Sea Otter Foraging Analysis (SOFA) V. 2.0, User Manual

SOFA created for U.S. Geological Survey and Seattle Aquarium

by M.T. Tinker, Nhydra Ecological

SEATTLE AQUARIUM

Nhydra

USGS science for a changing world

1. Overview

This manual provides instructions for installing and using the "SOFA" program for analysis of sea otter observation-based foraging data. The details of analysis methods are provided in an attached appendix. The manual provides detailed step-by step instructions for installing the necessary software, downloading the current version of SOFA from GitHub, setting up a raw data file for analysis, and running SOFA from R (which is actually a 3-step process). The last step involves the generation of an R markdown report to present results: this report can then be opened as an html file on any web browser.

2. Installation of Programs Necessary to run SOFA

1. Install up to date version of R:
    - https://cran.r-project.org/bin/windows/base/
2. Install Rstudio:
    - https://rstudio.com/products/rstudio/download/
    - (scroll down to the "Download Rstudio for Windows" button)
3. Install Rstan and/or cmdstan (which is the "native" version of stan, more efficient memory use):
    a) First you should install Rtools 4.2 (uninstall old versions if necessary)
        - https://cran.r-project.org/bin/windows/Rtools/
        - Select "Windows 64 bit" version
        - After installing Rtools, open Rstudio and run this line of code from the console:
          writeLines('PATH="${RTOOLS40_HOME}\\usr\\bin;${PATH}"', con = "~/.Renviron")
          (then restart R)
    b) Next install Rstan from within Rstudio by running this line of code from the console:
        - install.packages("rstan", repos = "https://cloud.r-project.org/", dependencies = TRUE)
    c) Test Rstan installation by running these two lines of code from the console:
        - library(rstan)
        - example(stan_model,run.dontrun = TRUE)
    d) Optionally, install cmdstan and cmdstanR: see instructions here:
        - https://mc-stan.org/cmdstanr/articles/cmdstanr.html
4. Install the remaining necessary libraries (or packages) from within R (use "Install Packages"):
    - readxl
    - openxlsx
    - rstudioapi
    - svDialogs
    - tcltk
    - parallel
    - dplyr
    - stats
    - MASS
    - gtools
    - ggplot2
    - gridExtra
    - bayesplot
    - rmarkdown
    - knitr

3. Download current version of SOFA program from GitHub

The current version of SOFA can be downloaded at any time from the GitHub repository:

https://github.com/mttinker/SOFA.git

If you are not familiar with GitHub, the easiest way to handle this is to click on the Green "Code" button and select "Download Zip". This will download a zip archive to your computer, which you can then un-zip to an appropriate location in your workspace. This will result in a "SOFA-master" folder with other folders and files nested within it. Rename the top-level folder from "SOFA-master" to "SOFA".

If you are already a GitHub user, then you know your other alternative is to create a clone of the SOFA repository, which will allow you to more easily keep your version up to date (by doing "Pull requests"). If the phrase "Pull request" sounds bizarre or inappropriate to you, then maybe stick with option 1.

NOTE: the directory structure (folder names and locations) of the repository should be left as-is: DO NOT change the locations or names of any of these folders, or SOFA will stop working. As you run SOFA (see below), new sub-folders will appear within the "projects" folder, which contain your data and results – you can edit the files within these individual project directories as you wish, but do not change or edit files within the higher-level folders.

4. Setup your Foraging Data File

Different observers and different studies have employed slightly different formats for recording foraging data. Generally, these different formats can be simplified or translated into a common format for analysis. The table below describes the data fields that must be included in a data file intended for analysis by SOFA. Other fields may also be included in the spreadsheet (they will be ignored), and the order of columns is not important. However, for the fields described in the table below, the column names in your spreadsheet must match exactly (although the program is not case sensitive so upper case or lower case letters do not matter), and the field types (text, numeric, date) must also match. For text fields with letter codes, the code values must be the same as those described below (e.g. for qualifier, the codes must be "a", "b", "c" or "z"). **Errors in data formatting are the most common reason a SOFA analysis fails**, so please check over your data file carefully.

