Package 'FAIBCompiler'

March 28, 2025

Title FAIB compilers

Version 3.0000

Description Home for VRI and PSP compilers

License Apache License (== 2.0) | file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports methods,
dplyr,
data.table,
fpCompare,
rmarkdown,
bcmaps,
bcdata,
nlme,
spatstat,
jsonlite,
openxlsx,
haven,
MuMIn,
SIndexR,
FAIBOracle,
FAIBBase,
sf,
fs,
lmfor,
reshape

Suggests knitr

Contents

---

ageByForester   *Site age by experienced forester*

Description

This function derives the site age based on experienced forester. It is equivalent to `vri_age` macro in original SAS compiler.

Usage

```
ageByForester(projectID, sampleNumber, sampleTypeCode)
```

Arguments

projectID	character, Project ID.
sampleNumber	character, Sample number.
sampleTypeCode	character, Sample type code.

Value

Age provided by experienced forester

Note

Contact Bob Krahn for details

Author(s)

Yong Luo

---

ageRangeClassifier   *Derive age range code*

Description

This function derives age range code based on `age`, `species` and `FIZ`. The returned age range code includes: 1-young(immature), 2-older(immature), 3-mature and 4-overmature. This function is equivalent to age_rng.sas macro.

Usage

```
ageRangeClassifier(age, species, FIZ)
```

Arguments

age	numeric, Usually layer mean age. what does this mean?
species	character, Tree basic species code, which is SP0 in VRI original data.
FIZ	character, BC forest inventory zone.

Value

DWB age range code

Author(s)

Yong Luo

---

applyDWB	*Apply decay, waste and breakage percentage to gross merchantable volume-VRI specific*

Description

This function calculates merchantable volume after removing decay, waste and breakage in VRI compiler. The function is part of `cp_vegi_2017.sas` to derive `tree_ms7`.

Usage

```
applyDWB(treeMS)
```

Arguments

treeMS   data.table, Compiled full and enhanced trees with percentage of decay, waste and breakage.

Value

A data table that contains VOL_NTWB (net volume that waste 2 wood and breakage), VOL_D (merchantable volume after removing decay), VOL_DW (merchantable volume after removing decay and waste) and VOL_DWB (merchantable volume after removing decay, waste and breakage).

Author(s)

Yong Luo

---

assignChangeComponent	*Assign component change for repeatedly-measured trees*

Description

This function is to assign component change for repeatedly-measured trees based on trees' live and dead, and visit number

Usage

```
assignChangeComponent(treelist, samples)
```

Arguments

treelist   data.table, A tree list data table contains trees' unique tree id, i.e., site_identifier,
                plot, and tree_number, trees' diameters (DIAMETER), live and dead status (LV_D),
                and MEASUREMENT_ANOMALY_CODE

samples   data.table, Table contains if a sample is measured (SAMP_TYP) in fixed area plot
                (F) or variable area plot (V).

Value

one table that contains tree id and component change (column COMPONENT_CHANGE).

- NA: at the site first measurement, A tree was measured
- I: ingrowth, a tree passed the size shreshold in the sites' remeasurements
- S: suivival, a tree is live in both previous measurement and current measurement
- M: mortality, a tree is dead in current measurement but alive in the previous measurement
- D: dead, a tree is dead in current measurement and dead in previous measurement
- H: harvested, tree is harvested
- DROP: droped, a tree is dropped for some reason when it is live

Author(s)

Yong Luo

---

assignMeasInt   *assign measurement intensity based on field measured attributes*

Description

This function it to assign measurement intensity based on field measured attributes.

Usage

```
assignMeasInt(compilationType, vic, vii, vid)
```

Arguments

compilationType

                character, either PSP or nonPSP. If it is PSP, it is consistent with original PSP
                compiler, otherwise, it is consistent with VRI compiler.

vic   data.table, vic.

vii   data.table, vii.

vid   data.table, vid.

Value

Two data tables: fullDimTrees contains all the trees that have both DBH and HT, nonHTTrees
contains the trees that have only DBH infor.

Author(s)

Yong Luo

| auxiTreeCompiler | *Derive volume components for H-enhanced and non-enhanced trees using external coeff and ratio-VRI specific* |

Description

Estimates volume components for H-enhanced and non-enhanced trees using regression and ratio methods. For H-enhanced trees, the whole stem volume and gross merchantable volume are already calculated directly using taper equations; and rest of volume components will be calculated using ratio method in this function. For non-enhanced trees, the whole stem volume is derived using regression equation between basal area and whole stem volume and the rest of volume components will be computed using ratio method in this function.

Usage

```
auxiTreeCompiler(
  fullMeasuredTrees,
  auxiTrees,
  clusterPlotHeader,
  fixedCoeff,
  randomCoeff,
  ratios
)
```

Arguments

fullMeasuredTrees

> Compiled tree-level data in vi_c, which contains full measured trees, enhanced trees and H-enhanced trees. This data is output of DWBCompiler

auxiTrees   data.table, Non-enhanced trees in anxilirary plots, however, it may have enhanced trees and H-enhanced trees. An output from VRIInit_auxTree.

clusterPlotHeader

> data.table, Cluster and plot-level information. An output of VRIInit_clusterplot.

fixedCoeff   data.table, Specifies the WSV-BA equations by project group PRJ_GRP, live and dead status LV_D, stand and falling status SF_COMPILE and species code SP0. The table can be generated using WSV_BARegression.

randomCoeff   data.table, Specifies the WSV-BA equations by project group PRJ_GRP, live and dead status LV_D, stand and falling status SF_COMPILE and species code SP0. The table can be generated using WSV_BARegression.

ratios   data.table, Specifies toWSV ratio by project group PRJ_GRP, live and dead status LV_D, stand and falling status SF_COMPILE and species code SP0. The table can be generated using toWSVRatio.

Value

A list of four tables: 1. fullenhancedtrees: full and enhanced trees; 2. HnonenhancedTrees: Height enhanced and non-enhanced trees;

Note

The data selection procedure for regression has been standardized as following:

1. Start from vi_c, which has all trees have minimum information of DBH and Height;

2. Select all the full, enhanced and H-enhanced trees;

3. Remove observations in Audit plots and have zero whole stem volume;

4. Select the latest observation for each tree by live_dead status. A tree's identity is considered same when it is from same proj_id, samp_no and plot.

The data selection for ratio has been standardized as following:

1. Start from the that used for regression;

2. Select all the full and enhanced trees;

3. Select the trees with DBH >= 10cm

Author(s)

Yong Luo

---

boredAgeCalculator_Bore

Derive bored age using office and field bored age

Description

This function is to derive bore age based on either office bored age (officeBoredAge) or field bored age (fieldBoredAge). When both bore age information are available, the function takes officeBoredAge as priority. The function is one of the four functions that derive bored age using different method. The rests are boredAgeCalculator_Total, boredAgeCalculator_Phys and boredAgeCalculator_Prorated.

Usage

```
boredAgeCalculator_Bore(officeBoredAge, fieldBoredAge)

## S4 method for signature 'numeric,numeric'
boredAgeCalculator_Bore(officeBoredAge, fieldBoredAge)

## S4 method for signature 'numeric,missing'
boredAgeCalculator_Bore(officeBoredAge)

## S4 method for signature 'missing,numeric'
boredAgeCalculator_Bore(fieldBoredAge)
```

Arguments

officeBoredAge   numeric, Office bored age, which is measured in lab by professionals.

fieldBoredAge   numeric, Field bored age, estimated in field by field crew.

Value

bored age

Author(s)

Yong Luo

See Also

[boredAgeCalculator_Total](#) [boredAgeCalculator_Phys](#) [boredAgeCalculator_Prorated](#)

---

boredAgeCalculator_Crted

Derive age at breast height

Description

This function uses site tools to derive age at breast height for the bored age that is not taken at breast height, i.e., 1.3 m. The process was documented in BC VRI Sample Data Compilation Process.

Usage

```
boredAgeCalculator_Crted(
  boredAge,
  boredHeight,
  treeHeight,
  species,
  FIZ,
  compilationType
)

## S4 method for signature
## 'numeric,numeric,numeric,character,character,character'
boredAgeCalculator_Crted(
  boredAge,
  boredHeight,
  treeHeight,
  species,
  FIZ,
  compilationType
)
```

Arguments

boredAge	numeric, Age at bored height
boredHeight	numeric, Height at the bore core.
treeHeight	numeric, Total tree height
species	character, Species code, must be consistent with the species code in site tools. Can be derived using [siteToolsSpeciesConvertor](#).
FIZ	character, Forest inventory zone.
compilationType	
	character, Compilation type.

Value

Tree age at breast height (1.3 m). For the estimated age <= 0.1, the bored age will be used as breast age and a warning message will be given.

Author(s)

Yong Luo

---

boredAgeCalculator_Phys

Derive bored age using physiological age

Description

This function is to derive bore age based on physiological age (physAge). The function is one of the four functions that derive bored age using different method. The rests are boredAgeCalculator_Bore, boredAgeCalculator_Total and boredAgeCalculator_Prorated.

Usage

```
boredAgeCalculator_Phys(physAge)

## S4 method for signature 'numeric'
boredAgeCalculator_Phys(physAge)
```

Arguments

physAge   numeric, Pysiological age.

Value

bored age

Author(s)

Yong Luo

See Also

boredAgeCalculator_Total boredAgeCalculator_Phys boredAgeCalculator_Prorated

boredAgeCalculator_Prorated

Derive bored age using pro-rated age

Description

This function is to derive bore age based on diameter at bore (boreDiameter), bark thickness (barkThickness), pro-rated ring length (ringLength_prorated) and pro-rated ring count (ringCount_prorated). The function is one of the four functions that derive bored age using different method. The rests are boredAgeCalculator_Bore, boredAgeCalculator_Total and boredAgeCalculator_Phys.

Usage

```
boredAgeCalculator_Prorated(
  ringLength_prorated,
  ringCount_prorated,
  boreDiameter,
  barkThickness
)

## S4 method for signature 'numeric,numeric,numeric,numeric'
boredAgeCalculator_Prorated(
  ringLength_prorated,
  ringCount_prorated,
  boreDiameter,
  barkThickness
)

## S4 method for signature 'numeric,numeric,numeric,missing'
boredAgeCalculator_Prorated(
  ringLength_prorated,
  ringCount_prorated,
  boreDiameter
)
```

Arguments

ringLength_prorated
                numeric, Pro-rated ring length in cm

ringCount_prorated
                numeric, Pro-rated ring count

boreDiameter   numeric, Diameter at bore in cm

barkThickness   numeric, Bark thickness in mm. If missing, 0.05 is used in the function.

Value

bored age

Author(s)

Yong Luo

See Also

[boredAgeCalculator_Total](#) [boredAgeCalculator_Phys](#) [boredAgeCalculator_Prorated](#)

---

boredAgeCalculator_Total

Derive bored age using total age

Description

This function is to derive bore age based on total age (totalAge).The function is one of the four functions that derive bored age using different method. The rests are [boredAgeCalculator_Bore](#), [boredAgeCalculator_Phys](#) and [boredAgeCalculator_Prorated](#).

Usage

```
boredAgeCalculator_Total(totalAge)

## S4 method for signature 'numeric'
boredAgeCalculator_Total(totalAge)
```

Arguments

totalAge   numeric, Total tree age, ie., age at height of 0.

Value

bored age

Author(s)

Yong Luo

See Also

[boredAgeCalculator_Total](#) [boredAgeCalculator_Phys](#) [boredAgeCalculator_Prorated](#)

---

checkMaps   *Load maps from map source*

Description

This function is to load maps from mapSourcePath and save them to mapPath. For TSA and BEC, the maps are direct from bcmaps package.

Usage

```
checkMaps(mapPath)
```

Arguments

mapPath   character, The path to save all the maps. Note that all the saved maps have time
                 stemps, which suggest when the files have been modified in `mapSourcePath`.

Value

no value is returned from this function.

Note

all the maps relies on the bcdata package.

