HOW TO

install libraries to compile/run Eco3M in the mini-CLUSTER (IEO)

Step-by-step guide to setting up an environment where you use the **Intel ifort compiler** for fortran with MPI and have support for **HDF5**, **NetCDF**, and **PnetCDF** libraries.

This setup uses **conda** for managing dependencies but Intel compilers and PnetCDF are not available via conda, therefore they need to be installed separately.

Conda is not available in the cluster, you can install miniconda following these instructions:
`/home/alvarez/EVA/Eco3M-MAR_MENOR/instructions_conda.txt`

1. Install Intel OneAPI:

The Intel oneAPI toolkit includes ifort, icc, icpc, and Intel MPI. Go to the Intel OneAPI page (https://www.intel.com/content/www/us/en/developer/tools/oneapi/toolkits.html#gs.fk8h75) and download the "Base Toolkit" and "HPC Toolkit", for Linux and the Offline Installer.

Follow Intel's official installation guide:
https://www.intel.com/content/www/us/en/docs/oneapi/installation-guide-linux/2024-2/install-with-command-line.html#GUID-56B16998-1363-40F5-A6D5-6A3D5B877F37

which, if you do not have root permission, summarizes in doing:

```
sh ./l_BaseKit_p_2024.2.1.100_offline.sh -a --silent --eula accept
sh ./l_HPCKit_p_2024.2.1.79_offline.sh -a --silent --eula accept
```

Find the files here: `/home/alvarez/EVA/Eco3M-MAR_MENOR/intel-oneAPI`

After installation, find where /intel/oneapi/ was installed (if you installed as user, it will be at $HOME) and source the setvars.sh script by doing:

```
source ~/intel/oneapi/setvars.sh
```

This script sets up the environment for Intel compilers and MPI. At any time, you can source again the script by including the flag `--force`.

Add the line to your `.bashrc` file in case you want to set the variables every time you enter the cluster.

2. Create the Conda Environment:

Create a new conda environment using the .yml file available in the repository (https://github.com/ealvarez-s/Eco3M-MAR_MENOR.git).

```
cd /home/alvarez/EVA/Eco3M-MAR_MENOR/
conda env create -f environment.yml
conda activate belich-legos
```

Find the files here: `/home/alvarez/EVA/Eco3M-MAR_MENOR`

3. Set Compilers and MPI Environment Variables:
With the conda environment active! ensure that mpiicc and mpiifort use the Intel ifort compiler and Intel MPI. You'll need to override the conda MPI wrappers as follows:

```
export CC=gcc
export CXX=g++
export FC=ifort
export MPICC=mpiicc
export MPICXX=mpiicpc
export MPIFC=mpiifort

export I_MPI_CC=gcc
export I_MPI_CXX=g++
export I_MPI_FC=ifort
```

Conda prioritizes its own wrappers when using mpifort, to ensure that this does not create conflicts **do not use mpifort** and use mpiifort instead. For security, you can rename the mpifort script in the conda environment to something like mpifort_backup.

```
~/miniconda3/envs/belich-legos/bin/mpifort -> mpifort_backup
```

Check before continuing

`nc-config --has-parallel` → must say yes

`which mpiicc` → it must point to the Intel compiler (something like `~/intel/oneapi/mpi/2021.13/bin/mpiicc`)

`which mpiifort` → it must point to the intel compiler (something like `~/intel/oneapi/mpi/2021.13/bin/mpiifort`)

4. Install PnetCDF:
You need to build and install PnetCDF from the source, as it is not available directly through conda channels. Download PnetCDF by doing:

```
wget https://parallel-netcdf.github.io/Release/pnetcdf-1.12.3.tar.gz
tar -xvf pnetcdf-1.12.3.tar.gz
cd pnetcdf-1.12.3
```

Find the files here: `/home/alvarez/EVA/Eco3M-MAR_MENOR/pnetcdf-1.12.3`

Before running `configure`, ensure any previous failed configuration attempt is completely removed. Just in case, run:

```
make clean
rm -rf config.log
rm -rf autom4te.cache
```

Make sure you are installing PnetCDF within the conda environment, and `which mpiicc` points to the Intel compiler.

Configure PnetCDF with Intel compilers and MPI by doing:

```
./configure --prefix=$CONDA_PREFIX --enable-parallel --enable-shared --enable-fortran CC=mpiicc CXX=g++ FC=ifort  MPICC=mpiicc MPIFC=mpiifort
```

If the `configure` step fails, use these flags for a debug and more verbose version:

```
CFLAGS="-O2" --enable-debug
```

After `configure`, the option enabled shared libraries must say yes.

Build and install it, by doing:

```
make
make install
```

If installation is successful, it will tell you the LIB and NETINC paths. They should be the same for **netcdf**, but just in case double check by running:

```
nc-config --libdir
nc-config --includedir
```

Those paths need to go to the makefile.inc in:

```
~/EVA/Eco3M-MAR_MENOR/SYMPHONIE_368/UDIR/BGC_MAR_MENOR2/makefile.inc
```

Example:

```
NETINC=-I/home/alvarez/miniconda3/envs/belich-legos/include
LIB=-L/home/alvarez/miniconda3/envs/belich-legos/lib -lnetcdff -lnetcdf -lpnetcdf
```

Other tips in the makefile.inc:
to avoid messages about the intel MPI version, use `-diag-disable=10448` as compiling flag.
F90 needs to be set to mpiifort (do not use mpifort!).

To compile:

`conda activate belich-legos` (make sure the environment is active!)

You may need to update the LD_LIBRARY_PATH if the shared libraries are not found, by doing:

```
export LD_LIBRARY_PATH=/home/alvarez/miniconda3/envs/belich-legos/lib:$LD_LIBRARY_PATH
```

Compile by doing:

```
cd ~/EVA/Eco3M-MAR_MENOR/SYMPHONIE_368/UDIR/BGC_MAR_MENOR2
make
```

This will create the executable in:

```
~/EVA/Eco3M-MAR_MENOR/SYMPHONIE_368/RDIR/BGC_MAR_MENOR2/S26.exe
```