

# Package: CBMutils (via r-universe)

January 23, 2025

**Type** Package

**Title** Utilities for modelling carbon based on CBM-CFS3

**Description** Implementation of several components of the Carbon Budget Model of the Canadian Forest Service (v3).

**URL** <https://CBMutils.predictiveecology.org/>,  
<https://github.com/PredictiveEcology/CBMutils>

**Date** 2023-02-28

**Version** 0.0.7.9013

**License** GPL-3

**Depends** R (>= 4.2)

**LinkingTo** Rcpp

**Imports** crayon, data.table, fasterize, ggforce, ggplot2, magrittr, methods, quickPlot (>= 0.1.8.9001), raster, reproducible, robustbase, Rcpp, RSQLite, scales, sf, sp, terra

**Suggests** covr, devtools, knitr, rmarkdown, testthat, withr

**Remotes** PredictiveEcology/quickPlot@development,  
PredictiveEcology/reproducible@development

**VignetteBuilder** knitr, rmarkdown

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**Config/pak/sysreqs** libfontconfig1-dev libfreetype6-dev libgdal-dev  
gdal-bin libgeos-dev make libssl-dev libproj-dev libsqlite3-dev  
libudunits2-dev

**Repository** <https://predictiveecology.r-universe.dev>

**RemoteUrl** <https://github.com/PredictiveEcology/CBMutils>

**RemoteRef** development

**RemoteSha** cb812bbe4b9fa407f5a1339357dfa0fc4c333fce

## Contents

CBMutils-package . . . . .	3
.pooldef . . . . .	4
barPlot . . . . .	4
biomassTurnoverMatrix . . . . .	5
biomProp . . . . .	5
b_m . . . . .	6
calcC . . . . .	7
calcTurnoverRates . . . . .	7
carbonOutPlot . . . . .	8
checkProp . . . . .	8
checkTransactions . . . . .	9
computeBioTurnoverMatrices . . . . .	10
computeDomDecayMatrices . . . . .	11
computeDomTurnoverMatrices . . . . .	11
ComputeGrowthAndDeclineMatrices . . . . .	12
ComputeGrowthAndDeclineMatrices2 . . . . .	12
ComputeGrowthCoordinateMatrices . . . . .	13
ComputeGrowthIncrements . . . . .	13
ComputeOvermatureDecline . . . . .	14
ComputeOverMatureDeclineCoordinateMatrices . . . . .	14
computeSlowDecayMatrices . . . . .	15
computeSlowMixingMatrix . . . . .	15
convertM3biom . . . . .	16
cumPoolsCreate . . . . .	17
cumPoolsSmooth . . . . .	18
dataset-class . . . . .	18
decayRate . . . . .	19
domDecayMatrix . . . . .	19
domDecayMatrixItem . . . . .	20
domTurnOverMatrix . . . . .	20
gcidsCreate . . . . .	21
getDecayRates . . . . .	21
getIdentityCoordinateMatrix . . . . .	22
getTable . . . . .	22
hash . . . . .	23
histDist . . . . .	23
loadDisturbanceMatrixIds . . . . .	24
m3ToBiomIncOnlyPlots . . . . .	24
m3ToBiomPlots . . . . .	25
m3ToVolCheckPlots . . . . .	26
matrixDT . . . . .	26
nmfac . . . . .	27
NPP . . . . .	27
NPPplot . . . . .	28
plotCarbonRasters . . . . .	29
prepInputsEcozones . . . . .	29

prepInputsVRI . . . . .	30
prepInputsVRlAge . . . . .	30
query . . . . .	31
readSqlFile . . . . .	31
retrieveSpuRaster . . . . .	32
sapfac . . . . .	32
seeDist . . . . .	33
simDist . . . . .	34
slowDecayMatrix . . . . .	34
spatialPlot . . . . .	35
spatialRaster . . . . .	35
spatialUnitDecayRates . . . . .	36
Spinup . . . . .	36
spuDist . . . . .	38
StepPools . . . . .	38
StepPoolsRef . . . . .	39
ws3Build . . . . .	40
<b>Index</b>	<b>41</b>

---

CBMutils-package	CBMutils
------------------	----------

---

## Description

Utilities for modelling carbon in R/SpaDES based on the Carbon Budget Model of the Canadian Forest Service v3 (CBM-CFS3).

## Author(s)

**Maintainer:** Alex M Chubaty <achubaty@for-cast.ca> ([ORCID](#))

Authors:

- Céline Boisvenue <celine.boisvenue@nrca-nrcan.gc.ca>
- Eliot McIntire <eliot.mcintire@nrca-nrcan.gc.ca>

Other contributors:

- Ian Eddy <ian.eddy@nrca-nrcan.gc.ca> [contributor]

## See Also

Useful links:

- <https://CBMutils.predictiveecology.org/>
- <https://github.com/PredictiveEcology/CBMutils>

---

<code>.pooldef</code>	<i>Disturbance matrix pool names and ids</i>
-----------------------	--

---

**Description**

TODO: confirm these

**Usage**

```
.pooldef
.pooldefids
.poolnames
.poolids
```

**Format**

An object of class character of length 26.  
 An object of class environment of length 26.  
 An object of class character of length 25.  
 An object of class environment of length 25.

**Details**

NOTE: A reminder that indexing in R starts at 1, whereas in C++ it starts at 0. Several C++ data structures do not include the Input category, so the indices there are defined using `.poolids - 1`.

---

<code>barPlot</code>	<code>barPlot</code>
----------------------	----------------------

---

**Description**

`barPlot`

**Usage**

```
barPlot(cbmPools)
```

**Arguments**

`cbmPools`      `TODO`

**Value**

`TODO`

---

 biomassTurnoverMatrix *Biomass turnover matrix*


---

**Description**

Biomass turnover matrix

**Usage**

```
biomassTurnoverMatrix(turnoverParam, PoolCount)
```

**Arguments**

turnoverParam	DESCRIPTION NEEDED
PoolCount	DESCRIPTION NEEDED

**Value**

DESCRIPTION NEEDED

---

biomProp	<i>Proportions of total tree biomass in stemwood, bark, branches, and foliage</i>
----------	---

---

**Description**

Implements equations 4-7 of Boudewyn et al. (2007), used to determine the proportions of total tree biomass in stemwood, bark, branches, and foliage ( $p_{stemwood}$ ,  $p_{bark}$ ,  $p_{branches}$ ,  $p_{foliage}$ , respectively), using parameters  $a$ ,  $b$  from Table 6 (table6) and volume-proportion caps from Table 7 (table7).

