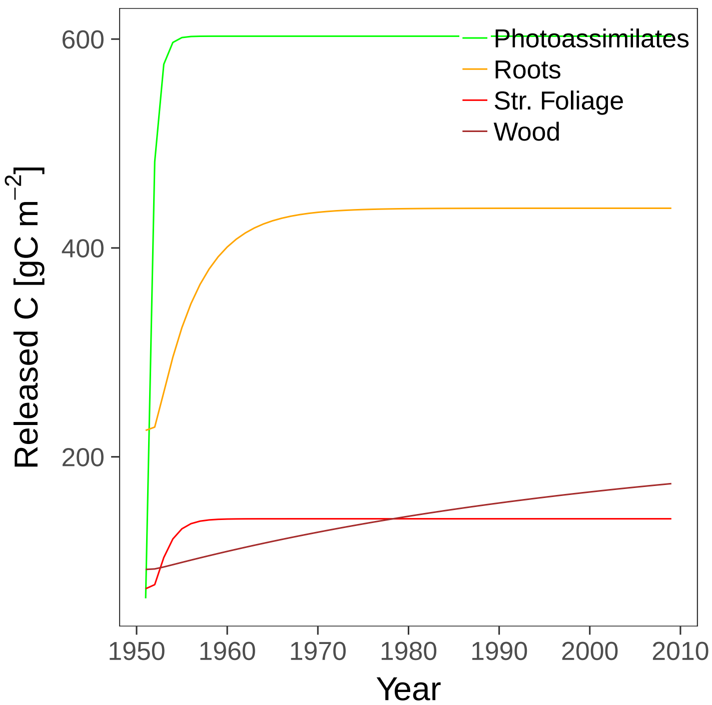

Figure 4: plot of chunk sp_C_released_S0

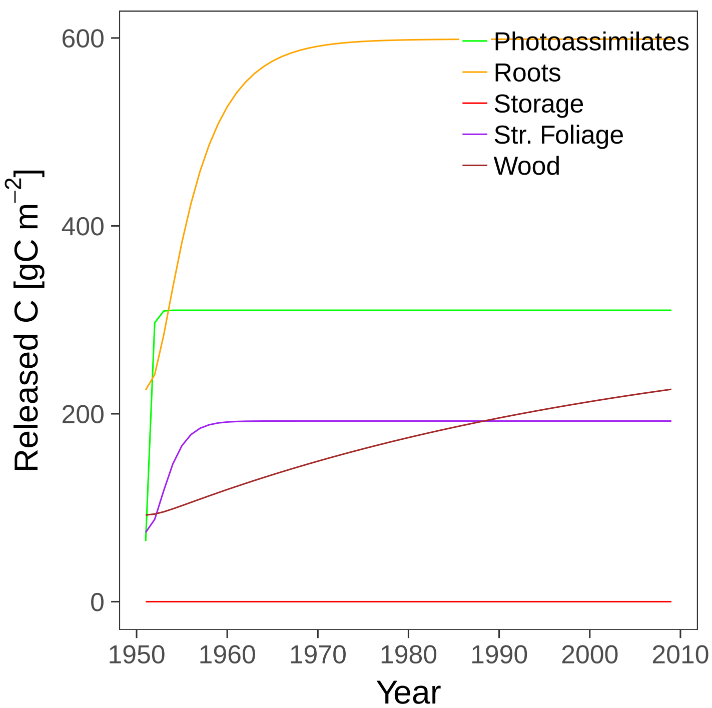

Figure 5: plot of chunk sp_C_released_S1

```
knitr::opts_chunk$set(dev = 'pdf')
ggplot(df_c, aes(x=time, y=rel))+
  geom_line(aes(colour=col)) +
  theme_bw(23) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = c(1,1), legend.justification = c(1,1),
        #plot.margin = unit(c(0.1, 6, 0.3, 0.5), "cm"), # top, right, bottom, left
        legend.background = element_rect(fill = 'transparent'),
        legend.title = element_blank()) +
  xlab("Year") +
  ylab(bquote('Released C ['*gC~m^-2*']')) +
  scale_color_manual(values=c("green","orange","red","blue","purple","brown")) +
  guides(color=guide_legend(
    keywidth=0.3,
    keyheight=0.3,
    default.unit="inch"))
```

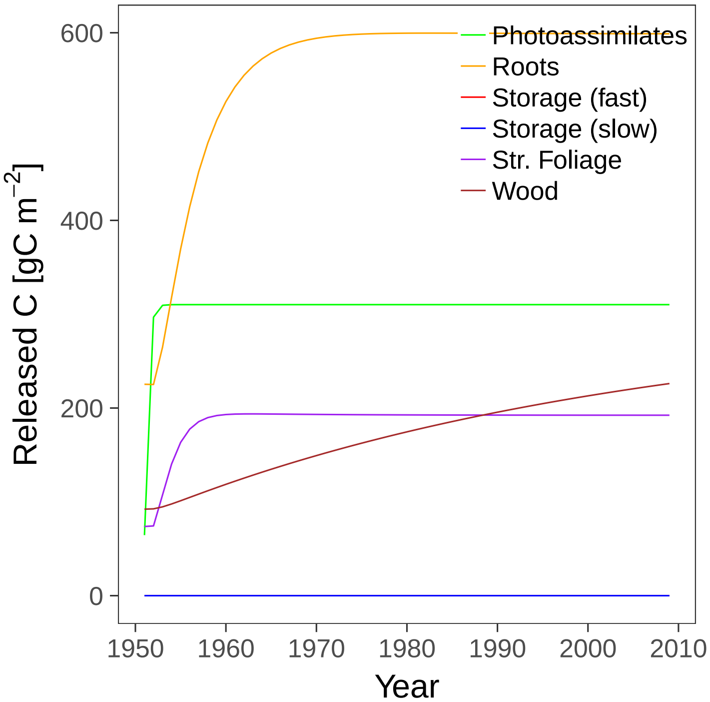

Figure 6: plot of chunk sp_C_released_S2

Radiocarbon content in each compartment

```
knitr::opts_chunk$set(dev = 'pdf')
plotC14Pool(t=sequence, mat=fC14ta, inputFc=Hua2013$NHZone2[,1:2], xlab="years",
            ylab="Radiocarbon in Delta 14C", col=2:5, cex.axis = 1.5, cex.lab = 1.6)
title(main = "Storage: 0", cex = 5, font = 10, col = 3)
legend("topright", inset=.01, legend=c("Photoassimilates", "Str. Foliage", "Wood", "Roots"),
       lty=1, col=2:5, horiz=FALSE, bty="n", cex = 1.5)

knitr::opts_chunk$set(dev = 'pdf')
plotC14Pool(t=sequence, mat=fC14tb, inputFc=Hua2013$NHZone2[,1:2], xlab="years",
            ylab="Radiocarbon in Delta 14C", col=c(2,6,3,4,5), cex.axis = 1.5, cex.lab = 1.6)
title(main = "Storage: 1", cex = 5, font = 10, col = 3)
legend("topright", inset=.01, legend=c("Photoassimilates", "Storage", "Str. Foliage", "Wood", "Roots"),
       lty=1, col=c(2,6,3,4,5), horiz=FALSE, bty="n", cex = 1.5)

knitr::opts_chunk$set(dev = 'pdf')
plotC14Pool(t=sequence, mat=fC14tc, inputFc=Hua2013$NHZone2[,1:2], xlab="years",
```

