Manuscript prepared for Biogeosciences
with version 2015/09/17 7.94 Copernicus papers of the LATEX class copernicus.cls.
Date: 29 December 2015

SUPPLEMENTARY MATERIAL: Underestimation of boreal soil carbon stocks by mathematical soil carbon models linked to soil nutrient status

B. Ťupek[1], C. Ortiz[2], S. Hashimoto[3], J. Stendahl[2], J. Dahlgren[4], E. Karltun[2], and A. Lehtonen[1]

[1]Natural Resources Institute Finland, P.O. Box 18, 01301 Vantaa, Finland
[2]Swedish University of Agricultural Sciences, P.O. Box 7014, 75007 Uppsala, Sweden
[3]Forestry and Forest Products Research Institute, Tsukuba, Ibaraki 305-8687, Japan
[4]Swedish University of Agricultural Sciences, Skogsmarksgränd, 90183 Umeå, Sweden

Correspondence to: B. Ťupek (boris.tupek@luke.fi)

Content:

1 Source codes of the Yasso07, Q, and CENTURY model

1.1 Yasso07 model

https://code.google.com/p/yasso07ui/source/browse/trunk/y07_subroutine.f90

```
MODULE yasso
IMPLICIT NONE
CONTAINS
  SUBROUTINE mod5c(a,t,cl,init,inf,s,leac,z)
  !components separately
  IMPLICIT NONE
    !********************************************* &
     !GENERAL DESCRIPTION FOR ALL THE MEASUREMENTS
    !********************************************* &
    !returns the model prediction for given parameters
    ! 1-16 matrix A entries: 4*k, 12*p
    !17-19 Climate-dependence parameters: b1, b2, g1
    !20-22 Leaching parameters: f1, f2, f3 IGNORED IN THE Y07-UI
    !23-25 Woody parameters
```

```
!26-27 Humus parametens: kH, pH
REAL,DIMENSION(27),INTENT(IN) :: a !parameters
REAL,INTENT(IN) :: t,s,leac !time,size,leaching
REAL,DIMENSION(3),INTENT(IN) :: cl !climatic conditions
REAL,DIMENSION(5),INTENT(IN) :: init
REAL,DIMENSION(5),INTENT(IN) :: inf !infall
REAL,DIMENSION(5),INTENT(OUT) :: z
REAL,DIMENSION(5,5) :: m,mt,m2,mi
INTEGER :: i
REAL,PARAMETER :: pi=3.1415926535
REAL :: tem
REAL,DIMENSION(5) :: te
REAL,DIMENSION(5) :: z1,z2
!temperature annual cycle approximation
te(1)=cl(1)+4*cl(3)*(1/SQRT(2.0)-1)/pi
te(2)=cl(1)-4*cl(3)/SQRT(2.0)/pi
te(3)=cl(1)+4*cl(3)*(1-1/SQRT(2.0))/pi
te(4)=cl(1)+4*cl(3)/SQRT(2.0)/pi
tem=0.0
DO i=1,4 !Annual cycle, different models
  tem=tem+EXP(a(17)*te(i)+a(18)*te(i)**2.0)/4.0 !Gaussian
END DO
!Precipitation dependence
tem=tem*(1.0-EXP(a(19)*cl(2)/1000))
!Size class dependence - - no effect if sc = 0.0
m(1,1)=a(1)*tem*MIN(1.0,(1.0+a(23)*s+a(24)*s**2.0)**a(25))
m(2,2)=a(2)*tem*MIN(1.0,(1.0+a(23)*s+a(24)*s**2.0)**a(25))
m(3,3)=a(3)*tem*MIN(1.0,(1.0+a(23)*s+a(24)*s**2.0)**a(25))
m(4,4)=a(4)*tem*MIN(1.0,(1.0+a(23)*s+a(24)*s**2.0)**a(25))
!Calculating matrix M, normal
m(2,1)=a(5)*ABS(m(2,2))
m(3,1)=a(6)*ABS(m(3,3))
m(4,1)=a(7)*ABS(m(4,4))
m(5,1)=0.0
m(1,2)=a(8)*ABS(m(1,1))
m(3,2)=a(9)*ABS(m(3,3))
m(4,2)=a(10)*ABS(m(4,4))
```

```
  m(5,2)=0.0
  m(1,3)=a(11)*ABS(m(1,1))
  m(2,3)=a(12)*ABS(m(2,2))
  m(4,3)=a(13)*ABS(m(4,4))
  m(5,3)=0.0
  m(1,4)=a(14)*ABS(m(1,1))
  m(2,4)=a(15)*ABS(m(2,2))
  m(3,4)=a(16)*ABS(m(3,3))
  m(5,4)=0.0
  m(5,5)=a(26)*tem !no size effect in humus
  DO i=1,4
    m(i,5)=a(27)*ABS(m(i,i)) !mass flows EWAN -> H
  END DO
  !Leaching
  m(1,1)=m(1,1)+leac*cl(2)/1000
  m(2,2)=m(2,2)+leac*cl(2)/1000
  m(3,3)=m(3,3)+leac*cl(2)/1000
  m(4,4)=m(4,4)+leac*cl(2)/1000
  !DY solution starts here...
  DO i=1,5
    z1(i)=DOT_PRODUCT(m(:,i),init)+inf(i)
  END DO
  mt=m*t
  CALL matrixexp(mt,m2)
  DO i=1,5
    z2(i)=DOT_PRODUCT(m2(:,i),z1)-inf(i)
  END DO
  CALL inverse(m,mi)
  DO i=1,5
    z1(i)=DOT_PRODUCT(mi(:,i),z2)
  END DO
  z=z1
CONTAINS
    SUBROUTINE matrixexp(a,b)
    IMPLICIT NONE
      !returns approximated matrix exponential
      !Taylor approximation.. another algorithm perhaps?
      REAL,DIMENSION(5,5),INTENT(IN) :: a
      REAL,DIMENSION(5,5),INTENT(OUT) :: b
      REAL,DIMENSION(5,5) :: c,d
      REAL :: p,normiter
      INTEGER :: i,q,j
      q=10
      b=0.0
      DO i=1,5
        b(i,i)=1.0
      END DO
      normiter=2.0
      j=1
      CALL matrix2norm(a, p)
      DO
        IF(p< normiter)THEN
          EXIT
        END IF
        normiter=normiter*2.0
        j=j+1
      END DO
      c=a/normiter
      b=b+c
      d=c
      DO i=2,q
        d=MATMUL(c,d)/REAL(i)
        b=b+d
      END DO
      DO i=1,j
        b=MATMUL(b,b)
      END DO
    END SUBROUTINE matrixexp
    SUBROUTINE matrix2norm(a,b)
    IMPLICIT NONE
      !returns matrix 2-norm with cartesian vector x,
      !| | x| | = 1
      !square matrix input (generalize if necessary)
      REAL,DIMENSION(5,5),INTENT(IN) :: a
```

```
    REAL,INTENT(OUT) :: b
    INTEGER :: n,i
    n=SIZE(a(1,:))
    b=0.0
    DO i=1,n
      b=b+SUM(a(:,i))**2.0
    END DO
    b=SQRT(b)
  END SUBROUTINE matrix2norm
  SUBROUTINE inverse(a,b)
  IMPLICIT NONE
    !returns an inverse of matrix a
    !(column elimination strategy)
    !input has to be a square matrix, otherwise erroneous
    REAL,DIMENSION(5,5),INTENT(IN)  :: a
    REAL,DIMENSION(5,5),INTENT(OUT)  :: b
    REAL,DIMENSION(5,5) :: c
    INTEGER :: n,m,i,j
    n=SIZE(a(1,:))
    m=SIZE(a(:,1))
    IF(m/=n) THEN
      WRITE(*,*) " Does not compute."
      WRITE(*,*) " No square matrix input."
      WRITE(*,*) " Error in function: inverse"
    ELSE
!      ALLOCATE(b(n,n),c(n,n))
      c=a
      b=0.0
      DO i=1,n !setting b a unit matrix
        b(i,i)=1.0
      END DO
      DO i=1,n
      !what if diagonal values are zeros?
        IF(c(i,i)==0.0)THEN!case of singuar matrix, is it?
          b(i,:)=0.0
          c(i,:)=0.0
          b(:,i)=0.0
```

```
            c(:,i)=0.0
            !        b(i,i)=1.0
            !        c(i,i)=1.0
          ELSE
            b(i,:)=b(i,:)/c(i,i)
            c(i,:)=c(i,:)/c(i,i)
          END IF
          DO j=1,i-1
            b(j,:)=b(j,:)-b(i,:)*c(j,i)
            c(j,:)=c(j,:)-c(i,:)*c(j,i)
          END DO
          DO j=i+1,n
            b(j,:)=b(j,:)-b(i,:)*c(j,i)
            c(j,:)=c(j,:)-c(i,:)*c(j,i)
          END DO
        END DO
        IF(c(n,n)==0.0)THEN
          b(n,:)=0.0
          b(:,n)=0.0
          !      b(n,n)=1.0
        ELSE
          b(n,:)=b(n,:)/c(n,n)
        END IF
      !now, b is supposed to be the requested inverse
      END IF
    END SUBROUTINE inverse
END SUBROUTINE mod5c
END MODULE yasso
```