**Usage**

```
biomProp(table6, table7, vol)
```

**Arguments**

table6	data.frame corresponding to Table 6 from Boudewyn et al. (2007), available from <a href="https://nfi.nfis.org/resources/biomass_models/appendix2_table6.csv">https://nfi.nfis.org/resources/biomass_models/appendix2_table6.csv</a> .
table7	data.frame corresponding to Table 7 from Boudewyn et al. (2007), available from <a href="https://nfi.nfis.org/resources/biomass_models/appendix2_table7.csv">https://nfi.nfis.org/resources/biomass_models/appendix2_table7.csv</a> .
vol	gross merchantable volume per hectare ( $m^3/ha$ )

**Details**

TODO: will eventually add species, ecozone

**Value**

four-column matrix will columns corresponding to  $p_{stemwood}$ ,  $p_{bark}$ ,  $p_{branches}$ , and  $p_{foliage}$

**References**

Boudewyn, P., Song, X., Magnussen, S., & Gillis, M. D. (2007). Model-based, volume-to-biomass conversion for forested and vegetated land in Canada (BC-X-411). Natural Resource Canada, Pacific Forestry Centre. <https://cfs.nrcan.gc.ca/pubwarehouse/pdfs/27434.pdf>

---

b\_m

*Calculate stemwood biomass (per ha) of live merchantable trees*


---

**Description**

Implements equation 1 of Boudewyn et al. (2007) to determines the total stemwood biomass of merchantable trees (in metric tonnes per hectare;  $T/ha$ ), using parameters  $a$  and  $b$  from Table 3 (table3).

**Usage**

```
b_m(table3, vol)
```

**Arguments**

table3	data.frame corresponding to Table 3 from Boudewyn et al. (2007), available from <a href="https://nfi.nfis.org/resources/biomass_models/appendix2_table3.csv">https://nfi.nfis.org/resources/biomass_models/appendix2_table3.csv</a> .
vol	gross merchantable volume per hectare ( $m^3/ha$ )

**Value**

stemwood biomass of merchantable trees ( $b_m$  in units  $T/ha$ )

**References**

Boudewyn, P., Song, X., Magnussen, S., & Gillis, M. D. (2007). Model-based, volume-to-biomass conversion for forested and vegetated land in Canada (BC-X-411). Natural Resource Canada, Pacific Forestry Centre. <https://cfs.nrcan.gc.ca/pubwarehouse/pdfs/27434.pdf>

---

calcC	<i>Sum carbon for totalCarbon or aboveGround or belowGround</i>
-------	---

---

**Description**

Sum carbon for totalCarbon or aboveGround or belowGround

**Usage**

calcC(cbmPools, poolToSum, masterRaster)

**Arguments**

cbmPools	DESCRIPTION NEEDED
poolToSum	DESCRIPTION NEEDED
masterRaster	A SpatRaster object to use as the template

**Value**

DESCRIPTION NEEDED

---

calcTurnoverRates	<i>Calculate turnover rates</i>
-------------------	---------------------------------

---

**Description**

For all spatial units in the simulation, calculate turnover rates.

**Usage**

calcTurnoverRates(turnoverRates, spatialUnitIds, spatialUnits)

**Arguments**

turnoverRates	DESCRIPTION NEEDED
spatialUnitIds	DESCRIPTION NEEDED
spatialUnits	DESCRIPTION NEEDED

**Value**

extracts the turnover rates for the specific SPU you are in. These are used to in the core module to calculate the specific rates, which are the used to calculate Net Primary Productivity (NPP) both in the core module and in the next function.

---

carbonOutPlot	carbonOutPlot
---------------	---------------

---

**Description**

carbonOutPlot

**Usage**

carbonOutPlot(emissionsProducts)

**Arguments**

emissionsProducts  
 TODO

**Value**

invoked for side effect of creating plot

---

checkProp	<i>check that the proportions add up to 1</i>
-----------	---

---

**Description**

check that the proportions add up to 1

**Usage**

checkProp(DTofMatrices, tarCols)

**Arguments**

DTofMatrices	DESCRIPTION NEEDED
tarCols	DESCRIPTION NEEDED

**Value**

DESCRIPTION NEEDED



---

checkTransactions	<i>Verify and check transactions</i>
-------------------	--------------------------------------

---

### Description

Post-simulations, this function takes the last two years of simulations, and verifies that the C transfers are as expected on  $n$  random pixelGroups. It randomly selects three pixelGroups, calculated the expected transactions from time `end(sim)` and time `end(sim) - 1`, and compares them to the last year of simulations. Cases of disturbed, non disturbed, sw, hw, young and old stands were tested.

### Usage

```
checkTransactions(
  cbmPools,
  pixelKeep,
  opMatrixCBM,
  allProcesses,
  pixelGroupC,
  pooldef,
  timeSim,
  outputsSim,
  n = 3
)
```

### Arguments

<code>cbmPools</code>	from <code>sim\$cbmPools</code>
<code>pixelKeep</code>	from <code>sim\$pixelKeep</code>
<code>opMatrixCBM</code>	from <code>sim\$opMatrixCBM</code>
<code>allProcesses</code>	from <code>sim\$allProcesses</code>
<code>pixelGroupC</code>	from <code>sim\$pixelGroupC</code>
<code>pooldef</code>	from <code>sim\$pooldef</code>
<code>timeSim</code>	from <code>time(sim)</code>
<code>outputsSim</code>	from <code>outputs(sim)</code>
<code>n</code>	number of pixelGroups (i.e., size of subset) to check

### Structure

1. Read-in data
2. randomly select 3 pixelGroups
3. read-in carbon pools at `timeSim` and `timeSim - 1` for the three pixelGroup
4. pull-out matrices applied to these three pixelGroups between `t0` and `t1`

5. matrices into a data.table
6. prep pools at t0
7. merge three pools at t0 with their respective transaction matrices ids
8. apply all matrices (8.1:8.9)

TODO: I am passing a whole simList here for simplification, but only these objects needed:

1. \$cbmPools;
2. \$pixelKeep;
3. \$opMatrixCBM;
4. \$allProcesses;
5. \$pooldef.