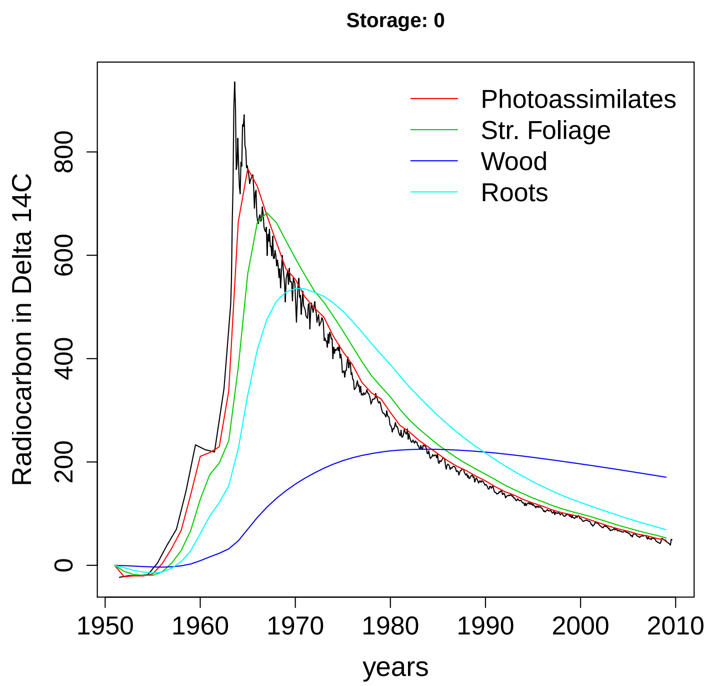

Figure 7: plot of chunk sp_C14_S0

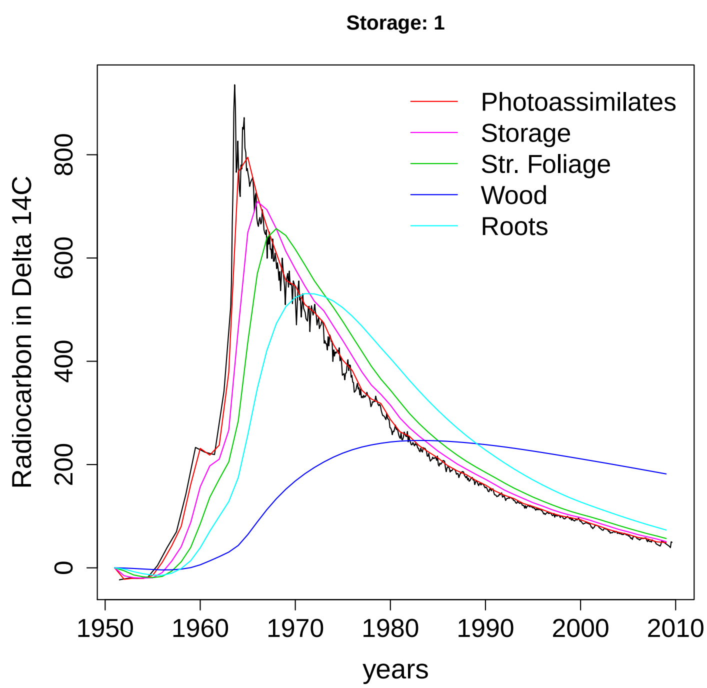

Figure 8: plot of chunk sp_C14_S1

```
          ylab="Radiocarbon in Delta 14C", c(2,6,9,3,4,5), cex.axis = 1.5, cex.lab = 1.6)
title(main = "Storage: 2", cex = 5, font = 10, col = 3)
legend("topright", inset=.01, legend=c("Photoassimilates", "Storage (fast)", "Storage (slow)", "Str. Foliage", "Wood",
       "Roots"), lty=1, col=c(2,6,9,3,4,5), horiz=FALSE, bty="n", cex = 1.5)
```

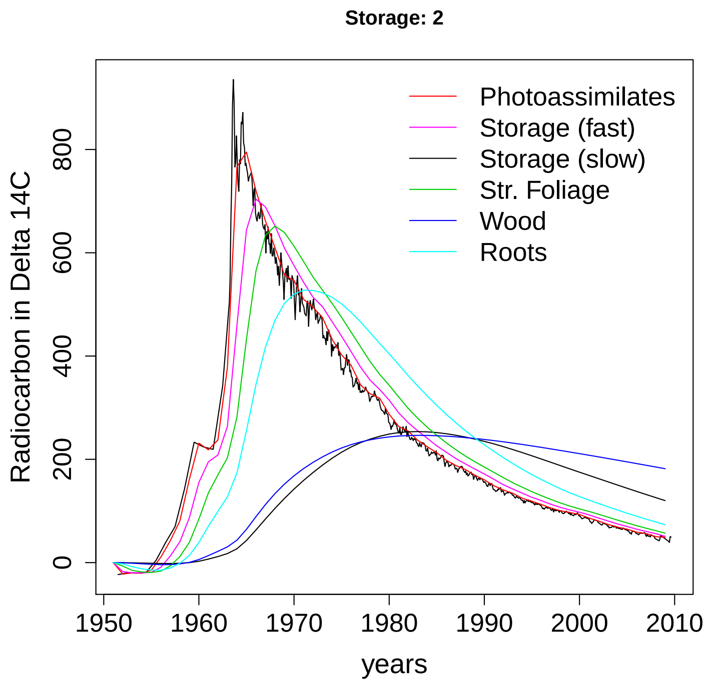

Figure 9: plot of chunk sp_C14_S2

Ages and transit times

We need the matrix **A** and the input vector **s** for a system in equilibrium (no time-dependencies). First, we calculate system and compartment ages using the function `systemAge` and for the transit times we use the function `transitTime`

```
# Model without storage compartment:
AgeVEGa=systemAge(A=A0(finpars),u=s(finpars), a=Age, q=c(0.05, 0.5, 0.95))
ttVEGa=transitTime(A=A0(finpars),u=s(finpars), a=Age, q=c(0.05, 0.5, 0.95))

# Model with one storage compartment:
AgeVEGb=systemAge(A=Ab0(finpars),u=sb(finpars), a=Age, q=c(0.05, 0.5, 0.95))
ttVEGb=transitTime(A=Ab0(finpars),u=sb(finpars), a=Age, q=c(0.05, 0.5, 0.95))

# Model with two storage compartments:
AgeVEGc=systemAge(A=Ac0(finpars),u=sc(finpars), a=Age, q=c(0.05, 0.5, 0.95))
ttVEGc=transitTime(A=Ac0(finpars),u=sc(finpars), a=Age, q=c(0.05, 0.5, 0.95))
```