Author(s)

Yong Luo

---

compilerPathSetup   *Setup output paths of the compiler*

Description

This function does two things: 1. create folders that will store compiled data; 2. return paths that
directs the compiled folder.

Usage

```
compilerPathSetup(
  compilationPath = ".",
  compilationDate,
  recompile,
  archiveDate
)
```

Arguments

compilationPath

                 character, Specifies a path to store whole compilation process. If missing, the
                 current work directory will be used.

compilationDate

                 character, Specifies a compilation date. It should be in format of YYYYMMDD
                 It will be used for archive the compilation outputs.

recompile   logical, Defines whether we want to use existing data that downloaded previ-
                 ously.

archiveDate   character, Defines on which archive date the raw data were downloaded. These
                 raw data will be used for recompilation. Format is YYYYMMDD.

Value

Seven paths will be returned as following:

- raw_from_oracle Path to save all data that read from both oracle and txt database without merging.
- compilation_sa Path to save merged data for key tables (i.e., vi_a to vi.g) from both oracle and txt database.
- compilation_db Path to save compiled outputs.
- compilation_archive Path to archive all compilation process.
- compilation_report Path to report compilation process.
- compilation_map Path to archive all maps for compilation process.
- compilation_coeff Path to archive all coefficients for compilation process.
- compilation_last Path to archive last compilation process.

Note

Could overwrite the existing output folder, depending on user's choise, i.e., yes or no.

Author(s)

Yong Luo

---

compilerPathSetup_new   *Setup output paths of the compiler*

Description

This function does two things: 1. create folders that will store compiled data; 2. return paths that directs the compiled folder.

Usage

```
compilerPathSetup_new(
  compilationPath = ".",
  compilationDate,
  compilationType,
  recompile,
  download,
  archiveDate
)
```

Arguments

compilationPath

                character, Specifies a path to store whole compilation process. If missing, the current work directory will be used.

compilationDate

                character, Specifies a compilation date. It should be in format of YYYYMMDD It will be used for archive the compilation outputs.

compilationType

> character, either PSP or nonPSP. If it is PSP, it is consistent with original PSP compiler, otherwise, it is consistent with VRI compiler.

recompile   logical, Defines whether we want to use existing data that downloaded previously.

download   logical, Defines whether we want to use existing data that downloaded previously.

archiveDate   character, Defines on which archive date the raw data were downloaded. These raw data will be used for recompilation. Format is YYYYMMDD.

Value

Seven paths will be returned as following:

- raw_from_oracle Path to save all data that read from both oracle and txt database without merging.
- compilation_sa Path to save merged data for key tables (i.e., vi_a to vi.g) from both oracle and txt database.
- compilation_db Path to save compiled outputs.
- compilation_archive Path to archive all compilation process.
- compilation_report Path to report compilation process.
- compilation_map Path to archive all maps for compilation process.
- compilation_coeff Path to archive all coefficients for compilation process.
- compilation_last Path to archive last compilation process.

Note

Could overwrite the existing output folder, depending on user's choise, i.e., yes or no.

Author(s)

Yong Luo

---

dataPrepSample   *prepare raw data from ismc for ISMCCompiler*

Description

This function is to prepare ismc data for compilation and save them in outputPath folder

Usage

```
dataPrepSample(compilationType, inputPath, outputPath, coeffPath)
```

Arguments

compilationType

character, either PSP or nonPSP. If it is PSP, it is consistent with original PSP
compiler, otherwise, it is consistent with VRI compiler.

inputPath   character, Specifies the path that stores data from oracle data base.

outputPath   character, Specifies the path to save your outputs. If missing, the current working
                directory will be choose.

coeffPath   character, Specifies the path where the crosswalk table is stored.

Value

no item returned

Author(s)

Yong Luo

---

dataPrepTree   *prepare raw data from ismc for ISMCCompiler*

Description

This function is to prepare ismc data for compilation and save them in outputPath folder

Usage

```
dataPrepTree(compilationType, inputPath, outputPath, coeffPath, sampleMsmts)
```

Arguments

compilationType

character, either PSP or nonPSP. If it is PSP, it is consistent with original PSP
compiler, otherwise, it is consistent with VRI compiler.

inputPath   character, Specifies the path that stores data from oracle data base.

outputPath   character, Specifies the path to save your outputs. If missing, the current working
                directory will be choose.

coeffPath   character, Specifies the path where the crosswalk table is stored.

Value

no item returned

Author(s)

Yong Luo

```
DBHGrowthRateGenerator
```
Derive DBH growth rate by DBH class, growth length class, species and bec zone

Description

Derive DBH growth rate by DBH class, growth length class, species and bec zone

Usage

```
DBHGrowthRateGenerator(treeData, minObs = 100)
```

Arguments

minObs	numeric, Specify minimum number of observation for calculating mean growth rate. If missing, default is 100.
treesData	data.table, Contains both height trees and nonHT trees in PSPs.

Value

Growth rate lookup table

Author(s)

Yong Luo

```
DBH_Height_MEM
```
Develop the mixed effect models between DBH and height by strata.

Description

Develop the mixed effect model, and select the best models per strata.

Usage

```
DBH_Height_MEM(compilationPath, coeffSavePath, fityear)
```

Arguments

compilationPath	
	character, Specifies the path that stores all the data/processes.
coeffSavePath	character, Specifies the path to save your outputs.
fityear	numeric, Specifies the year of the fit.

Value

no item returned

Author(s)

Yong Luo

DWBCompiler   *Compile decay, waste and breakage for standard tables*

Description

This function compiles decay, waste and breakage for standard tables in VRI compiler. The function is equivalent to dwb_vri_2017.sas.

Usage

```
DWBCompiler(compilationType, treeMS, siteAge, treeLossFactors, equation)

## S4 method for signature
## 'character,data.table,data.table,data.table,character'
DWBCompiler(compilationType, treeMS, siteAge, treeLossFactors, equation)

## S4 method for signature 'character,data.table,data.table,data.table,missing'
DWBCompiler(compilationType, treeMS, siteAge, treeLossFactors)
```

Arguments

compilationType

> character, either PSP or nonPSP. If it is PSP, it is consistent with original PSP compiler, otherwise, it is consistent with VRI compiler.

treeMS   data.table, Tree-level data that has been compiled whole stem volume and gross merchantable volume for full and enhanced trees.

siteAge   data.table, Cluster-level summaries of age and height. This table is an output from [siteAgeSummary](#)

treeLossFactors

> data.table, The tree loss factor data, an output of [VRIInit_lossFactor](#). In this funtion, this table provides loss indicator.

equation   character, Specifies whether the compiler is based on KFIZ or KBEC. Default is set as KBEC.

Value

A compiled volume after removing decay, waste and breakage; a log file

Author(s)

Yong Luo

DWBCompiler_PSP   *Compile decay, waste and breakage for standard tables-VRI specific*

Description

This function compiles decay, waste and breakage for standard tables in VRI compiler. The function is equivalent to dwb_vri_2017.sas.

Usage

```
DWBCompiler_PSP(treeMS, siteAge, treeLossFactors, equation)

## S4 method for signature 'data.table,data.table,data.table,character'
DWBCompiler_PSP(treeMS, siteAge, treeLossFactors, equation)

## S4 method for signature 'data.table,data.table,data.table,missing'
DWBCompiler_PSP(treeMS, siteAge, treeLossFactors)
```

Arguments

treeMS   data.table, Tree-level data that has been compiled whole stem volume and gross
                merchantable volume for full and enhanced trees.

siteAge   data.table, Cluster-level summaries of age and height. This table is an output
                from siteAgeSummary

treeLossFactors
                data.table, The tree loss factor data, an output of VRIInit_lossFactor. In this
                funtion, this table provides loss indicator.

equation   character, Specifies whether the compiler is based on KFIZ or KBEC. Default is
                set as KBEC.

Value

A compiled volume after removing decay, waste and breakage; a log file

Author(s)

Yong Luo

DWBCompiler_VRI   *Compile decay, waste and breakage for standard tables-VRI specific*

Description

This function compiles decay, waste and breakage for standard tables in VRI compiler. The function is equivalent to dwb_vri_2017.sas.

Usage

```
DWBCompiler_VRI(treeMS, siteAge, treeLossFactors, equation)

## S4 method for signature 'data.table,data.table,data.table,character'
DWBCompiler_VRI(treeMS, siteAge, treeLossFactors, equation)

## S4 method for signature 'data.table,data.table,data.table,missing'
DWBCompiler_VRI(treeMS, siteAge, treeLossFactors)
```

Arguments

treeMS   data.table, Tree-level data that has been compiled whole stem volume and gross
                merchantable volume for full and enhanced trees.

siteAge   data.table, Cluster-level summaries of age and height. This table is an output
                from siteAgeSummary

treeLossFactors
                data.table, The tree loss factor data, an output of VRIInit_lossFactor. In this
                funtion, this table provides loss indicator.

equation   character, Specifies whether the compiler is based on KFIZ or KBEC. Default is
                set as KBEC.

Value

A compiled volume after removing decay, waste and breakage; a log file

Author(s)

Yong Luo

DWBGenerator_BEC   *Collect decay, waste and breakage factor in BEC routine*

Description

This function is to collect the dead, waste and breakage factor from lookup table and join them into
tree data. Instead of reading the lookup table from disk, the function uses hard-coded the lookup
table. This function is equivalent to dwb_v3.sas macro. For FIZ routine, the decay, waste and
breakage are collected using funtion DWBGenerator_FIZ

Usage

```
DWBGenerator_BEC(DBH, height, species, meanAge, BEC, riskGroup, adjustID)

## S4 method for signature
## 'numeric,numeric,character,numeric,character,character,character'
DWBGenerator_BEC(DBH, height, species, meanAge, BEC, riskGroup, adjustID)
```

Arguments

DBH	numeric, Tree DBH.
height	numeric, Tree height.
species	character, Tree basic species code, which is SP0 in VRI original data.
meanAge	numeric, Mean site age.
BEC	character, BC BEC zone.
riskGroup	character, Specifies the risk group. It must be one of 1, 2 or 3. It can be derived from `riskGroupDeriver`.
adjustID	character, Adjustment identifier. Blank is no adjustment; QCI is queen charlottes; WET is wetbelt and GLD_NW golden ?.

Value

A list of `decay`, `waste` and `breakage` percentage.

Author(s)

Yong Luo

DWBGenerator_FIZ   *Collect decay, waste and breakage factor in FIZ routine*

Description

This function is to collect the dead, waste and breakage factor from lookup table and join them into tree data. Instead of reading the lookup table from disk, the function uses hard-coded the lookup table. This function is equivalent to second part of dwb_fct.sas macro. For BEC routine, the decay, waste and breakage are collected using funtion `DWBGenerator_BEC`

Usage

```
DWBGenerator_FIZ(DBHClass, tabNumber, riskGroup)

## S4 method for signature 'numeric,character,character'
DWBGenerator_FIZ(DBHClass, tabNumber, riskGroup)
```

Arguments

DBHClass	numeric, Tree DBH class. Currently, The function is capatable to the DBH class with 5 cm interval. It can be derived using `DBHClassifier`.
tabNumber	character, This character consist of 4 numbers. The first two number is the species number, and the last is the series number.
riskGroup	character, Specifies the risk group. It must be one of 1, 2 or 3. It can be derived from `riskGroupDeriver`.

Value

A list of `decay`, `waste` and `breakage` percentage.

Author(s)

Yong Luo

---

getDWBSeries   *Get local DWB series*

Description

Join local DWB (old/imperical factors) by species, PSYUB, ageRangeClass and FIZ. This function is the first part of dwb_fct.sas.

