**Resource**

<https://github.com/femeunier/ED2support> -> to branch and extend and collect all tutorials here then link to q-ForestLab GitHub

.

**Step 1 download and compile ED2**

ml intel-compilers/2023.2.1 HDF5/1.14.3-iimpi-2023b UDUNITS/2.2.28-GCCcore-13.2.0; ulimit -s unlimited

mkdir ED2.2

cd ED2.2

git clone [git@github.com:EDmodel/ED2.git](mailto:git@github.com:EDmodel/ED2.git)

OR

git clone https://github.com/EDmodel/ED2.git

####################################################

git config --global user.name "FULLNAME"  
git config --global user.email [you@yourdomain.example.com](mailto:you@yourdomain.example.com)

####################################################

cd ED2/ED/build/make

cp include.mk.intel include.mk.intel\_hpc

Edit include.mk.intel\_hpc:

HDF5\_INCS=-I/apps/gent/RHEL9/skylake-ib/software/HDF5/1.14.3-iimpi-2023b/include

HDF5\_LIBS=-lm -lz -L/apps/gent/RHEL9/skylake-ib/software/HDF5/1.14.3-iimpi-2023b/bin -lhdf5 -lhdf5\_fortran -lhdf5\_hl

cd ..

./install.sh -k E -p intel\_hpc

k: E or I, I is faster but not understandable if crashes, it doesn’t do some checks

p: refers to extension of files in make folder

./install.sh -k E -p intel\_hpc –clean (to repeat compilation)

**Step 2: prepare a simple run**

First we have to get some climate drivers for the model, these can be downloaded from <https://github.com/femeunier/ED2support/tree/main/outputs/drivers>

Put these on the cluster in an appropriate directoiry e.g. under $VSC\_DATA\_VO/your\_name/ED\_drivers/test

Then download the ED2IN file from <https://github.com/femeunier/ED2support/tree/main/files>

then adapt:

NL%ED\_MET\_DRIVER\_DB = '/scratch/gent/465/vsc46573/test\_ED2/site.lat9.25N.lon79.75W/ED2/'

NL%FFILOUT = '/scratch/gent/465/vsc46573/test\_ED2/analysis/analysis'

NL%SFILOUT = '/scratch/gent/465/vsc46573/test\_ED2/history/history'

**Step 3: run the model**

Download the Job.sh file from <https://github.com/femeunier/ED2support/tree/main/files>

Or inside an interactive job (qsub -I)

cd /user/gent/465/vsc46573/ED2.2/ED2/ED/run

../build/ed\_2.2-opt-master-8d4c3aff -f ED2IN\_history /user/gent/465/vsc46573/ED2.2/ED2/ED/build/ed\_2.2-opt-master-8d4c3aff -f ED2IN

**Step 4: Postprocess model**

install.packages(c("abind", "agricolae", "akima", "beanplot", "boot", "callr", "car", "caTools", "chron", "cluster", "compiler", "data.table", "devtools","FAdist", "fields", "gbm", "gdalUtils", "geoR", "gpclib", "grDevices", "gstat", "Hmisc", "klaR", "kriging", "leaps", "maps", "mapdata", "maptools", "MASS", "MCMCpack", "nlme", "numDeriv", "onls", "PBSmapping","plotrix", "pls", "proto", "raster", "rgdal", "rgeos", "rlas", "robustbase","rworldmap", "RSEIS", "R.utils","smatr"))

# rhdf5 is a specific package that needs to be installed separately

if (!require("BiocManager", quietly = TRUE))

install.packages("BiocManager")

BiocManager::install("rhdf5")

And manuel changes in files as written in comments script

Additionally comment ‘survival’ ‘ggplot2’ and forecast in load.everything.r

Make functions local in operator.r

source("post.process.ED2.outputs.R")

**Step 5: Plot model outputs (generally done locally)**

load("./outputs/analysis.RData")

load("C:\Users\sdeherto\OneDrive - Vrije Universiteit Brussel\postdoc\ED2

")

matplot(datum$szpft$gpp[,12,c(2,3,4,18)],type = "l")

**Step 2.1 (download forcing data and convert to ED compatible format)**

cd

ml purge; ml R-bundle-Bioconductor/3.15-foss-2021b-R-4.2.0

R

source("download.and.convert.input.ED2.R")

install.packages("udunits2")

install.packages('PEcAn.utils', repos = c('https://pecanproject.r-universe.dev', '<https://cloud.r-project.org>'))

install.packages('PEcAn.remote', repos = c('https://pecanproject.r-universe.dev', '<https://cloud.r-project.org>'))