For the mean, median, and confidence intervals of system age and transit time of each model we use the attributes `$mean` and `$quantiles`. We also prepare data frames for plotting age and transit time distributions

```
at_a <- data.frame("x"=Age,"age"=AgeVEGa$systemAgeDensity,"tt"=ttVEGa$transitTimeDensity)
at_a$name <- "Storage: 0"
at_b <- data.frame("x"=Age,"age"=AgeVEGb$systemAgeDensity,"tt"=ttVEGb$transitTimeDensity)
at_b$name <- "Storage: 1"
at_c <- data.frame("x"=Age,"age"=AgeVEGc$systemAgeDensity,"tt"=ttVEGc$transitTimeDensity)
at_c$name <- "Storage: 2"
DF <- do.call(rbind, list(at_a,at_b,at_c))

atm_a <- data.frame("mean_age"=AgeVEGa$meanSystemAge, "median_age"=AgeVEGa$quantilesSystemAge[2],
         "age CI.d"=AgeVEGa$quantilesSystemAge[1], "age CI.u"=AgeVEGa$quantilesSystemAge[3],
         "mean_tt"=ttVEGa$meanTransitTime, "median_tt"=ttVEGa$quantiles[2],
```

```
        "tt CI.d"=ttVEGa$quantiles[1], "tt CI.u"=ttVEGa$quantiles[3])
is.num <- sapply(atm_a, is.numeric)
atm_a[is.num] <- lapply(atm_a[is.num], round, 2)
# Info needed to limit geom_segment for median and means:
atm_a$mean_a_dens <- as.numeric(subset(DF, name == "Storage: 0" & x==atm_a$mean_age, select = age))
atm_a$median_a_dens <- as.numeric(subset(DF, name == "Storage: 0" & x==atm_a$median_age, select = age))
atm_a$mean_tt_dens <- as.numeric(subset(DF, name == "Storage: 0" & x==atm_a$mean_tt, select = tt))
atm_a$median_tt_dens <- as.numeric(subset(DF, name == "Storage: 0" & x==atm_a$median_tt, select = tt))
atm_a$name <- "Storage: 0"

atm_b <- data.frame("mean_age"=AgeVEGb$meanSystemAge, "median_age"=AgeVEGb$quantilesSystemAge[2],
        "age CI.d"=AgeVEGb$quantilesSystemAge[1], "age CI.u"=AgeVEGb$quantilesSystemAge[3],
        "mean_tt"=ttVEGb$meanTransitTime, "median_tt"=ttVEGb$quantiles[2],
        "tt CI.d"=ttVEGb$quantiles[1], "tt CI.u"=ttVEGb$quantiles[3])
is.num <- sapply(atm_b, is.numeric)
atm_b[is.num] <- lapply(atm_b[is.num], round, 2)
atm_b$mean_a_dens <- as.numeric(subset(DF, name == "Storage: 1" & x==atm_b$mean_age, select = age))
atm_b$median_a_dens <- as.numeric(subset(DF, name == "Storage: 1" & x==atm_b$median_age, select = age))
atm_b$mean_tt_dens <- as.numeric(subset(DF, name == "Storage: 1" & x==atm_b$mean_tt, select = tt))
atm_b$median_tt_dens <- as.numeric(subset(DF, name == "Storage: 1" & x==atm_b$median_tt, select = tt))
atm_b$name <- "Storage: 1"

atm_c <- data.frame("mean_age"=AgeVEGc$meanSystemAge, "median_age"=AgeVEGc$quantilesSystemAge[2],
        "age CI.d"=AgeVEGc$quantilesSystemAge[1], "age CI.u"=AgeVEGc$quantilesSystemAge[3],
        "mean_tt"=ttVEGc$meanTransitTime, "median_tt"=ttVEGc$quantiles[2],
        "tt CI.d"=ttVEGc$quantiles[1], "tt CI.u"=ttVEGc$quantiles[3])
is.num <- sapply(atm_c, is.numeric)
atm_c[is.num] <- lapply(atm_c[is.num], round, 2)
atm_c$mean_a_dens <- as.numeric(subset(DF, name == "Storage: 2" & x==atm_c$mean_age, select = age))
atm_c$median_a_dens <- as.numeric(subset(DF, name == "Storage: 2" & x==atm_c$median_age, select = age))
atm_c$mean_tt_dens <- as.numeric(subset(DF, name == "Storage: 2" & x==atm_c$mean_tt, select = tt))
atm_c$median_tt_dens <- as.numeric(subset(DF, name == "Storage: 2" & x==atm_c$median_tt, select = tt))
atm_c$name <- "Storage: 2"

DF_m <- do.call(rbind, list(atm_a,atm_b,atm_c))
DF_m

##   mean_age median_age age.CI.d age.CI.u mean_tt median_tt tt.CI.d tt.CI.u
## 1    44.33      25.82     0.64   150.62   10.66      1.57    0.08   62.40
## 2    44.15      25.43     0.88   150.21   14.91      3.77    0.09   80.20
## 3    44.12      25.40     0.88   150.16   14.94      3.77    0.09   80.26
##   mean_a_dens median_a_dens mean_tt_dens median_tt_dens       name
## 1 0.006549452   0.009314739  0.009963071     0.14624465 Storage: 0
## 2 0.006522671   0.009346455  0.008019344     0.07651512 Storage: 1
## 3 0.006524344            NA  0.008029032     0.07629909 Storage: 2
```