Usage

```
getDWBSeries(species, ageRangeClass, PSYUB, FIZ, source)

## S4 method for signature 'character,character,character,character,character'
getDWBSeries(species, ageRangeClass, PSYUB, FIZ, source)

## S4 method for signature 'character,character,character,missing,character'
getDWBSeries(species, ageRangeClass, PSYUB, source)

## S4 method for signature 'character,character,missing,character,character'
getDWBSeries(species, ageRangeClass, FIZ, source)
```

Arguments

species	character, Tree basic species code, which is SP0 in VRI original data.
ageRangeClass	character, The classified age range into 1 to 4. Output from [ageRangeClassifier](#).
PSYUB	character, Unique PSYU+PSYU_BLK code.
FIZ	character, BC forest inventory zone.
source	character, Series source, must one of local, zonal and reversingZonal. local is based on species, PSYUB and ageRangeClass; zonal is based on FIZ, species and ageRangeClass; and reversingZonal is based on reversing FIZ zones, i.e., changing interior to coastal and changing coastal to interior.

Value

DWB series, a two number character.

Author(s)

Yong Luo

---

grossVolCal_kozak   *Calcualte gross volume using kozak equations*

Description

This function use kozak BEC and species-specific taper equation to calculate tree volume. This functions also assigns the volume multiplier adjustment.

Usage

```
grossVolCal_kozak(
  compilationType,
  fullDimTreeData,
  logMinLength,
  stumpHeight,
  breastHeight,
  UTOPDIB
)
```

Arguments

compilationType

> character, either PSP or nonPSP. If it is PSP, it is consistent with original PSP compiler, otherwise, it is consistent with VRI compiler.

fullDimTreeData

> data.table, This is full dimension tree data, i.e., it has both DBH and height.

logMinLength   numeric, Specifies a minimum length for a log.

stumpHeight   numeric, Specifies stump height. If missing, 0.3 m will be used.

breastHeight   numeric, Specifies breast height. 1.3 m will be used when this arguement is missing.

UTOPDIB   numeric, Specifies minimum merchantable inside bark diameter. 10 cm is used as a default.

Value

A data table

Author(s)

Yong Luo

GYS_oracle_org1   *title GYS_oracle_org1*

Description

This function is to merge oracle and ascii data before the GYS compiler, the function below is modified from ld_gysfx_pt1_rene_10jun2017.sas The major changes from the original codes are 1. removed spatial attributes in GYS and updated from the most recent map 2. clearified the processes in sample_id, plot and tree_no order, as the original sas version creates massive tables, and confuses the compilation. 3. removed some pre-compiled results as inputs

Usage

```
GYS_oracle_org1(oracleSourcePath, outputPath)
```

Arguments

oracleSourcePath

        character, Specifies the path that stores data from oracle data base. In VRI compiler, this should be the savePath for `loadVGIS`.

outputPath   character, Specifies the path to save your outputs. If missing, the current working directory will be choosed.

Value

no item returned

heightEstimate_byHeightModel
                    Estimate height using height-DBH curves

Description

Estimate height using height-DBH curves by bec subzone and species

Usage

```
heightEstimate_byHeightModel(beczone, subzone, species, DBH, heightModels)
```

Arguments

beczone   character, BEC zone.

subzone   character, BEC subzone.

species   character, Species.

DBH   numeric, Diameter at breast height.

heightModels   data.table, Specifies the best model and coefficients by each BEC subzone and species.

Value

projected total height

Author(s)

Yong Luo

---

heightEstimate_mixedEffect

Estimate height using mixed effect height-DBH curves

Description

Estimate height using height-DBH curves by bec and species

Usage

```
heightEstimate_mixedEffect(BEC, siteID, sp0, DBH, fixedEffects, randomEffects)
```

Arguments

BEC	character, BEC zone.
siteID	character, site identifier
DBH	numeric, Diameter at breast height (cm).
fixedEffects	data.table, Fixed effect of the best model by each BEC and sp0.
randomEffects	data.table, Random effects of the best model by each BEC, sp0 and site identifier.
species	character, Species.

Value

projected total height

Author(s)

Yong Luo

heightEstimate_mixedEffect_nlme

Estimate height using mixed effect height-DBH curves using nlme

Description

Estimate height using height-DBH curves by species

Usage

```
heightEstimate_mixedEffect_nlme(
  siteID,
  unitreeid,
  species,
  DBH,
  fixedEffects,
  randomEffects_site,
  randomEffects_tree
)
```

Arguments

siteID	character, site identifier
species	character, Species.
DBH	numeric, Diameter at breast height (cm).
fixedEffects	data.table, Fixed effect of the best model by each sp0.
randomEffects_site	
	data.table, Random effects of the best model at site level.
randomEffects_tree	
	data.table, Random effects of the best model at tree level.

Value

projected total height

Author(s)

Yong Luo

---

heightSmry_byC   *Summarize mean and lorry's height by cluster-VRI specific*

Description

Summarizes mean and lorry's height by cluster for standing trees, standing + live trees, and standing + live + non-broken top trees. The function is improved version to calculate mean height in `vol_ha_2017.sas` by outputing lorey's height. For both fixed and variable area plots, the function computes mean height by using plot weight (`PLOT_WT`) weighted height. For lorey's height computation, the function treats variable and fixed area plots differently. Specifically, the function uses the mean height as lorey's height for variable plots, while uses height that weighted both by plot weight (`PLOT_WT`) and basal area (`BA_TREE`) for fixed area plots.

Usage

```
heightSmry_byC(treeMC)

## S4 method for signature 'data.table'
heightSmry_byC(treeMC)
```

Arguments

treeMC   data.table, Compiled tree-level data that contains both measured trees and counted trees.

Value

A table contains computed mean height `MN` or `MEAN` and lorey's height `LRY` for all standing trees `ALL`, standing and live trees 1 and standing and non-broken top trees 2. The output is equevalent to `height` table in original compiler.

Author(s)

Yong Luo

ISMCCompiler   *ISMC compiler*

Description

This compiler is a general compiler to compile field data from either PSP or other programs.

Usage

```
ISMCCompiler(
  compilationType,
  ismcUserName,
  ismcPassword,
  oracleEnv = "PROD",
  bcgwUserName,
```

```
    bcgwPassword,
    compilationPath,
    syncTo = as.character(NA),
    equation = "KBEC",
    walkThru = TRUE,
    logMinLength = 0.1,
    stumpHeight = 0.3,
    breastHeight = 1.3,
    UTOPDIB = 10,
    utilLevel = 4,
    weirdUtil = "4",
    download = TRUE,
    saveCSV = TRUE,
    recompile = FALSE,
    archiveDate = as.character(NA)
)
```

Arguments

compilationType

character, either PSP or nonPSP. If it is PSP, it is consistent with original PSP compiler, otherwise, it is consistent with VRI compiler.

ismcUserName   character, User name to access to ISMC database.

ismcPassword   character, Password to access to ISMC database.

oracleEnv   character, Specify which environment of ISMC database the data download from. Currently, it supports 1) INT for intergration environment; 2) TST for test environment; 3) PROD for final production environment.

bcgwUserName   character, User name to access to bcgw database.

bcgwPassword   character, Password to access to bcgw database.

compilationPath

character, Specifies the path that stores all the data/processes. By specifying this, four folders will be created to record all the data/processes. Specifically, raw_from_oracle stores the data just after oracle and ascii without editing; compilation_sa stores key data (not all) that after editing and before volume and age compilation; compilation_db stores compiled results for volume and age compilation at both tree level and cluater level; Archive_YYYYMMDD achives all the data mentioned above for the future use or reference.

syncTo   character, Specifies the path, i.e., network drive, that user wants to share the compilation outputs with coworkers.

equation   character, Specifies the taper equation that is used for compiler. Currently supports BEC-based (KBEC) and FIZ-based (KFIZ).

walkThru   logical, Speciefies whether the data had been collected using work through method. Default is TRUE, if it is not specified.

logMinLength   numeric, Specifies minimum length of log when doing log length adjustment, see logMatrixAdjustment for details. If missing 0.1 is used.

stumpHeight   numeric, Stump height. If missing 0.3 is used.

breastHeight   numeric, Breast height. If missing 1.3 is used.

UTOPDIB   numeric, Threshold inside-bark diameter for merchantable volume. If missing, UTOPDIB is 10.

utilLevel	numeric, Specifies utilization level in summrizing tree volumes at cluster and species level. Default is 4.
weirdUtil	character, Specifies weird utilization in summarizing tree volumes at cluster and species level. Default is no, if missing. Otherwise, a number should be provided.
download	logical, Specifies if the data from ISMC need to be downloaded. Default is TRUE, which m means need download.
saveCSV	logical, Specifies if the outputs need to be saved into CSV. Default is save.
recompile	logical, Defines whether we want to recompile data using archived raw data. Default is FALSE, which means the compiler needs to download data from ISMC database. When it is TRUE, a folder will be created in format of Archive_YYYYMMDD(archiveDate)_date) to save the all the compilation.
archiveDate	character, Defines on which archive date the raw data were downloaded. These raw data will be used for recompilation. Format is YYYYMMDD.

Value

This function compiles data and save outputs in `compilationPaths$compilation_db` and no file is returned.

Note

Improvements include:

1. specifies trees in B plots as height enhanced trees
2. removes the sas-dependent sindex functions
3. introduce the SIndexR package

Currently, the compiler supports compilation for the below sample types:

- Q: Regular VRI sample with five point clusters design
- T: ??, sample protocol and sample design are same as Q samples
- B: ??, same plot layout as Q sample, with height is measured for all trees and no call grading information
- M: CMI sample, with all trees have call grading information in the field
- L: LiDAR project, same plot layout and same design but without call grading information
- Y: YSM plots, population between 15 and 50 years
- F: NFI plots, trees measured all DBH, height and call grading
- N: NVAF plots
- A: VRI audit plots

Author(s)

Yong Luo

References

VRI compiler manual

ISMC_VGISTranslator   *Translate ismc data formats into vgis formats*

Description

This function is to translate ismc data formats into vgis formats

Usage

```
ISMC_VGISTranslator(inputPath, outputPath, coeffPath)
```

Arguments

inputPath	character, Specifies the path that stores data from oracle data base.
outputPath	character, Specifies the path to save your outputs. If missing, the current working directory will be choose.
coeffPath	character, Specifies the path where the crosswalk table is stored.

Value

no item returned

Author(s)

Yong Luo

loadASCII   *Load the data from ASCII files*

Description

This function is to read the data from ASCII files (in txt format).

Usage

```
loadASCII(txtLocation, saveThem = FALSE, savePath)
```

Arguments

txtLocation	character, Specifies the location of ASCII files.
saveThem	logical, Specifies whether the loaded data should be saved or returned. The default value is FALSE, which means the function will not save files for you.
savePath	character, Specifies the path that directs to the VRI original data soruce, i.e., //Mayhem/GIS_TIB/RDW/RDW_Data2/Work_Areas/VRI_ASCII_PROD/vri_sa.

Value

no files

Author(s)

Yong Luo

Examples

```
## Not run:
loadASCII(txtLocation = "Q:/RDW_Data2/Work_Areas/VRI_ASCII_PROD/vri_raw/to_be_added_next_compilation",
          saveThem = TRUE,
          savePath = "D:/vritest/forascii")


## End(Not run)
```

---

logAdjustment   *Adjust log length - VRI specific*

Description

The function is to adjust the log length to fit actual height. This function is equivalent to `log_adj_new` macro in original VRI compiler

Usage

```
logAdjustment(treeData, stumpHeight)

## S4 method for signature 'data.table,numeric'
logAdjustment(treeData, stumpHeight)

## S4 method for signature 'data.table,missing'
logAdjustment(treeData)
```

Arguments

treeData   data.table, Must have tree data information. The table is an output of `VRIInit_measuredTree`.

stumpHeight   numeric, Length of stump. As default, this arguement is set as 0.3 m.

Value

Data table that contains the adjusted log length

Note

Please see Bob for details about input files

Author(s)

Yong Luo

---

logAttributesLongForm   *Transpose wide form table to long form table-VRI specific*

Description

This function transposes wide form outputs to long form outputs. This function is not included in the VRI compiler anymore.