TODO: there are still a bunch of checks that need to be changed to assertions (if statements?)

All arguments come from a simList object of the same name, except timeSim (which comes from time(sim)) and outputsSim (which comes from outputs(sim)).

---

computeBioTurnoverMatrices

*Compute Turnover Matrices*

---

## Description

Compute Turnover Matrices

## Usage

```
computeBioTurnoverMatrices(turnoverParameters, PoolCount)
```

## Arguments

turnoverParameters	DESCRIPTION NEEDED
PoolCount	DESCRIPTION NEEDED

## Value

DESCRIPTION NEEDED

---

 computeDomDecayMatrices

*Compute all dom decay matrices in coordinate matrix format*


---

**Description**

The first column in the specified decayRates parameter acts as the key to each matrix.

**Usage**

```
computeDomDecayMatrices(decayRates, decayParameters, PoolCount)
```

**Arguments**

decayRates	matrix of decay rates column 1 is the key for the values in columns 1:n and columns 1:n are the DOM pool specific decay rates.
decayParameters	table of CBM decay parameters
PoolCount	DESCRIPTION NEEDED

**Value**

DESCRIPTION NEEDED

---

computeDomTurnoverMatrices

*Compute DOM turnover matrices*


---

**Description**

Compute DOM turnover matrices

**Usage**

```
computeDomTurnoverMatrices(turnoverParameters, PoolCount)
```

**Arguments**

turnoverParameters	DESCRIPTION NEEDED
PoolCount	DESCRIPTION NEEDED

**Value**

DESCRIPTION NEEDED

---

ComputeGrowthAndDeclineMatrices

*Compute growth and decline matrices*

---

### Description

DESCRIPTION NEEDED

### Arguments

growthIncrements	Matrix of growth increments
decline	Matrix containing proportion decline for each pool

---

ComputeGrowthAndDeclineMatrices2

*Compute growth and decline matrices*

---

### Description

DESCRIPTION NEEDED

### Usage

```

ComputeGrowthAndDeclineMatrices2(
  growthIncrements,
  ages,
  gcids,
  pools,
  rootParameters,
  turnoverParams,
  biomassToCarbonRate,
  swMult = 1,
  hwMult = 1
)

```

### Arguments

growthIncrements	a hash table of growth increments by gcid, by age
ages	the stand age, (the inventory age) stands will be simulated to this age in the final pass
gcids	integer vector of growth curve ids
pools	Matrix (DESCRIPTION NEEDED)

rootParameters data.frame specifying root parameters  
 turnoverParams 'data.frame specifying turnover parameters  
 biomassToCarbonRate  
                   numeric DESCRIPTION NEEDED  
 swMult          softwood multiplier? (DESCRIPTION NEEDED)  
 hwMult          hardwood multiplier? (DESCRIPTION NEEDED)

---

ComputeGrowthCoordinateMatrices

*Compute growth coordinate matrices*

---

### Description

Transforms growth matrix into coordinate format matrix in terms of CBM pools.

### Arguments

pn                  PoolNames object  
 growthIncrements  
                   Matrix of growth increments

---

ComputeGrowthIncrements

*Compute growth increments*

---

### Description

Compute the total growth (above ground and belowground) increment matrices for all stands based on age and hashtable of growth increments by growth curve id.

### Usage

```

ComputeGrowthIncrements(
  growthIncrements,
  ages,
  gcids,
  pools,
  rootParameters,
  biomassToCarbonRate,
  swMult = 1,
  hwMult = 1
)

```

**Arguments**

growthIncrements	a hash table of growth increments by gcid, by age
ages	the stand age, (the inventory age) stands will be simulated to this age in the final pass
gcids	DESCRIPTION NEEDED
pools	DESCRIPTION NEEDED
rootParameters	data.frame (DESCRIPTION NEEDED)
biomassToCarbonRate	DESCRIPTION NEEDED
swMult	softwood multiplier? (DESCRIPTION NEEDED)
hwMult	hardwood multiplier? (DESCRIPTION NEEDED)

---

ComputeOvermatureDecline

*Compute overmature decline*

---

**Description**

Compute flows to dom pools that occur on a growth curve decline.

**Usage**

ComputeOvermatureDecline(growthIncrements, turnoverParams)

**Arguments**

growthIncrements	Matrix of growth increments
turnoverParams	data.frame specifying turnover parameters

---

ComputeOverMatureDeclineCoordinateMatrices

*Compute overmature decline coordinate matrices*

---

**Description**

Transforms overmature decline matrix into coordinate format matrix in terms of CBM pools.

**Arguments**

pn	PoolNames object
decline	Matrix containing proportion decline for each pool

---

computeSlowDecayMatrices  
*Compute slow decay matrices*

---

**Description**

Compute slow decay matrices

**Usage**

```
computeSlowDecayMatrices(decayRates, decayParameters, PoolCount)
```

**Arguments**

decayRates	DESCRIPTION NEEDED
decayParameters	DESCRIPTION NEEDED
PoolCount	DESCRIPTION NEEDED

**Value**

DESCRIPTION NEEDED

---

computeSlowMixingMatrix  
*Compute slow mixing matrix*

---

**Description**

Compute slow mixing matrix

**Usage**

```
computeSlowMixingMatrix(slowMixingRate, PoolCount)
```

**Arguments**

slowMixingRate	DESCRIPTION NEEDED
PoolCount	DESCRIPTION NEEDED

**Value**

DESCRIPTION NEEDED

---

 convertM3biom

---

*Calculate biomass from gross merchantable volume*


---

### Description

Implements the flowchart from figure 3 of Boudewyn et al. (2007) to determined the total above ground biomass ( $T/ha$ ) from gross merchantable volume ( $m^3/ha$ ).

