WORKFLOW

loop_over_all_ARIANE_runs.sh

Loop/parallelize by larval parameters which are specified in loop_over_all_ARIANE_runs.sh.

build_larval_behvaviors.sh

Create output directory for this combination of larval behaviors

Loop/parallelize to split up simulations by year and season.

For each larval parameter

These loops and loops above are all in loop_over_all_ARIANE_runs.sh.

Note: When running the full calculation, the postprocessing will be slightly more complicated due to spitting apart the 12 case study sites (which are simulated together) into separate t, hist, and alive files. However, since we are running only in a very small subdomain, I did not include this postprocessing step, which would come between step 0 and 1. The code for the splitting is robust, well tested, and not expected to be a big computational overhead.

INPUTS

Larval behaviors definition file changes with each combination of the 5 larval parameters.
larval_behaviors.txt

Input data driving simulaion, 7592 files of which only a subset of 146 are used in one simulation.
VIKING20_10001_grid_U.nc
VIKING20_10001_grid_V.nc
…
VIKING20_10073_grid_U.nc
VIKING20_10073_grid_V.nc

Unchanging, static input files:
initial_positions.txt
namelist
mesh_mask.nc
V20_bottom_properties.nc

Set which input data drive the simulation, mklist_VIKING20.sh

It is the input data (VIKING20) that sets what year and season the larvae "experience" during the simulation.

ARIANE (the simulation)

ariane2tcdf (postprocessing 0)

tcdfhist (postprocessing 1)

tcdfsurvive (postprocessing 2)

OUTPUTS

- ariane_trajectories_qualitative.nc
- t.cdf
- hist.nc
- alive.nc

For each larval parameter for each year for each season - 6400 total instances. The simulation and postprocessing steps are all in the Docker container with output files going to /app/run.

Rename output files by year/season, move to output directory for that run. t.cdf, hist.nc, alive.nc

Linearly superpose the output files after all parallel runs are done.