This document describes the set of functions I’ve developed to read, edit, analyze, and write studbook files. This set of functions constitutes the beginnings of a package that I am developing to analyze studbook data in R. Currently, these functions are all housed within the “Rstudbooks\_YYYY\_MM\_DD.R” script.

I, Ms. Dr. Joseph L. Simonis, developed these functions and wrote the code in 2013 and 2014 while working at the Lincoln Park Zoo in Chicago, IL.

**Function List**

These are the functions so far put together in the script. Draft help files are on the pages following (outdated!!).

PackageLoad()

SBsetup()

overlayExamine()

overlayApply()

overlayMasterApply()

overlayEventApply()

overlaySexApply()

SBpedig()

SBinbreeding()

institutionWindow()

inOutInstitution()

dateWindow()

UDFexamine()

UDFapply()

**Function**

*Description*

*Usage*

*Arguments*

*Details*

*Value*

*Example*

*To-do*

**Function**

*Description*

*Usage*

*Arguments*

*Details*

*Value*

*Example*

*To-do*

**PackageLoad**

*Description*

This function loads all of the relevant packages for the ensuing analyses

*Usage*

PackageLoad(Packages = c("RODBC", "pedigree"))

*Arguments*

Packages: required packages for these analyses

*Details*

Should be run as PackageLoad()

*Value*

NULL

*Examples*

PackageLoad()

**SBsetup**

*Description*

This function grabs the studbook data from the SQL database

*Usage*

SBsetup(DBname, DBtablenames = c("Master", "Sex", "Event"), Overlay=TRUE)

*Arguments*

DBname: character of the name of the SQL database that houses the studbook

DBtablenames: character vector of the name(s) of the table(s) to grab from the SQL database

Overlay: logical for whether or not to grab the associated overlay tables

Packages: required packages for these analyses

*Details*

The only argument that needs to be controlled by the user right now is DBname.

Returns with nothing if not given a DBname

*Value*

Returns a list of database tables, named from the SQL database (so, currently Master, Sex, and Event) and their related overlay tables.

*Example*

SBsetup(DBname=“AORYX2013”)

**overlayExamine**

*Description*

Determines how many overlays exist in the studbook and parses the overlay tables as necessary.

*Usage*

overlayExamine(Studbook)

*Arguments*

Studbook: character, the name of the R studbook list that is the output from SBsetup

*Details*

Prints the number of overlays in the studbook to the console

*Value*

The overlay tables, split by overlay. The returned object is a list of overlays, each element of which is its own list that contains the overlay datatables.

*Example*

SB<- SBsetup(DBname=“AORYX2013”)

overlays<-overlayExamine(Studbook=SB)

overlays

**overlayApply**

*Description*

Applies the overlay indicated by the user to the studbook, returning the “analytical studbook”. Also allows user to strip off the overlay elements if they are not to be applied to the studbook (or if they are empty).

This function uses a suite of subfunctions to apply the overlay to each table (overlayMasterApply, overlayEventApply, and overlaySexApply).

*Usage*

overlayApply(Studbook, OverlayToUse)

*Arguments*

Studbook: character, the name of the R studbook list that is the output from SBsetup

OverlayToUse: Numeric, references the overlays as numbered by overlayExamine

*Details*

Applies the overlays to the raw studbook to create the analytical studbook. Specifics regarding each datatable are covered in the subfunction descriptions. Each table is also amended with the ‘IsHypothetical’ column, which allows downstream analyses to remove hypothetical animals and events from the data.

*Value*

The analytical studbook, which is the studbook (Master, Event, Sex tables) with the overlay applied

*Example*

SB<- SBsetup(DBname=“AORYX2013”)

overlays<-overlayExamine(Studbook=SB)

overlays

SBwithOL<-overlayApply(Studbook=SB, OverlayToUse=1)

**overlayMasterApply**

*Description*

Subfunction that applies the overlay to the Master table in the studbook as part of the overlayApply function. Shouldn’t be called on its own.

*Usage*

overlayMasterApply(Studbook, overlayUIDs\_toApply)

*Arguments* (all are passed from overlayApply)

Studbook: character, the name of the R studbook list that is the output from SBsetup

overlayUIDs\_toApply: the UID that’s related to the overlay the user indicated should be applied in the argument to overlayApply

*Details*

Altered entries are overwritten in the table as per the information in the overlay. New entries just get added on to the end of the current table. In addition, a column is added to indicate whether the entry is hypothetical (so that it can be removed as necessary from downstream analyses).