### Usage

```
convertM3biom(
  meta,
  gCvalues,
  spsMatch,
  ecozones,
  params3,
  params4,
  params5,
  params6,
  params7
)
```

### Arguments

meta	DESCRIPTION NEEDED
gCvalues	DESCRIPTION NEEDED
spsMatch	DESCRIPTION NEEDED
ecozones	DESCRIPTION NEEDED
params3	data.frame corresponding to Table 3 from Boudewyn et al. (2007), available from <a href="https://nfi.nfis.org/resources/biomass_models/appendix2_table3.csv">https://nfi.nfis.org/resources/biomass_models/appendix2_table3.csv</a> .
params4	data.frame corresponding to Table 4 from Boudewyn et al. (2007), available from <a href="https://nfi.nfis.org/resources/biomass_models/appendix2_table4.csv">https://nfi.nfis.org/resources/biomass_models/appendix2_table4.csv</a> .
params5	data.frame corresponding to Table 5 from Boudewyn et al. (2007), available from <a href="https://nfi.nfis.org/resources/biomass_models/appendix2_table5.csv">https://nfi.nfis.org/resources/biomass_models/appendix2_table5.csv</a> .
params6	data.frame corresponding to Table 6 from Boudewyn et al. (2007), available from <a href="https://nfi.nfis.org/resources/biomass_models/appendix2_table6.csv">https://nfi.nfis.org/resources/biomass_models/appendix2_table6.csv</a> .
params7	data.frame corresponding to Table 7 from Boudewyn et al. (2007), available from <a href="https://nfi.nfis.org/resources/biomass_models/appendix2_table7.csv">https://nfi.nfis.org/resources/biomass_models/appendix2_table7.csv</a> .



**Value**

three-column matrix with columns corresponding to biomass ( $T/ha$ ) for total merchantable, foliage, and other.

**References**

Boudewyn, P., Song, X., Magnussen, S., & Gillis, M. D. (2007). Model-based, volume-to-biomass conversion for forested and vegetated land in Canada (BC-X-411). Natural Resource Canada, Pacific Forestry Centre. <https://cfs.nrcan.gc.ca/pubwarehouse/pdfs/27434.pdf>

---

cumPoolsCreate	<i>Create cumPools data.table</i>
----------------	-----------------------------------

---

**Description**

Create cumPools data.table

**Usage**

```
cumPoolsCreate(
  fullSpecies,
  gcMeta,
  userGcM3,
  stable3,
  stable4,
  stable5,
  stable6,
  stable7,
  thisAdmin
)
```

**Arguments**

fullSpecies	DESCRIPTION NEEDED
gcMeta	DESCRIPTION NEEDED
userGcM3	DESCRIPTION NEEDED
stable3	DESCRIPTION NEEDED
stable4	DESCRIPTION NEEDED
stable5	DESCRIPTION NEEDED
stable6	DESCRIPTION NEEDED
stable7	DESCRIPTION NEEDED
thisAdmin	DESCRIPTION NEEDED

**Value**

cumPools data.table

---

cumPoolsSmooth	<i>Smooth the cumPools data.table</i>
----------------	---------------------------------------

---

### Description

This uses the Chapman Richards equation to smooth the curves that are in colsToUse.

### Usage

```
cumPoolsSmooth(
  cumPoolsRaw,
  colsToUse = c("totMerch", "fol", "other"),
  colsToUseNew = paste0(colsToUse, "_New")
)
```

### Arguments

cumPoolsRaw	Data.table with a numeric column called age`` and columns called colsToUse`.
colsToUse	A character vector of columns to smooth.
colsToUseNew	A character vector of column names for the new smoothed columns

### Value

#' A data.table with original columns plus new columns named 'colsToUseNew'.

### Author(s)

Celine Boisvenue and Eliot McIntire

---

dataset-class	<i>Dataset class</i>
---------------	----------------------

---

### Description

DESCRIPTION NEEDED

---

decayRate	<i>Calculate the decay rate based on mean annual temperature and other parameters</i>
-----------	---

---

**Description**

This is the ecological theory for decomposing being used in CBM.

**Usage**

```
decayRate(meanAnnualTemp, baseDecayRate, q10, tref, max)
```

**Arguments**

meanAnnualTemp	scalar temperature in degrees Celsius
baseDecayRate	scalar base decay rate constant
q10	the scalar q10 value
tref	the reference temperature
max	the maximum allowed decay rate

**Value**

the scalar rate of decay

---

domDecayMatrix	<i>Compute a single dom decay matrix based on the specified table of decay rates</i>
----------------	--

---

**Description**

Compute a single dom decay matrix based on the specified table of decay rates

**Usage**

```
domDecayMatrix(decayRates, decayParameters, PoolCount)
```

**Arguments**

decayRates	vector of decay rates (each element represents a dom pool)
decayParameters	table of cbm decay parameters
PoolCount	DESCRIPTION NEEDED

**Value**

DESCRIPTION NEEDED

---

domDecayMatrixItem      *Calculate a portion of the DOM decay matrix*

---

### Description

Calculate a portion of the DOM decay matrix

### Usage

domDecayMatrixItem(mat, decayRates, propToAtmosphere, src, dst, emission)

### Arguments

mat	the datatable
decayRates	vector of annual decay rates by dom pool
propToAtmosphere	vector of the proportions of decay emitted to the atmosphere as CO2 by this process
src	the integer code for the dom pool being decayed
dst	the integer code for the dom pool receiving non-emitted decayed matter
emission	the integer code for the CO2 pool

### Value

A modified copy of the input mat

---

domTurnOverMatrix      *DOM turnover matrix*

---

### Description

DOM turnover matrix

### Usage

domTurnOverMatrix(turnoverParam, PoolCount)

### Arguments

turnoverParam	DESCRIPTION NEEDED
PoolCount	DESCRIPTION NEEDED

### Value

DESCRIPTION NEEDED

---

gcidsCreate	<i>Standardized way to create gcids</i>
-------------	---

---

**Description**

This uses the underscore delimiter between column names.

**Usage**

```
gcidsCreate(...)
```

**Arguments**

... The data.table with ONLY the columns on which to determine unique gcids

---

getDecayRates	<i>Get decay rates</i>
---------------	------------------------

---

**Description**

Returns a vector of decay rates where the indices of the vector are the DOM pools of CBM.