1.2 Q model

```
!********************************************* &
 !     Main program to calculate carbon store in forest soils.
 !     SPRUCE
!********************************************* &
!     Edited by Carina Ortiz, Version 2015-11-16
!     Understory vegetation, variable temp, variable litter
```

```
      real, dimension(1) :: yr(0:0),carb(0:0),
     +                        totc(0:0),u0y(0:0)
  real, dimension(1,5) :: lit(0:0,5),cssn(0:0,1),
     + vtempsite(0:0,1:244),cssnund(0:0,1),ssi0b(0:0,1),
     + ssi0scr(0:0,1),ssi0s(0:0,1),alfanss(0:0,1),
z(0:0,1),u0(0:0,1)
*    ,ssi0b(0:0,5),ssi0scr(0:0,5),ssi0s(0:0,5)
*     + ssi0scr(0:0,5),cssnund(0:0,5)
      real, dimension (244,0:0) :: nefr  !(jj,ii)
  real, dimension (244,0:0) :: br  !(jj,ii)
  real, dimension (244,0:0) :: st  !(jj,ii)
  real, dimension (244,0:0) :: stpcr  !(jj,ii)
  real, dimension (244,0:0) :: und  !(jj,ii)
  real, dimension (244,0:0) :: vtemp  !(jj,ii)
!  needlesfineroots jj columns(MCSIMS),  rows(yr) ii
      real i0, u0sum, ts
      character (len=10) :: dumtext
  integer :: r
!     Monte Carlo Simulations
!     Read file with parameter setups
!     Open parameter file
  OPEN (unit=111, file=’1parvaraccregsprucemean.dat’,status=’old’)
      read(111,*) dumtext
!  Open output file
  open(unit=11, file=’q_soil.dat’)
!      Read matrix file with litter production in N simulations
!      (Monte Carlo), each fraction one separate file
      open(unit=1111,file=’biom70reg1SpruceNeedlesandfineroots.dat’,

     +     status=’old’)
    read(1111,*) dumtext
    read(1111,*) nefr
  open(unit=11111,file=’biom70reg1SpruceBranches.dat’,
     +     status=’old’)
      read(11111,*) dumtext
           read(11111,*) br
  open(unit=1111111,file=’biom70reg1SpruceStumpCoarseroots.dat’,
     +     status='old')
      read(1111111,*) dumtext
      read(1111111,*) stpcr
      open(unit=111111,file='biom70reg1SpruceStem.dat',
     +     status='old')
      read(111111,*) dumtext
      read(111111,*) st
  open(unit=11111111,file='biom70reg1SpruceUnderveg.dat ',
     +     status='old')
      read(11111111,*) dumtext
           read(11111111,*) und
!  Read variable temperature file yearly means
  open(unit=1, file='reftempreg1spruce.dat',status='old')
    read(1,*) dumtext
    read(1,*) vtemp
!  Estimates the mean temperature for steady state
*         ts=ts+vtemp
*          ii=ii+1
*    temp=ts/ii  !mean temp
!  Main loop Number of Monte Carlo  simulations
  DO ii=1,244
  read(111,*) q0n,q0w,e0,eta11,beta,maxb,maxs,u00,u01
  ! Temperature for steady state.
  ! First row temperature, only for one step
    do jj=0,0!first row is dumtext !
      vtempsite(jj,ii) = vtemp(ii,jj)!
      u0(jj,1)=u00+u01*vtempsite(jj,ii)
          do i=0,0
          u0y(i)=u00+u01*vtempsite(jj,ii) ! each year new temp.
        enddo
      enddo
! Set some parameters for decomposition functions
  ssi0n=0
  fC=0.5
  do jj=0,0
  alfanss(jj,1)=fC*beta*eta11*u0(jj,1)*q0n**beta
  z(jj,1)=(1.-e0)/(beta*eta11*e0)
  end do
! Create litter matrix
    do jj=0,0!first row is dumtext
      lit(jj,1) = nefr(ii,jj)
      !litter fraction 1 needles & fine roots
      lit(jj,2) = br(ii,jj)
      !litter fraction 2 branches & roots
      lit(jj,3) = st(ii,jj)
      !litter fraction 3 stems
      lit(jj,4) = stpcr(ii,jj)
      !litter fraction 4 stump & coarse roots
      lit(jj,5) = und(ii,jj)
      !litter fraction 5 understorey
    enddo
!Calculates steady state litter fraction for needles
  do jj=0,0
  cssn(jj,1)=(lit(jj,1))*1/(alfanss(jj,1)*(z(jj,1)-1))
  enddo
!Steady state understorey vegetation
  do jj=0,0
  cssnund(jj,1)=(lit(jj,5))*1/(alfanss(jj,1)*(z(jj,1)-1))
  enddo
!Steady state input branches
  do jj=0,0
  ssi0b(jj,1)=lit(jj,2)
  enddo
!Steady state input stump and coarse roots
  do jj=0,0
  ssi0scr(jj,1)=lit(jj,4)
  enddo
!Steady state input stem
  do jj=0,0
  ssi0s(jj,1)=lit(jj,3)
  enddo
!  Fixed parameters
  fC=0.5
!     Call for soil decomposition model
```

```
      call soildecomp(fC,u0,u0y,eta11,e0,beta,ssi0b,ssi0s,ssi0scr,
     +                q0n,cssn,q0w,maxb,maxs,itend,lit,carb,nitr,
     +                cssnund)
!  WRITE TO OUTPUTFILE
*  do i=0,0
*  write(11,*) carb(i)
*  enddo
!  End main loop
*  enddo !input litter loop
  ENDDO !parameter loop
*!  Close Temperature file
  close(unit=1)
!  Close litterfiles
  close(unit=1111)
        close(unit=11111)
  close(unit=111111)
  close(unit=1111111)
  close(unit=11111111)
!  Close OUTPUT document
  close(unit=11)
!  Close parameterrange file
  close(unit=111)
!********************************************* &
!     Subroutine to calculate carbon stores in forest soils
!     Version 2015-11-16
!     Undervegetation, variable temp, variable litter
      subroutine soildecomp (fC,u0,u0y,eta11,e0,beta,ssi0b,ssi0s,
     +  ssi0scr,q0n,cssn,q0w,maxb,maxs,itend,lit,carb,nitr,cssnund)
      real, dimension(1) :: carb(0:0), totc(0:0),
     +            u0y(0:0)
       real, dimension(1,5) :: lit(0:0,5), vtempsite(0:0,1:225)
        real, dimension (225,0:0) :: nefr  !(jj,ii)
  real, dimension (225,0:0) :: br  !(jj,ii)
  real, dimension (225,0:0) :: st  !(jj,ii)
  real, dimension (225,0:0) :: stpcr  !(jj,ii)
  real, dimension (225,0:0) :: und  !(jj,ii)
  real, dimension (225,0:0) :: vtemp  !(jj,ii)
```

```
*  needlesfineroots jj columns(MCSIMS),  rows(yr) ii
      real i0, u0sum, t, gnv, gnold, gbv, gbold, gsv, gsold
      dimension gn(0:0), gb(0:0), gs(0:0)
      dimension hn(0:0), hb(0:0), hs(0:0), qn(0:0)
*     real, dimension (101,100):: a(0:100,1:100)
      integer :: r
!     Litter and OM decomposition parameters
  z=(1.-e0)/(beta*eta11*e0)
  zn=1./(beta*eta11*e0)
  alfa =fC*beta*eta11*u0*q0w**beta
  !Alfa with constant temp!
  alfav=fC*beta*eta11*q0w**beta
  !Alfa with variable temperature vtemp!
         alfan=fC*beta*eta11*u0*q0n**beta
         !Old alfa with constant temp!
  alfanv=fC*beta*eta11*q0n**beta
  !Alfa with variable temperature vtemp!
         alfa0n=fN/fC-(beta*eta11*e0+e0-1)*(fn/fc-r0)/(beta*eta11*e0-1)
!     Old SOM from steady state org.mat. is calculated and summed
      do i=0,0
  u0sum=0
  do k=1,i !No decomposition the same year as we assess
     u0sum=u0sum+u0y(k)
     !Integral of u0 when using variable temperature
  enddo
  gnold= (1+alfanv*u0sum)**(1-(1-e0)/
  !Decomposition of old needles
     +  (e0*eta11*beta))
  t=(i)
!     Call for decomposition fuctions
!     that calculates the remaining mass of the o.m.
      gsold=gold(t,maxs,alfa,alfav,u0sum,z)
  gbold=gold(t,maxb,alfa,alfav,u0sum,z)
!      Old SOM is summed
        carb(i)=
     +  gnold*cssn !part of o.m left  and the steady state input
in fraction
```

```
     +  +gnold*cssnund
     +    +gbold*ssi0b
     +    +gsold*ssi0s
     +    +gsold*ssi0scr
  write(11,*) carb (i)
  enddo
!   New SOM from is calculated
!   and summed with the olds steady state carbon
  do i=0,0
   do mt=0,i
   u0sum=0
      if (i.ne.mt) then
      !No decomposition the same year as we assess
    do k=(mt+1),i
       u0sum=u0sum+u0y(k)
       !Integral of u0 when using variable temperature
    enddo
      endif
  gnv=1./(1.+alfanv*u0sum)**z
  !Decomposition of new needles!
  t=(i-mt)
  gbv=gbran(t,maxb,alfa,alfav,u0sum,z)
  !Decomposition of new branches!
  if (gbv.gt.1) then
  gbv=1
  endif
  gsv=gbran(t,maxs,alfa,alfav,u0sum,z)
  !Decomposition of new stems!
  if (gbs.gt.1) then
  gbs=1
  endif
!     SOM from old steady state and new litter is summed
  carb(i)=carb(i)+
     +       lit(mt,1)*gnv+
     +       lit(mt,2)*gbv+
     +       (lit(mt,3)+lit(mt,4))*gsv+
     +    lit(mt,5)*gnv
  enddo
       enddo
!      Litter fractions are 1=needles 2=branches
!      3=stem 4=stump & course roots 5=under vegetation
9991   continue
       return
999    end
!     Function for decomposition of old steady state carbon
!     for branches (2) and stems(3+4)
      function gold(t,itmax,alfa,alfav,u0sum,z)
      !Decomposition of old branches & stem!
  tmax=real(itmax)
  if (t.le.tmax) then
  gold=-2.*(1.+alfav*u0sum)**(2.-z)/(tmax*alfa**2*(1.-z)*(2.-z))
     ++2.*((1.+alfav*u0sum)**(3.-z)-1.)
     +  /(tmax**2*alfa**3*(1.-z)*(2.-z)*(3.-z))
     +  +(1.-t/tmax)**3*tmax/3.-(1.-t/tmax)**2/(alfa*(1.-z))
     +  +2.*(tmax-t)/(tmax**2*alfa**2*(1.-z)*(2.-z))
  return
  else
  gold=-2.*(1+alfav*u0sum)**(2.-z)/(tmax*alfa**2*(1.-z)*(2.-z))
     +   2.*((1.+(alfav*u0sum-alfa*(tmax)))**(3.-z)-
     +  (1+alfav*u0sum)**(3.-z))
     +  /(tmax**2*alfa**3*(1.-z)*(2.-z)*(3.-z))
  return
  endif
  end
      function gbran(t,itmax,alfa,alfav,u0sum,z)
      !Decomposition of new branches!
      tmax=real(itmax)
      if (t.le.tmax) then
      gbran=2.*((1+alfav*u0sum)**(1.-z)-(1.-t/tmax))/(tmax*alfa*(1.-z))
     +   +2.*(1.-(1.+alfav*u0sum)**(2.-z))/((tmax**2)*(alfa**2)*
     +   (1.-z)*(2.-z))+(1.-(t/tmax))**2
      return
      else
      gbran=2.*(1.+alfav*u0sum)**(1.-z)/(tmax*alfa*(1.-z))
     + +2.*((1.+(alfav*u0sum-alfa*tmax))**(2.-z)-
     +(1.+alfav*u0sum)**(2.-z))/
     +    ((tmax**2)*(alfa**2)*(1.-z)*(2.-z))
      return
      endif
      end
```