A warning is thrown if any individuals have more than one entry in the overlay master table. In these cases, just the first entry is used (see to-do).

*Value*

The Master table with the overlay applied (i.e. the “analytical” Master table)

*Example*

# Not run by itself

#

**overlayEventApply**

*Description*

Subfunction that applies the overlay to the Event table in the studbook as part of the overlayApply function. Shouldn’t be called on its own.

*Usage*

overlayEventApply(Studbook, overlayUIDs\_toApply)

*Arguments* (all are passed from overlayApply)

Studbook: character, the name of the R studbook list that is the output from SBsetup

overlayUIDs\_toApply: the UID that’s related to the overlay the user indicated should be applied in the argument to overlayApply

*Details*

Assumption now is that all entries are new (see “to-do”), so basically the new entries just get added on to the end of the current table. In addition, a column is added to indicate whether the entry is hypothetical (so that it can be removed as necessary from downstream analyses).

*Value*

The Event table with the overlay applied (i.e. the “analytical” Event table)

*Example*

# Not run by itself

#

**overlaySexApply**

*Description*

Subfunction that applies the overlay to the Sex table in the studbook as part of the overlayApply function. Shouldn’t be called on its own.

*Usage*

overlaySexApply(Studbook, overlayUIDs\_toApply)

*Arguments* (all are passed from overlayApply)

Studbook: character, the name of the R studbook list that is the output from SBsetup

overlayUIDs\_toApply: the UID that’s related to the overlay the user indicated should be applied in the argument to overlayApply

*Details*

Assumption now is that all entries are new (see “to-do”), so basically the new entries just get added on to the end of the current table. In addition, a column is added to indicate whether the entry is hypothetical (so that it can be removed as necessary from downstream analyses).

*Value*

The Sex table with the overlay applied (i.e. the “analytical” Sex table)

*Example*

# Not run by itself

#

**SBpedig**

*Description*

This function takes the studbook (typically with overlays applied if interested in using them), creates a pedigree table, and determines inbreeding values for each individual in the population.

*Usage*

SBpedig(Studbook)

*Arguments*

Studbook: the studbook to use

*Details*

The assumption I am using right now is that all MULTs and UNKs that the user wanted to replace with more specific (assumption-based) individuals were done in the overlay. Thus, I am replacing all MULT, UNK, and WILD parents with “NA” values, which is what the pedigree functions use for unknown/wild parents.

Sources functions from ‘pedigree’ package (orderPed and calcInbreeding) to do the actual work.

*Value*

List with two components: [1] the pedigree table (includes the hypothetical individuals and has all UNK, WILD, and MULT replaced with NA) and [2] a table of inbreeding values for each individual (hypothetical individuals are not included, but the rest of the studbook is, including animals outside of any institutional or time window).

*Example*

SB<- SBsetup(DBname=“AORYX2013”)

overlays<-overlayExamine(Studbook=SB)

SBwithOL<-overlayApply(Studbook = SB, OverlayToUse = 1)

Pedig<-SBpedig(Studbook = SBwithOL)

Pedig$Pedigree

Pedig$Inbreeding

**inbreedingHistory**

*Description*

This function produces historical summaries of the inbreeding in the population (looking at both the population as a whole and the new births every year). The institutional and time windows are applied.

This function uses the subfunctions InstWindow and inOut to apply the institutional window to the population.

*Usage*

inbreedingHistory (Studbook, InbreedingTable, SBcurrentnessDate, startingDate, institutions, ADDinstitutions = NULL, DROPinstitutions = NULL)

*Arguments*

Studbook: studbook to be used

InbreedingTable: a table of inbreeding values for the individuals in the population

SBcurrentnessDate: POSIX or character in “YYYY-MM-DD” format. currentness date of the studbook (default gets set to today’s date)

startingDate: POSIX or character in “YYYY-MM-DD” format. starting date of the analyses (default gets set to 1980-01-01)

institutions: character (can be multiple in a vector) of the name(s) of the .FED file(s) that are to be used for the institutional definition.

ADDinstitutions: character (can be multiple in a vector) of the name(s) of institutions that are to be added to the list of included institutions

DROPinstitutions: character (can be multiple in a vector) of the name(s) of institutions that are to be dropped from the list of included institutions

*Details*

Applies the institutional window on the population via InstWindow and inOut functions

Summarizes the inbreeding values of the new births/hatches for each year (all from Jan 1 to Dec 31 are combined for each year) and the population on every Jan 1.