**Usage**

```
getDecayRates(meanAnnualTemp, decayParameters, domPools)
```

**Arguments**

meanAnnualTemp scalar temperature in deg Celcius  
 decayParameters table of decay parameters for calculating the temperature dependent decay rate  
 domPools DESCRIPTION NEEDED

**Value**

the vector of decay rates

---

getIdentityCoordinateMatrix  
*getIdentityCoordinateMatrix*

---

**Description**

DESCRIPTION NEEDED

**Usage**

getIdentityCoordinateMatrix(size)

**Arguments**

size            Numeric indicating the number of rows and columns in the matrix.

---

getTable            *getTable*

---

**Description**

DESCRIPTION NEEDED

**Usage**

getTable(filename, dbPath, sqlDir)

**Arguments**

filename            DESCRIPTION NEEDED  
dbPath              DESCRIPTION NEEDED  
sqlDir              DESCRIPTION NEEDED

---

hash	<i>Hashing functions</i>
------	--------------------------

---

**Description**

DESCRIPTION NEEDED

**Usage**

hash(x)

matrixHash(x)

**Arguments**

x	DESCRIPTION NEEDED
---	--------------------

---

histDist	<i>CBM-CFS3 Historical Disturbances</i>
----------	---

---

**Description**

Identifies the stand-replacing wildfire disturbance in each spatial unit.

**Usage**

histDist(spuIDs, dbPath = NULL, localeID = 1, listDist = NULL)

**Arguments**

spuIDs	Spatial unit ID(s)
dbPath	Path to CBM-CFS3 SQLite database file
localeID	CBM-CFS3 locale_id
listDist	data.table. Optional. Result of a call to <a href="#">spuDist</a> . A list of possible disturbances in the spatial unit(s) with columns 'spatial_unit_id', 'disturbance_type_id', 'disturbance_matrix_id', 'name', 'description'. If provided, the dbPath and localeID arguments are not required.

**Details**

Historical disturbances in CBM-CFS3 are used for "filling-up" the soil-related carbon pools. Boudewyn et al. (2007) translate the m3/ha curves into biomass per ha in each of four pools: total biomass for stem wood, total biomass for bark, total biomass for branches and total biomass for foliage. Biomass in coarse and fine roots, in aboveground- and belowground- very-fast, -fast, -slow, in medium-soil, and in snags still needs to be estimated. In all spatial units in Canada, the historical disturbance is set to fire. A stand-replacing fire disturbance is used in a disturb-grow cycle, where stands are disturbed and regrown with turnover, overmature, decay, functioning until the dead organic matter pools biomass values stabilize (+/- 10%).

**TODO: (I think but that is in the Rcpp-RCMBGrowthIncrements.cpp so can't check):**

By default the most recent is selected, but the user can change that.

---

```
loadDisturbanceMatrixIds
```

*Load disturbance matrix IDs*

---

**Description**

Load disturbance matrix IDs

**Usage**

```
loadDisturbanceMatrixIds(disturbanceMatrixValues, dbPools)
```

**Arguments**

disturbanceMatrixValues	DESCRIPTION NEEDED
dbPools	DESCRIPTION NEEDED

**Value**

DESCRIPTION NEEDED

---

```
m3ToBiomIncOnlyPlots m3ToBiomIncOnlyPlots
```

---

**Description**

m3ToBiomIncOnlyPlots

**Usage**

```
m3ToBiomIncOnlyPlots(inc)
```



**Arguments**

inc            TODO

**Value**

TODO

---

m3ToBiomPlots	<i>Plot all columns that are not id_col</i>
---------------	---

---

**Description**

Plot all columns that are not id\_col

**Usage**

```
m3ToBiomPlots(
  inc = "increments",
  id_col = "gcids",
  nrow = 5,
  ncol = 5,
  filenameBase = "rawCumBiomass_",
  path = NULL,
  title = "Cumulative merch fol other by gc id",
  scales = "free_y"
)
```

**Arguments**

inc	DESCRIPTION NEEDED
id_col	DESCRIPTION NEEDED
nrow	DESCRIPTION NEEDED
ncol	DESCRIPTION NEEDED
filenameBase	DESCRIPTION NEEDED
path	DESCRIPTION NEEDED
title	DESCRIPTION NEEDED
scales	DESCRIPTION NEEDED

---

m3ToVolCheckPlots	<i>m3ToVolCheckPlots</i>
-------------------	--------------------------

---

**Description**

Check the growth increments by above ground pool. This is necessary because the Boudewyn et al parameters used to translate the m3/ha into biomass/ha do not always work. Returns a list of plots, each plot show the merch, fol, and other increments for a specific growth curve.

**Usage**

```
m3ToVolCheckPlots(sim)
```

**Arguments**

sim	A SpaDES CBM simulation object (simList).
-----	---

---

matrixDT	<i>Make a data.table out of the carbon transfer matrices</i>
----------	--

---

**Description**

Make a data.table out of the carbon transfer matrices

**Usage**

```
matrixDT(matricesIn, indicesIn)
```

**Arguments**

matricesIn	DESCRIPTION NEEDED
indicesIn	DESCRIPTION NEEDED

**Value**

a data.table object summarizing the carbon transfers

---

nmfac	<i>Expansion factor for non-merchantable live tree biomass</i>
-------	--

---

**Description**

Implements equation 2 of Boudewyn et al. (2007), used to determine the total stem wood biomass (in metric tonnes per hectare;  $T/ha$ ) of non-merchantable trees ( $B_n$ ), together with the stemwood biomass of live merchantable and non-merchantable trees ( $B_{nm}$ ), using parameters  $a$ ,  $b$ , and  $k$  from Table 4 (table4).

