ESP (figures and analysis)

Fig 1 – made by tracing Google map with inkscape & used open source dolphin silhouette from phylopic

Fig 2 – made using “plot\_dolphins\_vs\_ESP\_B.py”.

Input:

* ddPCR\_data\_2024-03-12\_dPB105.csv

ddPCR quants of T Tursiops, received directly from Megan Shaffer. Datasheet didn’t have time steps.

* MURI\_Module1\_datasheets - 03\_ESP1\_Feb2023.csv

Used just for the time steps of ESP quants. Downloaded from the MURI drive (MURI Project 2022-2027/Modules – Project Plans & Notes/Module 1 – Hood Canal dolphins/01 M1 Lab/MURI\_Module1\_datasheets.xlsx (sheet called “03\_ESP1\_Feb2023”) <https://docs.google.com/spreadsheets/d/1p8M0_PJOpVJpVYAjRic67Ak7A2-T42tB0i7HOFwxdME/edit?gid=460468686#gid=460468686>

Would probably be better to use the updated/maintained version of the data in “MURI Project 2022-2027/Modules – Project Plans & Notes/All\_Mod1\_Molecula\_Data.xlsx”, but it’s just the timesteps, so it’s probably same same

* dolphin\_presence.csv

Saved in “MURI Project 2022-2027/Modules – Project Plans & Notes/Module 1 – Hood Canal dolphins/03 ESP/NOAA ESP Deployment/DolphinPresenceData\_Jan2023\_ESP\_deployment.csv”. Digitized version of dolphin occupancy datasheet from Bangor Marine Mammal team personnel. Original datasheet photo saved in same folder as “NOAA Data Sheets\_Ttru occupancy\_ESPdeployment\_Jan2023.pdf”

Code plots ddPCR quantified dolphin DNA in ESP samples against reported dolphin occupancy, representing the mismatch between the time the ESP is actively sampling and the dolphin occupancy dataset.

Fig 3 – made using “plot\_nimble\_models.py”. All nimble models and data are available at https://github.com/eabrasse/eDNA\_nimble

Input:

* m30lnormobservation\_plotting.csv, m31lineardolphin\_plotting.csv, m32expdolphin\_plotting.csv, m33mixeddolphin\_plotting.csv, m34tidalloss\_plotting.csv, m35randomloss\_plotting.csv

Saved the subset of nimble model output relevant for plotting. Matching the prefix of the output to the generating models and model runfiles…  
Nimble models are m30lnormobservation.R, m31lineardolphin.R, m32expdolphin.R, m33mixeddolphin.R, m34tidalloss.R, and m35randomloss.R  
Run files are runm30lnormobservation.R, runm31lineardolphin.R, runm32expdolphin.R, runm33mixeddolphin.R, runm34tidalloss.R, runm35randomloss.R  
Models are run with these input files, both produced with “process\_boxmodel\_data.py”:

* + ESP\_timestamps\_mLseawater.csv
  + dolphin\_tide\_minutefreq.csv

for runm34tidalloss.R, input also includes

* + ESP\_box\_model\_terms\_ESPtimesteps.csv

I can’t find the piece of code that generated that one. It must have been an older version of process\_boxmodel\_data.py. It’s just the NOAA SSH data interpolated to the same timesteps as the ESP data. Even though the processing code is missing, the data (and the other input files) are available at https://github.com/eabrasse/eDNA\_nimble

* ESP\_timestamps\_mLseawater.csv

Made using process\_boxmodel\_data.py

Code plots the output of 6 nimble models (saved as “code/R/nimble/Results/m3\*[modelname]\_plotting.csv”) against the mean of the ESP triplicates.

Note: other details of the nimble model results are saved in <https://github.com/eabrasse/eDNA_nimble/blob/main/Results/model_comparison_v2.xlsx>

Fig 4 – made with “plot\_Feb2023\_const\_extended\_F.py

Input:

* All\_Mod1\_Molecular\_Data - all\_data.csv

Available on MURI drive <https://docs.google.com/spreadsheets/d/1tCg9GWKrK1eHJ-xDaEXG2wO8CyTY2um9wkvBYRFO1xw/edit?gid=0#gid=0>

The next few inputs are processed versions of my particle release experiments using Parker MacCready’s tracker code. Tracker is available as part of Parker’s LiveOcean software suite: <https://github.com/parkermac/LO>, with good notes on installation in the Readme. I ran 96 releases from four days before the sampling campaign to the day of the sampling campaign, each with a run command that looked like:

python tracker.py -gtx hc11\_v01\_uu0k -exp hc\_dolph -3d True -dtt 4 -clb True -d 2023.01.31 -sh 16 -sub\_tag 2023.01.31\_sh16 > /data2/pmr4/eab32/LO\_output/track\_logs/hcdolphJune2024\_log16.txt &

(Run commands were generated in sets of 10 in a bash file using gen\_Feb2023\_bash.py) The tracker notes explain what all the flags mean. The hc\_dolph “experiment” is outlined in /data2/pmr4/eab32/LO\_user/tracker/experiments.py. Some metadata is saved in the particle release results folders as well under “exp\_info.csv”. The results from each release went to /data2/pmr4/eab32/LO\_output/tracks/hc\_dolph\_3d\_sh16\_2023.01.31/release\_2023.01.31.nc (or equivalent date & start hour combo). I processed these in a couple of ways to create a picture of the accumulation of hourly releases over four days leading up to the transect samples. All of these were generated by code run on Parker’s perigee machine where the tracker run results are – contact me if you need these to be rerun or modified.

* Feb2023\_hist\_counts\_dev2.p – a 2d histogram (or heatmap) of surface (upper 2m) particles, generated using “extract\_Feb2023\_releases.py”
* Feb2023\_moor\_radius\_counts\_dev2.p – calculated surface (upper 2m) particles near the February transect sampling locations (I called them “moorings” for some reason I can’t recall now. But I mean the February transect sampling locations). I tested including particles from a bunch of different radiuses from 15 m to 150 m. I also calculated the stats of particle ages at the time and location of sampling.
* all\_3d\_hc\_dolph\_releases.p – in my comments, I say I only use this file because it included the grid & grid mask, so I must have forgotten to include those in my other extractions. That was dumb of me. This file was generated using “process\_tracks\_at\_refT.py” but you could probably use any file from the high resolution HC grid to get those.
* Feb2023\_surfaceparticlepositionstats.p – this was my attempt to quantify the rate that the diffusion process reduces particle concentration for each pulse of particles released. I used the increase in distance between particles as a metric for decreased concentration (particles/vol = particles/distance^3, as distance between particles increases, particle/vol goes down) I don’t feel great about that metric, tbh – mostly because it’s not obvious that the diffusion of a pulse will be the same as the diffusion of a continuously released thing

This figure has four plots. (1) The biggest one demonstrates the comparison of DNA observed at the transect samples against environmental transport model predictions, with (2) an inset showing a map with the full heatmap of particle concentrations and where the samples were taken relative to the husbandry area. (3) shows the distribution of particle ages at each location, and (4) does a really rough estimate of lab estimated decay vs model estimated diffusion vs observed concentration decrease vs modeled concentration decrease.