Usage

```
logAttributesLongForm(treeData, maximumLogNO)

## S4 method for signature 'data.table,numeric'
logAttributesLongForm(treeData, maximumLogNO)

## S4 method for signature 'data.table,missing'
logAttributesLongForm(treeData, maximumLogNO)
```

Arguments

treeData   data.table, an output from `logValueCalculator` function, i.e., tree_ms6. This
                table currently has top diameter (LOG_D_X), length (LOG_L_X), volume (LOG_V_X),
                merchantable volume (LOG_VM_X), grade (LOG_G_X), sound percentage (LOG_S_X)
                and value (LOG_c_x). X is log number from 1 to maximum log number.

maximumLogNO   numeric, determine the maximum number of logs. In VRI compiler, it is 9.
                Therefore, 9 is default.

Value

A data table and a log file

Author(s)

Yong Luo

---

logFileProducer   *To display a table to log file*

Description

Simple tool to display a table to log file.

Usage

```
logFileProducer(
  reason,
  action,
  displayTable,
  displayColumn,
  changedVariable,
  fromTo
)

## S4 method for signature
## 'character,character,data.table,character,character,character'
logFileProducer(
  reason,
  action,
  displayTable,
  displayColumn,
  changedVariable,
  fromTo
)

## S4 method for signature
## 'character,character,data.table,missing,character,character'
logFileProducer(reason, action, displayTable, changedVariable, fromTo)

## S4 method for signature
## 'character,character,data.table,character,missing,missing'
logFileProducer(reason, action, displayTable, displayColumn)

## S4 method for signature
## 'character,character,data.table,missing,missing,missing'
logFileProducer(reason, action, displayTable)
```

Arguments

`reason`	character, Reason to trigger an action.
`action`	character, Specifies action from one of `removed`, `no` and `changed`.
`displayTable`	data.table, A table of interest
`displayColumn`	character, Specifies which column(s) will be displayed in the log file.
`changedVariable`	charcater, Specifies the variable that has been modified, must be present if action is changed.
`fromTo`	character, This is two vectors character. Specifies columns that before and after alteration. Must be present when action is set as changed.

Value

A tring of text

Author(s)

Yong Luo

---

logMatrixAdjustment   *Adjust log length matrix*

Description

This function is to adjust log length matrix based on tree height, minimum log length and default log length. This function is equivalent to vol_tree_log_validation macro in original sas compiler.

Usage

```
logMatrixAdjustment(
  logLengthMatrix,
  height,
  stumpHeight,
  logMinLength,
  logDefaultLength
)

## S4 method for signature 'data.table,numeric,numeric,numeric,numeric'
logMatrixAdjustment(
  logLengthMatrix,
  height,
  stumpHeight,
  logMinLength,
  logDefaultLength
)

## S4 method for signature 'data.table,numeric,missing,numeric,numeric'
logMatrixAdjustment(logLengthMatrix, height, logMinLength, logDefaultLength)

## S4 method for signature 'data.table,numeric,numeric,missing,numeric'
logMatrixAdjustment(logLengthMatrix, height, stumpHeight, logDefaultLength)

## S4 method for signature 'data.table,numeric,numeric,numeric,missing'
logMatrixAdjustment(logLengthMatrix, height, stumpHeight, logMinLength)

## S4 method for signature 'data.table,numeric,missing,missing,numeric'
logMatrixAdjustment(logLengthMatrix, height, logDefaultLength)

## S4 method for signature 'data.table,numeric,missing,numeric,missing'
logMatrixAdjustment(logLengthMatrix, height, logMinLength)

## S4 method for signature 'data.table,numeric,missing,missing,missing'
logMatrixAdjustment(logLengthMatrix, height)
```

Arguments

logLengthMatrix

> data.table, A matrix of log length for each tree, NA is accepted in the matrix. The order of log from bottom to top must be presented from left to right in the table

`height`	numeric, Tree height
`stumpHeight`	numeric, Stump height. If missing, 0.3 m is used.
`logMinLength`	numeric, Minimum log length. If missing, 3 m is used.
`logDefaultLength`	
	numeric, Default log length. If missing, 5 m is used.

Value

A data.table that contains the matrix of adjusted log length

Author(s)

Yong Luo

`lookupCheck`	*Test whether the lookup table is updated*

Description

Reports whether a lookup table is updated. This function is highly recommended before proceed the VRI compiler.

Usage

```
lookupCheck(lookupName, lookupPath)

## S4 method for signature 'character,character'
lookupCheck(lookupName, lookupPath)

## S4 method for signature 'character,missing'
lookupCheck(lookupName)
```

Arguments

`lookupName`	character, Specifies the name of lookup table.
`lookupPath`	character, Path that directs to lookup tables.

Value

Not value returned. A warning message is given if the lookup table is changed.

Author(s)

Yong Luo

---

measNumCorrect_PSP   *Measurement number correction for raw PSP data*

Description

This function is to correct the measurement number for the remeasured PSP data. The remeasured PSP data is defined as PSP that sits outside of GYS oracle database. In those remeasurement data, some measurement number has not been correctly assigned, and conflicts the measurement number for the same samples in GYS oracle database.

Usage

```
measNumCorrect_PSP(sampleID, measureNumber)
```

Arguments

sampleID	character, Specifies how the PSP sample ID.
measureNumber	numeric, Specifies measurement number before correction.

Value

corrected measurement number

Note

with confirmation by Rene

Author(s)

Yong Luo

---

mergeAllVolTrees   *Merge all volume trees-VRI specific*

Description

Merge all the volume trees, including full trees (fully-measure trees in IPC), enhanced trees (fully-measured trees in auxi plots), H-enhanced trees (Height measured in auxi plots) and non-enhanced trees (only DBH measured in auxi plots). The function is part of `vol_ha_2017.sas` and modified dramatically in R compiler.

Usage

```
mergeAllVolTrees(treeMS, treeAX)

## S4 method for signature 'data.table,data.table'
mergeAllVolTrees(treeMS, treeAX)
```

Arguments

treeMS	data.table, Compiled full, enhanced and H-enhanced trees. This data should be listed in vi_c table. This data is an output of DWBCompiler
treeAX	data.table, Non-enhanced trees in anxilirary plots (vi_i). Supposedly, the table only contains non-enhanced tree list. However, some enhanced and H-enhanced trees also been stored in this dataset. An output from VRIInit_auxTree.

Value

A data table that contains all volume trees without duplicates. Equivalent to tree_vb table.

Author(s)

Yong Luo

---

mergeOAData   *Merge oracle and ascii data for vi_(a, ..i)*

Description

This function is to merge oracle and ascii data before the VRI compiler

Usage

```
mergeOAData(
  oracleSourcePath,
  asciiSourcePath,
  coeffPath,
  fizmapPath,
  fizmapName,
  fizmapFormat,
  outputPath
)
```

Arguments

oracleSourcePath	
	character, Specifies the path that stores data from oracle data base. In VRI compiler, this should be the savePath for loadVGIS.
asciiSourcePath	
	character, Specifies the path that stores data from ascii data base. In VRI compiler, this should be the savePath for loadASCII.
coeffPath	character, Specifies the path that stores coefficients and spatial lookup tables.
fizmapPath	character, Specifies the path to forest inventory zone map. By default, it is set to //spatialfiles2.bcgov/work/for/vic/hts/dam/workarea/data/infrastructure, which is maintained by FAIB employee.
fizmapName	character, Specifies the name of forest inventory zone map. By default, it is set to FIZ_REG_COMPARTMENT, which is maintained by FAIB employee.

fizmapFormat	character, Specifies the format of forest inventory zone map. Currently, it can be specified as gdb for geodatabase format and shp for shapefile format. By default, it is set to gdb, which is maintained by FAIB employee.
outputPath	character, Specifies the path to save your outputs. If missing, the current working directory will be choosed.

Value

no item returned

Author(s)

Yong Luo

---

netVolumeCalculator   *Calcualtes total net volume and merchantable volume-VRI specific*

Description

This function calculates total net volume and net merchantable volume for each tree based on ground called sound percentage. From the second column to the last column, the grossVolMatrix table should have same dimensions (i.e., number of rows and columns) of netFactorMatrix and grossMerchVolMatrix if they are provided. Furthermore, be aware of the correspondingness among the matrix. This function is part of log_valu_2017.sas.

Usage

```
netVolumeCalculator(grossVolMatrix, grossMerchVolMatrix, netFactorMatrix)

## S4 method for signature 'data.table,data.table,data.table'
netVolumeCalculator(grossVolMatrix, grossMerchVolMatrix, netFactorMatrix)

## S4 method for signature 'missing,data.table,data.table'
netVolumeCalculator(grossMerchVolMatrix, netFactorMatrix)

## S4 method for signature 'data.table,missing,data.table'
netVolumeCalculator(grossVolMatrix, netFactorMatrix)

## S4 method for signature 'data.table,data.table,missing'
netVolumeCalculator(grossVolMatrix, grossMerchVolMatrix)
```

Arguments

grossVolMatrix	data.table, Calculated gross volume for each log. The first column of this table is the volume for the stump. If missing, the function calculates the total net merchantable volume.
grossMerchVolMatrix	
	data.table, Calculated gross merchantable volume for each log. If missing, all the merchantable volume is assigned as 0.
netFactorMatrix	
	data.table, Ground call for sound percentage. If missing, the net factoring will be assigned as 100.

Value

Data table that contains total net volume (`VOL_NET`) and total net merchantable volume (`VOL_NETM`) for each tree.

Author(s)

Yong Luo

---

pathIndicatorGenerator

Generate path indicator string for risk group

Description

This function to generate a length of eight character path indicator string. The path indicator string is an input for rish group for Jim's decay, waste and breakage function. The function varies with PSP compiler and VRI compiler.

Usage

```
pathIndicatorGenerator(
  lossIndicatorMatix,
  lossIndicatorLocMatrix,
  merchantableHeight,
  compiler
)
```

Arguments

lossIndicatorLocMatrix
:   data.table, Contains the location for each corresponding loss indicator in `lossIndicatorMatrix`. This matrix will be used for VRI compiler.

merchantableHeight
:   numeric, Specifies the maximum height for merchantable volume. This arguement will be used for VRI compiler.

compiler
:   character, Specifies in which compiler the path indicator will be generated. It can be either `PSP` or `VRI`.

lossIndicatorMatrix
:   data.table, Contains eight columns of loss indicator, i.e., LOSS1...8_IN.

Value

A length of eight character that contains 0 and 1, e.g., 10010000.

Note

For PSP compiler, the path indicator is only based on loss indicator. However, for VRI compiler, the path indicator also based on loss indicator location and merchantable height.

Author(s)

Yong Luo

plotDynamics	*Summarize the dynamic table between two measurements for repeatedly measured samples*

Description

This function takes tree-level component change data and summarizes the data to a plot-level dynamics.

Usage

```
plotDynamics(treelist, samples)
```

Arguments

treelist	data.table, Tree measurements with component change, which is the output from `assignChangeComponent`.
samples	data.table, Site visit information including SITE_IDENTIFIER and VISIT_NUMBER.

Value

return a summary table

Author(s)

Yong Luo

preparePublishData	*Prepare the compiled data for publish*

Description

To prepare the compiled data for publish, currently supporting nonPSP part

Usage

```
preparePublishData(
  compilationPath,
  compilationDate,
  publishPath,
  compilationType
)
```

Arguments

compilationPath

character, The path to the compiled data, which is configured from ISMCCompiler.

compilationDate

numeric, The date of the compiled data.

publishPath   character, The path to save prepared data.

compilationType

character, Specifies the compilation type either nonPSP or codePSP.

Value

no value returned. Instead, all the files will be saved into the `tempPath` including a readme file.

Note

The compilationPath must have all the outputs from `ISMCCompiler`.