Graphically, we can see:

```
knitr::opts_chunk$set(dev = 'pdf')
ggplot(data=DF_m) +
#  geom_linerange(aes(x=mean_age, ymin=0, ymax=mean_a_dens, linetype="Mean",colour=name)) +
  geom_vline(aes(xintercept=mean_age,linetype = "Mean",colour=name)) +
  geom_vline(aes(xintercept=median_age,linetype = "Median",colour=name)) +
  scale_linetype_manual(values=c("longdash","dotted")) +
  #geom_segment(data = DF_m, aes(x = mean_age, y = mean_a_dens, xend = mean_age, yend = Inf), colour="white") +
  #geom_segment(data = DF_m, aes(x = median_age, y = median_a_dens, xend = median_age, yend = Inf), colour="white") +
  geom_line(data=DF, aes(x=x,y=age,colour=name)) +
  theme_bw(23) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = c(1,1), legend.justification = c(1,1),
        legend.background = element_rect(fill = 'transparent'),
        legend.title = element_blank()) +
  xlim(0,70) +
  xlab("Age [years]") +
  ylab("Age density") +
  scale_color_manual(values=c("#009999","#FF6633","#660099")) +
```

```
  guides(color=guide_legend(
    keywidth=0.33,
    keyheight=0.33,
    default.unit="inch"))
```

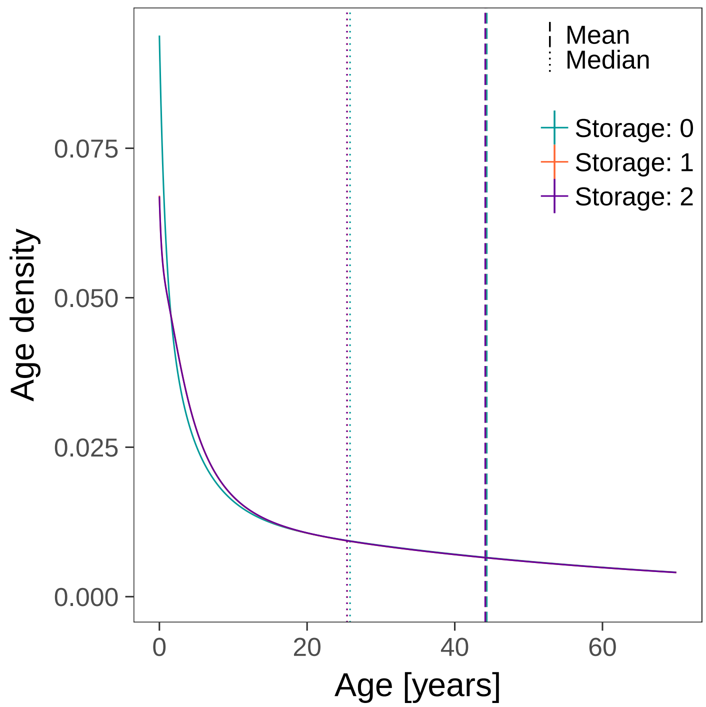

Figure 10: plot of chunk sp_Age_dist_3_models

Comparing the compartments Cp, Cf, Cw, and Cr of the 3 models:

```
ap_a <- data.frame("x"=Age, "Cp"=AgeVEGa$poolAgeDensity[,1], "Cf"=AgeVEGa$poolAgeDensity[,2],
               "Cw"=AgeVEGa$poolAgeDensity[,3], "Cr"=AgeVEGa$poolAgeDensity[,4])
ap_a$name <- "Storage: 0"
ap_b <- data.frame("x"=Age, "Cp"=AgeVEGb$poolAgeDensity[,1], "Cf"=AgeVEGb$poolAgeDensity[,3],
               "Cw"=AgeVEGb$poolAgeDensity[,4], "Cr"=AgeVEGb$poolAgeDensity[,5])
ap_b$name <- "Storage: 1"
ap_c <- data.frame("x"=Age, "Cp"=AgeVEGc$poolAgeDensity[,1], "Cf"=AgeVEGc$poolAgeDensity[,4],
               "Cw"=AgeVEGc$poolAgeDensity[,5], "Cr"=AgeVEGc$poolAgeDensity[,6])
ap_c$name <- "Storage: 2"
DF_p <- do.call(rbind, list(ap_a, ap_b, ap_c))

apm_a <- data.frame("Cp"=AgeVEGa$meanPoolAge[1,1], "Cf"=AgeVEGa$meanPoolAge[2,1],
                    "Cw"=AgeVEGa$meanPoolAge[3,1], "Cr"=AgeVEGa$meanPoolAge[4,1])
is.num <- sapply(apm_a, is.numeric)
apm_a[is.num] <- lapply(apm_a[is.num], round, 2)
apm_a$Cp_dens <- as.numeric(subset(DF_p, name == "Storage: 0" & x==apm_a$Cp, select = Cp))
apm_a$Cf_dens <- as.numeric(subset(DF_p, name == "Storage: 0" & x==apm_a$Cf, select = Cf))
apm_a$Cw_dens <- as.numeric(subset(DF_p, name == "Storage: 0" & x==apm_a$Cw, select = Cw))
apm_a$Cr_dens <- as.numeric(subset(DF_p, name == "Storage: 0" & x==apm_a$Cr, select = Cr))
apm_a$name <- "Storage: 0"
apm_b <- data.frame("Cp"=AgeVEGb$meanPoolAge[1,1], "Cf"=AgeVEGb$meanPoolAge[3,1],
                    "Cw"=AgeVEGb$meanPoolAge[4,1], "Cr"=AgeVEGb$meanPoolAge[5,1])
is.num <- sapply(apm_b, is.numeric)
apm_b[is.num] <- lapply(apm_b[is.num], round, 2)
apm_b$Cp_dens <- as.numeric(subset(DF_p, name == "Storage: 1" & x==apm_b$Cp, select = Cp))
apm_b$Cf_dens <- as.numeric(subset(DF_p, name == "Storage: 1" & x==apm_b$Cf, select = Cf))
apm_b$Cw_dens <- as.numeric(subset(DF_p, name == "Storage: 1" & x==apm_b$Cw, select = Cw))
apm_b$Cr_dens <- as.numeric(subset(DF_p, name == "Storage: 1" & x==apm_b$Cr, select = Cr))
apm_b$name <- "Storage: 1"
apm_c <- data.frame("Cp"=AgeVEGc$meanPoolAge[1,1], "Cf"=AgeVEGc$meanPoolAge[4,1],
                    "Cw"=AgeVEGc$meanPoolAge[5,1], "Cr"=AgeVEGc$meanPoolAge[6,1])
```

```r
is.num <- sapply(apm_c, is.numeric)
apm_c[is.num] <- lapply(apm_c[is.num], round, 2)
apm_c$Cp_dens <- as.numeric(subset(DF_p, name == "Storage: 2" & x==apm_c$Cp, select = Cp))
apm_c$Cf_dens <- as.numeric(subset(DF_p, name == "Storage: 2" & x==apm_c$Cf, select = Cf))
apm_c$Cw_dens <- as.numeric(subset(DF_p, name == "Storage: 2" & x==apm_c$Cw, select = Cw))
apm_c$Cr_dens <- as.numeric(subset(DF_p, name == "Storage: 2" & x==apm_c$Cr, select = Cr))
apm_c$name <- "Storage: 2"
DF_pm <- do.call(rbind, list(apm_a, apm_b, apm_c))
#DF_pm <- as.data.frame(DF_pm)

# The mean age of each compartment:
DF_pm
```

```
##     Cp   Cf    Cw   Cr   Cp_dens   Cf_dens     Cw_dens    Cr_dens
## 1 0.67 2.02 54.89 5.11 0.5490695 0.2569879 0.006785126 0.08372964
## 2 0.34 2.87 55.74 5.95 1.0819440 0.2125708 0.006786105 0.08413411
## 3 0.34 2.91 55.78 5.99 1.0819440 0.2087594 0.006786452 0.08323389
##         name
## 1 Storage: 0
## 2 Storage: 1
## 3 Storage: 2
```