1.3 CENTURY model

```
########################################################
##
##   SOC sub-model of the CENTURY version 4.0
##
########################################################
#
# Coded in R by Shoji Hashimoto (shojih@ffpri.affrc.go.jp)
# edited by Boris Tupek (boris.tupek@luke.fi)
# original model available at
# https://www.nrel.colostate.edu/projects/century/obtain2.htm
######################################################
# Related source files in the original CENTURY model
# Please see /original/source/*.f
#
# adjlig.f, anerob.f, csa_detiv.f, csa_main.f, cycle.f, declig.f
# decomp.f, eachyr.f, h2olos.f, litdec.f, partit.f
# pevap.f, prelim.f, simsom.f, somdec.f, tcalc.f
# wdeath.f, woodec.f, and so on.
#
######################################################
# Simplification:
#
# only for forest ecocystem (not grass, savanna etc)
# no irrigation
# not floating C/N ratio for plant organs.
# cnr_max=cnr_min=cnr_initial in tree.100
# no mineral N cycling: constant N at surface soil
# (xNmineral in f_site.100)
# drain=1, anerb=1
# idef=2 in fix.100 (water function for calculating defac)
# no CO2 effect
#
#####################################################
# A bug in the original CENTURY
#
# a bug (please see calfc_wtpt function below)
# The difference in results was small,
# but it depends on the climate and soil.
#
# BFix<-0: with bug as the original CENTURY
# BFix<-1: the bug was fixed
rm(list=ls())
options(digits=12)
BFix<-1
#DEFINE number of years for spinup simulations!
TSTART=1
TEND=500 # 5000 for steady state
#################################
# Read data
#################################
# parameters from fix.100 in the original CENTURY
# environments (site specific temperature,
# precipitation from SMHI), site.100 in the original CENTURY
# parameters describing site conditions(site specific sand,
# silt,clay,bulk density from SFSI data)
# see file site.100 in the original CENTURY
# parameters describing tree,
# see tree.100 in the original CENTURY
# "AND H_J ANDREWS" for conifers
# "Coweeta" for deciduous
# initial conditions from site.100
## READ SITE SPECIFIC data ######################################
#general parameters (fix.100)
parameters.names<-c("adep1","adep2","adep3","adep4","adep5",
                     "adep6","adep7","adep8","adep9","adep10",
                     "awtl1","awtl2","awtl3","awtl4","awtl5",
                     "awtl6","awtl7","awtl8","awtl9","awtl10",
                     "damr11","damr21","damrmn","dec11,
                     Asrfstr_0","dec21,Asrfmet_0","dec12,
                     Abelstr_0","dec22,Abelmet_0","dec31,
                     Asrfmic_0","dec32,kactv_0","dec5,kslow_0",
                     "dec4,kpass_0","Edepth","Elitst",
                     "Fwloss1","Fwloss2","Fwloss3","Fwloss4",
                     "ntspm,CYCL","OMLECH(1)","OMLECH(2)",
                     "OMLECH(3)","P1CO2A1","P1CO2A2","P1CO2B1",
                     "P1CO2B2","P2CO2","P3CO2","pabres",
                     "Peftxa","Peftxb","pligst1","pligst2",
                     "PMCO21","PMCO22","PmnTmp","PmxBio",
                     "PmxTmp","PS1CO21","PS1CO22","PS1S31",
                     "PS1S32","PS2S31","PS2S32","Rsplig",
                     "spl1","spl2","strmax1","strmax2",
                     "teff1","teff2","teff3","Tmelt1","Tmelt2")
parameters.values <-c(15,15,15,15,30,30,30,30,0,0,0.8,
                      0.6,0.4,0.3,0.2,0.2,0.2,0.2,0,0,0,
                      0.02,15,3.9,14.8,4.9,18.5,6,7.3,
                      0.2,0.0045,0.2,0.4,0.8,0.8,0.65,
                      0.9,4,0.03,0.12,60,0.6,0.17,0,
                      0.68,0.55,0.55,100,0.25,0.75,3,
                      3,0.55,0.55,0.004,600,-0.0035,
                      0.45,0.55,0.003,0.032,0.003,
                      0.009,0.3,0.85,0.013,5000,
                      5000,0,0.125,0.07,-8,4)
parameters <- data.frame(V1=parameters.values,
                         V2=parameters.names)
#initial parameters (site.100)
init.names<-c("xsrfstr","xsrfmet","xsrfmic","xbelstr",
               "xbelmet","xactv","xslow","xpass",
               "xwood1","xwood2","xwood3",
               "rwcf_initial1","rwcf_initial2",
               "rwcf_initial3","rwcf_initial4",
               "rwcf_initial5","rwcf_initial6",
               "rwcf_initial7","rwcf_initial8",
               "rwcf_initial9","rwcf_initial10",
               "asmos1","asmos2","asmos3","asmos4",
               "asmos5","asmos6","asmos7","asmos8",
               "asmos9","asmos10","asmos11","snql",
               "snow","srfstrlig","belstrlig")
init.values <-c(240,60,60,186.5,113.4,130,2570,
                1596,500,500,500,0.5,0.5,0.5,0.5,
                0.5,0.5,0.5,0.5,0.5,0.5,0.2,
                0.2,0.2,0.2,0.2,0.2,0.2,0.2,
                0.2,0.2,0.2,0,0,0.275,0.354)
init <- data.frame(V1=init.values,V2=init.names)
#site (site.100)
site.parameters.names <-c("sitlat","sitlog",
                          "sand","silt", "clay", "bd",
                          "nlayer","nlaypg", "drain",
                          "basef","stormf",
                          "SWFLAGflag_fc_wtpt(0useactual,1.0Guputa)",
                          "AWILT1","AWILT2", "AWILT3",
                          "AWILT4","AWILT5", "AWILT6",
                          "AWILT7","AWILT8", "AWILT9",
                          "AWILT10",
                          "AFIEL1","AFIEL2", "AFIEL3",
                          "AFIEL4","AFIEL5", "AFIEL6",
                          "AFIEL7","AFIEL8", "AFIEL9",
                          "AFIEL10","elev", "xNmineral")
#site sand,silt, clay, bulk density
site.parameters.values <-c(59.36,13.47,0.55,0.15,0,1.226,
                           8,5,1,0.5,0.9,1,0.2,0.2,0.2,0.2,
                           0.2,0.2,0.2,0.2,0.2,0.2,0.3,0.3,
                           0.3,0.3,0.3,0.3,0.3,0.3,0.3,0.3,
                           50,1.65)
site.parameters <- data.frame(V1=site.parameters.values,
                              V2=site.parameters.names)
#climate environment (site.100)
envi.parameters.names <-c("Prec(1)cm","Prec(2)cm",
                          "Prec(3)cm","Prec(4)cm", "Prec(5)cm",
                          "Prec(6)cm","Prec(7)cm",
                          "Prec(8)cm","Prec(9)cm", "Prec(10)cm",
                          "Prec(11)cm","Prec(12)cm",
                        "Tmin(1)degree","Tmin(2)degree",
                        "Tmin(3)degree","Tmin(4)degree",
                        "Tmin(5)degree","Tmin(6)degree",
                        "Tmin(7)degree","Tmin(8)degree",
                        "Tmin(9)degree","Tmin(10)degree",
                        "Tmin(11)degree","Tmin(12)degree",
                        "Tmax(1)degree","Tmax(2)degree",
                        "Tmax(3)degree","Tmax(4)degree",
                        "Tmax(5)degree","Tmax(6)degree",
                        "Tmax(7)degree","Tmax(8)degree",
                        "Tmax(9)degree","Tmax(10)degree",
                        "Tmax(11)degree","Tmax(12)degree")
envi.parameters.values <-c(3.395,2.695,2.884,3.051,3.306,
                           4.471,4.623,6.016,5.494,5.221,
                           5.659,3.858,-6.647,-7.235,-4.201,
                           -0.121,4.97,9.538,11.74,11.038,7.266,
                           3.379,-1.027,-5.32,-0.782,-0.375,3.613,
                           9.369,15.549,19.758,21.351,20.219,
                           15.489,9.915,4.286,0.584)
envi.parameters <- data.frame(V1=envi.parameters.values,
                              V2=envi.parameters.names)
#tree
tree.parameters.names <-c("cerfor(1:2:3,1,1),cnr_fol",
                          "cerfor(1:2:3,3,1),cnr_bra",
                          "cerfor(1:2:3,4,1),cnr_ste",
                          "cerfor(1:2:3,2,1),cnr_fir",
                          "cerfor(1:2:3,5,1),cnr_cor",
                          "DECW1,kwood1_0,bra","DECW2,
                          kwood2_0,ste","DECW3,kwood3_0,cor",
                          "forrtf","leafdr1", "leafdr2",
                          "leafdr3","leafdr4", "leafdr5",
                          "leafdr6","leafdr7", "leafdr8",
                          "leafdr9","leafdr10", "leafdr11",
                          "leafdr12",
                          "wdlig1,cfol_lig","wdlig3,cbra_lig",
                         "wdlig4,cste_lig","wdlig2,cfir_lig",
                          "wdlig5,ccor_lig",
                          "wooddr1fol","wooddr3bra",
                          "wooddr4ste","wooddr2fir", "wooddr5cor")
tree.parameters.values <-c(20,99,140,40,83,1.5,0.5,0.6,
                           0.5,0,0,0,0.002,0.006,0.012,
                           0.013,0.039,0.175,0.664,0.066,
                           0.023,0.223,0.25,0.25,0.25,0.25,
                           1,0.01,0.002,0.04,0.004)
tree.parameters <- data.frame(V1=tree.parameters.values,
                              V2=tree.parameters.names)
# biomass components gC.m-2
biomass.in <- data.frame(id=1,
                foliage.tot70=795.954,
                branch.tot70=1241.235,
                wood.tot70=5110.385,
                fineroot.tot70=251.318,
                root.tot70=1652.101)
# litter components gC.m-2
litter.in <- data.frame(id=1,
                        foliage.lit.tot70=116.804,
                        branch.lit.tot70=15.515,
                        wood.lit.tot70=12.447,
                        fineroot.lit.tot70=131.778,
                        root.lit.tot70=20.651)
# Define objects from  SITE SPECIFIC PARAMETERS: ################
# environment(meteo), site, and tree parameters #################
#site specific parameters
envi <- envi.parameters
tree <- tree.parameters
site <- site.parameters
## define environment ################
# prec: monthly precipitation, cm
# atempmin: monthly minimum air temperature
# atempmax: monthly maximum air temperature
prec<-matrix(0,nrow=12,ncol=1)
atempmin<-matrix(0,nrow=12,ncol=1)
```

```
atempmax<-matrix(0,nrow=12,ncol=1)
for(m in 1:12)
{
  prec[m]<-envi[m,1]
  atempmin[m]<-envi[m+12,1]
  atempmax[m]<-envi[m+24,1]
}
## define site parameters ###########
# awilt: wilting point
# afiel: field capacity
sitlat<-site[1,1]
sitlog<-site[2,1]
sand<-site[3,1]
silt<-site[4,1]
clay<-site[5,1]
bd<-site[6,1]
#use mean soil parameters for swedish soils
#(if soil data is not available)
if (is.na(bd)){
  bd<-1.2
}
if(sum(sand,silt,clay)==0){
  silt<-0.45
  clay<-0.179
  bd<-0.029
}
nlayer<-as.integer(site[7,1])
nlaypg<-as.integer(site[8,1])
drain<-site[9,1]
basef<-site[10,1]
stormf<-site[11,1]
flag_fc_wtpt<-as.integer(site[12,1])
awilt<-matrix(0,nrow=10,ncol=1)
afiel<-matrix(0,nrow=10,ncol=1)
for(i in 1:10)
{
  awilt[i]<-site[12+i,1]
  afiel[i]<-site[22+i,1]
}
elev<-site[33,1]
xNmineral<-site[34,1]
## define init parameters ##########
# xsrfstr: surface structural
# xsrfmet: surface metabolic
# xsrfmic: surface microbe
# xbelstr: belowground structural
# xbelmet: belowground metabolic
# xactv: actic pool
# xslow: slow pool
# xpass: passive pool
# xwood1: branch litter
# xwood2: stem litter
# xwood3: coase root litter
# rwcf: volumetric soil water content
# asmos: soil water content of the ith soil layer cmh2o
xsrfstr<-init[1,1]
xsrfmet<-init[2,1]
xsrfmic<-init[3,1]
xbelstr<-init[4,1]
xbelmet<-init[5,1]
xactv<-init[6,1]
xslow<-init[7,1]
xpass<-init[8,1]
xwood1<-init[9,1]
xwood2<-init[10,1]
xwood3<-init[11,1]
tawood <- xwood1 + xwood2
tbwood <- xwood3
talit <- xsrfstr + xsrfmet + xsrfmic
tblit <- xbelstr + xbelmet
somsc <- xactv + xslow + xpass
somtc <- xactv + xslow + xpass + xbelstr + xbelmet
rwcf<-matrix(0.1,nrow=10,ncol=1)
for(j in 1:nlayer)
```

```
{
  rwcf[j]<-init[11+j,1]
}
asmos<-matrix(0.1,nrow=11,ncol=1)
for(j in 1:(nlayer+1))
{
  asmos[j]<-init[21+j,1]
}
snlq<-init[33,1]
snow<-init[34,1]
srfstrlig<-init[35,1]
belstrlig<-init[36,1]
##
## define tree parameters #############
# CN ratio of foliage, branch stem, fine roots, coarse roots
# Decomposition constant
# Translocation of N
# Lignin ratios
# Death rate
cnr_fol<-tree[1,1]
cnr_bra<-tree[2,1]
cnr_ste<-tree[3,1]
cnr_fir<-tree[4,1]
cnr_cor<-tree[5,1]
kwood1<-tree[6,1]
kwood2<-tree[7,1]
kwood3<-tree[8,1]
forrtf<-tree[9,1]
leafdr<-matrix(0,nrow=12,ncol=1)
for(j in 1:12)
{
  leafdr[j]<-tree[j+9,1]
}
cfol_lig<-tree[22,1]
cbra_lig<-tree[23,1]
cste_lig<-tree[24,1]
cfir_lig<-tree[25,1]
```

```
ccor_lig<-tree[26,1]
wooddr<-matrix(0,nrow=5,ncol=1)
for(j in 1:5)
{
  wooddr[j]<-tree[j+26,1]
}
## define main (FIX) parameters ################
# A: decomposition constant
# k: decomposition constant
adep<-matrix(0.1,nrow=10,ncol=1)
for(j in 1:10)
{
  adep[j]<-parameters[j,1]
}
#
#
awtl<-matrix(0,nrow=10,ncol=1)
for(j in 1:10)
{
  awtl[j]<-parameters[10+j,1]
}
damr11<-parameters[21,1]
damr21<-parameters[22,1]
damrmn<-parameters[23,1]
Asrfstr<-parameters[24,1]
Asrfmet<-parameters[25,1]
Abelstr<-parameters[26,1]
Abelmet<-parameters[27,1]
Asrfmic<-parameters[28,1]
kactv<-parameters[29,1]
kslow<-parameters[30,1]
kpass<-parameters[31,1]
Edepth<-parameters[32,1]
elitst<-parameters[33,1]
fwloss1<-parameters[34,1]
fwloss2<-parameters[35,1]
fwloss3<-parameters[36,1]
```

```
fwloss4<-parameters[37,1]
CYCL<-as.integer(parameters[38,1])
omlech<-matrix(0,nrow=3,ncol=1)
omlech[1]<-parameters[39,1]
omlech[2]<-parameters[40,1]
omlech[3]<-parameters[41,1]
P1CO2A1<-parameters[42,1]
P1CO2A2<-parameters[43,1]
P1CO2B1<-parameters[44,1]
P1CO2B2<-parameters[45,1]
Psrfmic<-P1CO2A1
Pactv<-P1CO2A2+P1CO2B2*sand
Pslow<-parameters[46,1]
Ppass<-parameters[47,1]
pabres<-parameters[48,1]
Peftxa<-parameters[49,1]
Peftxb<-parameters[50,1]
pligst1<-parameters[51,1]
pligst2<-parameters[52,1]
Psrfstr<-parameters[53,1]
Psrfmet<-parameters[54,1]
Pbelstr<-Psrfstr
Pbelmet<-Psrfmet
PmnTmp<-parameters[55,1]
PmxBio<-parameters[56,1]
PmxTmp<-parameters[57,1]
PS1CO21<-parameters[58,1]
PS1CO22<-parameters[59,1]
ps1s31<-parameters[60,1]
ps1s32<-parameters[61,1]
ps2s31<-parameters[62,1]
ps2s32<-parameters[63,1]
RSPLIG<-parameters[64,1]
spl1<-parameters[65,1]
spl2<-parameters[66,1]
strmax1<-parameters[67,1]
strmax2<-parameters[68,1]
```

```
teff1<-parameters[69,1]
teff2<-parameters[70,1]
teff3<-parameters[71,1]
Tmelt1<-parameters[72,1]
Tmelt2<-parameters[73,1]
#Biomass data from Swe Forest and Soil Inventory ##############
#biomass components gC.m-2
#biomass.in
pools.bfol<- biomass.in[1,2]
pools.bbra<-biomass.in[1,3]
pools.bste<-biomass.in[1,4]
pools.bfir<-biomass.in[1,5]
pools.bcor<-biomass.in[1,6]
#Litterfall SITE SPECIFIC data
litter.in <- litter.in
# Initialization #################
stempave<-0.0
defac<-0.0
pet<-0.0
anerb<-0.0
CO2out<-0.0
leaching<-0.0
pet<-matrix(0,nrow=12,ncol=1)
avh2o<-matrix(0.0,nrow=3,ncol=1)
amov<-matrix(0.0,nrow=11,ncol=1)
tran<-0.0
evap<-0.0
stream1<-0.0
cleach<-0.0
tcleach<-0.0
#######################################################################
#
## Functions of the CENTURY ###########################################
#
#######################################################################
## function (calpet ) #######
## potential evapotranspiration
```

```
calpet<-function()
{
  # Linacre(1977) from CENTURY source
  #As in the CENTURY
  elev<-0.0
  ave<-matrix(0,nrow=12,ncol=1)
  ave[1]<-(atempmax[1]+atempmin[1])/2.0
  highest<-ave[1]
  lowest<-ave[1]
  for(k in 2:12)
  {
    ave[k]<-(atempmax[k]+atempmin[k])/2.0
    if(ave[k]>highest)
    {
      highest<-ave[k]
    }
    if(ave[k]<lowest)
    {
      lowest<-ave[k]
    }
  }
  if(lowest< (-10.0))
  {
    lowest<- (-10.0)
  }
  ra<-abs(highest-lowest)
  for(k in 1:12)
  {
    if(atempmin[k]<(-10.0))
    {
      tr<-atempmax[k]-(-10.0)
    }
    else
    {
      tr<-atempmax[k]-atempmin[k]
    }
    t<-tr/2.0+atempmin[k]
    tm<-t+0.006*elev
    td<-0.0023*elev+0.37*t+0.53*tr+0.35*ra-10.9
    e<-((700.0*tm/(100.0-abs(sitlat)))+15.0*td)/(80.0-t)
    monpet<-(e*30.0)/10.0
    if(monpet < 0.5)
    {
      pet[k]<<-0.5*fwloss4
    }
    else
    {
      pet[k]<<-monpet*fwloss4
    }
  }
}
## function (calstemp)  #########################
## soil temperature
calstemp<-function(month)
{
  #For Forest only (e.g. no savana)
  stdead<-0.0
  bio<-(pools.bfol)*2.5+stdead+(xsrfstr+xsrfmet)*2.0*elitst
  if(bio>PmxBio)
  {
    bio<-PmxBio
  }
else {
    bio<-bio
  }
  stempmax <<-atempmax[month]+
              (25.4/(1+18.0*exp(-0.20*atempmax[month])))*
              (exp(PmxTmp*bio)-0.13)
  stempmin <<-atempmin[month]+PmnTmp*(bio)-1.78
  stempave <<-(stempmax+stempmin)/2.0
}
## function (calfc_wtpt)  #########
##  field capacity and wilting point
calfc_wtpt<-function()
```

```
{
  #From CENTURY source
  #swflag lets the model user choose between using actual data
  #for awilt and afiel or equations from Gupta and Larson (1979)
  #or Rawls et al (1982).
  #swflag=0
  #Use actual data
  #swflag=1
  #Use G&L for both awilt (-15 bar) and afiel (-0.33 bar)
  #swflag=2
  #Use G&L for both awilt (-15 bar) and afiel (-0.10 bar)
  #swflag=3
  #Use Rawls for both awilt (-15 bar) and afiel (-0.33 bar)
  #swflag=4
  #Use Rawls for both awilt (-15 bar) and afiel (-0.10 bar)
  #swflag=5
  #Use Rawls for afiel (-0.33 bar) and actual data for awilt
  #swflag=6
  #Use Rawls for afiel (-0.10 bar) and actual data for awilt
  fcsa<-c( 0.3075,    0.5018,   -0.20,   -0.30,  -0.19,   0.31)
  fcsi<-c( 0.5886,    0.8548,    0.0,     0.0,    0.0,    0.0)
  fccl<-c( 0.8039,    0.8833,    0.36,    0.23,   0.0,    0.0)
  fcom<-c( 2.208E-03, 4.966E-03, 0.0299,  0.0317, 0.0210, 0.0260)
  fcbd<-c(-0.1434,   -0.2423,    0.0,     0.0,    0.0,    0.0)
  fcwp<-c( 0.0,       0.0,       0.0,     0.0,    0.72,  0.41)
  fcin<-c( 0.0,       0.0,       0.2576,  0.4118, 0.2391, 0.4103)
  wpsa<-c(-0.0059,   -0.0059,    0.0,     0.0,    0.0,    0.0)
  wpsi<-c( 0.1142,    0.1142,    0.0,     0.0,    0.0,    0.0)
  wpcl<-c( 0.5766,    0.5766,    0.50,    0.50,   0.0,    0.0)
  wpom<-c( 2.228E-03, 2.228E-03, 0.0158,  0.0158, 0.0,    0.0)
  wpbd<-c( 0.02671,   0.02671,   0.0,     0.0,    0.0,    0.0)
  wpwp<-c( 0.0,       0.0,       0.0,     0.0,    1.0,   1.0)
  wpin<-c( 0.0,       0.0,       0.0260,  0.0260, 0.0,    0.0)
  #print(somsc)
  ompc <- somsc*1.724/(10000*bd*Edepth)
  swflag<-flag_fc_wtpt
  for(lyr in 1:nlayer)
  {
    #Please note:
    #In the original CENTURY model,
    #somsc was not calculated before the call of the prelim.f,
    #so afiel is calculated using somsc=ompc=0.
    #This is a bug of the original CENTURY model
    if(BFix==0)
    {
      ompc<-0.0
    }
    afiel[lyr] <<- fcsa[swflag]*sand  + fcsi[swflag]*silt +
                    fccl[swflag]*clay + fcom[swflag]*ompc +
                    fcbd[swflag]*bd + fcwp[swflag]*awilt[lyr] +
                    fcin[swflag]
    awilt[lyr] <<- wpsa[swflag]*sand  + wpsi[swflag]*silt+
                    wpcl[swflag]*clay + wpom[swflag]*ompc +
                    wpbd[swflag]*bd + wpwp[swflag]*awilt[lyr] +
                    wpin[swflag]
    ompc<-ompc*0.85
  }
}
## function (calwater)  #####################
## soil water content
calwater<-function(month)
{
  #Initialize
  add<-0.0
  amelt<-0.0
  asimx<-0.0
  avh2o[1]<<-0.0
  avh2o[2]<<-0.0
  avh2o[3]<<-0.0
  avap<-0.0
  evl<-0
