Initial Conditions EAAM

Javier Porobic

November 27, 2023

Contents

Contents

1 Getting the templates

In this step, your objective is to obtain the core template that will serve as the foundation for creating the initial conditions of your simulation. To create these templates, you will require two essential components: the BGM file, which contains the polygon definitions, and the group definition files in CSV format (aka 'groups csv'). The code to create these templates is the ShinyRAtlantis. the following is the code and how to use it :

```
## creating the templates for the EAAM
library(shinyrAtlantis)
library(stringr)
library(tidyverse)
library(dplyr)
library(ncdf4)
grp.file = 'source_files/AntarcticGroups_v2.csv'
bgm.file_29 = 'EAAM_29_polygons_xy.bgm'
bgm.file_28 = 'EAAM_28_polygons_xy.bgm'
cum.depths = c(0, 20, 50, 100, 200, 300, 400, 750, 1000, 2000, 5000)
csv.name = 'template/EAAM_28'
source("/home/por07g/Documents/Code_Tools/shiny-Shane/Fork_git/shinyrAtlantis/R/initGen.R")

make.init.csv(grp.file, bgm.file_28, cum.depths, csv.name, ice_model=TRUE)
```

After creating the template there is few changes that need to be done to get the right template for the model :

1. Filling up the templates witht he right information for the init and horizontal
2. Make sure the default values make sense (temp, salinity and so on)
3. In your case, you will need to divide the box 21 in 2 boxes when using the 29 polygons version

2 Getting the previous initial condition

In this step, the primary objective is to extract and gather the values for Reserve and Structural Nitrogen, as well as the Abundance for each functional group as well as the information of nutrients and other variables. The underlying concept behind this approach is to delegate the handling of the vertical distribution of these values to the Atlantis itself. While it is indeed possible to manually force the vertical distribution, it would necessitate the creation of a new vertical distribution file (which is possible to do, but unnecessary).

```
## external set of tools that harvest the old.nc file
source('tools.R')
old_init_file='source_files/init_squid_230809.nc'
## Getting the total values, total sum and abundance as well as defaul values
harvest_ini(old_init_file, fillVal=TRUE, output_file = 'template/old_EAAM28')
## Getting the mean for RN and SN
harvest_ini(old_init_file, calculate_mean=TRUE, output_file = 'template/old_EAAM28')
```

3 Populating the template

In this step, you will take the values calculated in the previous step and input them into the templates. For abundance, you will use the sum, and for Reserve and structural nitrogen, as well as certain other required variables, you will use the mean. Please note that for the current configuration of your model, you need to divide box 21 from the old model into boxes 21 and 22.

- I will keep the configuration for 28 polygons if you want to change that, you can either update the template yourself or use the modified template.

```r
## Load the original data from 'old_EAAM_Sums.csv' into 'orig' dataframe
orig <- read.csv('template/old_EAAM28_Sums.csv', sep = ',')
## Load the template data from 'EAAM_horiz.csv' into 'template.h' dataframe
template.h <- read.csv('template/EAAM_28_horiz.csv')

## Get column names of the 'orig' dataframe
col <- colnames(orig)
## Iterate through each row of 'template.h' to match and update values
for (i in 1 : nrow(template.h)){
  ## Find the positions of matching column names in 'orig'
  pos <- which(col %in% template.h$Variable[i])
  ## Skip to the next row if no matching column found
  if (length(pos) == 0 | is.null(pos)) next()
  ## Update the values in 'template.h' with corresponding values from 'orig'
  template.h[i, 2 : ncol(template.h)] <- orig[, pos]
}

## Load default values from 'old_EAAMFillValues.csv' into 'orig.default'
orig.default <- read.csv('template/old_EAAM28FillValues.csv', sep = ',')

## Load means data from 'old_EAAM_Means.csv' into 'orig.m' dataframe
orig.m  <- read.csv('template/old_EAAM28_Means.csv', sep = ',')
col.nam <- colnames(orig.m)

## Define a subset of columns to update in 'template.h'
col <- c('NH3', 'NO3', 'DON', 'Si', 'Chl_a')
## Iterate through each row of 'template.h' to match and update values
for (i in 1 : nrow(template.h)){
  ## Find the positions of matching column names in 'orig.m'
  pos <- which(col %in% template.h$Variable[i])
  ## Skip to the next row if no matching column found
  if (length(pos) == 0 | is.null(pos)) next()
  ## Find the corresponding positions in 'col.nam' for the matched columns
  t.pos <- which(col.nam %in% col[pos])
  ## Update the values in 'template.h' with corresponding values from 'orig.m'
  template.h[i, 2 : ncol(template.h)] <- orig.m[, t.pos]
}

## Write the updated 'template.h' to 'EAAM_horiz_filled.csv' file
write.table(template.h, file = 'template/EAAM_horiz_28_filled.csv', sep = ',', row.names=FALSE)
## Define default and required values to fill
default.val <- c(grep('*_StructN', orig.default$Variables), grep('*_ResN', orig.default$Variables), grep('*_F$', colnames(orig.m)))
required_vals <- c(grep('*_StructN', colnames(orig.m)), grep('*_ResN', colnames(orig.m)), grep('*_F$', colnames(orig.m)))
## Load the initial condition template data from 'EAAM_init.csv' into 'template.ini'
template.ini <- read.csv('template/EAAM_28_init.csv')
## Get column names from 'orig.m' for required values
col.names <- colnames(orig.m)[required_vals]
## Extract fill values from 'orig.default' based on default values
fillvals <- orig.default[default.val, ]
## Iterate through each row of 'template.ini' to update sediment and wc.hor.scalar values
for (i in 1 : nrow(template.ini)){
  ## Find the positions of matching column names in 'col.names'
  pos <- which(col.names %in% template.ini[i, 1])
  ## Skip to the next row if no matching column found
  if (length(pos) == 0) next()
  ## Extract the values from 'orig.m' for the matching columns
  values <- orig.m[1 : 2, required_vals[pos]]
  ## Replace missing values with corresponding fill values
  if (any(is.na(values))){
    replace <- fillvals$fillvalues[which(fillvals$Variables %in% template.ini[i, 1])]
    values[is.na(values)] <- replace
  }
  ## Update 'template.ini' with the calculated values
  template.ini[i, c("sediment", "wc.hor.scalar")] <- values
}
## Write the updated 'template.ini' to 'EAAM_init_filled.csv' file
write.table(template.ini, 'template/EAAM_init_28_filled.csv', row.names = FALSE, sep = ',')
```

4 Creating the New initial condition files

- After filling in the templates with the necessary data, which includes reserves, structural nitrogen, species abundance, and all other required variables, you will be able to generate the new initial conditions file. Ensure that the initial conditions match the previous ones.

```r
init.file<- '/home/por07g/Documents/Projects/Supervision/Ilaria/Initial_conditions/template/EAAM_init_28_filled.csv'
horiz.file<-'/home/por07g/Documents/Projects/Supervision/Ilaria/Initial_conditions/template/EAAM_horiz_28_filled.csv'
```

```
source('/home/por07g/Documents/Code_Tools/shiny-Shane/Fork_git/shinyrAtlantis/R/initGen.R')
#debug(make.init.nc)
make.init.nc(bgm.file_28, cum.depths, init.file, horiz.file, 'EAAM_28_init.nc', ice_model=TRUE)
```