Table 1. Database format for sea otter foraging data. Spreadsheets must have fields with these field (or column) names and data types. Note that field names and text codes are not case sensitive.

Field name	Data type	Description
Region	text	Largest Spatial Designation, 4-letter code (WASH, SEAK, SANI, CECA, etc.)
Area	text	Intermediate Spatial grouping, designating Study area (or "sub-region")
Site	text	Smallest Spatial grouping, designating specific data collection site (possibly an Island name or coastal headland)
Period	text	Time Period, Categorical, User-defined (e.g. "2005-2012", or "years 1-5" or "years 5-10")
Date	date	mm/dd/yyyy (e.g. 5/14/2007, or 10/31/2020)
Season	text	user-defined: e.g. "summer" for May - Oct, "winter" = for all other months
Sex	text	m = male, f = female, u = unknown
AgeClass	text	j = juvenile, sa = subadult, a = adult, aa = aged adult, u = unknown.
Pup	text	y = yes focal otter has pup but pup size not known or not recorded, n = focal otter has no pup, u = unknown if focal otter has pup or not, s = focal otter has small pup, m = focal otter has medium size pup, l = focal otter has large pup
Ottername	text	for studies without marked otters use generic names: **adfem** = adult female, **admale** = adult male, **adunk** = adult unknown sex, **subfem** = subadult/juv female, **submale** = subadult/juv male, **subunk** = subadult/juv sex unknown.

Field name	Data type	Description
Bout	text	unique bout identifier (may consist of date, time, observer name, etc.)
Subbout	integer	Sequential Integer for non-contiguous sub-sets of recorded dives within a bout: not used in AK studies, so only 1 subbout for all bouts
TimeStart	time	hh:mm 24 hr military time that observation of foraging bout began
TimeEnd	time	hh:mm 24 hr military time that observation of foraging bout ended
Divenum	integer	sequential number of dives within a bout (or within a subbout within a bout)
DT	number	time in seconds that focal otter is under water
ST	number	time in seconds that focal otter is on surface in between dives
Success	text	what is the outcome of the dive? y = yes, focal otter was successful in retrieving prey, n = otter did not retrieve prey, c = otter surfaced with prey carried over from an earlier dive that was not (completely) consumed, u = unknown success. Some projects also use "t" and "x" dives. NOTE: for carry-over dives (success "c"), a dive prior to the "c" dive **must** have success "y" and the **same prey item** that is listed on the carry-over dive. This is true even if there are multiple carry-over dives (e.g. if there is a "y" dive followed by 5 "c" dives, all should list the same prey type, size and number items).
Prey	text	3-letter code for prey item retrieved (code systems often vary across studies)
N_Items	integer	integer number of prey items retrieved during dive (though not necessarily eaten - see "Prop_Lost" field)
Size	integer	size category of prey item, relative to the sea otters paw size. Possible values are 1, 2, 3 or 4, where 1 mean <= 1 paw width, 2 = between 1 and 2 paw widths, 3 = between 2 and 3 paw widths, 4 = >3 paw widths (for larger prey use the "Est_cm" field to record). Size or (size + qualifier) are translated into millimeters by the SOFA analysis program
Qualifier	text	Letter code to record finer scale size info, effectively dividing each size class into 3 equal sub-divisions (a, b or c). For size-4 prey, you can also enter a "z", where "4z" indicates prey larger than 4 paw widths.
HT	number	handling time-- amount of time in seconds that an otter uses to handle and consume the prey item-- often equal to surface time. For dives with multiple prey types, the observer attempts to determine how the total handling time is partitioned between prey types and lists those values on the separate rows for each prey type. (e.g. if a dive resulted in two size 2a urchins and one size 3a crab, and the total handling time for all prey was 230s, then the observer might divide this into HT = 120 for the two urchins listed on the first row and HT = 110 for the crab listed on the second row).
Prop_Lost	number	proportion of prey not consumed by focal otter. Example: focal otter retrieves 1 clam and gives the entire clam to her pup, 1.0 would be entered here. If the focal otter retrieves 2 clams and gives 1 of them to her pup, 0.5 would be entered here.
How_Lost	text	If some proportion of prey not consumed, explanation of why: possible causes include to pup, dropped, stolen, not consumed, carried over
Est_kg	number	when prey is too large for established size categories, observer should estimate weight and/or size, and record here
Est_cm	number	when prey is too large for established size categories, observer should estimate weight and/or size, and record here