  pevp<-0.0
  pttr<-0.0
  rwc1<-0.0
```

```
tran<<-0.0
trap<-0.01
aabs<-0.0
evsnow<-0.0
evap<<-0.0
petrem<-pet[month]
awwt<-matrix(0.0,nrow=11,ncol=1)
#CO2 effect was not included here
co2val<-1.0
irract<-0.0
inputs<-prec[month]+irract
winputs<-inputs
atempave<<-(atempmax[month]+atempmin[month])/2.0
aliv<-pools.bfol*2.5
alit<-(xsrfstr+xsrfmet)*2.0
adead<-0.0
#************
#Snow
#Snowfall
if(atempave <= 0.0)
{
  #snow <- snow + prec[month]
  snow <<- snow + inputs
  winputs<-0.0
}
# melt
if(atempave >= Tmelt1)
{
  melt <- Tmelt2 *(atempave -Tmelt1)
  if(melt>snow)
  {
    melt<-snow
  }
  snow <<-snow-melt
  ##......................
  if((atempave > 0.0) && (snow > 0.0))
  {
    snlq<<-snlq+inputs
  }
  snlq<<-snlq+melt
  if(snlq >= (0.05*snow))
  {
    add<-snlq -0.05*snow
    snlq<<-snlq-add
  }
}
if(snow > (0.0))
{
  evsnow<-petrem*0.87
  snow1<-snow+snlq
  if(evsnow > snow1)
  {
    evsnow<-snow1
  }
  snow<<-snow-evsnow*(snow/snow1)
  snlq<<-snlq-evsnow*(snlq/snow1)
  evap<<-evap+evsnow
  petrem<-petrem-evsnow/0.87
  if(petrem < 0.0)
  {
    petrem<-0.0
  }
}
if(snow <= 0.0)
{
  sd<-aliv+adead
  if(sd > 800.0)
  {
    sd<-800.0
  }
  if(alit > 400.0)
  {
    alit<-400.0
  }
  aint<-(0.0003 * alit +0.0006 *sd) *fwloss1
  aabs<-0.5*exp((-0.002*alit)-(0.004*sd))*fwloss2
  if((aabs+aint)*inputs<0.4*petrem)
  {
    evl<-(aabs+aint)*winputs
  }
  else
  {
    evl<-0.4*petrem
  }
  evap<<-evap+evl
  add<-add+winputs -evl
  trap<-petrem-evl
}
if(atempave < 2.0)
{
  pttr<-0.0
}
else
{
  pttr<-petrem *0.65 *(1.0 -exp(-0.020 *aliv)) *co2val
}
if(pttr <= trap){trap<-pttr}
if(trap <= 0.0){trap<-0.01}
##....................
#hpttr is not included
pevp<-petrem -trap -evl
if(evap<0.0){pevp<-0.0}
if((trap-0.01) < add)
{
  #print(add)
  tran<<- trap-0.01
}
else
{
  tran<<- add
}
```

```
trap<-trap-tran
add<-add-tran
strm<-0.0
base<-0.0
stream1<<-0.0
for(j in 1:nlayer)
{
  asmos[j]<<-asmos[j]+add
  afl<-adep[j]*afiel[j]
  if(asmos[j]>afl)
  {
    amov[j]<<-asmos[j]-afl
    asmos[j]<<-afl
    if(j == nlayer)
    {
      strm<-amov[j]*stormf
    }
  }
  else
  {
    amov[j]<<-0.0
  }
  add<-amov[j]
}
asmos[nlayer+1]<<-asmos[nlayer+1]+add-strm
base<-asmos[nlayer+1]*basef
asmos[nlayer+1]<<-asmos[nlayer+1]-base
stream1<<-strm+base
asimx<-asmos[1]
rwc1<-0.0
tot<-0.0
tot2<-0.0
for(j in 1:nlayer)
{
  avw<-asmos[j]-awilt[j]*adep[j]
  if(avw < 0.0)
  {
    avw<-0.0
  }
  awwt[j]<-avw*awtl[j]
  tot<-tot+avw
  tot2<-tot2+awwt[j]
}
if(tot<trap)
{
  trap<-tot
}
else
{
  trap<-trap
}
if(tot2 > 0.0)
{
  for(j in 1:nlayer)
  {
    avinj<-asmos[j]-awilt[j]*adep[j]
    if(avinj < 0.0)
    {
      avinj<-0.0
    }
    trl<-(trap*awwt[j])/tot2
    if(trl > avinj)
    {
      trl<-avinj
    }
    asmos[j]<<-asmos[j]-trl
    #if(year==5 && month==1){cat(asmos[j], trl,"\n")}
    avinj<-avinj-trl
    tran<<-tran+trl
    rwcf[j]<-(asmos[j]/adep[j]-awilt[j])/(afiel[j]-awilt[j])
    if(j<=nlaypg)
    {
      avh2o[1]<<-avh2o[1]+avinj
    }
```

```
      avh2o[2]<<-avh2o[2]+avinj
      if(j <= (2))
      {
        avh2o[3]<<-avh2o[3]+avinj
      }
    }
  }
  fwlos<-0.25
  evmt<-(rwcf[1]-fwlos)/(1.0-fwlos)
  if(evmt <= (0.01))
  {
    evmt<-0.01
  }
  evlos<-evmt*pevp*aabs*0.10
  avinj<-asmos[1]-awilt[1]*adep[1]
  if(avinj < 0.0)
  {
    avinj<-0.0
  }
  if(evlos > avinj)
  {
    evlos<-avinj
  }
  asmos[1]<<-asmos[1]-evlos
  evap<<-evap+evlos
  avhsm<-(asmos[1]+rwc1*asimx)/(1.0+rwc1)
  rwcf[1]<<-(avhsm/adep[1]-awilt[1])/(afiel[1]-awilt[1])
  avh2o[1]<<-avh2o[1]-evlos
  avh2o[2]<<-avh2o[2]-evlos
  avh2o[3]<<-avh2o[3]-evlos
}
## function (caldefac)  ###############
## decomposition factor
caldefac<-function()
{
  if(snow > 0.0)
  {
    stempave<-0.0
  }
  # Cal defac
  tfunc<-teff1+teff2*exp(teff3 * stempave)
  rprpet <<- (avh2o[3] + prec[month] ) / pet[month]
  #* idef in fix.100 in Century control linear 1 or ratio 2 option
  #* this is idef==2
  if(rprpet > 9.0 )
  {
    wfunc<<-1.0
  }
  else
  {
    wfunc<<-1.0/(1.0+30.0*exp(-8.5*rprpet))
  }
  #if(wfunc>1.0)
  #{
  #  wfunc<<-1.0
  #}
  defac<<-tfunc*wfunc
  ##
  # If you want to use the defac from the original CENTURY, then.
  # defac<<-centdefac[1+(year-1)*12+month,2]
  #*** Cal anerb
  anerb <<- 1.0
}
## functions to Divide litter inputs##################################
##
## function (calcenturyinput) ################
## litter inputs into each soil carbon pool:1
calcenturyinput<-function()
{
  if(flows.lfinfol>0.0)
  {
    #centurypartit(1, cnr_srflit)
    centurypartit(1, cnr_fol)
  }
  else
  {
    usrfstr <<-0.0
    usrfmet <<-0.0
    usrfstr_lig <<- 0.0
  }
  if(flows.lfinfir>0.0)
  {
    #centurypartit(2, cnr_bellit)
    centurypartit(2, cnr_fir)
  }
  else
  {
    ubelstr <<- 0.0
    ubelmet <<- 0.0
    ubelstr_lig <<- 0.0
  }
  uwood1 <<- flows.lfinbra
  uwood2 <<- flows.lfinste
  uwood3 <<- flows.lfincor
  uwood1_lig <<- flows.lfinbra * cbra_lig
  uwood2_lig <<- flows.lfinste * cste_lig
  uwood3_lig <<- flows.lfincor * ccor_lig
}
## function (centurypartit) #################
## litter inputs into each soil carbon pool:2
centurypartit<-function(p, cnr)
{
  #translocation
  #forrtf
  if(p==1)
  {
    cpart<- flows.lfinfol
    epart<- flows.lfinfol*(1.0/cnr)*(1.0-forrtf)
    #amax1
    if(cpart/pabres > 1.0)
    {
      s<-cpart/pabres
    }
    else
    {
      s<-1.0
    }
    #damr11<-0.0
    dirabs<- damr11 * xNmineral * s
    if((epart+dirabs) <= 0.0)
    {
      rcetot<-0.0
    }
    else
    {
      rcetot<-cpart/(epart+dirabs)
    }
    if(rcetot < damrmn)
    {
      dirabs<-cpart/damrmn-epart
    }
    if(dirabs<0.0)
    {
      dirabs<-0.0
    }
    frlign<- cfol_lig
  }
  else if (p==2)
  {
    cpart<- flows.lfinfir
    #epart<- flows.lfinfir*(1.0/cnr)*(1.0-forrtf)
    epart<- flows.lfinfir*(1.0/cnr)
    #amax1
    if(cpart/pabres > 1.0)
    {
      s<-cpart/pabres
    }
    else
    {
      s<-1.0
    }
    dirabs<- damr21 * xNmineral * s
    if((epart+dirabs) <= 0.0)
    {
      rcetot<-0.0
    }
    else
    {
      rcetot<-cpart/(epart+dirabs)
    }
    if(rcetot < damrmn)
    {
      dirabs<-cpart/damrmn-epart
    }
    if(dirabs<0.0)
    {
      dirabs<-0.0
    }
    frlign<- cfir_lig
  }
  else
  {
    printf("error")
  }
  ###
  frn<- (epart + dirabs)/(cpart*2.5)
  rlnres<-frlign/frn
  frmet<- spl1 -spl2 *rlnres
  if(frlign > (1.0-frmet))
  {
    frmet<- 1.0-frlign
  }
  if(frmet<0.20)
  {
    frmet<-0.20
```

```
    }
    caddm <- cpart*frmet
    if(caddm < 0.0)
    {
      caddm<-0.0
    }
    cadds <- cpart-caddm
    fligst <- frlign/(cadds/cpart)
    if(fligst > 1.0)
    {
      fligst <- 1.0
    }
    if(p==1)
    {
      usrfstr <<- flows.lfinfol *(1.0-frmet)
      usrfmet <<- flows.lfinfol *frmet
      usrfstr_lig <<- fligst
    }
    else if (p==2)
    {
      ubelstr <<- flows.lfinfir *(1.0-frmet)
      ubelmet <<- flows.lfinfir *frmet
      ubelstr_lig <<- fligst
    }
  }
  ##############################################################
  ## functions (calcentury)
  ## to calculate soil carbon dynamics
  ## **********************************************************
  calcentury<-function()
  {
    uwood1<-flows.lfinbra
    uwood2<-flows.lfinste
    uwood3<-flows.lfincor
    #********************************************
    # ** Dead branch = Wood 1
    #strlig=(xwood1*wood1strlig+uwood1_lig)/(xwood1+uwood1)
```

```
#wood1strlig= strlig
strlig <-cbra_lig
if(xwood1>0.000001)
{
  tcflow <- xwood1*defac*kwood1*exp(-pligst1*strlig)*DT
  if(tcflow>xwood1)
  {
    tcflow<-xwood1
  }
}
else
{
  tcflow<-0.0
}
tsom2_fwood1 <- tcflow * strlig
#*Respiration associated with decomposition to som2
co2los <- tsom2_fwood1 * RSPLIG
CO2out <<- CO2out+co2los
#*Net C flow to SOM2
tsom2_fwood1 <- tsom2_fwood1 - co2los
tsom1_fwood1 <- tcflow - tsom2_fwood1 - co2los
#*Respiration associated with decomposition to som1
co2los <- tsom1_fwood1 * PS1CO21
tsom1_fwood1 <- tsom1_fwood1 -co2los
CO2out<<-CO2out+co2los
#******
xwood1_new <- xwood1 + uwood1 - tcflow
#**********************************************
# ** Dead Stem = Wood 2
#strlig=(xwood2*wood2strlig+uwood2_lig)/(xwood2+uwood2)
#wood2strlig= strlig
strlig<-cste_lig
if(xwood2>0.000001)
{
  tcflow<- xwood2*defac*kwood2*exp(-pligst1*strlig)*DT
  if(tcflow>xwood2)
  {
    tcflow<-xwood2
  }
}
else
{
  tcflow<-0.0
}
tsom2_fwood2 <- tcflow * strlig
#*Respiration associated with decomposition to som2
co2los <- tsom2_fwood2 * RSPLIG
CO2out<<-CO2out+co2los
#*Net C flow to SOM2
tsom2_fwood2 <- tsom2_fwood2 - co2los
tsom1_fwood2 <- tcflow - tsom2_fwood2 - co2los
#*Respiration associated with decomposition to som1
co2los <- tsom1_fwood2 * PS1CO21
tsom1_fwood2 <- tsom1_fwood2 -co2los
CO2out<<-CO2out+co2los
#******
xwood2_new <- xwood2 + uwood2 - tcflow
#*******************************************
# ** Dead Coarse root = Wood 3
#strlig=(xwood3*wood3strlig+uwood3_lig)/(xwood3+uwood3)
#wood3strlig= strlig
strlig<-ccor_lig
if(xwood3>0.000001)
{
  tcflow<- xwood3*defac*kwood3*exp(-pligst2*strlig)*anerb*DT
  if(tcflow>xwood3)
  {
    tcflow<-xwood3
  }
}
else
{
  tcflow<-0.0
}
```

```
tsom2_fwood3 <- tcflow * strlig
#*Respiration associated with decomposition to som2
co2los <- tsom2_fwood3 * RSPLIG
CO2out<<-CO2out+co2los
#*Net C flow to SOM2
tsom2_fwood3 <- tsom2_fwood3 - co2los
tsom1_fwood3 <- tcflow - tsom2_fwood3 - co2los
#*Respiration associated with decomposition to som1
co2los <- tsom1_fwood3 * PS1CO21
tsom1_fwood3 <- tsom1_fwood3 -co2los
CO2out<<-CO2out+co2los
#******
xwood3_new <- xwood3 + uwood3 - tcflow
#*******************************************
# ** surface structural
#strlig=(pools.xlig1_fol + pools.xlig1_bra +
#        pools.xlig1_ste)/(pools.talit)
#srfstrlig = xsrfstr*srfstrlig/xsrfstr
#strlig=(xsrfstr*srfstrlig + usrfstr_lig)/(xsrfstr+usrfstr)
strlig<-(xsrfstr*srfstrlig + usrfstr_lig*usrfstr)/
        (xsrfstr + usrfstr)
srfstrlig <<- strlig
if(xsrfstr>0.000001)
{
  if(xsrfstr>strmax1)
  {
    mass<-strmax1
  }
  else
  {
    mass<-xsrfstr
  }
  tcflow <-mass*defac*Asrfstr*exp(-pligst1*strlig)*DT
  if(tcflow>xsrfstr)
  {
    tcflow<-xsrfstr
  }
```

```
}
else
{
  tcflow<-0.0
}
tsom2_fsrfstr <- tcflow * strlig
#*Respiration associated with decomposition to som2
co2los <- tsom2_fsrfstr * RSPLIG
CO2out <<- CO2out+co2los
#*Net C flow to SOM2
tsom2_fsrfstr <- tsom2_fsrfstr - co2los
tsom1_fsrfstr <- tcflow - tsom2_fsrfstr - co2los
#*Respiration associated with decomposition to som1
co2los <- tsom1_fsrfstr * PS1CO21
tsom1_fsrfstr <- tsom1_fsrfstr -co2los
CO2out <<- CO2out+co2los
#******
xsrfstr_new <- xsrfstr + usrfstr - tcflow
#**********************************************
# ** soil structural
#strlig=(pools.xlig1_fir+pools.xlig1_cor)/(pools.tblit)
#belstrlig = xbelstr*belstrlig/xbelstr
strlig<-(xbelstr*belstrlig + ubelstr_lig*ubelstr)/
        (xbelstr + ubelstr)
belstrlig<<-strlig
if(xbelstr>0.000001)
{
  if(xbelstr>strmax2)
  {
    mass<-strmax2
  }
  else
  {
    mass<-xbelstr
  }
  tcflow<-mass*defac*Abelstr*exp(-pligst2*strlig)*anerb*DT
  if(tcflow>xbelstr)
  {
    tcflow<-xbelstr
  }
}
else
{
  tcflow<-0.0
}
tsom2_fbelstr <- tcflow * strlig
#*Respiration associated with decomposition to som2
co2los <- tsom2_fbelstr * RSPLIG
CO2out <<- CO2out+co2los
#*Net C flow to SOM2
tsom2_fbelstr <- tsom2_fbelstr - co2los
tsom1_fbelstr <- tcflow - tsom2_fbelstr - co2los
#*Respiration associated with decomposition to som1
co2los <- tsom1_fbelstr * PS1CO22
tsom1_fbelstr <- tsom1_fbelstr -co2los
CO2out<<-CO2out+co2los
#******
xbelstr_new <- xbelstr + ubelstr - tcflow
#**********************************************
# ** surface metab
if(xsrfmet>0.000001)
{
  tcflow<-xsrfmet * defac * Asrfmet * DT
  if(tcflow>xsrfmet)
  {
    tcflow<-xsrfmet
  }
}
else
{
  tcflow<-0.0
}
co2los<-tcflow*Psrfmet
tsom1_fsrfmet <- tcflow-co2los
```

```
CO2out <<- CO2out+co2los
xsrfmet_new <- xsrfmet +usrfmet -tcflow
#*********************************************
# ** belowground metab
if(xbelmet>0.000001)
{
  tcflow<-xbelmet * defac * Abelmet* anerb * DT
  if(tcflow>xbelmet)
  {
    tcflow<-xbelmet
  }
}
else
{
  tcflow<-0.0
}
co2los<-tcflow*Pbelmet
tsom1_fbelmet<-tcflow-co2los
CO2out<<-CO2out+co2los
xbelmet_new<-xbelmet + ubelmet -tcflow
#*********************************************
#**** surface microbe
if(xsrfmic>0.000001)
{
  tcflow <- xsrfmic * defac * Asrfmic *DT
  if(tcflow>xsrfmic)
  {
    tcflow <- xsrfmic
  }
}
else
{
  tcflow<-0.0
}
co2los<-tcflow*Psrfmic
tsom2_fsrfmic<-tcflow-co2los
CO2out<<-CO2out+co2los
```

```
xsrfmic_new <- xsrfmic + tsom1_fsrfstr + tsom1_fsrfmet +
               tsom1_fwood1 + tsom1_fwood2 -tcflow
#xsrfmic_new= -tcflow + xsrfmic + tsom1_fsrfstr+tsom1_fsrfmet
#xsrfmic_new= -tcflow + xsrfmic + tsom1_fsrfmet
#**********************************************
#**** active
eftext <- Peftxa + Peftxb * sand
if(xactv>0.000001)
{
  tcflow<-xactv* defac * eftext * kactv * anerb *DT
  if(tcflow>xactv)
  {
    tcflow<-xactv
  }
}
else
{
  tcflow<-0.0
}
co2los<-tcflow*Pactv
#*cfsfs2=tcflow-co2los
#*tcflow=tcflow-co2los
CO2out<<-CO2out+co2los
fps1s3 <- ps1s31 + ps1s32 * clay
tsom3_factv<-tcflow * fps1s3
#leaching
if(amov[2]>0.0)
{
  orglch<-omlech[1]+omlech[2]*sand
  t<-1.0-(omlech[3]-amov[2])/omlech[3]
  if(t>1.0)
  {
    linten<-1.0
  }
  else
  {
    linten<-t
  }
  cleach<<-tcflow * orglch * linten
  }
else
{
  cleach<<-0.0
}
tcleach<<-tcleach+cleach
tsom2_factv<-tcflow -co2los -tsom3_factv -cleach
#* Updated at the end.
#xactv_new = xactv + tsom1_fbelstr +tsom1_fbelmet +
#            tsom1_fwood3 +tsom1_fslow +tsom1_fpass -tcflow
xactv_new <- xactv + tsom1_fbelstr +tsom1_fbelmet +
             tsom1_fwood3 -tcflow
#*********************************************
#**** Slow
if(xslow>0.000001)
{
  tcflow<-xslow *defac * kslow * anerb *DT
  if(tcflow>xslow)
  {
    tcflow<-xslow
  }
}
else
{
  tcflow<-0.0
}
co2los<-tcflow*Pslow
#*cfsfs2=tcflow-co2los
#*tsom3_fslow=tcflow-co2los
CO2out<<-CO2out+co2los
xslow_new <- xslow + tsom2_fsrfstr +tsom2_fsrfmic +
             tsom2_fbelstr +tsom2_factv + tsom2_fwood1 +
             tsom2_fwood2 + tsom2_fwood3 -tcflow
fps2s3 <- ps2s31 + ps2s32 * clay
tsom3_fslow<-tcflow * fps2s3
```

```
tsom1_fslow<-tcflow -co2los -tsom3_fslow
#**********************************************
#**** Passive
if(xpass>0.000001)
{
  tcflow<-xpass *defac  * kpass* anerb *DT
  if(tcflow>xpass)
  {
    tcflow<-xpass
  }
}
else
{
  tcflow<-0.0
}
co2los<-tcflow*Ppass
#*cfsfs2=tcflow-co2los
tsom1_fpass<-tcflow-co2los
CO2out<<-CO2out+co2los
xpass_new <- xpass + tsom3_factv +tsom3_fslow -tcflow
#******************************************************
#*********** Active new
#xactv_new = xactv + tsom1_fpass
#xactv_new = xactv + tsom1_fslow
xactv_new <- xactv_new + tsom1_fslow + tsom1_fpass
#**********************************************
#****** UPDATE
xsrfstr <<- xsrfstr_new
xsrfmet <<- xsrfmet_new
xsrfmic <<- xsrfmic_new
xbelstr <<- xbelstr_new
xbelmet <<- xbelmet_new
xactv <<- xactv_new
xslow <<- xslow_new
xpass <<- xpass_new
xwood1 <<- xwood1_new
xwood2 <<- xwood2_new
```

```
  xwood3 <<- xwood3_new
  #*********************************************
  somsc <<- xactv + xslow + xpass
  talit <<- xsrfstr + xsrfmet + xsrfmic
  tblit <<- xbelstr + xbelmet
  somtc <<-xactv + xslow + xpass + xbelstr + xbelmet
  tawood <<- xwood1 + xwood2
  tbwood <<- xwood3
}
#end of functions
## calculate field capacity, wilting point ############
awilt
afiel
if(flag_fc_wtpt>0.0)
{
  somsc <- xactv + xslow + xpass
  calfc_wtpt()
}
awilt
afiel
## Initialize soil water condition  ###################
# essential to calculate deep asmos correctly
pet
calpet()
pet
for(month in 1:12)
{
  calwater(month)
}
obj.s <- ls()
#obj.s
#####################################################################
##   MAIN CENTURY SIMULATION             ############################
carbon.out <- NULL
for(s in 1:1){
  id<-s
  soil.carbon.year.out <-NULL
```

```
for(year in TSTART:TEND){
  CO2out<-0.0
  calpet()
  #month loop
  for(month in 1:12){
    #month=1
    tcleach<-0.0
    DT<-1.0/(12.0*CYCL)
    ##........................................................
    #Litterfall SITE SPECIFIC data
    flows.lfinfol<-litter.in [1,2]*leafdr[month]*(1.0/CYCL)
    flows.lfinbra<-litter.in [1,3]*(1.0/(12*CYCL))
    flows.lfinste<-litter.in [1,4]*(1.0/(12*CYCL))
    flows.lfinfir<-litter.in [1,5]*(1.0/(12*CYCL))
    flows.lfincor<-litter.in [1,6]*(1.0/(12*CYCL))
    talit <-xsrfstr + xsrfmet +xsrfmic
    tblit <- xbelstr + xbelmet
    tawood <- xwood1 + xwood2
    tbwood <- xwood3
    somsc <- xactv + xslow + xpass
    somtc <- xactv + xslow + xpass + xbelstr + xbelmet
    ##........................................................
    calstemp(month)
    calwater(month)
    if(snow>0.0)
    {
      stempmax <-0.0
      stempmin <-0.0
      stempave <-0.0
    }
    ##........................................................
    caldefac()
    calcenturyinput()
    # CENTURY CARBON FUNCTION SIMULATIONS
    # updated 4 times per month (CYCL=4)
    for(i in 1:CYCL)
    {
        calcentury()
      }
#end of centurycal CYCL loop
    }
#end of month loop ()
    ##........................................................
    ## site specific output of CENTURY carbon
    if(year==year) #TEND)
    {
      soil.carbon0 <- data.frame(id,year, month,
                                 xsrfstr, xsrfmet,
                                 xsrfmic, xbelstr, xbelmet,
                                 xactv, xslow, xpass, somsc,
                                 xwood1, xwood2, xwood3,
                                 CO2out, somtc)
    }

    soil.carbon.year.out <- rbind(soil.carbon.year.out,
                                  soil.carbon0)
  }
#end of year loop
  carbon.out <-rbind(carbon.out,soil.carbon.year.out )
}
#end of site.group for loop
options(digit=12)
century.out <-carbon.out[,c("id","year","month",
                             "CO2out","somsc",
                             "xbelstr","xbelmet",
                             "xactv","xslow","xpass",
                             "somtc")]
#convert gC.m-2 to tC.ha-1 by 1/1e6*1e4
century.out[,4:11]<-century.out[,4:11]/100
##........................................................
#plot components of soil carbon stock
#somtc <- xactv + xslow + xpass + xbelstr + xbelmet
#figure
par(mfrow=c(1,1), mar=c(5,5,1,1))
```

```
plot(century.out$year,century.out$somtc,
     log="y", ylim=c(0.3,round(max(century.out$somtc),1)+50),
     ylab="CENTURY soil carbon pools (tC/ha)",
     xlab="year")
lines(century.out$year,century.out$somtc,
      lwd=2)
lines(century.out$year,century.out$xactv,
      col="blue", lwd=2)
lines(century.out$year,century.out$xslow,
      col="red", lwd=2)
lines(century.out$year,century.out$xpass,
      col="orange", lwd=2)
lines(century.out$year,century.out$xbelstr,
      col="grey", lwd=2)
lines(century.out$year,century.out$xbelmet,
      col="magenta", lwd=2)
legend("bottomright",
       c("total","active","slow","passive",
         "bg.structural","bg.metabolic"),
       col=c("black","blue","red","orange","grey","magenta"),
       pch=c(1,NA,NA,NA,NA,NA),
       lwd=2, lty=1, border="white",bg="white")
```