**Usage**

```
nmfac(table4, eq1, vol)
```

**Arguments**

table4	data.frame corresponding to Table 4 from Boudewyn et al. (2007), available from <a href="https://nfi.nfis.org/resources/biomass_models/appendix2_table4.csv">https://nfi.nfis.org/resources/biomass_models/appendix2_table4.csv</a> .
eq1	stemwood biomass of merchantable trees ( $T/ha$ ) from equation 1 of Boudewyn et al. (2007) (i.e., the result of <code>b_m()</code> ).
vol	gross merchantable volume per hectare ( $m^3/ha$ )

**Value**

two-column matrix with columns corresponding to  $b_n$  and  $b_{nm}$

**References**

Boudewyn, P., Song, X., Magnussen, S., & Gillis, M. D. (2007). Model-based, volume-to-biomass conversion for forested and vegetated land in Canada (BC-X-411). Natural Resource Canada, Pacific Forestry Centre. <https://cfs.nrcan.gc.ca/pubwarehouse/pdfs/27434.pdf>

---

NPP	<i>Calculate post-simulation net primary productivity</i>
-----	---

---

**Description**

Calculate post-simulation net primary productivity

**Usage**

```
NPP(pools, pixelGroupSpu, year1, year2, turnoverRates)
```

**Arguments**

pools	DESCRIPTION NEEDED
pixelGroupSpu	DESCRIPTION NEEDED
year1, year2	Consecutive start and end years (e.g., 1990, 1991)
turnoverRates	DESCRIPTION NEEDED

**Examples**

```
## Not run:
spadesCBMout <- simInitAndSpades() ## TODO: fill this in
NPP(pools = spadesCBMout$cbmPools,
    pixelGroupSpu = spadesCBMout$pixelGroupC[, .(pixelGroup, spatial_unit_id)],
    year1 = 1990, year2 = 1991,
    turnoverRates = spadesCBMout$turnoverRates)

## End(Not run)
```

---

NPPplot

NPPplot

---

**Description**

NPPplot

**Usage**

```
NPPplot(spatialDT, NPP, masterRaster)
```

**Arguments**

spatialDT	TODO
NPP	TODO
masterRaster	A SpatRaster object to use as the template

**Value**

TODO

---

plotCarbonRasters      *Plotting pools*

---

**Description**

Plotting pools

**Usage**

```
plotCarbonRasters(pixelkeep, cbmPools, poolsToPlot, years, masterRaster)
```

**Arguments**

pixelkeep	DESCRIPTION NEEDED
cbmPools	DESCRIPTION NEEDED
poolsToPlot	DESCRIPTION NEEDED
years	DESCRIPTION NEEDED
masterRaster	A SpatRaster object to use as the template

**Examples**

```
## Not run:
# include 'totalCarbon' in poolsToPlot to add plot of total carbon
plotCarbonRasters(
  cbmPools = spadesCBMout$cbmPools,
  poolsToPlot = c("totalCarbon", "BelowGroundSlowSoil"),
  masterRaster = spadesCBMout$masterRaster,
  pixelkeep = spadesCBMout$pixelKeep,
  years = c(1990, 2000, 2005)
)

## End(Not run)
```

---

prepInputsEcozones      prepInputsEcozones

---

**Description**

prepInputsEcozones

**Usage**

```
prepInputsEcozones(url = NULL, dPath, rasterToMatch)
```

**Arguments**

url	A URL to the data
dPath	destination path
rasterToMatch	A RasterLayer with NA representing "off study area" and otherwise any arbitrary value for "in study area". Equivalent to rasterToMatch argument in <a href="#">reproducible::prepInputs()</a> .

---

prepInputsVRI	prepInputsVRI
---------------	---------------

---

**Description**

Read in the BC VRI, with growth curve information (from ws3), and creates a raster stack of the age and gcID.

**Usage**

```
prepInputsVRI(url, dPath, rasterToMatch)
```

**Arguments**

url	A URL to the data
dPath	destination path
rasterToMatch	A RasterLayer with NA representing "off study area" and otherwise any arbitrary value for "in study area". Equivalent to rasterToMatch argument in <a href="#">reproducible::prepInputs()</a> .

---

prepInputsVRIage	<i>Read in the BC VRI, with growth curve information (from ws3), and creates a raster stack of the age</i>
------------------	--

---

**Description**

Read in the BC VRI, with growth curve information (from ws3), and creates a raster stack of the age

**Usage**

```
prepInputsVRIage(
  VRIurl,
  dPath,
  rasterToMatch,
  targetFile,
  field = "PROJ_AGE_1"
)
```

**Arguments**

VRIurl	a URL to the data
dPath	destination path
rasterToMatch	A RasterLayer with NA representing "off study area" and otherwise any arbitrary value for "in study area". Equivalent to rasterToMatch argument in <a href="#">reproducible::prepInputs()</a> .
targetFile	A gdb.zip file that is the inventory file.
field	The age column name in the inventory file.

---

query	<i>query</i>
-------	--------------

---

**Description**

DESCRIPTION NEEDED

**Usage**

```
query(dbPath, sql)
```

**Arguments**

dbPath	Path to sqlite database file.
sql	SQL statement with to execute in the database.

---

readSqlFile	<i>readSqlFile</i>
-------------	--------------------

---

**Description**

DESCRIPTION NEEDED

**Usage**

```
readSqlFile(filePath)
```

**Arguments**

filePath	Path to sqlite database file.
----------	-------------------------------

---

retrieveSpuRaster	<i>Produce a raster with spUnits</i>
-------------------	--------------------------------------

---

### Description

Produce a raster with spUnits

### Usage

```
retrieveSpuRaster(
  spatialUnitsFile = shapefile("data/spUnit_Locator.shp"),
  UserArea,
  rasterRes = c(250, 250)
)
```

### Arguments

spatialUnitsFile	DESCRIPTION NEEDED
UserArea	DESCRIPTION NEEDED
rasterRes	The desired raster resolution.

### Examples

```
## Not run:
test1 <- raster::shapefile("data/forIan/SK_data/CBM_GIS/SpadesCBM_TestArea.shp")
out1 <- retrieveSpuRaster(UserArea = test1, rasterRes = c(250, 250))
if (interactive()) Plot(out1)

## End(Not run)
```

---

sapfac	<i>Expansion factor for sapling-sized trees</i>
--------	---

---

### Description

Implements equation 3 of Boudewyn et al. (2007), used to determine the total stem wood biomass (in metric tonnes per hectare;  $T/ha$ ) of sapling-sized trees ( $B_s$ ), using parameters  $a$ ,  $b$ , and  $k$  from Table 5 (table5).