5. Run the SOFA program

NOTE: The best way to open RStudio is to double click “SOFA.Rproj” from the SOFA folder, which opens up RStudio with SOFA as the working directory.

Using the SOFA program to analyze foraging data actually consists of 3 separate steps, each of which is accomplished by “Sourcing” one of the R scripts from within the main SOFA folder. The steps are:

1). Import the data file and prepare the "Project" folder by Sourcing "**SOFA_prep.R**"
2). Fit the SOFA model to foraging data by Sourcing "**SOFA_run.R**" (or "**SOFA_run_cmdstan.R**")
3). Summarize the results of a model fit by Sourcing "**SOFA_sum.R**"

For each of these steps, you the user just need to open the script in Rstudio and click the "Source" button. Pop-up menus and information boxes will guide you through the rest. Details of each step are provided below in case you are interested (or run into issues).

Step 1: SOFA_prep

In the preparation stage you will be asked to select a spreadsheet containing the raw foraging data. **Make sure this spreadsheet has been formatted properly, as described above in section "4. Setup Foraging Data File"**, otherwise the data processing steps will not work and the script will crash. You will next be asked to choose a name for the Project: this should be a brief but descriptive name that will be easily associated with the data, e.g. "BC_Ccoast_2014-2020". Note that once your project is set up, you can use it to do multiple SOFA analyses (e.g. with different data grouping options or other settings), so the project name should be associated with the data but not with a specific analysis of those data (that comes later).

Once you've selected the raw data file and a project name, the script will create a project directory with the name you provided, and then process the data file and save a "cleaned" version ("Forage_dat.xlsx") into the project directory. The script will also copy into that directory a number of other xlsx files required for analysis (Sizeclass_key, Pawsz, Prey_Spcs, Mass_lng_dat, Energy_dat). You can edit these files as appropriate for your analysis (e.g. to set appropriate paw sizes for your population, or to remove mass-length data records from some regions). Finally, it will create a spreadsheet called "Prey_Key", **which you are required to edit before continuing on with the next step**. Once the SOFA_prep script is complete (you will be notified by a pop-up message), go to your project directory (it will be located within the "projects" folder found within the main SOFA folder) and open up the Prey_Key spreadsheet for editing.

Editing Prey_Key.xlsx: General instructions for editing this file are included as comments in cells of the top rows of each of the 4 worksheets – the instructions here essentially duplicate those comments.

The first worksheet (Prey_code_key) is used to associate prey codes from the data file with a more limited set of prey types. Column 1 (PreyCode) is a list of all the prey codes that appear in the raw data file. **These values should not be edited or deleted**, although you can insert new items in the case of prey that you wish to divide into 2 prey classes. Specifically, if there is a particular prey code you'd like to divide into small and large categories, simply insert a row immediately below the prey code in question, and add a new prey code consisting of same letters but with a "2" suffix (e.g. if "pry" is the code you want to split, the prey code in the new line will be "pry2") . Associate the "pry2" with separate prey type (e.g. "prey_L") in ColumnE, and separate proxies in columns 6 – 25 (see example prey key table below).

Column 2 of worksheet 1 (SzBrkmm) can be left blank except for prey codes that you are dividing into small and large categories. To divide a prey type into large and small, add a size breakpoint (e.g.70mm) into this field for the prey code you wish to split, and then insert a row immediately after with the new prey code (see previous paragraph, and sample prey key table below). Do not enter anything into this field for the new prey code you have inserted.