Table S1. Statistical characteristics (mean, standard error) of basic variables for groups of soils derived by recursive partitioning including soil variables (see Fig. 2a), compared with interpretation of carbon, moisture, and fertility of groups.

N of soil samples in groups		959	909	136	335	182	296	180	8	142	83
Total SOC ($\mathrm{tC\,ha^{-1}}$)	Mean	65.1	81.8	130.2	86.2	126.4	103.9	136.8	268.6	143.7	203.1
	SE	1	1.3	5.1	2.6	4.8	2.8	4.2	23.7	6.7	9.8
SOC mineral ($\mathrm{tC\,ha^{-1}}$)	Mean	45.4	56.4	86.9	68.5	98.4	73.2	92.4	230.6	108.8	153.3
	SE	1.6	2.1	8.2	4.9	9.1	4.5	7	33	12.8	21.6
C/N	Mean	13.4	18.6	15.4	10.7	8.1	23.8	21.7	23.1	23.3	32.6
	SE	0.3	0.4	0.8	0.2	0.2	0.5	0.5	1.9	0.6	2
H100 (m)	Mean	20.8	24.3	24.7	25.4	26.6	22.2	29.9	31	23.6	24
	SE	0.2	0.2	0.5	0.4	0.4	0.3	0.3	0.5	0.5	0.6
Total Litter ($\mathrm{tC\,ha^{-1}}$)	Mean	2.7	3.3	3.4	3.5	3.6	3.1	4	3.8	3.2	3.5
	SE	0	0	0.1	0.1	0.1	0	0.1	0.2	0.1	0.1
Temperature air (C)	Mean	3.3	4.8	5.1	5.1	5.8	4.5	7.3	7.3	5.3	6.4
	SE	0.1	0.1	0.2	0.1	0.1	0.1	0.1	0.2	0.2	0.2
Long-term moisture (%)	Mean	20.2	22.4	26	23.6	26.2	22.9	22.9	21.3	21.8	25.7
	SE	0.2	0.2	0.6	0.3	0.5	0.4	0.4	1.3	0.5	0.7
Precipitation (mm y-1)	Mean	698.8	712.9	697.1	644.1	630.3	693	817.2	1173.2	687.9	619.4
	SE	5.7	7	18.8	10.4	12.5	10.8	21.2	162	18.1	27
CEC ($\mathrm{mmol_c\,kg^{-1}}$)	Mean	4.7	12.1	11.6	49.2	91.5	24	23.5	24.5	59.7	98.7
	SE	0.1	0.1	0.2	2.6	5.1	0.3	0.3	1.7	3.2	7.6
pH	Mean	5.2	5.1	5.1	5.5	5.6	5	4.8	4.6	4.9	6
	SE	0	0	0	0	0.1	0	0	0.1	0.1	0.2
Clay content (%)	Mean	0.8	1.1	4.2	5.9	21.5	1.7	2.4	2.5	2.4	8
	SE	0.1	0.1	0.5	0.5	1	0.3	0.3	1.3	0.4	1.2
Silt content (%)	Mean	15.1	14.5	29	27.2	57.8	16.4	17.9	18.8	17.1	32.5
	SE	0.3	0.3	1.5	1.2	1.4	0.7	0.7	3.2	1	3
Carbon		low	medium	high	medium	high	medium	high	extra	high	extra
Moisture		dry-fresh	fresh	moist-fresh	fresh	moist-fresh	fresh	fresh	fresh	fresh	moist-fresh
Fertility		low	medium	medium	medium	high	low	high	high	medium	medium
Soil group		1	2	3	4	5	6	7	8	9	10

