Directions for Inseason Estimates for Taku River Sockeye Salmon Using BTSPAS

Sara Miller

2019-07-15

Reference for BTSPAS function:
Bonner, S. J. and Schwarz, C. J. (2019). BTSPAS: Bayesian Time Stratified Petersen Analysis System.R package version 2019.01.07.

Install/Update R: https://cran.r-project.org/

Install/Update R Studio: https://www.rstudio.com/

Install JAGS from: https://sourceforge.net/projects/mcmc-jags/files/latest/download.

Project Folder Setup

Set up your Rproject to include code, data, document, data/prior_weeks_data, data/*yyyy*_inseason, and output folders. Make sure the .gitignore file contains:

```
Taku*
*.csv
*.pdf
```

so that ouput and data from inseason estimates is not committed to GitHub. The code file that are necessary to run the analysis are the *inseason_analysis.R* file. The *inseason_analysis.R* file will also load some helper functions to structure the data for *BTSPAS* and to do the actual model fitting.

Data Setup

Three csv files should be updated weekly and placed in the data folder in the correct statistical week folder. Old versions of weekly data should be kept in a prior weekly folder *data/yyyy_inseason/SWxx*. For example, the files for the 2019 season statistical week 28 should be placed here: data/2019_inseason/SW28. The data files should be 'clean' data and free of NAs or missing information (i.e. missing recovery dates or tag id numbers).

Data files will be provided on a weekly basis from the tagging crews and DFO and should be placed in the data folder. These include:
1. release data,
2. recapture data, and
3. commercial catch data.

Release data should include the variable names:

- *Year*,
- *TagID*,
- *ReleaseDate*, and
- *ReleaseStatWeek* (which starts on Sunday).

Recapture data from DFO should include the variable names:

- *Year*,
- *TagID*,
- *RecoveryDate*, and
- *RecoveryStatWeek* (starts on Sunday).

Recovery type should only include those records with `RecoveryType="Commerical"` and `TagPrefix ="s"` for sockeye salmon. It is assumed that the recovery date matches a commerical opening. For example, if a tag is returned after the opening is closed, it is assumed to have occurred during the opening (which is usually in the first half of the week). This is important to allow the half week analysis to work properly. **The commercial catch data from DFO should include commercial catch including recoveries of tagged fish**. The file should contain the following variable names:

- *Year*,
- *Date*,
- *StatWeek*, and
- *CdnCommCt*.

The recovery type should only include commercial sockeye salmon catch. If the fishery was not open or there were zero catches, a '0' should be placed in the cell. All dates should in mm-dd-yyyy format.

Download the latest version of BTSPAS from the GitHub site at https://github.com/cschwarz-stat-sfu-ca/BTSPAS using `devtools::install_github("cschwarz-stat-sfu-ca/BTSPAS", dependencies = TRUE, build_vignettes = TRUE)`. This could take up to 20 minutes because the vignettes take a long time to compile. This only needs to be done once at the start of the season (unless an important update is made to the BTSPAS package.)

Analysis

All required packages need for the analysis are loaded prior to running models. These names of these pacakges are located at the top of the *inseason_analysis.R* file. If you are missing a package, you will need to install it from *CRAN* in the usual fashion.

The helper functions are sourced within the *inseason_analysis.R file*. The helper functions are:

- `BTSPAS_input()` = creates the data structures required for *BTSPAS* for releases, recoveries, and catch data The data structures include the
- *stratum index*,
- *n1* which is the number of tagged releases
- *m2* which is a matrix with columns representing recoveries in the same stratum as release, the next stratum of release etc.
- *u2* whick is the number of recoveries.
- `fit.BTSPAS()` which takes the input data structures and fits the basic time stratified analysis using the *BTSPAS* package.
- `fit.BTSPAS.dropout()` which fits the time stratified model but allowing for dropout/fallback.

Inseason Estimates for Taku River

The inputs that need to be specified for the code to run are the * stat weeks to include in the analysis, * the year, *stat weeks sub folder,* sw.randomseed, and *data.directory to store the results.

These are specified at the top of the *inseason_analysis.R* code in the code fragments:

```
fw.stat.weeks <- 23:28   # stat weeks with releases and recoveries to  be included
Year<-2019 # input year

sw.subfolder <- "SW28"
sw.randomseed <- 2328.
```

Select the stat weeks for which you want the BTSPAS to provide estimates on a FW and HW basis. Change the above code to account for which stat weeks of data will be used in the estimate. There should be a folder in the data folder for *yyyy*_inseason.

The Bayesian estimates will differ due to simulation uncertainty when generating the posterior distribution. Bayesian methods use Markov Chain Monte Carlo (MCMC) method which involve simulations from a random number generator. To "force" the random numbers to be the same across runs ... in the BTSPAS call, there is an argument InitialSeed=xxxxx where you can specify a "random" integer between 1 and 1000000 to force the same random numbers to be used across different run. In the code, the InitialSeed is set to sw.randomseed. For consistency across runs, use the random seed of the weeks that are being used in the estimate. For example, if the estimate pertains to statistical weeks 23 to 27, the random seed should be set to 2327.

After reading in the data and doing various merges, the *inseason_analysis.R* code will create a series of directories in the current workspace that will accumulate as you run the code each week. This code will compute the Full Week and Half week stratified with and without dropout. The output files will appear in the output folder as:

- "Taku-FullWeek-Inseason-Wxx-Wxx–YYYY,"
- "Taku-HalfWeek-Inseason-Wxx-Wxx–YYYY,"
- "Taku-FullWeek-Inseason-Wxx-Wxx-fallback–YYYY," and
- "Taku-HalfWeek-Inseason-Wxx-Wxx-fallback-YYYY,"

where x is the stat week numbers and $YYYY$ is the year.

If the Canadian commercial sockeye fishery does not commence until 2 or more weeks after sockeye are tagged in the fishwheels, then the `add.ones.at.start=TRUE` argument should be specified in the `fit.BTSPAS()` and `fit.BTSPAS.dropout()` functions to force the fitted curve to go to zero in the weeks when the fishery was not operating. If the `add.ones.at.start=TRUE` argument is used in the first weeks of estimates, it should be used the entire season (along with postseason). There are four places in the code where the argument needs to changed from a *FALSE* to *TRUE* (Be careful that *TRUE* is all uppercase.)

- `fit.BTSPAS(fw.data,prefix=fw.prefix, add.ones.at.start=FALSE, InitialSeed=sw.randomseed)`
- `fit.BTSPAS.dropout(fw.data,prefix=fw.prefix.dropout, n=50, dropout=11, add.ones.at.start=FALSE, InitialSeed=sw.randomseed)`
- `fit.BTSPAS(hw.data,prefix=hw.prefix, add.ones.at.start=FALSE, InitialSeed=sw.randomseed)`
- `fit.BTSPAS.dropout(hw.data,prefix=hw.prefix.dropout, n=50, dropout=11, add.ones.at.start=FALSE, InitialSeed=sw.randomseed)`

The code to implement the inseason estimates are not expanded by fishwheel runtiming.

The weekly inseason estimate should be the mean estimate from the file/folder '*Taku-FullWeek-Inseason-Wxx-Wxx-fallback–yyyy*' output to the file 'Taku-Inseason-W*xx*-W*xx–yyyy*Inseason-run.size.csv.

Dropout rate

The protocol for the 2019 season was to model the year-to-year variation in drop out (including sampling uncertainty) as binomial with $n = 50$ and $x = 11$ which gives mean $p_{dropout} = 0.22$ with a standard deviation of 0.05938262.