Table 2. Sample Prey Key, worksheet 1. The row with red font was inserted in order to split the "unknown clam" prey code (cla) into large and small clams, with the break point set at 70mm. We then defined prey

type “cla_L” for large clams, and set this as the prey type for the new prey code “cla2” (and set the proxy species as *Tresus nuttallii*). Other prey codes can also be associated with the “cla_L “ prey type, such as giant rick scallops. Note that barnacles and chitons both had too few observations to analyze separately, and so were combined into the “oth” prey type.

PreyCode	SzBrkmm	Description	N	PreyType	Prox1	Prox2
bas		barnacle, various species	1	**oth**	LOGI	
cag		graceful crab, Cancer gracilis	7	can	CAPR	MEMA
cam		dungeness crab, Cancer magister	145	can	MEMA	
can		cancer crab, Cancer sp.	38	can	CAPR	MEMA
cap		Cancer productus	183	can	CAPR	
chi		chiton, various species	5	**oth**	CRST	
cla	**70**	clam, unidentified, SMALL	303	cla	CLNU	LEST
cla2		**clam, unidentified, LARGE**		**cla_L**	**TRNU**	
cra		crab, unidentified	405	cra	PUPR	BLOC
crg		giant rock scallop, Crassadoma gigantea	1	**cla_L**	CRGI	

Column 3 of worksheet 1 (Description) is where you can add a description for each prey code if desired (taxa name or general descriptor).

Column 4 of worksheet 1 (“N”) provides the number of observations for this prey code in the data set - **do not edit this, it is for reference only**.

Column 5 of worksheet 1 is “PreyType”. **In this column you must select a prey type for each prey code**, which is done by selecting from a drop-down list. HOWEVER, note that first you will need to edit the list of possible prey types in the worksheet 2 (see below). In some cases, a prey code may correspond one-to-one with a prey type, but in most cases there will be several prey codes that share a single prey type (refer to sample prey key table, above). Note that you should limit the number of possible prey types such that each type is represented by a reasonable number of instances in the data set (a minimum of 25 is a good rule of thumb, at least 50 preferred): prey codes with few observations should be combined with other codes into a single prey type. **Select "UNID" for the prey code that corresponds to un-identified prey. For prey types that are "non-edible" (e.g. rock or shell), leave this field blank.**

Columns 6 – 25 of worksheet 1 are columns in which you can add one or more “proxy” prey species from the prey library for biomass and energy data. Select prey species from a drop-down list of species currently available in the prey library (see 4th worksheet for details of these codes). You should select the "matching" prey species if available, otherwise select one or more similar prey species as proxies. You can weight certain proxy species more heavily than others by entering its code in two columns.

Worksheet 2 contains the list of prey types to choose from: **note that you must create this list**. The first row has been created for you: this is the “UNID” prey type, which has associated TypeN value of 0 – **leave this row as-is** and start entering remaining prey types in row 2. In the 1st column, add a sequential integer for each prey type (1,2,… N, see Table 3 for example). In the 2nd column, enter a list of codes that represent the mutually exclusive prey types for analysis, with a unique code for each (3 or 4 letter codes are

preferred). You should limit the number of possible prey types such that each type is represented by a reasonable number of observations in the data set (refer to "N" column in previous worksheet: a minimum of 25 observations is a good rule of thumb, at least 50 preferred). Thus, prey codes with few observations should be represented by a single prey type.

The 3rd column of worksheet 2 is used to enter a description for each of your prey types. The 4th column of worksheet 2 is used to set the class for each prey type: these classes are selected from a drop-down list of 15 standardized prey classes (which are defined for you in worksheet 3). See sample prey type list, Table 3.