Table S2. Statistical characteristics (mean, standard error) of basic environmental and soil variables for regression tree of data groups classified by recursive partitioning with data excluding soil variables (see Fig. 3).

Number of samples		735.0	932.0	796.0	711.0	56.0
Total soil carbon ($\mathrm{tC\,ha^{-1}}$)	Mean	67.1	85.4	96.5	120.1	179.0
	SE	1.4	1.5	2.0	2.4	12.5
C/N	Mean	15.0	15.8	16.8	18.1	35.2
	SE	0.3	0.3	0.4	0.4	3.2
H100 (m)	Mean	19.0	19.7	27.0	30.1	18.7
	SE	0.1	0.1	0.1	0.2	0.3
Total Litter ($\mathrm{tC\,ha^{-1}}$)	Mean	2.3	2.8	3.6	4.0	3.0
	SE	0.0	0.0	0.0	0.0	0.1
Soil water content (%)	Mean	20.2	24.2	22.0	22.6	22.3
	SE	0.2	0.2	0.2	0.2	1.0
Temperature air (C)	Mean	2.4	2.7	6.4	7.2	7.2
	SE	0.1	0.1	0.0	0.0	0.0
pH	Mean	5.2	5.2	5.3	4.9	7.2
	SE	0.0	0.0	0.0	0.0	0.1
Sand content (%)	Mean	52.7	51.5	47.2	55.1	36.2
	SE	0.5	0.5	0.8	0.6	4.0
Clay content (%)	Mean	0.8	2.6	6.1	3.0	5.8
	SE	0.1	0.2	0.4	0.2	1.1
Silt content (%)	Mean	14.6	19.2	25.3	20.0	29.5
	SE	0.3	0.5	0.8	0.5	4.1
Group		low-C	median-C	median-C	hig-C	extra-C
acronym		cold.pine	cold.other	warm.rainy low-N	warm.rainy high-N	warm.dry

Table S3. Species and classes of ground vegetation grouped into functional types (1-dwarfshrubs, 2-herbs, 3-grasses, 4-mosses and 5-lichens).

