Practicalities w.r.t. running SAM

Anders Nielsen & Olav Nikolai Breivik

DTU-Aqua

an@aqua.dtu.dk

Three main ways to run SAM

- The site `http://www.stockassessment.org`
- The R-package `stockassessment`
- Using and modifying the code directly (github and all that)

Stockassessment.org: Background

- Scientific software is a way communicate ideas
- We to want enable others to benefit from our work
- Peer review process is important at all levels
- Too often expert groups are reduced to one person running it all, while others are reviewing results only.
- Problem is that it is a lot of work just 'setting up':
  - Getting the data together
  - Getting the data in the appropriate format
  - Finding a) the correct tools, b) what they depend on, c) for your platform, and d) in the correct versions
  - Figuring out how to make it all communicate

Stockassessment.org: Aim

Openness All interested should be able to:

- See all details of the implementation
- Run it themselves
- Experiment with data
- Experiment with assumptions

Collaboration The interface should support

- Several working same stock
- Privately working on a stock
- Publicly visible stocks

Traceability Should always be possible to see:

- What was changed
- Who did it

Accessibility Should be available on all platforms.

(quick demo)

Exercise - try the online way

- Upload the test data (or some other stock) and run the default run.
- Set the run to ‘baserun’ (button on result page)
- Change a setting and rerun
- Which configuration was best?

The stockassessment R-package

- The web-version predates the R package, but now the web-page also runs via the r-package
- Makes it easy to run on your own computer
- Fast to experiment with settings / model configurations
- Install currently with:
  `devtools::install_github("fishfollower/SAM/stockassessment")`

Trick to quickly change something in an online assessment

```
library(stockassessment)
fit <- fitfromweb("Herring-corrObs-update")  ## get the fit from online
fit2 <- fit                                  ## take copy
fit2$conf$fracMixObs <- c(0.05, 0.05, 0.05)  ## example change
fit2<-runwithout(fit2)                       ## reruns with change
modeltable(c(fit, fit2))                     ## compares models
#       log(L) #par      AIC
#M1 -499.6152   28 1055.230
#M2 -495.8728   28 1047.746
```

- The code above allows you to quickly make a change

Running from scratch

- Assume we have the usual data files, then the following setup would do a first run:

```
library(stockassessment)
cn <- read.ices("testdata/cn.dat")
cw <- read.ices("testdata/cw.dat")
dw <- read.ices("testdata/dw.dat")
lf <- read.ices("testdata/lf.dat")
lw <- read.ices("testdata/lw.dat")
mo <- read.ices("testdata/mo.dat")
nm <- read.ices("testdata/nm.dat")
pf <- read.ices("testdata/pf.dat")
pm <- read.ices("testdata/pm.dat")
sw <- read.ices("testdata/sw.dat")
surveys <- read.ices("testdata/survey.dat")

dat<-setup.sam.data(surveys=surveys, residual.fleet=cn, prop.mature=mo, stock.mean.weight=sw,
                    catch.mean.weight=cw, dis.mean.weight=dw, land.mean.weight=lw, prop.f=pf,
                    prop.m=pm, natural.mortality=nm, land.frac=lf)

conf <- defcon(dat)
par <- defpar(dat,conf)
fit <- sam.fit(dat,conf,par)
```

- Then when you have the first run you can save the model configuration in a configuration file:

```
saveConf(conf, file="model.cfg")
```

- Then you can manually edit the options in the configuration file
- Read in the updated configuration and re-run with:

```
newConf <- loadConf(dat, file="model.cfg")
newPar <- defpar(dat,newConf)
newFit <- sam.fit(dat,newConf,newPar)
```

- The graphs and tables from online can mostly be done with simple functions from R, e.g:

```
ssbplot(newFit)
fbarplot(newFit)
catchplot(newfit)
summary(newfit)
...
```

Exercise: Trying the R-package

- Install the R-package
- Try running the build-in example

```
example(sam.fit)
```

- Run the test data locally, and mimic the configuration you did online
- Try to compare two or more runs by something like:

```
fits <- c(old=fit1, new=fit2)
modeltable(fits)
plot(fits, addCI=TRUE)
```

- Try to do a forecast via the build in `forecast` function

Modifying the package and development

- How many already use github?
- Quick look at what is on `https://github.com/fishfollower/SAM`
- Demo of, clone, pull, branch, commit, push
- Compiling and testing
- Talk about forks and pull request
- Making a small change to the package