----- Biosonar array toolbox -----

**Foreword:**

These tools, developed for Matlab 6.5 and above, have been written to facilitate studies of the echolocation parameters of free-ranging odontocetes using hydrophone arrays. The tools include support for automated click detection [a], click localization [b], click train visualization, identification of on-axis clicks and click parameter extraction [c], click parameter compilation [d] and biosonar beam quantification including bootstrap methods for quantifying confidence intervals [e].

Tools have been developed by Frants H. Jensen in collaborations with Kristian Beedholm, Prof. Magnus Wahlberg, and Prof. Peter T. Madsen

**Prerequisites:**

1) Calibration of individual hydrophones (use pistonphone or relative calibration)  
2) Calibration of localization accuracy using array (can be done using some of the included tools)  
3) Sea surface temperature and salinity or measured underwater sound speed in recording area   
 - see soundspeed.m [Mark Johnson, St. Andrews University] for quick estimate of sound speed based on leroy equation with temperature, salinity and depth input  
4) Array recordings of series of echolocation clicks from animals approaching recording array  
 - Careful notes of other species nearby should be kept while collecting recordings  
 - Works best when animals approach or pass the array - keep careful field notes of when animals have passed the array and exclude those recordings

Tools are meant to be modular, so as to facilitate gradual changes such as updated localization methods, added source parameter measurements or new options. All toolbox settings are adjusted in one file (toolbox\_settings.m) and passed as a structure to all toolbox scripts.

**----- Quick overview of toolbox -----**

**toolbox\_settings.m:** Contains settings for different toolbox scripts, including array configuration, environmental parameters, analysis parameters, information on where to save output files, and various cosmetic options. Make sure to configure this file correctly before analysing data. If some files have been recorded with different system amplification, an option to modify amplification and recalculate clip levels for all hydrophones is provided when running extractclicks.m

**extractclicks.m:** Goes through a single multichannel wave file and does automatic click detection (based on user defined threshold or automatic threshold defined by background noise), click localization, and click train visualization. Last step allows user to select on-axis clicks, check localization of that click, extract parameters and save to user-defined file.

***Instructions:*** Sequential plots of detected clicks (amplitude on all channels, estimated position) will be shown, separated up into click series following the maximum ICI criterion in the toolbox settings. Previously analysed clicks will be marked in plot (but see comments below). User will have a ginput interface (a crosshair will follow the mouse cursor and allow you to click in plot). Following options are available:

-left click: Locate nearest click and extract parameters for this click:  
First, source localization is shown. Check that surface reflection is never higher than highest direct click, that localization plot looks good, and that all cross correlation peaks (or envelope peaks) are found correctly. NBHF species will likely require different localization method (-6 dB envelope) which is chosen in toolbox settings.  
Once localization is accepted, the extracted waveforms and power spectral densities (based on toolbox settings) are shown and user can name the click (underscores not permitted, will be replaced with dash), after which it will be saved to \*\_params.mat file

-'f': Press the f button to go to next scan

-'t': Zoom in (x2), centered on cursor

-'y': Zoom out, centered on cursor

-'r': Reset zoom

-'s': Save ICIs of all clicks in primary (red colorcode) scan to click detector output

-'q': Press the q-button to immediately quit (not recommended)

End-of-file options: Once the end of a wave file is reached, user will have the option of going back to start of file to go through each scan again (either for verification purposes, to save corrected ICIs using the 's' button, or to add any missed on-axis clicks.

***Data output:*** All extracted parameters will be saved to individual click parameter files (\*\_params.mat) with measured source parameters, information on source file and location within file, and all toolbox settings used for click parameter extraction. Output files will be saved to data library defined in toolbox settings. Previously analysed clicks will be marked in plot - but ONLY if those parameter files are still located in the default output directory.

**compileclickparams.m:** Goes through a user defined library and loads in all click parameter files (ends with \_params.m), accumulates all source parameters into variables (ASLpp, range, etc), saves variables to file for further data analysis, statistics, etc, and shows summary statistics.

**pistonfit.m:** Loads in a file containing all on-axis clicks (output from compileclickparams.m) and does a parametric fit of a circular piston model to find the best-fitting piston size and consequently beam pattern parameters. Then gives user the option to continue with a bootstrap procedure to measure confidence around estimate. Once this is finished, beam parameter statistics (mean and confidence intervals) will be displayed and saved to file. Bootstrapping can be time-consuming - expect bootstrap to take a few hours. If you have access to MatLab parallel computing toolbox, you can accellerate bootstrapping by activating matlab workers (the different computing cores in your laptop) and switching the bootstrap 'for' loop to a parallel computing 'parfor' loop.

**showparams.m:** Displays summary statistics (tables with data values) for datasets.

Other scripts are support scripts in one way or another. Many of these can be run on their own, but see individual help menus to learn more.

----- Good practice and advice -----

Times change, and tools occasionally do too. When finishing up a project, it can be good to zip a copy of the tools used to create that project, and to save those tools along with all click detection files (\*\_click.mat) and click parameter files (\*\_params.mat), in a secure, backed-up location.

If you make changes to tools, keep logs of what changes are made, and always keep an original copy of script. Please forward change log and bug reports to frants.jensen@gmail.com.