*Value*

List of two tables: [1] inbreeding values of the births for every year and [2] inbreeding values of the population every year. These values are summarized (rather than given in full) by N, mean, standard deviation, min, and max. Those tables are also automatically written out to .csv files and graphed out to .tif files.

*Example*

SB<- SBsetup(DBname=“AORYX2013”)

overlays<-overlayExamine(Studbook=SB)

SBwithOL<-overlayApply(Studbook = SB, OverlayToUse = 1)

Pedig<-SBpedig(Studbook = SBwithOL)

inbreedingHist<-inbreedingHistory(Studbook = SBwithOL, InbreedingTable = Pedig, SBcurrentnessDate = "2013-04-30", startingDate = "1980-01-01", institutions = "AZA")

**infMort**

*Description*

This function determines if each individual in the studbook was in the population for the entire first X days of its life, and then whether or not it survived.

This function uses the subfunctions InstWindow and inOut to apply the institutional window to the population.

*Usage*

infMort(Studbook, mortWindow = 365, SBcurrentnessDate, startingDate, institutions, ADDinstitutions = NULL, DROPinstitutions = NULL)

*Arguments*

Studbook: studbook to be used

mortWindow: numeric. Number of days for calculating infant mortality (typically = 30 or 365). Defaults to 365.

SBcurrentnessDate: POSIX or character in “YYYY-MM-DD” format. currentness date of the studbook (default gets set to today’s date)

startingDate: POSIX or character in “YYYY-MM-DD” format. starting date of the analyses (default gets set to 1980-01-01)

institutions: character (can be multiple in a vector) of the name(s) of the .FED file(s) that are to be used for the institutional definition.

ADDinstitutions: character (can be multiple in a vector) of the name(s) of institutions that are to be added to the list of included institutions

DROPinstitutions: character (can be multiple in a vector) of the name(s) of institutions that are to be dropped from the list of included institutions

*Details*

Applies the institutional window on the population via InstWindow and inOut functions

Reasons an individual would be excluded from the mortality calculations:

1. No birth/hatch event is recorded
2. Birth/hatch was before the starting date for the analyses
3. The animal was born/hatched outside of the institutional window or moved outside of the institutional window within the first X days of its life (including LTF)
4. The animal has not yet been alive for X days
5. The animal is a hypothetical entry in the overlay

*Value*

Table with the following columns: StudbookID, Sex, InfantMort, ReasonforNA. InfantMort is a binary 0/1 as to whether the individual died in the first X days of its life (1) or survived (0). If an individual was excluded from InfantMort (i.e. it has an “NA”), the reasoning for that NA is given in ReasonforNA.

*Example*

SB<- SBsetup(DBname=“AORYX2013”)

overlays<-overlayExamine(Studbook=SB)

SBwithOL<-overlayApply(Studbook = SB, OverlayToUse = 1)

INFmort<-infMort(Studbook = SBwithOL, mortWindow = 365, SBcurrentnessDate = "2013-04-30", startingDate = "1980-01-01", institutions = "AZA")

INFmort

INFmort[-which(is.na(INFmort$InfantMort)),]

**institutionsList**

*Description*

Reads in the FED file(s) of interest and makes any amendments based on user definitions

*Usage*

institutionsList(institutions, ADDinstitutions=NULL, DROPinstitutions=NULL)

*Arguments*

institutions: character (can be multiple in a vector) of the name(s) of the .FED file(s) that are to be used for the institutional definition.

ADDinstitutions: character (can be multiple in a vector) of the name(s) of institutions that are to be added to the list of included institutions

DROPinstitutions: character (can be multiple in a vector) of the name(s) of institutions that are to be dropped from the list of included institutions

*Details*

The function grabs the .FED file(s) from the user’s PopLink/Federation Files folder. I’ve tried to write that grab so it works on any computer, but it might not yet.

*Value*

Character vector of institutions defining the population

*Example*

institutionsList(institutions=”AZA”)

institutionsList(institutions=c(“AZA”, “EAZA”))

**inOut**

*Description*

Applies the institutional window to a given data table with location information (typically the event table, but not limited), and amends the table with a logical vector column that indicates whether each entry’s location was in the population or not.

*Usage*

inOut(Table, institutionList )

*Arguments*

Table: single table from the studbook, must include location information

institutionList: character vector of institutions defining the population (likely as output from institutionList function, but not limited to such)

*Details*

Simple matching function.