Functional type	Ground vegetation
1	*Vaccinium myrtillus, Vaccinium vitis-idaea,Arctostaphylos uva-ursi, Empetrum nigrum ssp, Calluna vulgaris, Erica tetralix, Vaccinium uliginosum, Rhododendron tomentosum, Andromeda polifolia, Vaccinum oxycoccus/microcarpum*, Other field layer plants
2	*Gymnocarpium dryopteris, Oxalis acetosella, Anemone nemorosa, Maianthemum bifolium, Chamaenerion angustifolium, Anthriscus sylvestris, Melampyrum pratense/sylvaticum, Equisetum sylvaticum, Menyanthes trifoliata, Rubus chamaemorus, Phegopteris connectilis, Hepatica nobilis, Geum rivale, Urtica dioica, Rumex acetosa, Stellaria nemorum, Stellaria holostea, Silene dioica, Aconitum lycoctonum subsp. septentrionale, Actaea erythrocarpa, Trollius europaeus, Cardamine bulbifera, Filipendula ulmaria, Mercurialis perennis, Sanicula europaea, Aegopodium podagraria, Librar, Galium odoratum, Lamiastrum galeobdolon, Stachys sylvatica, Cirsium palustre, Cirsium heterophyllum, Lactuca alpina, Lactuca muralis, Crepis paludosa, Paris quadrifolia, Neottia ovata, Geranium sylvaticum, Rubus idaeus*, Other large grown ferns
3	Broad-leaved grass, Narrow-leaved grass, *Carex globularis,* Other sedges
4	*Pteridium aquilinum, Lycopdiaceae, Spagnum spp, Polytrichum commune, Pleurozium schreberi, Hylocomium splendens,* Other bryophytes
5	*Cladonia , Stereocaulon spp, Cladina spp, Cladonia and Cladina,* Other lichens

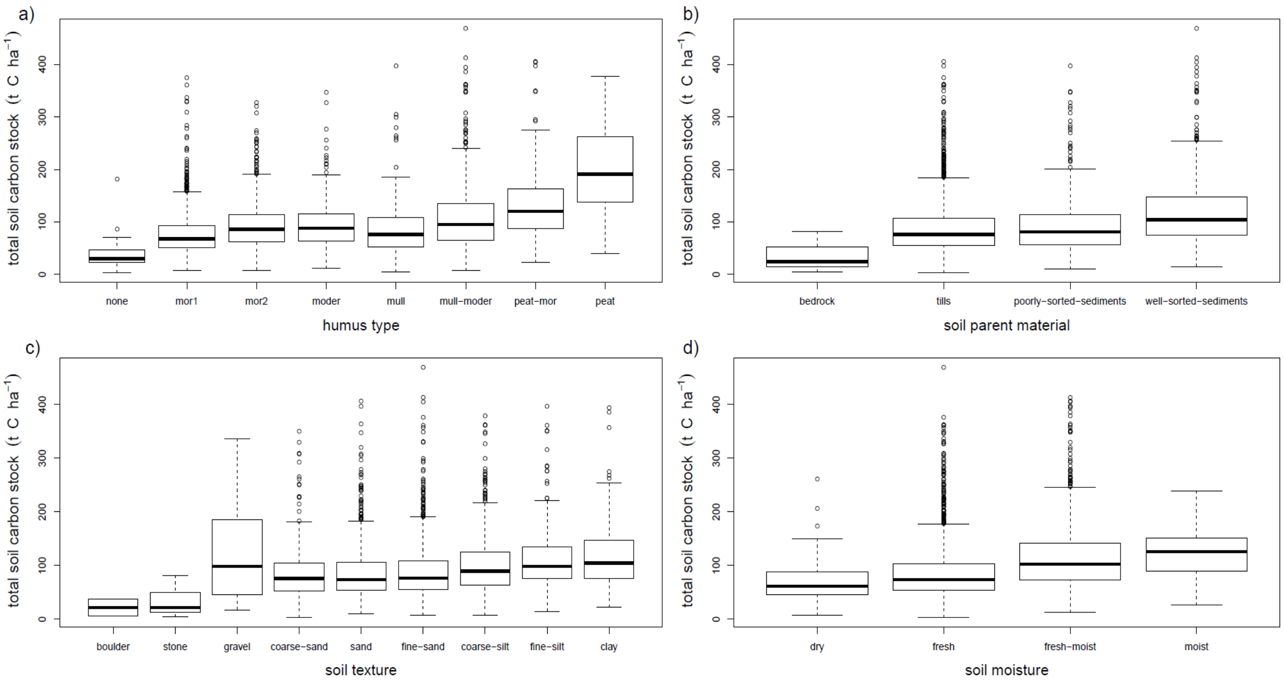

Figure S1. Boxplot main levels (minimum, 1st quantile, median, 3rd quantile, maximum, and dots for outliers) of the total soil carbon stock $(\mathrm{t\,C\,ha^{-1}})$ for SFSI categorical data on a) humus type, b) soil parent material, c) soil texture, and d) soil moisture.

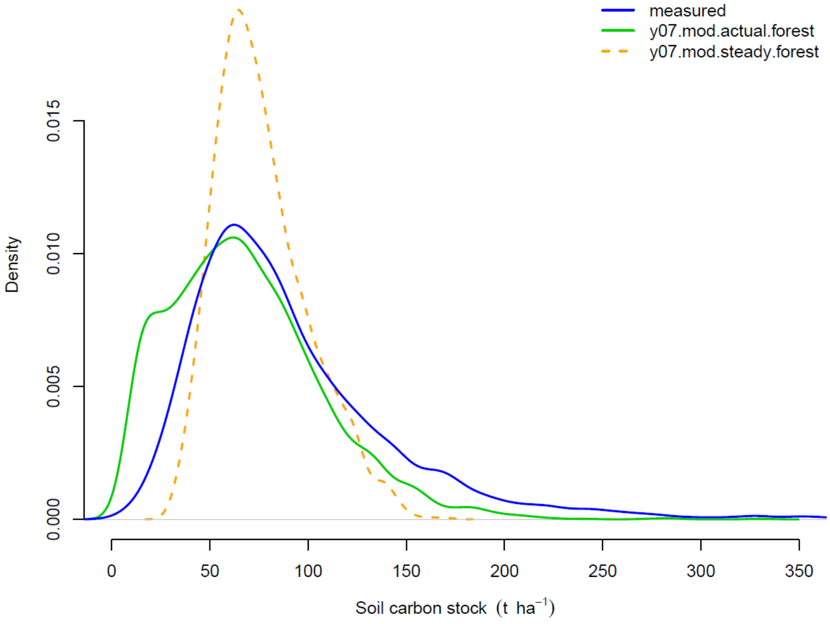

Figure S2. Density function of soil carbon stock measurements (measured) and the simulated soil carbon by the soil carbon model Yasso07 run with actual state forest and litter inputs (y07.mod.actual.forest) and with the steady state forest and litter inputs (y07.mod.steady.forest).

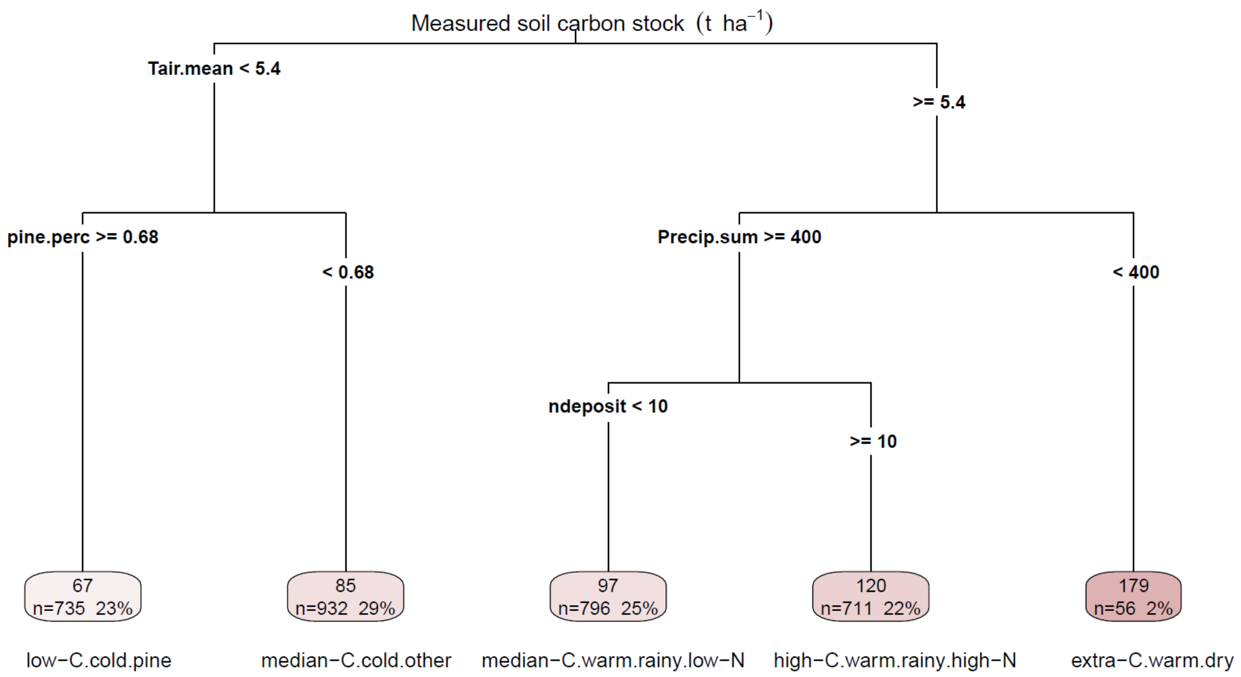

Figure S3. Classification/regression tree for the measured soil carbon stock ($\mathrm{tC\,ha^{-1}}$) and site environmental characteristics excluding soil physicochemical properties; the annual air temperature (Tair.mean, °C), the fraction of pine trees of the total canopy (pine.perc), the annual precipitation sum (Precip.sum, mm), and the nitrogen deposition (ndeposit, $\mathrm{kgN\,ha^{-1}\,y^{-1}}$). The values in the leaves of the tree show for the distinct environmental conditions mean soil carbon stock ($\mathrm{tC\,ha^{-1}}$), number and percentage of samples. The group acronyms are shown below the leaves of the regression tree.

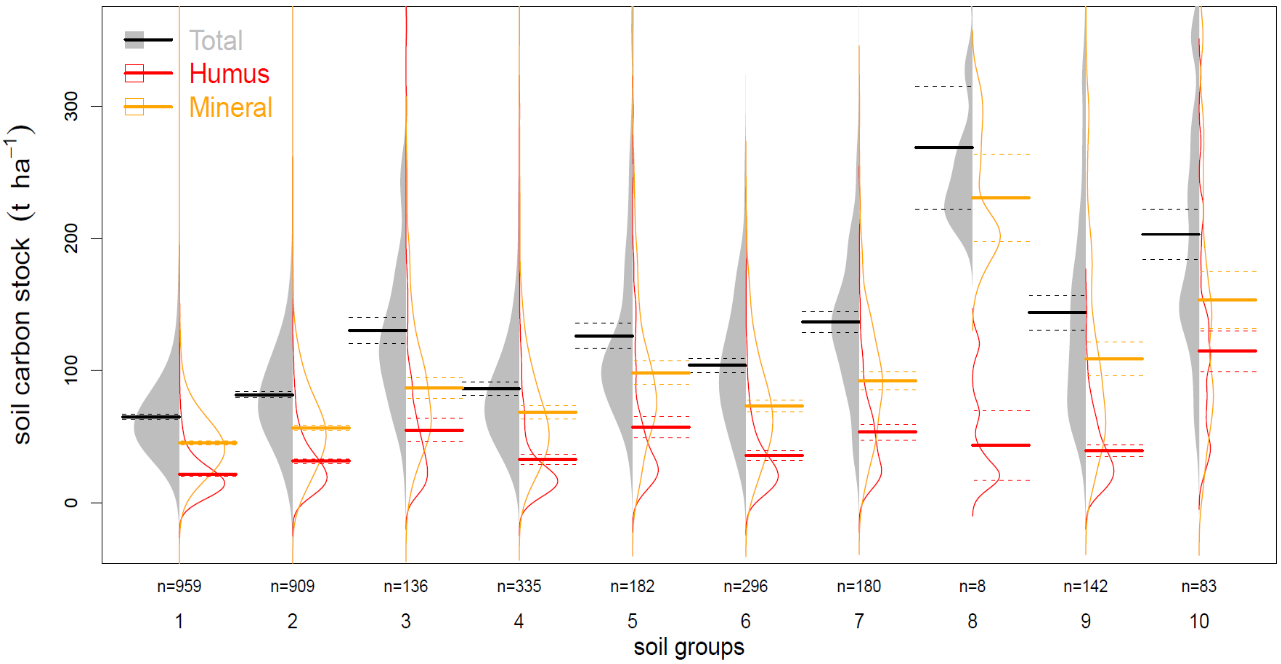

Figure S4. Density functions for 10 physicochemical groups of the soil carbon (SOC) stock ($\mathrm{t C\, ha^{-1}}$) Swedish forest soil inventory measurements for soil depth up to 1 m (total, grey fill) and for the soil humus horizon and mineral soil horizon. The thin lines are the density distributions. The thick lines are the group means and dashed lines are their confidence intervals. The n is number of samples. For description of group levels of SOC stocks, moisture, and fertility see Fig.2 and Table S1.