1. Photoassimilates

```r
knitr::opts_chunk$set(dev = 'pdf')
ggplot() +
  ggtitle("Photoassimilates") +
  geom_vline(data=DF_pm,aes(xintercept=Cp,linetype = "Mean",colour=name)) +
  scale_linetype_manual(values="longdash") +
  geom_segment(data = DF_pm, aes(x = Cp, y = Cp_dens, xend = Cp, yend = Inf), colour="white") +
  geom_line(data=DF_p, aes(x=x,y=Cp,colour=name)) +
  geom_text(data = DF_pm, aes(label = Cp, y=Cp_dens+0.25, x=Cp), size=5.5) +
  theme_bw(23) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = c(1,1), legend.justification = c(1,1),
        legend.background = element_rect(fill = 'transparent'),
        plot.title = element_text(size = rel(1), hjust = 0.5, colour = "black"),
        legend.title = element_blank()) +
  xlim(0,2) +
  xlab("Age [years]") +
  ylab("Age density") +
  scale_color_manual(values=c("#009999","#FF6633","#660099")) +
  guides(color=guide_legend(
    keywidth=0.33,
    keyheight=0.33,
    default.unit="inch"))
```

2. Foliage

```r
knitr::opts_chunk$set(dev = 'pdf')
ggplot() +
  ggtitle("Foliage") +
  geom_vline(data=DF_pm,aes(xintercept=Cf,linetype = "Mean",colour=name)) +
  scale_linetype_manual(values="longdash") +
  geom_segment(data = DF_pm, aes(x = Cf, y = Cf_dens, xend = Cf, yend = Inf), colour="white") +
  geom_line(data=DF_p, aes(x=x,y=Cf,colour=name)) +
  geom_text(data = DF_pm, aes(label = Cf, y=Cf_dens+0.1, x=Cf), size=5.5) +
  theme_bw(23) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = c(1,1), legend.justification = c(1,1),
        legend.background = element_rect(fill = 'transparent'),
        plot.title = element_text(size = rel(1), hjust = 0.5, colour = "black"),
        legend.title = element_blank()) +
  xlim(0,6) +
```

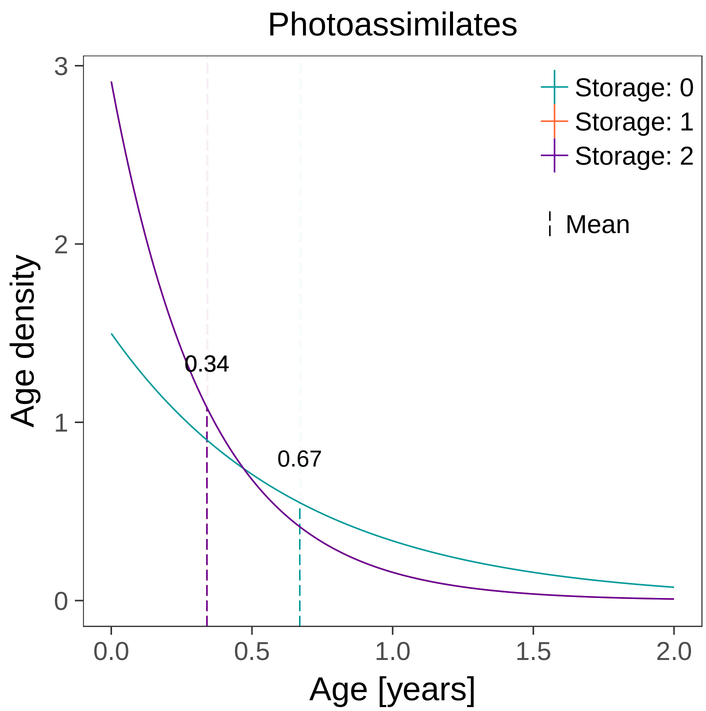

Figure 11: plot of chunk sp_Age_dist_Cp

```
  xlab("Age [years]") +
  ylab("Age density") +
  scale_color_manual(values=c("#009999","#FF6633","#660099")) +
  guides(color=guide_legend(
    keywidth=0.33,
    keyheight=0.33,
    default.unit="inch"))
```

3. Wood

```
knitr::opts_chunk$set(dev = 'pdf')
ggplot() +
  ggtitle("Wood") +
  geom_vline(data=DF_pm,aes(xintercept=Cw,linetype = "Mean",colour=name)) +
  scale_linetype_manual(values="longdash") +
  geom_segment(data = DF_pm, aes(x = Cw, y = Cw_dens, xend = Cw, yend = Inf), colour="white") +
  geom_line(data=DF_p, aes(x=x,y=Cw,colour=name)) +
  geom_text(data = DF_pm, aes(label = Cw, y=Cw_dens+0.005, x=Cw), size=5.5) +
  theme_bw(23) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = c(1,1), legend.justification = c(1,1),
        legend.background = element_rect(fill = 'transparent'),
        plot.title = element_text(size = rel(1), hjust = 0.5, colour = "black"),
        legend.title = element_blank()) +
  xlim(0,200) +
  xlab("Age [years]") +
  ylab("Age density") +
  scale_color_manual(values=c("#009999","#FF6633","#660099")) +
  guides(color=guide_legend(
    keywidth=0.33,
    keyheight=0.33,
    default.unit="inch"))
```