### Usage

```
sapfac(table5, eq2, vol)
```



**Arguments**

table5	data.frame corresponding to Table 5 from Boudewyn et al. (2007), available from <a href="https://nfi.nfis.org/resources/biomass_models/appendix2_table5.csv">https://nfi.nfis.org/resources/biomass_models/appendix2_table5.csv</a> .
eq2	two-column matrix giving stemwood biomass of non-merchantable trees (i.e., $b_n$ given in units $T/ha$ ), and merchantable + non-merchantable trees (i.e., $b_{nm}$ given in units $T/ha$ ), from equation 2 of Boudewyn et al. (2007) (i.e., the result of <code>nmfac()</code> ).
vol	gross merchantable volume per hectare ( $m^3/ha$ )

**Value**

stemwood biomass of sapling-sized trees ( $b_s$  in units  $T/ha$ )

**References**

Boudewyn, P., Song, X., Magnussen, S., & Gillis, M. D. (2007). Model-based, volume-to-biomass conversion for forested and vegetated land in Canada (BC-X-411). Natural Resource Canada, Pacific Forestry Centre. <https://cfs.nrcan.gc.ca/pubwarehouse/pdfs/27434.pdf>

---

seeDist	<i>See disturbances</i>
---------	-------------------------

---

**Description**

Get the descriptive name of the disturbance, the source pools, the sink pools, and the proportions transferred.

**Usage**

```
seeDist(distId, dbPath)
```

**Arguments**

distId	Description needed
dbPath	Path to sqlite database file.

**Value**

A list of data.frames, one per disturbance matrix id.

---

<code>simDist</code>	<i>get the descriptive name and proportions transferred for disturbances in a simulation requires a simulation list post simulations (from <code>spades()</code>) and returns a list of data.frames. Each data had the descriptive name of a disturbance used in the simulations, the disturbance matrix identification number from <code>cbm_defaults</code>, the pool from which carbon is taken (source pools) in this specific disturbance, the pools into which carbon goes, and the proportion in which the carbon-transfers are completed.</i>
----------------------	---

---

### Description

get the descriptive name and proportions transferred for disturbances in a simulation requires a simulation list post simulations (from `spades()`) and returns a list of data.frames. Each data had the descriptive name of a disturbance used in the simulations, the disturbance matrix identification number from `cbm_defaults`, the pool from which carbon is taken (source pools) in this specific disturbance, the pools into which carbon goes, and the proportion in which the carbon-transfers are completed.

### Usage

```
simDist(sim)
```

### Arguments

<code>sim</code>	A SpaDES CBM simulation ( <code>simList</code> ) object.
------------------	--

---

<code>slowDecayMatrix</code>	<i>Slow decay matrix</i>
------------------------------	--------------------------

---

### Description

Slow decay matrix

### Usage

```
slowDecayMatrix(decayRates, decayParameters, PoolCount)
```

### Arguments

<code>decayRates</code>	DESCRIPTION NEEDED
<code>decayParameters</code>	DESCRIPTION NEEDED
<code>PoolCount</code>	DESCRIPTION NEEDED

**Value**

DESCRIPTION NEEDED

---

spatialPlot	spatialPlot
-------------	-------------

---

**Description**

spatialPlot

**Usage**

spatialPlot(cbmPools, years, masterRaster, spatialDT)

**Arguments**

cbmPools	TODO
years	TODO
masterRaster	A SpatRaster object to use as the template
spatialDT	TODO

**Value**

TODO

---

spatialRaster	<i>Plot</i>
---------------	-------------

---

**Description**

the spatialPlot function plots rasters directly, but does not return rasters, whereas it does so here.

**Usage**

spatialRaster(pixelkeep, cbmPools, poolsToPlot, years, masterRaster)

**Arguments**

pixelkeep	DESCRIPTION NEEDED
cbmPools	DESCRIPTION NEEDED
poolsToPlot	DESCRIPTION NEEDED
years	DESCRIPTION NEEDED
masterRaster	A SpatRaster object to use as the template

**Value**

RasterLayer but invoked for its side-effect of plotting the rasters.

**See Also**

spatialPlot

---

spatialUnitDecayRates *Spatial unit decay rates*

---

**Description**

Spatial unit decay rates

**Usage**

```
spatialUnitDecayRates(climate, decayparameters, domPools)
```

**Arguments**

climate	DESCRIPTION NEEDED
decayparameters	DESCRIPTION NEEDED
domPools	DESCRIPTION NEEDED

**Value**

DESCRIPTION NEEDED

---

Spinup *Spinup*

---

**Description**

Spinup a landscape by running rotations of stand replacing disturbances repeatedly until the pre-disturbance slow pools and the last rotation pre-disturbance slow pools are within a tolerance. Stand then grows to inventory age, and is ready for CBM simulation.

**Usage**

```
Spinup(
  pools,
  opMatrix,
  constantProcesses,
  growthIncrements,
  ages,
  gcids,
  historicdmids,
  lastPassdmids,
  delays,
  minRotations,
  maxRotations,
  returnIntervals,
  rootParameters,
  turnoverParams,
  biomassToCarbonRate,
  debug = FALSE
)
```

**Arguments**

pools	Matrix (DESCRIPTION NEEDED)
opMatrix	Matrix (DESCRIPTION NEEDED)
constantProcesses	a list of constant process C dynamics matrices
growthIncrements	a hash table of growth increments by gcid, by age
ages	the stand age, (the inventory age) stands will be simulated to this age in the final pass
gcids	the growth curve ids (referenced by growthIncrements)
historicdmids	(DESCRIPTION NEEDED)
lastPassdmids	(DESCRIPTION NEEDED)
delays	(DESCRIPTION NEEDED)
minRotations	(DESCRIPTION NEEDED)
maxRotations	(DESCRIPTION NEEDED)
returnIntervals	(DESCRIPTION NEEDED)
rootParameters	data.frame specifying root parameters
turnoverParams	data.frame specifying turnover parameters
biomassToCarbonRate	(DESCRIPTION NEEDED)
debug	logical indicating whether to run spinup in debug mode.