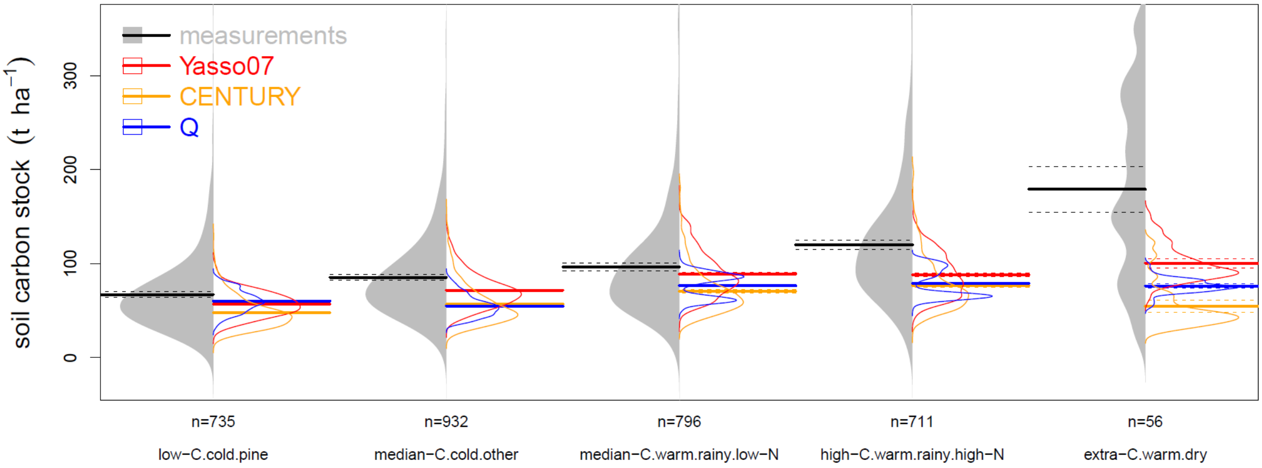

Figure S5. Bean plot of density functions for 5 groups of the soil carbon ($\mathrm{tC\,ha^{-1}}$) measurements (C.measured, grey fill) and soil carbon estimates simulated by the soil carbon models Yasso07, CENTURY, and Q with the litter input derived from the steady state forest. The thin lines are the density distributions. The thick lines are the group means and dashed lines are their confidence intervals. The n is number of samples. For description of group acronyms based on levels of SOC stocks, temperature, percentage of pine in canopy, precipitation, and nitrogen deposition see Fig.3 and Table S2.

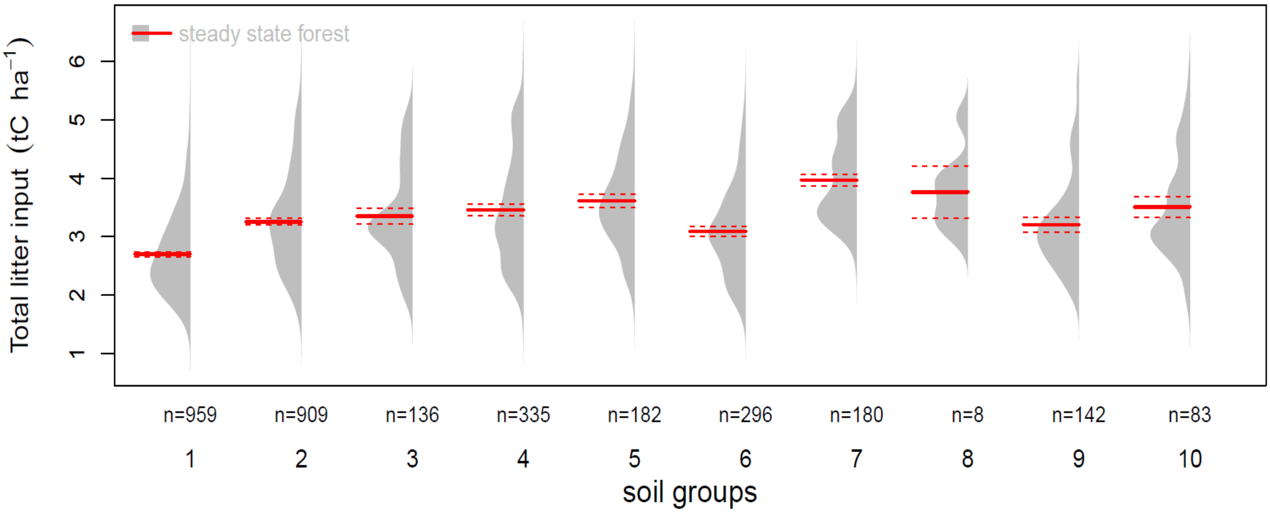

Figure S6. Density functions for 10 physicochemical groups of the total annual plant litter input ($\mathrm{tC\,ha^{-1}}$) of steady state forest. The thick lines are the group means and dashed lines are their confidence intervals. The n is number of samples. For description of group levels of SOC stocks, moisture, and fertility see Fig.2 and Table S1.

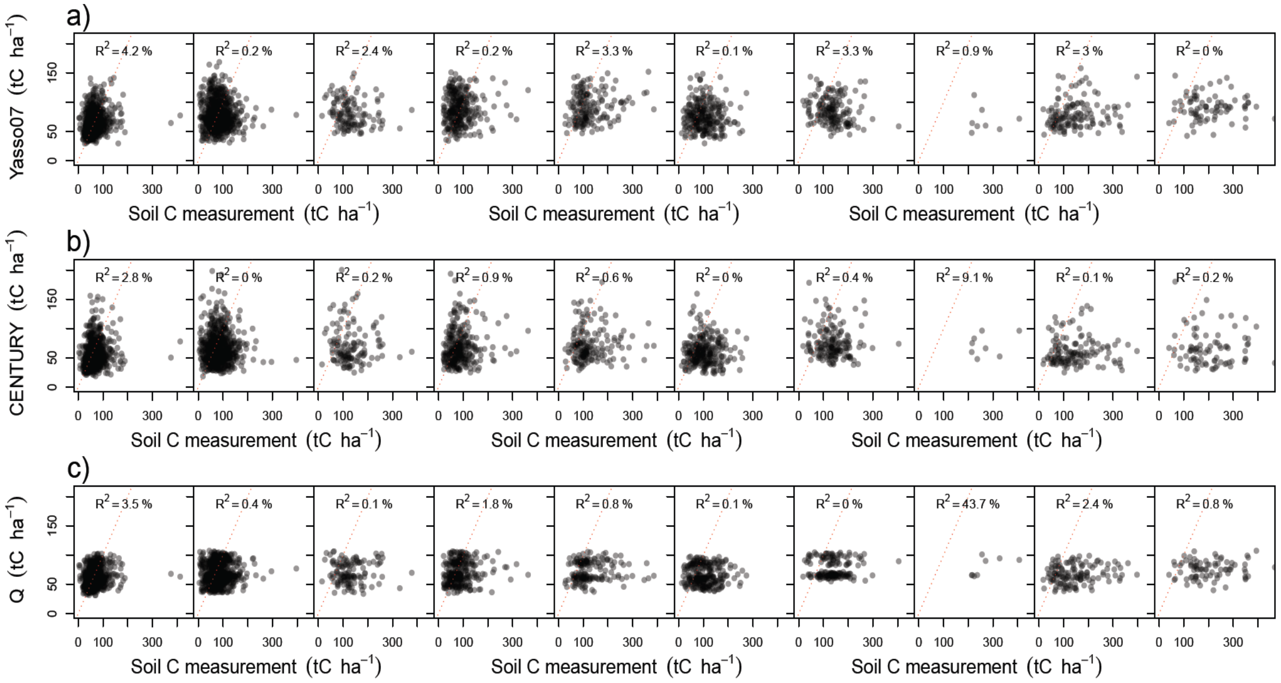

Figure S7. Scatter plots between model soil organic carbon stock ($\mathrm{tC\,ha^{-1}}$) measurements and a) Yasso07 and CENTURY, b) Yasso07 and Q, and c) CENTURY and Q for 10 physicochemical groups of Fig.4.

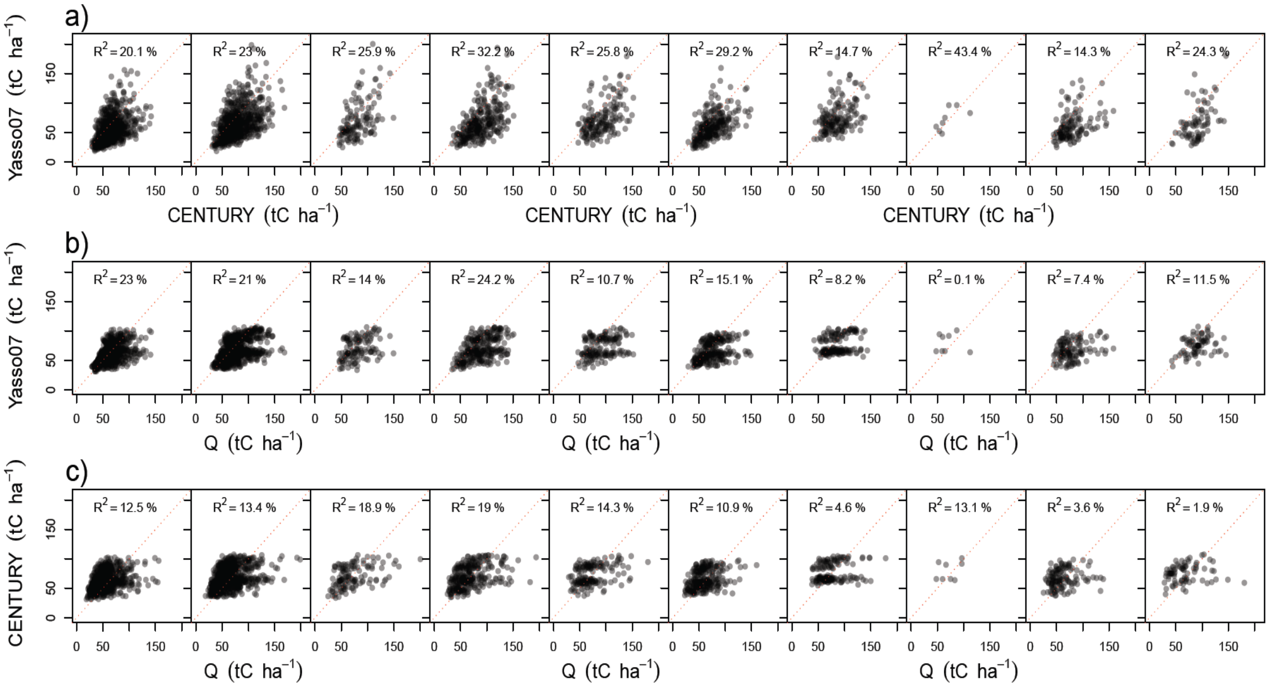

Figure S8. Scatter plots between model soil organic carbon estimates ($\mathrm{tC\,ha^{-1}}$) of a) Yasso07 and CENTURY, b) Yasso07 and Q, and c) CENTURY and Q for 10 physicochemical groups of Fig.4.

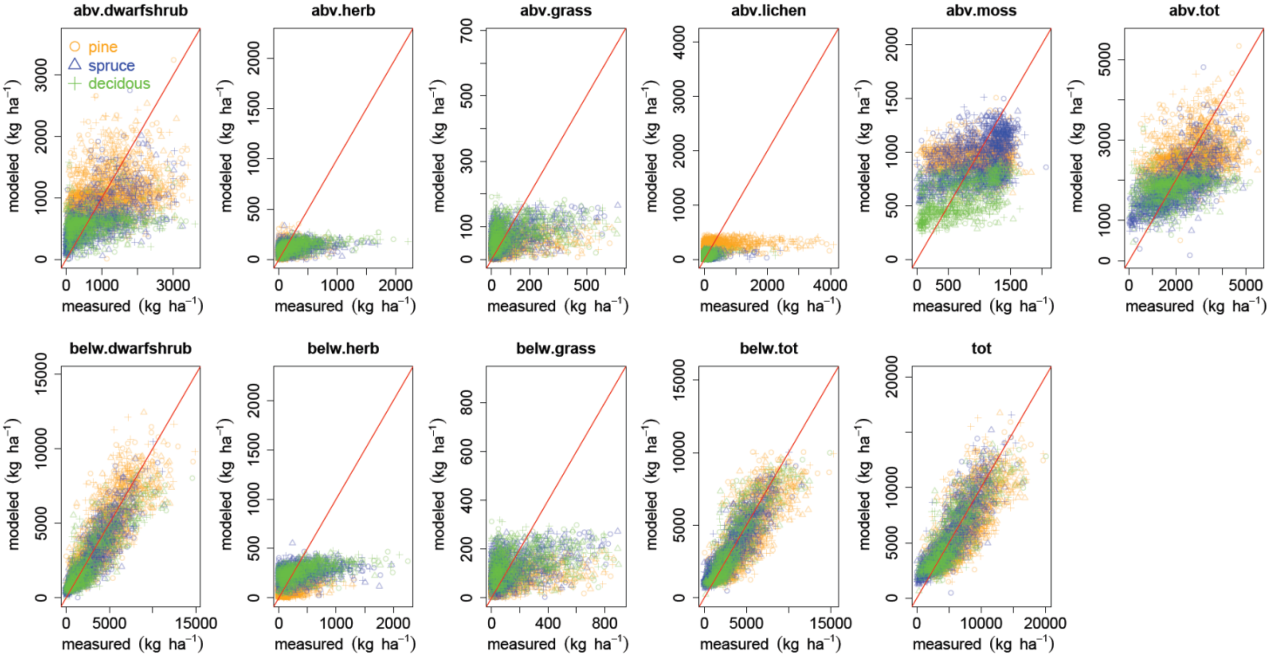

Figure S9. Scatter plots for the dry weight biomass $(\mathrm{kg\,ha^{-1}})$ of the functional types of ground vegetation. On x-axes is the measured coverage multiplied by the coverage/biomass conversion functions and y-axis is the biomass modelled by the ground vegetation dry weight biomass $(\mathrm{kg\,ha^{-1}})$ models. The abbreviations “abv”,”belw”, and ”tot” mean aboveground, belowground and total.