4. Roots

```
knitr::opts_chunk$set(dev = 'pdf')
ggplot() +
```

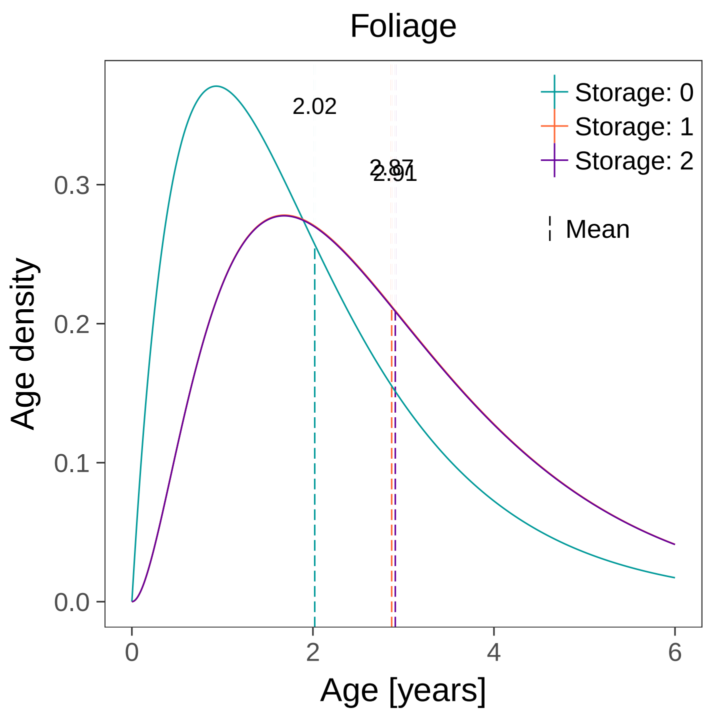

Figure 12: plot of chunk sp_Age_dist_Cf

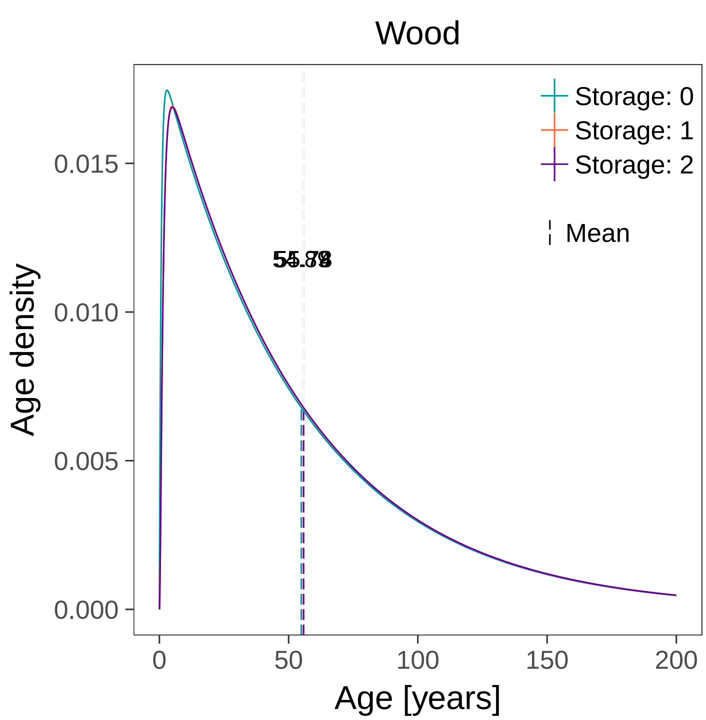

Figure 13: plot of chunk sp_Age_dist_Cw

```
  ggtitle("Roots") +
  geom_vline(data=DF_pm,aes(xintercept=Cr,linetype = "Mean",colour=name)) +
  scale_linetype_manual(values="longdash") +
  geom_segment(data = DF_pm, aes(x = Cr, y = Cr_dens, xend = Cr, yend = Inf), colour="white") +
  geom_line(data=DF_p, aes(x=x,y=Cr,colour=name)) +
  geom_text(data = DF_pm, aes(label = Cr, y=Cr_dens+0.05, x=Cr), size=5.5) +
  theme_bw(23) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = c(1,1), legend.justification = c(1,1),
        legend.background = element_rect(fill = 'transparent'),
        plot.title = element_text(size = rel(1), hjust = 0.5, colour = "black"),
        legend.title = element_blank()) +
  xlim(0,15) +
  xlab("Age [years]") +
  ylab("Age density") +
  scale_color_manual(values=c("#009999","#FF6633","#660099")) +
  guides(color=guide_legend(
    keywidth=0.33,
    keyheight=0.33,
    default.unit="inch"))
```