---

spuDist                      *CBM-CFS3 Spatial Unit Disturbances*

---

### Description

Identify the disturbances possible in spatial units.

### Usage

```
spuDist(spuIDs, dbPath, localeID = 1)
```

### Arguments

spuIDs	Spatial unit ID(s)
dbPath	Path to CBM-CFS3 SQLite database file
localeID	CBM-CFS3 locale_id

### Value

data.table with columns 'spatial\_unit\_id', 'disturbance\_type\_id', 'disturbance\_matrix\_id', 'name', 'description'

---

StepPools                      *Step through pools, computing carbon transfers*

---

### Description

Computes by row (stand/pixelGroup) across all columns (pools).

### Usage

```
StepPools(pools, opMatrix, flowMatrices)
```

### Arguments

pools	A numeric matrix describing the amount of carbon in each pool (column) across multiple stands/pixel groups (rows).
opMatrix	A numeric matrix with columns corresponding to flows, and rows corresponding to stands/pixel groups. Values are indices into the flowMatrices.
flowMatrices	A list of lists (or environments) containing all carbon flow matrices.

**Examples**

```

nPixGrp <- 3

pools <- matrix(c(1.0, 10.0, 0.0, 1.0, 20.0, 0.0, 1.0, 5.0, 0.0),
               ncol = 3, nrow = nPixGrp, byrow = TRUE)
colnames(pools) <- c("input", "pool1", "pool2")
pools

##      input pool1 pool2
## [1,]     1    10     0
## [2,]     1    20     0
## [3,]     1     5     0

op <- matrix(rep(c(1, 1), nPixGrp), ncol = 2, nrow = nPixGrp, byrow = TRUE)
colnames(op) <- c("disturbance", "growth")

cnames <- c("row", "col", "value")
dist <- matrix(c(2, 3, 1, 1, 1, 1), ncol = 3, nrow = 2, byrow = TRUE)
colnames(dist) <- cnames
grow <- matrix(c(1, 2, 0.1, 1, 3, 0.2, 2, 3, 0.3, 3, 3, 1.0), ncol = 3, nrow = 4, byrow = TRUE)
colnames(grow) <- cnames
flow <- list(Disturbance = list(dist), Growth = list(grow))

new_pools <- StepPools(pools, op, flow)
new_pools

##      input pool1 pool2
## [1,]     1  0.1 10.2
## [2,]     1  0.1 20.2
## [3,]     1  0.1  5.2

```

StepPoolsRef

*Step through pools, computing carbon transfers***Description**

Computes by row (stand/pixelGroup) across all columns (pools) in place.

**Usage**

```
StepPoolsRef(pools, opMatrix, flowMatrices)
```

**Arguments**

pools	A numeric matrix describing the amount of carbon in each pool (column) across multiple stands/pixel groups (rows).
opMatrix	A numeric matrix with columns corresponding to flows, and rows corresponding to stands/pixel groups. Values are indices into the flowMatrices.

flowMatrices    A list of lists (or environments) containing all carbon flow matrices.

---

ws3Build

ws3Build

---

### Description

ws3, the harvest optimizer created by Greg Paradis provided the BC disturbance rasters (harvest or harvest and fires) per TSA and not for the whole study area. This function reads-in the file structure, pulls in all the TSA-level rasters, knits them together and post-processes them to match the rasterToMatch.

### Usage

```
ws3Build(masterRaster, tsaDirs, years, pathsTifs)
```

### Arguments

masterRaster	A SpatRaster object to use as the template
tsaDirs	character vector of paths to the TSA directories
years	numeric or integer vector specifying the years for the data
pathsTifs	character vector of paths for the geotiff files



# Index

- \* **datasets**
  - .pooldef, 4
  - .pooldef, 4
  - .pooldefids (.pooldef), 4
  - .poolids (.pooldef), 4
  - .poolnames (.pooldef), 4
  
- b\_m, 6
- b\_m(), 27
- barPlot, 4
- biomassTurnoverMatrix, 5
- biomProp, 5
  
- calcC, 7
- calcTurnoverRates, 7
- carbonOutPlot, 8
- CBMutils (CBMutils-package), 3
- CBMutils-package, 3
- checkProp, 8
- checkTransactions, 9
- computeBioTurnoverMatrices, 10
- computeDomDecayMatrices, 11
- computeDomTurnoverMatrices, 11
- ComputeGrowthAndDeclineMatrices, 12
- ComputeGrowthAndDeclineMatrices2, 12
- ComputeGrowthCoordinateMatrices, 13
- ComputeGrowthIncrements, 13
- ComputeOvermatureDecline, 14
- ComputeOvermatureDeclineCoordinateMatrices, 14
- computeSlowDecayMatrices, 15
- computeSlowMixingMatrix, 15
- convertM3biom, 16
- cumPoolsCreate, 17
- cumPoolsSmooth, 18
  
- dataset-class, 18
- decayRate, 19
- domDecayMatrix, 19
- domDecayMatrixItem, 20
  
- domTurnOverMatrix, 20
  
- gcidsCreate, 21
- getDecayRates, 21
- getIdentityCoordinateMatrix, 22
- getTable, 22
  
- hash, 23
- histDist, 23
  
- loadDisturbanceMatrixIds, 24
  
- m3ToBiomIncOnlyPlots, 24
- m3ToBiomPlots, 25
- m3ToVolCheckPlots, 26
- matrixDT, 26
- matrixHash (hash), 23
  
- nmfac, 27
- nmfac(), 33
- NPP, 27
- NPPplot, 28
  
- plotCarbonRasters, 29
- prepInputsEcozones, 29
- prepInputsVRI, 30
- prepInputsVRIage, 30
  
- query, 31
  
- readSqlFile, 31
- reproducible::prepInputs(), 30, 31
- retrieveSpuRaster, 32
  
- sapfac, 32
- seeDist, 33
- simDist, 34
- slowDecayMatrix, 34
- spatialPlot, 35
- spatialRaster, 35
- spatialUnitDecayRates, 36

Spinup, [36](#)  
spuDist, [23](#), [38](#)  
StepPools, [38](#)  
StepPoolsRef, [39](#)  
  
ws3Build, [40](#)