Table 3. Sample Prey Key, worksheet 2. Sample prey type list

TypeN	PreyType	Description	Class
0	UNID	UN-IDENTIFIED	
1	red	Red urchin	urchin
2	pur	Purple urchin	urchin
3	urc	Unidentified urchin	urchin
4	sna	Snails	snail
5	cra	Crabs	other_crab
6	aba	Abalone	abalone
7	lob	Lobster	lobster
8	oct	Octopus	cephalapod
9	cla	Small clams	clam
10	cla_L	Large clams	clam
11	oth	All other prey types	other_hardsub

Worksheets 3 and 4 are for reference only, do not edit. Worksheet 3 defines prey classes, and worksheet 4 provides a complete list of the prey species currently available from the prey library for biomass and energetic data: the first column describes the general prey category, the 2nd column is the 4 letter code used to refer to each species (these are the codes you select as proxy species in worksheet 1) and the 3rd column provides the full species name for each proxy species.

Step 2: SOFA_run

This script is used to set up and run SOFA, fitting the model to a data project that has been prepared by the SOFA_prep script. The script will first prompt you to select a project folder to work with. It will then prompt you to set 4 different parameter values that affect the model fitting:

i. The minimum dives/bout that a prey type has to be observed before data from that bout will be used for calculating mean prey size, mean handling time and mean consumption rate (increasing this value above 1 can reduce sample size and thus speed up processing time for very large data sets)
ii. The minimum dives/bout with successful prey captures that are required for a bout to be included in the estimation of effort allocation (increasing this value above 1 can reduce sample size and thus speed up processing time for very large data sets)
iii. Number posterior samples desired from the Bayesian fitting (10000 default). Reducing this value will speed up processing, increasing it will improve chain convergence.

iv. Number of burn-in samples per chain for Bayesian fitting (750 default). Reducing this value will speed up processing, increasing it will improve chain convergence.

Next the script will ask you whether you wish to analyze ALL the data in the data set, or just a sub-set. If you chose to analyze a sub-set, the script will provide you a drop-down list of different variables you can use to select data by (e.g. Area, Site, Period, etc.). It will then prompt you to select (from a list) one or more values of the selection variable (e.g. you might select 3 different sites from the 10 sites available in the data set), and only those data will be included in the rest of the analysis.

After a bit more data processing, the script will give you an option to review mass-length data plots for the prey codes in your data set – note that this can take a little time depending on how many prey codes you have. Next it will ask you whether you wish to use **by-groups' for analysis**: this is where one or more categorical variables in your data file are used to divide the data into groups, and stats are generated for each group level. If you choose to analyze using by-groups, the script will provide you a drop-down list of different variables you can use to group data by (e.g. Area, Site, Period, Ottername, etc.). Note that you can select more than one field to group by combinations of Variables (e.g. you could select Area and Period, to get stats for each Area, by Period). Make sure there are enough data in each group level for analysis.

The script will use your responses to the above questions to complete the data processing required for analysis. **If an error occurs, it is likely due to errors in the data set or Prey_key**, so go back and check that these spreadsheets have been formatted and set up properly (following instructions above). If data processing is successful, the script will then ask you to confirm you are ready to begin Bayesian data fitting (using rstan) – this fitting step can take several hours to complete, depending on the size of your data file and speed of your computer. Note that the progress will show in the "Viewer" window, on the right-hand side of the Rstudio screen. You can hit the "refresh" button periodically (the little circular arrow at top right) to see what % of the iterations have completed.

Once fitting is complete, the results will be automatically saved into the "results" sub-folder of your project folder. The results files have an "*.rdata" file extension, and the file name generated will reflect both the date and hour completed, as well as the type of analysis (whether ALL data, by-groups, or data sub-set).

Step 3: SOFA_sum

This script is run once an analysis has been completed in Step 2. The script will first prompt you to select a project folder, after which it will prompt you to select one of the available "*.rdata" results files for that project. Note that if you have not yet completed a model fitting (see Step 2), no results files will be available, and the script will be terminated (return to Step 2). Once you select a results file to summarize, the script will check to see if there were by-groups used for the analysis, and if so it will ask you to select up to 3 representative levels of the group variable for plotting (tabular results will be output for ALL group levels, but for some of the graphs it is impractical to plot more than 3 group levels). After that it will "render" an R markdown report that is customized for your results file: the result will be an html report (located in the Project directory) in which you can view the results of your model.

Enjoy!