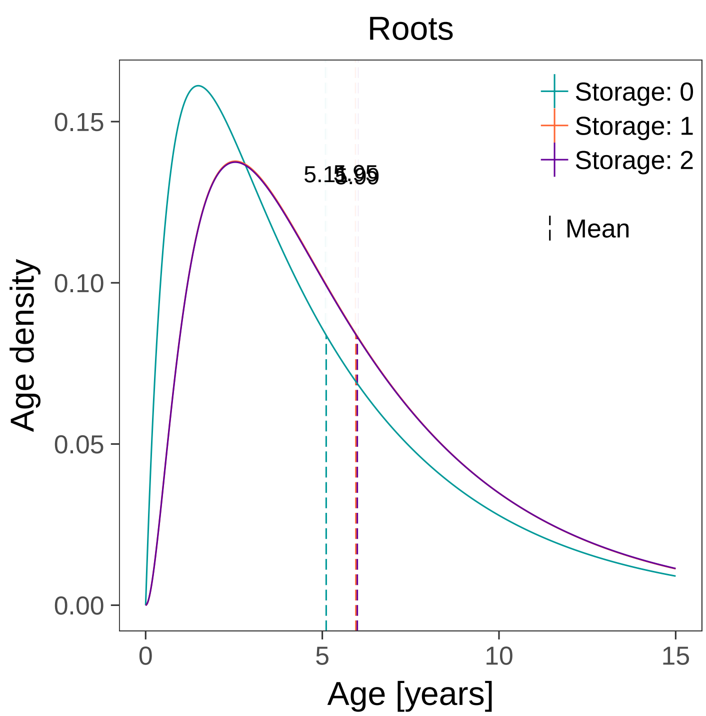

Figure 14: plot of chunk sp_Age_dist_Cr

Comparing fast cycling storage compartments:

```
knitr::opts_chunk$set(dev = 'pdf')
as_b <- data.frame("x"=Age, "Cs"=AgeVEGb$poolAgeDensity[,2])
as_b$name <- "Storage: 1"
as_c <- data.frame("x"=Age, "Cs"=AgeVEGc$poolAgeDensity[,2])
as_c$name <- "Storage: 2"
DF_s <- do.call(rbind, list(as_b, as_c))

asm_b <- data.frame("Cs"=AgeVEGb$meanPoolAge[2,1])
is.num <- sapply(asm_b, is.numeric)
asm_b[is.num] <- lapply(asm_b[is.num], round, 2)
asm_b$Cs_dens <- as.numeric(subset(DF_s, name == "Storage: 1" & x==asm_b$Cs, select = Cs))
asm_b$name <- "Storage: 1"
asm_c <- data.frame("Cs"=AgeVEGc$meanPoolAge[2,1])
is.num <- sapply(asm_c, is.numeric)
```

```
asm_c[is.num] <- lapply(asm_c[is.num], round, 2)
asm_c$Cs_dens <- as.numeric(subset(DF_s, name == "Storage: 2" & x==asm_c$Cs, select = Cs))
asm_c$name <- "Storage: 2"
DF_sm <- do.call(rbind, list(asm_b, asm_c))

ggplot() +
  ggtitle("Fast cycling storage") +
  geom_vline(data=DF_sm,aes(xintercept=Cs,linetype = "Mean",colour=name)) +
  scale_linetype_manual(values="longdash") +
  geom_segment(data = DF_sm, aes(x = Cs, y = Cs_dens, xend = Cs, yend = Inf), colour="white") +
  geom_line(data=DF_s, aes(x=x,y=Cs,colour=name)) +
  theme_bw(23) +
  geom_text(data = DF_sm, aes(label = Cs, y=Cs_dens+0.08, x=Cs), size=6) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = c(0.85,1), legend.justification = c(0.5,1),
        legend.background = element_rect(fill = 'transparent'),
        plot.title = element_text(size = rel(1), hjust = 0.5, colour = "black"),
        legend.title = element_blank()) +
  xlim(0,5) +
  xlab("Age [years]") +
  ylab("Age density") +
  scale_color_manual(values=c("#FF6633","#660099")) +
  guides(color=guide_legend(
    keywidth=0.33,
    keyheight=0.33,
    default.unit="inch"))
```

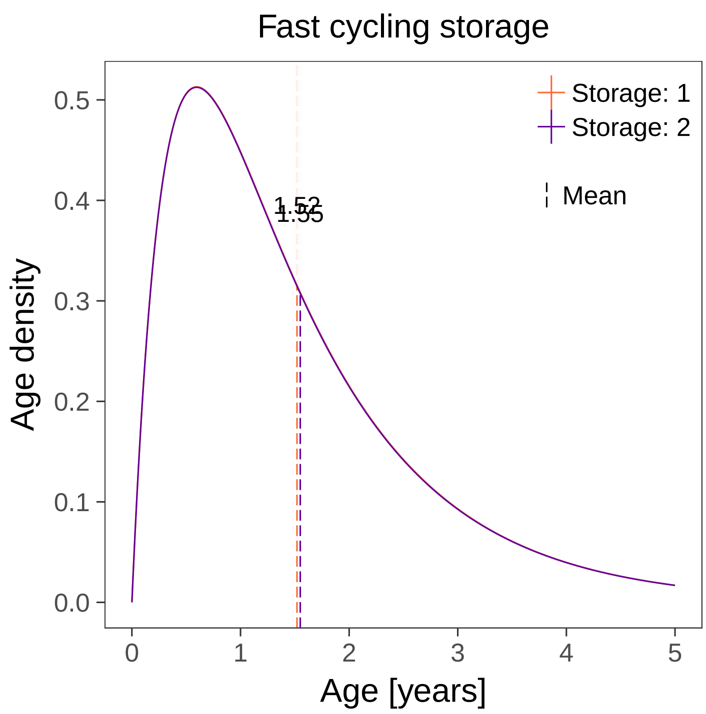

Figure 15: plot of chunk sp_df_storage_age

Plotting only the slow cycling storage compartment:

```
knitr::opts_chunk$set(dev = 'pdf')
DF_s <- data.frame("x"=Age, "Cs"=AgeVEGc$poolAgeDensity[,3])
DF_s$name <- "Storage: 2"

DF_sm <- data.frame("Cs"=AgeVEGc$meanPoolAge[3,1])
is.num <- sapply(DF_sm, is.numeric)
DF_sm[is.num] <- lapply(DF_sm[is.num], round, 2)
DF_sm$Cs_dens <- as.numeric(subset(DF_s, name == "Storage: 2" & x==DF_sm$Cs, select = Cs))
```

```
DF_sm$name <- "Storage: 2"

ggplot() +
  ggtitle("Slow cycling storage") +
  geom_vline(data=DF_sm,aes(xintercept=Cs,linetype = "Mean",colour=name)) +
  scale_linetype_manual(values="longdash") +
  geom_segment(data = DF_sm, aes(x = Cs, y = Cs_dens, xend = Cs, yend = Inf), colour="white") +
  geom_line(data=DF_s, aes(x=x,y=Cs,colour=name)) +
  theme_bw(23) +
  geom_text(data = DF_sm, aes(label = Cs, y=Cs_dens+0.01, x=Cs), size=6) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = c(0.85,1), legend.justification = c(0.5,1),
        legend.background = element_rect(fill = 'transparent'),
        plot.title = element_text(size = rel(1), hjust = 0.5, colour = "black"),
        legend.title = element_blank()) +
  xlim(0,100) +
  xlab("Age [years]") +
  ylab("Age density") +
  scale_color_manual(values=("#660099")) +
  guides(color=guide_legend(
    keywidth=0.33,
    keyheight=0.33,
    default.unit="inch"))
```