Author(s)

Yong Luo

---

prepareTASSInputs   *To prepare input data for TASS run*

Description

This function takes the compilation outputs and prepare stand age, loss factor, site index and stem mapping, so that the outputs are ready for TASS run

Usage

```
prepareTASSInputs(
  inputPath,
  outputPath,
  projectName,
  clstrIDs = NA,
  siteIndexTable = NA,
  siteIndexTableSource = "ISMCCompiler",
  siteIndexMethod,
  treeVigorMethod,
  vigorAdjust08 = TRUE,
  randomSeed = NA
)
```

Arguments

inputPath	character, The path to the compilation output.
outputPath	character, The path to save the outputs.
projectName	character, The name of a project, currently support YSM, Taan or `special`.
clstrIDs	character, A list of clsterid in a special project. It will be ignored if the `projectName` is YSM or Taan. As they are rule-based selection of clster id.
siteIndexTable	data.table, The table to contain site index.
siteIndexTableSource	
	character, Indicates where the site index table from either Rene or `ISMCCompiler`. If the source is ISMCCompiler, the function will ignore the site index table provided. Therefore, it can be missing. Default is `ISMCCompiler`.

siteIndexMethod

                character, The method to derive site index for repeatedly-visited sites. There are three methods can be used: `byvisit`, `average`, `firstvisit` or `closest50`. `byvisit` uses the site index for each visit. `average` takes the mean site index for a given site and species over multiple visits. `firstvisit` uses the site index from the first visit. `closest50` uses the site index which has the closest to the stand age of 50 years.

treeVigorMethod

                character, Method to derive a tree's vigor, currently support `mainsub`.

vigorAdjust08   logical, Indicates if the mean of mean of the tree height vigor needed to be adjusted to 0.8. Default is `TRUE`.

randomSeed   numeric, The random seeds for the stem mapping extension. Default is `NA`, which does not have a seed number.

Note

mainsub in treeVigorMethod: • For each combination of site_identifier by visit_number by species, select the 6 tallest trees from the main plot and the single tallest tree from the subplot. • From this subset, drop all trees with o BROKEN_TOP_IND = Y o CR_CL = I, S o RESIDUAL = Y o WALK-THRU = O • Compute site_height as the average height of remaining trees • If no trees remain, then compute an alternate site_height as the average height of all trees in the plot by site_identifier * visit_number * species (ie., no exclusion for height, broken top, crown class, residual class, or walkthru class.

Author(s)

Yong Luo

---

prj_ID2BEC   *Assign BEC based on project ID*

Description

This function takes lookup table that connects project id to BEC zone and joins BEC zone by project id. The function uses a hardcoded lookup table `vri_bec`. The function is equivalent to `group_bec.sas`.

Usage

```
prj_ID2BEC(projectID)

## S4 method for signature 'character'
prj_ID2BEC(projectID)
```

Arguments

projectID   character, Specifies project ID.

Value

BEC, `Unknown` will be return if project id does have any match in lookup table.

Author(s)

Yong Luo

---

prj_ID2Grp   *Group project ID into project group*

Description

This function takes lookup table that connect project id to project group and joins project group
by project id. The function uses hardcoded lookup table `vri_grp`. The function is equivalent to
`group_prj.sas`.

Usage

```
prj_ID2Grp(projectID)

## S4 method for signature 'character'
prj_ID2Grp(projectID)
```

Arguments

projectID   character, Specifies project ID.

Value

Project group, `Unknown` will be return if project id does have any match in lookup table.

Author(s)

Yong Luo

PSPCompilation_Init   *Initiate PSP data in ISMC for the compilation*

Description

This function is to organize ismc PSP data for the sequential compilation.

Usage

```
PSPCompilation_Init(inputPath, outputPath)
```

Arguments

inputPath   character, Specifies the path that stores data from oracle data base.

outputPath   character, Specifies the path to save your outputs. If missing, the current working
                 directory will be choose.

Value

no item returned

Author(s)

Yong Luo

---

pspHT   *Fit height for PSP nonHT data using different method*

Description

This function is to fit height for the nonHT trees in PSPs.

Usage

```
pspHT(treeData, method = "bestMEM", coeffs)
```

Arguments

method	character, Method to derive height, currently it supports `bestHeightModel`.
coeffs	data.table, Coefficient table.
treesData	data.table, Contains both height trees and nonHT trees in PSPs.

Value

Full list of trees with derived height.

Author(s)

Yong Luo

PSPInit_clusterplot   *Load and select cluster and plot level data for PSP compilation*

Description

This function prepares the cluster/plot-level inputs for PSP compiler. Specifically, it standardizes names for the variables; reports and removes the duplicate observations at cluster, cluster/plot.

Usage

```
PSPInit_clusterplot(dataSourcePath)
```

Arguments

dataSourcePath character, Specifies the path that directs to the VRI original data soruce, i.e.,
                `//Mayhem/GIS_TIB/RDW/RDW_Data2/Work_Areas/VRI_ASCII_PROD/vri_sa`.

Value

A data table that contains key information at cluster/plot level and compiler log file.

Author(s)

Yong Luo

PSPInit_lossFactor   *Load and select trees that have loss factor information-VRI specific*

Description

This function loads and selects trees that have loss factor information (vi_d, cardd) based on selected trees from vi_c.

Usage

```
PSPInit_lossFactor(fullMeasuredTrees, dataSourcePath)
```

Arguments

fullMeasuredTrees

                data.table, Selected trees in vi_c, which includes full, enhanced and H-enhanced trees. An output of VRIInit_measuredTree.

dataSourcePath character, Specifies the path that directs to the VRI original data soruce, i.e., //Mayhem/GIS_TIB/RDW/RDW_Data2/Work_Areas/VRI_ASCII_PROD/vri_sa.

Value

A data table that contains loss factor data. A log file documents the detailed process

Author(s)

Yong Luo

PSPInit_measuredTree   *Load and select fully measured tree data for PSP compilation*

Description

This function selects the tree-level data from vi_c (cardc) based on selected cluster/plot headers. Additonally, the function calculates basal area and tree per ha factor.

Usage

```
PSPInit_measuredTree(clusterplotHeader, dataSourcePath)
```

Arguments

clusterplotHeader

> data.table, Cluster and plot-level attributes, an output from VRIInit_clusterplot.

dataSourcePath   character, Specifies the path that directs to the VRI original data soruce, i.e.,
//Mayhem/GIS_TIB/RDW/RDW_Data2/Work_Areas/VRI_ASCII_PROD/vri_sa.

Value

A data table that contains tree-level information. A log file that describes the detailed process.

Author(s)

Yong Luo

PSPInit_nonHTTree   *Load nonHT trees for PSP compilation*

Description

This function loads nonHT tree based on cluster/plot header.

Usage

```
PSPInit_nonHTTree(clusterplotHeader, dataSourcePath)
```

Arguments

clusterplotHeader

> data.table, Cluster and plot level attributes, an output from VRIInit_clusterplot.

dataSourcePath   character, Specifies the path that directs to the VRI original data soruce, i.e.,
//Mayhem/GIS_TIB/RDW/RDW_Data2/Work_Areas/VRI_ASCII_PROD/vri_sa.

Value

A data table that contains auxiliary plot tree data.

Author(s)

Yong Luo

PSPInit_siteTree   *Load and select site trees-VRI specific*

Description

This function connects site tree data (vi_h, cardh) to selected cluster/plot-level data. Site tree data is located in `//Mayhem/GIS_TIB/RDW/RDW_Data2/Work_Areas/VRI_ASCII_PROD/vri_sa`

Usage

```
PSPInit_siteTree(clusterplotHeader, dataSourcePath)
```

Arguments

clusterplotHeader

data.table, contains cluster/plot-level attributes, an output from [VRIInit_clusterplot](#).

dataSourcePath   character, Specifies the path that directs to the VRI original data soruce, i.e.,
`//Mayhem/GIS_TIB/RDW/RDW_Data2/Work_Areas/VRI_ASCII_PROD/vri_sa`.

Value

A data table that contains site tree data information. A log file documents the detailed process

Note

VRI specific

Author(s)

Yong Luo

---

pspTreeCheck   *Check psp trees for the abnormal observations*

Description

This function is to check psp trees in terms of dramatic change of size, live dead code, species and missing observation.

Usage

```
pspTreeCheck(treeData)
```

Arguments

treesData   data.table, Contains both height trees and nonHT trees in PSPs.

Value

Full list of trees with flags.

Author(s)

Yong Luo

PSPVolTree   *Calcualte tree volume-PSP specific*

Description

This function use BEC(or FIZ) and species-specific taper equation to calculate tree volume. Before calculation, the function adjusts height for broken top trees: scenario 1 (D scenario): availability of DBH, DIB at broken height; scenario 2 (H scenario): availability of projected tree height in the field. This functions also assigns the volume multiplier adjustment.

Usage

```
PSPVolTree(
  treeData,
  equation,
  logMinLength,
  stumpHeight,
  breastHeight,
  UTOPDIB,
  HTEstimateMethod,
  htDBHCoeff
)
```

Arguments

treeData	data.table, Tree data
equation	character, Specifies which taper equation form will be used to calculate diameter inside bark for a given height. Must be either KBEC or KFIZ3. If missing, default is KBEC
stumpHeight	numeric, Specifies stump height. If missing, 0.3 m will be used.
breastHeight	numeric, Specifies breast height. 1.3 m will be used when this arguement is missing.
UTOPDIB	numeric, Specifies minimum merchantable inside bark diameter. 10 cm is used as a default.
bestHeightModels	
	data.table, External table that contains the best height/DBH model and coefficients by becsubzone and species.
HTBTOPModel	character, Specifies whether the height estimate for broken top trees either from `taper` or from `height`.

Value

A data table

Author(s)

Yong Luo

rankingMatrix	*This function is to calculate rating for faib sample sites and to rank them by rating values*

Description

To rank psp sample sites

Usage

```
rankingMatrix(archivedPSPPath, archivednonPSPPath = NULL, useOldCellKey = TRUE)
```

Arguments

archivedPSPPath

> character, The path to the compiled PSP data, which is configured and outputed from ISMCCompiler.

archivednonPSPPath

> character, The path to the compiled nonPSP data, which is configured and outputed from ISMCCompiler.

useOldCellKey logical, Indicates if using the old cell key from original sas ranking. Default is TRUE.

Value

A list of ranking tables including 1) Psp_netdown_summary: net down summary 2) Protected_psp_summary: a summary of protect psp sites by protect code 3) Remeasured_psp_summary: a summary of remeasured psp sites by last msmt year group 4) data_source: a description of data used for ranking 5) Ranking_psp: the ranking for psp sample sites 6) Ranking_matrix: all the raw data and intermediate data used for calculating the rating values

Note

As the ISMCCompiler could not produce the same cell key as the previously used. The function allows using a lookup table to populate the cell key to reach consistency. However, the cell key is further updated using the site status code from ISMC.

Author(s)

Xinjia(Bridget) Guo and Yong Luo

---

readHandHeld   *load json data from handheld and save tables into target folder*

Description

This function is to load JSON file data from FAIB new handheld.

Usage

```
readHandHeld(
  fileName,
  savePath = getwd(),
  saveFormat = "rdata",
  saveName = "myHandHeldFile",
  overWrite = FALSE
)
```

Arguments

fileName	character, JSON file name.
savePath	character, A folder path user wants to save all the outputs. Default is the current work directory.
saveFormat	character, Specifies the format user wants to save. Currently, the function supports xlsx and rdata. Default is rdata.
saveName	character, Specifies the save name.
overWrite	logical, Determine if the file with same name as user specifies will be overwritten. Default is FALSE.

Value

No value will be returned. All the outputs will be saved in a target folder.

Author(s)

Yong Luo

Examples

```
## Not run:
## given, in the current work directory, there is a ismc-file-transfer.json file
  icmcjsonFile <- "ismc-file-transfer.json"
## create a folder in current work directory and save the outputs into it
  dir.create("ismc_test_folder")

  readHandHeld(fileName = icmcjsonFile,
               savePath = file.path(getwd(), "ismc_test_folder"))

## End(Not run)
```

readHandHeld_mod *load json data from handheld and save tables into target folder*

Description

This function is to load JSON file data from FAIB new handheld.

Usage

```
readHandHeld_mod(
  fileName,
  savePath = getwd(),
  saveFormat = "rdata",
  saveName = "myHandHeldFile",
  overWrite = FALSE
)
```

Arguments

fileName	character, JSON file name.
savePath	character, A folder path user wants to save all the outputs. Default is the current work directory.
saveFormat	character, Specifies the format user wants to save. Currently, the function supports xlsx and rdata. Default is rdata.
saveName	character, Specifies the save name.
overWrite	logical, Determine if the file with same name as user specifies will be overwritten. Default is FALSE.