*Value*

The data table with the logical column regarding whether the entry was in or out of the population.

*Example*

SB<- SBsetup(DBname=“AORYX2013”)

overlays<-overlayExamine(Studbook=SB)

SBwithOL<-overlayApply(Studbook = SB, OverlayToUse = 1)

inOut(SBwithOL$Event, institutionsList(institutions=”AZA”))

# compare to

SBwithOL$Event

**infMortSexSummary**

*Description*

Summarizes the infant mortality data into sex-specific rates, based on a cut-off F value, if the user so chooses.

*Usage*

infMortSummary (IMtable, InbreedingTable = NULL, cutoff = NULL, direction = NULL, writeOut=TRUE)

*Arguments*

IMtable: infant mortality table, as generated by infMort

InbreedingTable: a table of inbreeding values for the individuals in the population. Required only if the data are to be summarized based on the cutoff

cutoff: numerical value for comparing to individual F values (must be between 0 and 1, inclusive).

direction: The direction to be used for the cutoff/comparison. Character input of one of the following: “LessThan”, “LessThanOrEqualTo”, “GreaterThan”, “GreaterThanOrEqualTo”, “EqualTo”, or “NotEqualTo”. Required only if the data are to be summarized based on the cutoff

writeOut: logical for whether or not to write a .csv file of the data table

*Details*

It prints the cutoff used (or says whole population if no cutoff used) and the table.

It also saves the table to a .csv unless wireOut=FALSE

*Value*

Table of infant mortality rates (as defined by infMort) split out into different sexes (Male, Female, Unknown, Other, and Total)

*Example*

SB<- SBsetup(DBname=“AORYX2013”)

overlays<-overlayExamine(Studbook=SB)

SBwithOL<-overlayApply(Studbook = SB, OverlayToUse = 1)

INFmort<-infMort(Studbook = SBwithOL, mortWindow = 365, SBcurrentnessDate = "2013-04-30", startingDate = "1980-01-01", institutions = "AZA")

Pedig<-SBpedig(Studbook = SBwithOL)

eq0<-infMortSexSummary(IMtable = INFmort, InbreedingTable = Pedig$Inbreeding, cutoff = 0.0, direction = "EqualTo")

**infMortAnalysis**

*Description*

Runs statistical analyses and creates graphs for relationship between inbreeding and infant mortality

*Usage*

infMortAnalysis(IMtable, InbreedingTable)

*Arguments*

IMtable: infant mortality table, as generated by infMort

InbreedingTable: a table of inbreeding values for the individuals in the population

*Details*

This function conducts a set of logistic regressions on the relationship between inbreeding and infant mortality:

1. Infant Mortality ~ 1 (intercept-only model)
2. Infant Mortality ~ inbreeding
3. Infant Mortality ~ sex
4. Infant Mortality ~ inbreeding + sex
5. Infant Mortality ~ inbreeding \* sex

Currently, I am running a “crappy” version of model selection here, not based on AIC/LRT, but just whether or not terms are significant. Initially, the models are fit with Unknown and Other sex individuals removed. If either model 1 or model 2 is deemed the “best” of those five, then models 1 and 2 are re-run with the Unknown and Other individuals included.

The “take homes” from the “best” model are printed to the screen, also with the summary of the “best” model.

The function also writes the “best” model parameter table and the text description (the same that’s printed to the screen) out to .csv files and produces a graph of the relationship between mortality and inbreeding. The specifics of the graph (how many lines are fit, how points are distinguished, etc.) are determined by which model is the “best” fit.

*Value*

A list that has two components: [1] all of the model summaries (for all five models) and [2] a description of the best-fit model. **NOTE**: all the models in [1] might not be directly comparable (if model 1 or 2 was the best fit and there are Unknown or Other sex individuals in the population).

*Example*

SB<- SBsetup(DBname=“AORYX2013”)

overlays<-overlayExamine(Studbook=SB)

SBwithOL<-overlayApply(Studbook = SB, OverlayToUse = 1)

INFmort<-infMort(Studbook = SBwithOL, mortWindow = 365, SBcurrentnessDate = "2013-04-30", startingDate = "1980-01-01", institutions = "AZA")

Pedig<-SBpedig(Studbook = SBwithOL)

iMA<-infMortAnalysis(IMtable = INFmort, InbreedingTable = Pedig$Inbreeding)

iMA

**Function**

*Description*

*Usage*

*Arguments*

*Details*

*Value*

*Example*

*To-do*