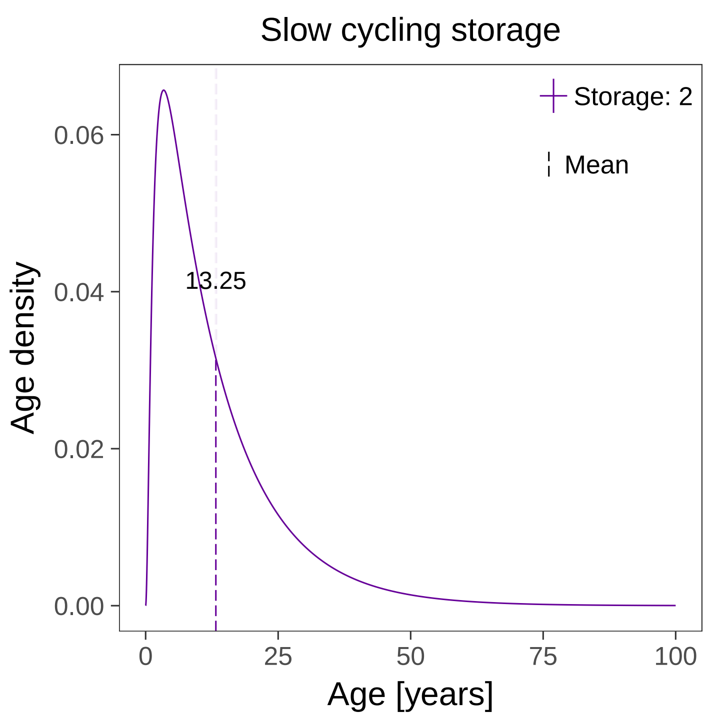

Figure 16: plot of chunk sp_df_slow_storage_age

And the transit time distribution

```
knitr::opts_chunk$set(dev = 'pdf')
ggplot() +
  geom_vline(data=DF_m,aes(xintercept=mean_tt,linetype = "Mean",colour=name)) +
  geom_vline(data=DF_m,aes(xintercept=median_tt,linetype = "Median",colour=name)) +
  scale_linetype_manual(values=c("longdash","dotted")) +
  #geom_segment(data = DF_m, aes(x = mean_tt, y = mean_tt_dens, xend = mean_tt, yend = Inf), colour="white") +
  #geom_segment(data = DF_m, aes(x = median_tt, y = median_tt_dens, xend = median_tt, yend = Inf), colour="white") +
  geom_line(data = DF, aes(x=x,y=tt,colour=name)) +
  theme_bw(23) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
```

```r
      legend.position = c(0.6,1), legend.justification = c(1,1),
      legend.background = element_rect(fill = 'transparent'),
      legend.title = element_blank()) +
  xlim(0,16) +
  xlab("Time [years]") +
  ylab("Transit time density") +
  scale_color_manual(values=c("#009999","#FF6633","#660099")) +
  guides(color=guide_legend(
    keywidth=0.33,
    keyheight=0.33,
    default.unit="inch"))
```

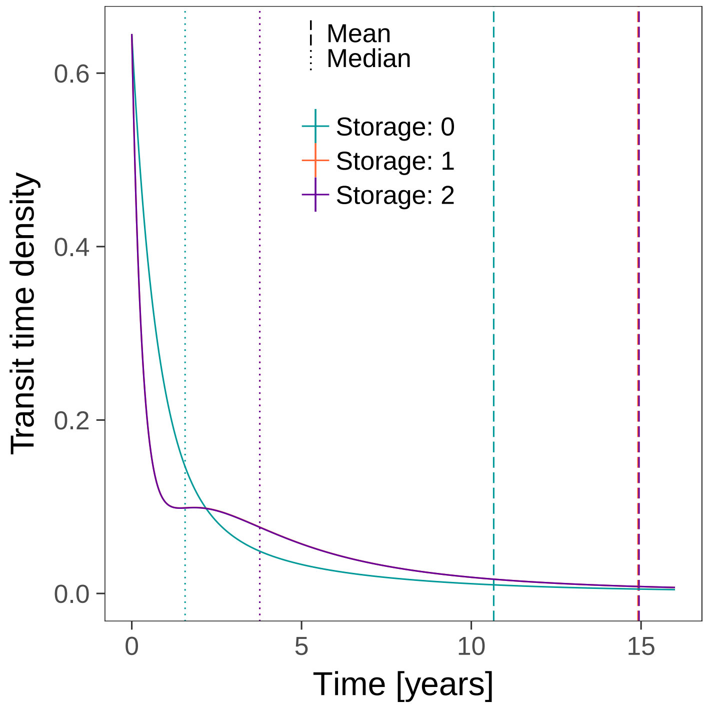

Figure 17: plot of chunk sp__Transit__time__distribution

Summary

This workflow shows how to implement three models with different CA schemes and run them in order to obtain important model diagnostics. It was possible to fit the models to the available data, but it is to be taken into account that the collinearity was very high. This problem may be solved by restructuring the models in order to eliminate parameter redundancies or compensations, or by constraining the parameter estimations with more data on C stocks and ages. The difference in the model structures had a lesser impact in some properties, but overall it resulted in different predictions of age and transit time distributions.

References

Bartlett, Megan K., Scott V. Ollinger, David Y. Hollinger, Haley F. Wicklein, and Andrew D. Richardson. 2011. "Canopy-Scale Relationships Between Foliar Nitrogen and Albedo Are Not Observed in Leaf Reflectance and Transmittance Within Temperate Deciduous Tree Species." *Botany* 89 (7): 491–97. doi:10.1139/b11-037.

Canadell, Josep G., Corinne Le Quéré, Michael R. Raupach, Christopher B. Field, Erik T. Buitenhuis, Philippe Ciais, Thomas J. Conway, Nathan P. Gillett, R. A. Houghton, and Gregg Marland. 2007. "Contributions to Accelerating Atmospheric CO_2 Growth from Economic Activity, Carbon Intensity, and Efficiency of Natural Sinks." *Proceedings of the National Academy of Sciences* 104 (47): 18866–70.

Fox, Andrew, Mathew Williams, Andrew D. Richardson, David Cameron, Jeffrey H. Gove, Tristan Quaife, Daniel Ricciuto, et al. 2009. "The REFLEX Project: Comparing Different Algorithms and Implementations for the Inversion of a Terrestrial Ecosystem Model Against

Eddy Covariance Data.” *Agricultural and Forest Meteorology* 149 (10): 1597–1615. doi:http://dx.doi.org/10.1016/j.agrformet.2009.05.002.

Lacointe, André. 2000. “Carbon Allocation Among Tree Organs: A Review of Basic Processes and Representation in Functional-Structural Tree Models.” *Annals of Forest Science* 57 (5-6): 521–33. doi:10.1051/forest:2000139.

Luo, Yiqi, Luther W. White, Josep G. Canadell, Evan H. DeLucia, David S. Ellsworth, Adrien Finzi, John Lichter, and William H. Schlesinger. 2003. “Sustainability of Terrestrial Carbon Sequestration: A Case Study in Duke Forest with Inversion Approach.” *Global Biogeochemical Cycles* 17 (1): n/a–n/a. doi:10.1029/2002GB001923.

Schiestl-Aalto, Pauliina, Liisa Kulmala, Harri Mäkinen, Eero Nikinmaa, and Annikki Mäkelä. 2015. “CASSIA – a Dynamic Model for Predicting Intra-Annual Sink Demand and Interannual Growth Variation in Scots Pine.” *New Phytologist* 206 (2): 647–59. doi:10.1111/nph.13275.

Sitch, Stephen, B. Smith, I. C. Prentice, A. Arneth, A. Bondeau, W. Cramer, J. O. Kaplan, et al. 2003. “Evaluation of Ecosystem Dynamics, Plant Geography and Terrestrial Carbon Cycling in the LPJ Dynamic Global Vegetation Model.” *Global Change Biology* 9 (2): 161–85. doi:10.1046/j.1365-2486.2003.00569.x.