Value

No value will be returned. All the outputs will be saved in a target folder.

Author(s)

Yong Luo

Examples

```
## Not run:
## given, in the current work directory, there is a ismc-file-transfer.json file
   icmcjsonFile <- "ismc-file-transfer.json"
## create a folder in current work directory and save the outputs into it
   dir.create("ismc_test_folder")

   readHandHeld_mod(fileName = icmcjsonFile,
              savePath = file.path(getwd(), "ismc_test_folder"))

## End(Not run)
```

| regBA_WSV | *To fit basal area and whole stem volume regression using mixed effect model* |

Description

This function takes the selected data for fitting regression model between basal area and whole stem volume using mixed effect model. The models are fitted based on strata of BEC+SP0+LV_D

Usage

```
regBA_WSV(inputData, needCombs)
```

Arguments

inputData	data.table, The data for fitting regression.
needCombs	data.table, The combinations of BEC+SP0+LV_D are needed to derive coefficients.

Value

coefficient table

Author(s)

Yong Luo

| regRatioDataSelect | *Select the data for regression and ratio* |

Description

This function selects the data for fitting regression model between basal area and whole stem volume, and for ratios to whole stem volume in VRI compiler. The regreesion method is used for derive whole stem volume for the trees that just have DBH information. The ratio method is used to derive netted merchantable volume for the trees that do not have call grading information. The data selection should be done annually at the end of every March to allow newer and higher quaulity data enter the data.

Usage

```
regRatioDataSelect(sampledata, alltreedata, usage)
```

Arguments

sampledata	data.table, The data contains the sample level information.
alltreedata	data.table, The is the data pool which contains the full/enhanced trees and H-enhanced trees.
usage	character, Specifies compiler name. It supports ismc and vgis.

Value

Selected data for regression and ratio.

Author(s)

Yong Luo

---

riskGroupDeriver   *Derive risk group for standard sample compilation/data*

Description

will refine. This function is equivalent to risk_grp.sas for fiz-based process, risk_v3.sas for bec-based process.

Usage

```
riskGroupDeriver(species, pathIndex, series, height, method)

## S4 method for signature 'character,character,character,numeric,character'
riskGroupDeriver(species, pathIndex, series, height, method)

## S4 method for signature 'character,character,missing,missing,character'
riskGroupDeriver(species, pathIndex, series, height, method)
```

Arguments

species	character, Tree basic species code, which is SP0 in VRI original data.
pathIndex	character, A character with length of 8, consists of 0 or 1.
series	character, DWB series. It is a lenght of 2 number character and can be derived using getDWBSeries function.
height	numeric, Total tree height.
method	character, Specifies the method between FIZ and KBEC to categorize the risk group. The FIZ method derives risk group by species, pathIndex, series and height. KBEC method derives the risk group using species and pathIndex.

Value

Risk group, which is character

Author(s)

Yong Luo

---

samplePlotCompilation   *Compile sample and plot level information*

Description

This function is to compile sample and plot information.

Usage

```
samplePlotCompilation(compilationType, dataSourcePath, mapPath, coeffPath)
```

Arguments

compilationType

        character, either PSP or nonPSP. If it is PSP, it is consistent with original PSP
        compiler, otherwise, it is consistent with VRI compiler.

dataSourcePath   character, Specifies the path that contains prepared data from raw data.

mapPath   character, Specifies the path dependent maps are stored.

coeffPath   character, Specifies the path dependent coeffs are stored.

Value

A data table that contains key information at cluster/plot level and compiler log file.

Author(s)

Yong Luo

---

siteAgeCompiler   *Compile breast age, total age, and site index where possible-VRI spe-*
                      cific

Description

This function takes site age tree data ie., vi_h, an output of `VRIInit_siteTree` to compute the
breast height age, total age, and site index where possible. This function is equivalent to site_age.sas.
The function heavily depends on site tools program.

Usage

```
siteAgeCompiler(siteAgeData, compilationType)

## S4 method for signature 'data.table,character'
siteAgeCompiler(siteAgeData, compilationType)
```

Arguments

siteAgeData   data.table, Site age data with plot header information. An output from `VRIInit_siteTree`
        function.

compilationType

        character, Compilation type.

Value

A data table and a log file.

Author(s)

Yong Luo

---

siteAgeCorrection	*Bored age correction for multiple measurments based on measurement interval*

Description

Correct the bored age if the difference of ages between two measurements does not match the years of measurement interval

Usage

```
siteAgeCorrection(vih)
```

Arguments

vih   data.table, The table contains the site trees in the database.

Value

A site tree table that is equivalent to original vih, but with bored age corrected

Note

1. for the site trees with the age measurement code of PRE, the current age is corrected based on the previous bored age. 2) for the site trees with multiple drills, the age of last measurement is used as a reference point, and the previous ages are adjusted. 3) some ages of site trees are manuallly corrected due to complexity of measurements (see comments in the codes)

Author(s)

Yong Luo

---

siteAgeSummary   *Summarize site age data by cluster and cluster/species-VRI specific*

Description

This function takes compiled site age tree data, an output of `siteAgeCompiler`, to derive mean age and height results. The compiled data must have breast height age, total age, and site index. This function is equivalent to `mean_htl.sas`.

Usage

```
siteAgeSummary(cpldSiteAgeData)

## S4 method for signature 'data.table'
siteAgeSummary(cpldSiteAgeData)
```

Arguments

cpldSiteAgeData

                data.table, Compiled site age tree data, an output of `siteAgeCompiler`.

Value

Two data tables: cl_ah is the age/height summary at cluster level and spc_ah is the age/height summary at cluster and species level

Author(s)

Yong Luo

---

siteAgeSummary_PSP   *Summarize site age data by cluster and cluster/species for PSP data*

Description

This function takes compiled site age tree data, an output of `siteAgeCompiler`, to derive mean age and height results. The compiled data must have breast height age, total age, and site index.

Usage

```
siteAgeSummary_PSP(cpldSiteAgeData, treemsmt)
```

Arguments

cpldSiteAgeData

                data.table, Compiled site age tree data, an output of `siteAgeCompiler`.

treemsmt   data.table, Tree measurement data.

Value

Two data tables: cl_ah is the age/height summary at cluster level and spc_ah is the age/height summary at cluster and species level

Author(s)

Yong Luo

---

siteToolsSpeciesConvertor

Convert VRI species code to site tools species code

Description

This function converts BC VRI species code to site tools species code. The convertion is based on a hardcoded lookup table spv_frd.

Usage

```
siteToolsSpeciesConvertor(species)

## S4 method for signature 'character'
siteToolsSpeciesConvertor(species)
```

Arguments

species   character, Species code in VRI data sets.

Value

Converted species codes that can be recognized by site tools program. NA is the species that failed to be converted.

Author(s)

Yong Luo

SiteTools_HTBoredAge2SI

Calculate site index using site tools

Description

This function calculates site index based on bored age (boredAge), tree height (height), species (species) and region (ICRegion) using site tools program. This function is equivalent to sindex_httoage.sas.

Usage

```
SiteTools_HTBoredAge2SI(boredAge, height, species, ICRegion, ageType, estType)

## S4 method for signature
## 'numeric,numeric,character,character,numeric,numeric'
SiteTools_HTBoredAge2SI(boredAge, height, species, ICRegion, ageType, estType)
```

Arguments

boredAge	numeric, Age at bored height.
height	numeric, Total tree height.
species	character, Species code, must be consistent with the species code in site tools, which can be converted from the original species code by using siteToolsSpeciesConvertor.
ICRegion	character, Must be either I (interior) and C (coastal). IC regions can be derived using BEC2IC.
ageType	numeric, Must be either 0 or 1. 0 stands for total age, for which site index is calculated for 50 years of total tree age. While 1 stands for breast height age, for which site index is calculated for 50 year old at breast height.
estType	numeric, Defines how the site tools estimate site index. Valued as 0 and 1, 0 is interative and while 1 is directive. Default is 1, which is directive.

Value

Site index

Author(s)

Yong Luo

SiteTools_Y2BH   *Derive years to breast height using site tools*

Description

Derive years to breast height based on species (species), region (ICRegion) and site index (siteIndex) using site tools. This function is equivalent to sindex_httoage.sas.

Usage

```
SiteTools_Y2BH(species, ICRegion, siteIndex)

## S4 method for signature 'character,character,numeric'
SiteTools_Y2BH(species, ICRegion, siteIndex)
```

Arguments

species	character, Species code, must be consistent with the species code in site tools, which can be converted from the original species code by using siteToolsSpeciesConvertor.
ICRegion	character, Must be either I (interior) and C (coastal). IC regions can be derived using BEC2IC function.
siteIndex	numeric, Site index. Defined as tree height at 50 years old.

Value

Years to breast height

Author(s)

Yong Luo

---

smallTreeSmry   *Summarizes the tally for small trees - VRI specific*

Description

Calculates and summarizes tally of small trees at both cluster/species level and cluster. This function is equivalent to `sml_tree.sas` in original compiler.

Usage

```
smallTreeSmry(smallTreeData, smallTreePlotHeader)

## S4 method for signature 'data.table,data.table'
smallTreeSmry(smallTreeData, smallTreePlotHeader)
```

Arguments

smallTreeData   data.table, Small tree data. This data is from card f, i.e., `vi_f`.

smallTreePlotHeader

data.table, Plot header data for stump and small tree data. The data is from card e, i.e., `vi_e`.

Value

Two tables: stmp_c is summarized volume at cluster level; stmp_cs is summarized volume at cluster/species level.

Author(s)

Yong Luo

---

spAttriFromRegCompt   *get tsa, bec, fiz and ownership attributes based on region and compartment map*

Description

get tsa, bec, fiz and ownership attributes based on region and compartment map

Usage

```
spAttriFromRegCompt(regionCompartMap, spAttributeMap, mapName)
```

Arguments

regionCompartMap

    sf, regioncompartment map.

spAttributeMap   sf, spatial attribute map.

mapName   character, spatial attribute map name, must be one of BEC, TSA, FIZ, TFL, and Ownership.

Value

table of spatial attributes corresponding to region and compartment.

Note

The spatial attributes must be one-on-one match to a unique combination of region and compartment

Author(s)

Yong Luo

---

speciesCode2speciesType

         Group species into deciduous and coniferous species group

---

Description

This function is to group species into deciduous and coniferous species types based on BC species code and sp_type lookup table. This function uses hardcoded sp_type lookup table.

Usage

```
speciesCode2speciesType(species)

## S4 method for signature 'character'
speciesCode2speciesType(species)
```

Arguments

species   character, Tree basic species code, which is SPECIES in VRI original data.

Value

Species type: D-deciduous species and C-coniferous species. NA, with a warning message, is given if a species fails to be grouped.

Author(s)

Yong Luo

---

speciesComp_byC   *Calculates species composition based on cluster/species summary*

Description

Calculates species composition at cluster level based on cluster/species summary. The cluster/species-level summaries is an output of volSmry_byCS function. This function is equivalent to sp_comp.sas in original compiler.

Usage

```
speciesComp_byC(CSSmryTable, basedOn, speciesMaxNO, smallTreeCompile)

## S4 method for signature 'data.table,character,numeric,logical'
speciesComp_byC(CSSmryTable, basedOn, speciesMaxNO, smallTreeCompile)

## S4 method for signature 'data.table,character,numeric,missing'
speciesComp_byC(CSSmryTable, basedOn, speciesMaxNO)
```

Arguments

CSSmryTable   data.table, Summarized volume components for both measured and counted trees at cluster and species level. See volSmry_byCS for details.

basedOn   character, Specifies which component is used for species composition summary.

speciesMaxNO   numeric, Maximum number of species entries to calculate.

smallTreeCompile

     logical, Defines whether the function is used for calculate species composition for small trees. If missing, FALSE is used.

Value

A data table that contains species composition at cluster level.

Author(s)

Yong Luo

| speciesCorrection | *Species correction, based on Rene and Dan's correction routine. See emails on May 12th, 2021.* |

Description

This function is to correct species codes

Usage

```
speciesCorrection(species, BEC, BEC_subzone)
```

Arguments

species	character, Specifies the original species.
BEC	character, BEC zone.
BEC_subzone	character, BEC subzone.

Value

corrected species code

Author(s)

Yong Luo

| stumpVolSmry | *Summarizes the volume for stumps - VRI specific* |

Description

Calculates stump volume at cluster/species level and cluster. This function is equivalent to stmpvol.sas in original compiler.

Usage

```
stumpVolSmry(stumpData, stumpPlotHeader)

## S4 method for signature 'data.table,data.table'
stumpVolSmry(stumpData, stumpPlotHeader)
```

Arguments

stumpData	data.table, Stump data. This data is from card g, i.e., vi_g.
stumpPlotHeader	
	data.table, Plot header data for stump and small tree data. The data is from card e, i.e., vi_e.

Value

Two tables: stmp_c is summarized volume at cluster level; stmp_cs is summarized volume at cluster/species level.

Author(s)

Yong Luo

| toWSVRatio | *To toWSV ratio for each of strata* |

Description

This function takes the selected data for derive ratio for each of strata by BEC+SP0+LV_D

Usage

```
toWSVRatio(inputData, needCombs, minDBH = 10, minObs = 30)
```

Arguments

inputData	data.table, The data for deriving ratios, that must contain full and enhanced trees.
needCombs	data.table, The combinations of BEC+SP0+LV_D are needed to derive coefficients.
minDBH	numeric, The minimum DBH for selecting trees to derive ratios. If missing, 10 cm will be used.
minObs	numeric, The minimum samples size for a stratum. If missing, 30 observations will be used.

Value

ratio table

Author(s)

Yong Luo

---

toWSVRatio_curve   *To toWSV ratio for each of strata*

Description

This function takes the selected data for derive ratio for each of strata by BEC+SP0+LV_D

Usage

```
toWSVRatio_curve(inputData, needCombs, minDBH = 10, minObs = 100)
```

Arguments

inputData   data.table, The data for deriving ratios, that must contain full and enhanced trees.

needCombs   data.table, The combinations of BEC+SP0+LV_D are needed to derive coefficients.

minDBH   numeric, The minimum DBH for selecting trees to derive ratios. If missing, 10 cm will be used.

minObs   numeric, The minimum samples size for a stratum. If missing, 30 observations will be used.

Value

ratio table

Author(s)

Yong Luo

---

treemsmtEditing   *Edit tree measurements for a repeatedly measured data.*

Usage

```
treemsmtEditing(compilationType, treemsmts, sitevisits)
```

Arguments

compilationType

character, Compilation type, either nonPSP or PSP.

treemsmts   data.table, Tree measurements. The table must contain unique tree id, i.e., SITE_IDENTIFIER, PLOT, and TREE_NUMBER, and VISIT_NUMBER. And key measurements including live/dead ((TREE_EXTANT_CODE)), diameter information (DIAMETER and DIAMETER_MEASMT_HEIGHT), TREE_CLASS_CODE, BROKEN_TOP_IND, CROWN_CLASS_CODE, and TREE_SPECIES_CODE.}

\item{sitevisits}{data.table, Site visit information including SITE_IDENTIFIER and VISIT_NUMBER.} } { return a table after editing. } { This function takes tree-level measurements and edits live/dead codes, diameter, species, and add tree measurements if the measurements are missing. } {

1. For missing live/dead code, using the next live/dead code if it is present. Otherwise, assign dead (D). 2. In the case a tree is observed dead at a visit and change to live in the next visit, the dead code (D) will be changed to live (L). 3. If there is a missing measurement in between two visits. A measurement will be added with diameter assigned as mean of previous and next diameter and live/dead will be populated using next measurement. 4. If there is a missing measurement at tail, i.e., a tree was seen live for a given visit but without reaching the last visit for a given site. A measurement for next visit will be added with diameter populated using previous diameter and live/dead code will be assigned as D. 5. If species code changes during visits, the species code at last visit will be used to correct species code throughout all visits. 6. If at a tree's last visit, there is no diameter information and it is marked as live tree. The live/dead code will be changed to D. 7. If a tree is dead at a given visit and without diameter, and a tree was live at previous visit with diameter, the diameter at previous visit will be used to populate diameter at this visit. 8. If a tree is broken top tree for a given visit, this tree must be broken top tree since then. 9. If crown_class_code is missing at a visit and it was present at previous visit, the previous crown_class_code will be used to populate for this visit.

} { Yong Luo }

---

treeVolEst_RegRatio *Estimate volume for H-enhanced and non-enhanced trees-VRI specific*

Description

This function estimates the volumes for JH-enhanced and non-enhanced trees using BA-WSV equation and toWSV ratio methods. For H-enhanced trees, the whole stem volume and gross merchantable volume are already calculated directly using taper equations; and rest of volume components will be calculated using ratio method in this function. For non-enhanced trees, the whole stem volume is derived using regression equation between basal area and whole stem volume and the rest of volume components will be computed using ratio method in this function.

Usage

```
treeVolEst_RegRatio(nonVolTrees, fixedCoeffTable, randomCoeffTable, ratioTable)
```

Arguments

nonVolTrees   data.table, H-enhanced trees and non-enhanced trees.

fixedCoeffTable

data.table, Specifies the WSV-BA equations by project group PRJ_GRP, live and dead status LV_D, stand and falling status SF_COMPILE and species code SP0. The table can be generated using WSV_BARegression.

randomCoeffTable

data.table, Specifies the WSV-BA equations by project group PRJ_GRP, live and dead status LV_D, stand and falling status SF_COMPILE and species code SP0. The table can be generated using WSV_BARegression.

ratioTable   data.table, Specifies toWSV ratio by project group PRJ_GRP, live and dead status LV_D, stand and falling status SF_COMPILE and species code SP0. The table can be generated using toWSVRatio.

Value

A data table that has compiled non volume trees.

Author(s)

Yong Luo

---

treeVolEst_RegRatio_new

Estimate volume for H-enhanced and non-enhanced trees-VRI specific

---

Description

This function estimates the volumes for JH-enhanced and non-enhanced trees using `BA-WSV` equation and `toWSV` ratio methods. For H-enhanced trees, the whole stem volume and gross merchantable volume are already calculated directly using taper equations; and rest of volume components will be calculated using ratio method in this function. For non-enhanced trees, the whole stem volume is derived using regression equation between basal area and whole stem volume and the rest of volume components will be computed using ratio method in this function.

Usage

```
treeVolEst_RegRatio_new(nonVolTrees, BA_WSVCoeff, ratioTable)
```

Arguments

nonVolTrees	data.table, H-enhanced trees and non-enhanced trees.
BA_WSVCoeff	list, Contains fixed and random coefficients the WSV-BA equations by project group `PRJ_GRP`, live and dead status `LV_D`, stand and falling status `SF_COMPILE` and species code `SP0`. The table can be generated using [WSV_BARegression]().
ratioTable	list, Specifies `toWSV` ratio by project group `PRJ_GRP`, live and dead status `LV_D`, stand and falling status `SF_COMPILE` and species code `SP0`. The table can be generated using [toWSVRatio]().

Value

A data table that has compiled non volume trees.

Author(s)

Yong Luo

---

updateSA_vegcomp *Update projected stand age from veg comp rank 1 layer*

Description

This function is to update projected stand age from vegcomp rank 1 layer.

Usage

```
updateSA_vegcomp(
  compilationType,
  coeffPath,
  bcgwUserName,
  bcgwPassword,
  sampleSites,
  sampleMsmts
)
```

Arguments

compilationType

         character, either PSP or nonPSP. If it is PSP, it is consistent with original PSP compiler, otherwise, it is consistent with VRI compiler.

coeffPath   character, Specifies the path dependent coeffs are stored.

bcgwUserName   character, User name to access to bcgw database.

bcgwPassword   character, Password to access to bcgw database.

sampleSites   data.table, sample sites A data table must contain site_identifier, bc albers coordinates.

sampleMsmts   data.table, sample measurements A data table must contain site_identifier, bc albers coordinates.

Value

A data table with feature_id, proj_age_1 and projected_date.

Author(s)

Yong Luo

---

updateSpatial *Used for updating spatial attributes based on locations*

Description

Used for updating spatial attributes based on locations.

Usage

```
updateSpatial(compilationType, samplesites, mapPath)
```

Arguments

`compilationType`	
	character, either PSP or nonPSP. If it is PSP, it is consistent with original PSP compiler, otherwise, it is consistent with VRI compiler.
`samplesites`	data.table A table that contains spatial loctions. Currently it is designed for accepting UTM coordinates. Therefore, zone, northing and easting must be provided.
`mapPath`	character The path to fiz map.

Value

a table that contains spatial attributes of bc albers, longitude/latitude, TSA, BEC, FIZ, TFL and OWNERSHIP.

Author(s)

Yong Luo

`updateSpatial_badUTM_PSP`

Used for updating spatial attributes for the sites with bad utm

---

Description

Used for updating spatial attributes for the sites with bad utm.

Usage

```
updateSpatial_badUTM_PSP(mapPath, samplesites)
```

Arguments

`mapPath`	character The path to fiz map.
`samplesites`	data.table A table that contains spatial loctions. Currently it is designed for accepting UTM coordinates. Therefore, zone, northing and easting must be provided.
`compilationType`	
	character, either PSP or nonPSP. If it is PSP, it is consistent with original PSP compiler, otherwise, it is consistent with VRI compiler.

Value

a table that contains spatial attributes of bc albers, longitude/latitude, TSA, BEC, FIZ, TFL and OWNERSHIP.

Note

This routine is based on discussion with Dan and Anya on 2023-10-05

Author(s)

Yong Luo

---

valueCalculator   *Calcualte tree value-VRI specific*

Description

This function calculates tree value for each tree based on ground called grade. From the second column to the last column, the `grossVolMatrix` table should have same dimensions (i.e., number of rows and columns) of `callGradeMatrix` and `grossMerchVolMatrix` if they are provided. Furthermore, be aware of the correspondingness among the matrix. In the function, two lookup table are hardcoded (i.e., `spv_spc` and `sp_cost`). This function is part of the `log_valu_2017.sas`.

Usage

```
valueCalculator(species, grossVolMatrix, grossMerchVolMatrix, callGradeMatrix)

## S4 method for signature 'character,data.table,data.table,data.table'
valueCalculator(species, grossVolMatrix, grossMerchVolMatrix, callGradeMatrix)
```

Arguments

species	character, Species codes in BC inventory system.
grossVolMatrix	data.table, Calculated gross volume for each log. The first column of this table is the volume for the stump. If missing, the function calculates the total net merchantable volume.
grossMerchVolMatrix	
	data.table, Calculated gross merchantable volume for each log. If missing, all the merchantable volume is assigned as 0.
callGradeMatrix	
	data.table, Ground call grading table.

Value

Data table that contains net value (`VAL_NET`) and net merchantable value (`VAL_MER`).

Author(s)

Yong Luo

---

vicPrep   *prepare vi_c for compilation*

Description

This function selects the tree-level data from vi_c (cardc) based on selected cluster/plot headers. Additonally, the function calculates basal area and tree per ha factor.

Usage

```
vicPrep(compilationType, clusterplotHeader, dataSourcePath, walkThru = TRUE)
```

Arguments

compilationType

                character, either PSP or nonPSP. If it is PSP, it is consistent with original PSP compiler, otherwise, it is consistent with VRI compiler.

clusterplotHeader

                data.table, Cluster and plot-level attributes.

dataSourcePath   character, Specifies the path that directs to the compilation_sa.

walkThru   logical, Indicates whether walkthrough sampling protocal is used, Tree weight is determined by walkthrough method. In walkthrough method, a tree is identified as NA (no walkthrough applied), O for out tree (not counted), and W for double counted tree.

Value

A data table that contains tree-level information. A log file that describes the detailed process.

Author(s)

Yong Luo

---

vidPrep   *Load and select trees that have loss factor information-VRI specific*

Description

This function loads and selects trees that have loss factor information (vi_d, cardd) based on selected trees from vi_c.

Usage

```
vidPrep(dataSourcePath)
```

Arguments

dataSourcePath   character, Specifies the path that directs to the VRI original data soruce, i.e., //Mayhem/GIS_TIB/RDW/RDW_Data2/Work_Areas/VRI_ASCII_PROD/vri_sa.

Value

A data table that contains loss factor data. A log file documents the detailed process

Author(s)

Yong Luo

vihPrep	*to populate bored age for site trees using last measurements, set meas_cod = "FROM_LAST and to correct bored age using last measurements, set meas_cod = *-ADJ_FROM_LAST*

Description

This function is to populate bored age for site trees using last measurements, set meas_cod = "FROM_LAST and to correct bored age using last measurements, set meas_cod = *-ADJ_FROM_LAST

Usage

```
vihPrep(msmtInterval, siteAgeTrees)
```

Arguments

msmtInterval	data.table, contains sample visits.
siteAgeTrees	data.table, sample trees.

Value

A data table

Author(s)

Yong Luo

viiPrep	*Load and select auxiliary plot trees-VRI specific*

Description

This function loads and selects auxiliary data (`vi_i`, cardi) based on cluster/plot header.

Usage

```
viiPrep(compilationType, clusterplotHeader, dataSourcePath)
```

Arguments

compilationType

character, either PSP or nonPSP. If it is PSP, it is consistent with original PSP compiler, otherwise, it is consistent with VRI compiler.

clusterplotHeader

data.table, Cluster and plot level attributes, an output from `VRIInit_clusterplot`.

dataSourcePath   character, Specifies the path that directs to the VRI original data soruce.

Value

A data table that contains auxiliary plot tree data.

Author(s)

Yong Luo

---

volSmry_byC	*Summarize volume components at cluster level-VRI specific*

Description

This function summarizes the cluster-level volume components using cluster/species-level summaries. The cluster/species-level summaries is an output of volSmry_byCS function.

Usage

```
volSmry_byC(volSmryByCS)

## S4 method for signature 'data.table'
volSmry_byC(volSmryByCS)
```

Arguments

volSmryByCS   data.table, Summarized volume components for both measured and counted
                   trees. See volSmry_byCS for details.

Value

A data table

Author(s)

Yong Luo

---

volSmry_byCS	*Summarize volume components per hectare by cluster and species-VRI specific*

Description

Summarizes volume components per hectare by cluster and species. The function is last part of
vol_ha_2017.sas.

Usage

```
volSmry_byCS(treeMC, utilLevel, weirdUtil, equation)

## S4 method for signature 'data.table,numeric,character,character'
volSmry_byCS(treeMC, utilLevel, weirdUtil, equation)

## S4 method for signature 'data.table,missing,character,character'
volSmry_byCS(treeMC, weirdUtil, equation)

## S4 method for signature 'data.table,numeric,missing,character'
volSmry_byCS(treeMC, utilLevel, equation)

## S4 method for signature 'data.table,numeric,numeric,character'
volSmry_byCS(treeMC, utilLevel, weirdUtil, equation)

## S4 method for signature 'data.table,numeric,character,missing'
volSmry_byCS(treeMC, utilLevel, weirdUtil)

## S4 method for signature 'data.table,missing,missing,missing'
volSmry_byCS(treeMC)
```

Arguments

treeMC	data.table, Tree-level compiled data for all volume trees.
utilLevel	numeric, Utilization levels. Default is 4.
weirdUtil	character, Weird util. Default is 4. Otherwise need to be specified as a number.
equation	character, Specifies whether the compiler is based on KBEC or KFIZ. Default is KBEC.

Value

A data table summarizes volume components by cluster and species. Equevalent to smy_cs.

Author(s)

Yong Luo

VolumeSummaries_PSP *Summarize the PSP tree-level data at cluster or cluster/species level*

Description

Summarizes the compiled tree data (including both enhanced tree data and non-enhanced tree data) at cluster level. This function is equevalent to the summary part in sas compiler in cp_vegi_2017.sas. Different from the original compiler, this function outputs the summaries by summarized components, rather than putting all together.

Usage

```
VolumeSummaries_PSP(
  allVolumeTrees,
  clusterPlotHeader,
  utilLevel,
  weirdUtil,
  equation
)

## S4 method for signature 'data.table,data.table,numeric,character,character'
VolumeSummaries_PSP(
  allVolumeTrees,
  clusterPlotHeader,
  utilLevel,
  weirdUtil,
  equation
)
```

Arguments

allVolumeTrees   data.table, Compiled tree-level volumes data.

clusterPlotHeader

                 data.table, Cluster and plot-level information. An output of `VRIInit_clusterplot`.

utilLevel   numeric, Utilization levels.

weirdUtil   character, Weird util. Default is No. Otherwise need to be specified as a number.

equation   character, Specifies whether the compiler is based on KBEC or KFIZ.

Value

Cluster and species-level volume summaries; cluster-level volume summaries; cluster-level height summaries; cluster-level species composition summaries and log file.

Author(s)

Yong Luo

VRIInit_auxTree   *Load and select auxiliary plot trees-VRI specific*

Description

This function loads and selects auxiliary data (`vi_i`, `cardi`) based on cluster/plot header.

Usage

```
VRIInit_auxTree(clusterplotHeader, dataSourcePath)
```

Arguments

clusterplotHeader
data.table, Cluster and plot level attributes, an output from `VRIInit_clusterplot`.

dataSourcePath character, Specifies the path that directs to the VRI original data soruce, i.e., //Mayhem/GIS_TIB/RDW/RDW_Data2/Work_Areas/VRI_ASCII_PROD/vri_sa.

Value

A data table that contains auxiliary plot tree data.

Author(s)

Yong Luo

VRIInit_clusterplot   *Load and select cluster and plot level data- VRI specific*

Description

This function prepares the cluster/plot-level inputs for VRI compiler. Specifically, it standardizes names for the variables; reports and removes the duplicate observations at cluster, cluster/plot.

Usage

```
VRIInit_clusterplot(dataSourcePath)
```

Arguments

dataSourcePath character, Specifies the path that directs to the VRI original data soruce, i.e., //Mayhem/GIS_TIB/RDW/RDW_Data2/Work_Areas/VRI_ASCII_PROD/vri_sa.

Value

A data table that contains key information at cluster/plot level and compiler log file.

Author(s)

Yong Luo

VRIInit_lossFactor   *Load and select trees that have loss factor information-VRI specific*

Description

This function loads and selects trees that have loss factor information (vi_d, cardd) based on se-lected trees from vi_c.

Usage

```
VRIInit_lossFactor(fullMeasuredTrees, dataSourcePath)
```

Arguments

fullMeasuredTrees

data.table, Selected trees in vi_c, which includes full, enhanced and H-enhanced trees. An output of VRIInit_measuredTree.

dataSourcePath character, Specifies the path that directs to the VRI original data soruce, i.e., //Mayhem/GIS_TIB/RDW/RDW_Data2/Work_Areas/VRI_ASCII_PROD/vri_sa.

Value

A data table that contains loss factor data. A log file documents the detailed process

Author(s)

Yong Luo

VRIInit_measuredTree   *Load and select fully measured tree data-VRI specific*

Description

This function selects the tree-level data from vi_c (cardc) based on selected cluster/plot headers. Additonally, the function calculates basal area and tree per ha factor.

Usage

```
VRIInit_measuredTree(clusterplotHeader, dataSourcePath, walkThru = TRUE)
```

Arguments

clusterplotHeader

data.table, Cluster and plot-level attributes, an output from VRIInit_clusterplot.

dataSourcePath character, Specifies the path that directs to the VRI original data soruce, i.e., //Mayhem/GIS_TIB/RDW/RDW_Data2/Work_Areas/VRI_ASCII_PROD/vri_sa.

walkThru   logical, Indicates whether walkthrough sampling protacal is used, Tree weight is determined by walkthrough method. In walkthrough method, a tree is identified as NA (no walkthrough applied), O for out tree (not counted), and W for double counted tree.

Value

A data table that contains tree-level information. A log file that describes the detailed process.

Author(s)

Yong Luo

VRIInit_siteTree   *Load and select site trees-VRI specific*

Description

This function connects site tree data (vi_h, cardh) to selected cluster/plot-level data. Site tree data is located in `//Mayhem/GIS_TIB/RDW/RDW_Data2/Work_Areas/VRI_ASCII_PROD/vri_sa`

Usage

```
VRIInit_siteTree(clusterplotHeader, dataSourcePath)
```

Arguments

clusterplotHeader

                data.table, contains cluster/plot-level attributes, an output from `VRIInit_clusterplot`.

dataSourcePath character, Specifies the path that directs to the VRI original data soruce, i.e., `//Mayhem/GIS_TIB/RDW/RDW_Data2/Work_Areas/VRI_ASCII_PROD/vri_sa`.

Value

A data table that contains site tree data information. A log file documents the detailed process

Note

VRI specific

Author(s)

Yong Luo

VRISummaries | *Summarize the tree-level data at cluster or cluster/species level-VRI specific*

Description

Summarizes the compiled tree data (including both enhanced tree data and non-enhanced tree data) at cluster level. This function is equevalent to the summary part in sas compiler in cp_vegi_2017.sas. Different from the original compiler, this function outputs the summaries by summarized components, rather than putting all together.

Usage

```
VRISummaries(
  allVolumeTrees,
  clusterPlotHeader,
  utilLevel,
  weirdUtil,
  equation,
  nvafRatio
)

## S4 method for signature
## 'data.table,data.table,numeric,character,character,data.table'
VRISummaries(
  allVolumeTrees,
  clusterPlotHeader,
  utilLevel,
  weirdUtil,
  equation,
  nvafRatio
)
```

Arguments

allVolumeTrees data.table, All tree data from vi_c and vi_i that have been compiled with tree volume.

clusterPlotHeader

data.table, Cluster and plot-level information. An output of VRIInit_clusterplot.

utilLevel numeric, Utilization levels.

weirdUtil character, Weird util. Default is No. Otherwise need to be specified as a number.

equation character, Specifies whether the compiler is based on KBEC or KFIZ.

nvafRatio data.table, NVAF adjustment table based on BEC, Species, and LV_D.

Value

Cluster and species-level volume summaries; cluster-level volume summaries; cluster-level height summaries; cluster-level species composition summaries and log file.

Author(s)

Yong Luo

VRIVolTree   *Calcualte tree volume-VRI specific*

Description

This function use BEC(or FIZ) and species-specific taper equation to calculate tree volume. Before calculation, the function adjusts height for broken top trees: scenario 1 (D scenario): availability of DBH, DIB at broken height; scenario 2 (H scenario): availability of projected tree height in the field. This functions also assigns the volume multiplier adjustment. The function is equivalent to `vir_vol_tree_2011` macro in orignal SAS compiler.

Usage

```
VRIVolTree(
  treeData,
  equation,
  logMinLength,
  stumpHeight,
  breastHeight,
  UTOPDIB,
  bestHeightModels,
  HTBTOPModel
)
```

Arguments

treeData	data.table, An output from [VRIInit_measuredTree](#) function, i.e., vi_c data.
equation	character, Specifies which taper equation form will be used to calculate diameter inside bark for a given height. Must be either KBEC or KFIZ3. If missing, default is KBEC
logMinLength	numeric, Specifies a minimum length for a log.
stumpHeight	numeric, Specifies stump height. If missing, 0.3 m will be used.
breastHeight	numeric, Specifies breast height. 1.3 m will be used when this arguement is missing.
UTOPDIB	numeric, Specifies minimum merchantable inside bark diameter. 10 cm is used as a default.
bestHeightModels	data.table, External table that contains the best height/DBH model and coefficients by becsubzone and species.
HTBTOPModel	character, Specifies whether the height estimate for broken top trees either from `taper` or from `height`.

Value

A data table

Author(s)

Yong Luo

Index

81