An online tool to easily run stock assessment models, using SS3 and SWO as an example

Anne-Elise Nieblas[1], Sylvain Bonhommeau[2], Taha Imzilen[3], Dan Fu[4], Fabio Fiorellato[4], Julien Barde[5]

[1]IRD, UMR MARBEC (IRD/Ifremer/Univ.Montpellier/CNRS), IRD Réunion, 97744 St Denis, La Réunion, France.

[2] IFREMER- DOI, rue Jean Bertho, 97822 LE PORT CEDEX, La Réunion, France.

[3] IRD, UMR MARBEC (IRD/Ifremer/Univ.Montpellier/CNRS), IRD Sète, 34200, France.

[4] IOTC, Le Chantier Mall, Victoria Mahé, Seychelles.

[5] IRD, UMR MARBEC (IRD/Ifremer/Univ.Montpellier/CNRS), IOC, Rue de l'Institut, Ebène, Maurice.

ABSTRACT

Stock assessment software are complex and advanced technical skills are required to develop the models. Producing output becomes time-intensive and even more complex as thousands of simulations must be run on supercomputers in order to include the multiple sources of uncertainty in assessment results. As few stock assessment participants have the specific technical skills required to reproduce these outputs, our aim has been to develop a Virtual Research Environment (VRE) that enables any user to easily parameterize, execute and edit online various steps of the stock assessment work flow using SS3 (a widely-used statistical catch-at-age model), with standardized data outputs. The VRE will provide various collaborative web services, including: (i) a workspace to share documents or data, (ii) webpages or an RStudio server to process data online, and (iii) an automated reporting service to dynamically generate documents to package the results. Here, we will show a mock-up of the VRE, using the last stock assessment of swordfish, provided by the IOTC, as an example. Using an R-shiny application, we explain the procedure of inputting new data, changing parameters, and quickly and automatically viewing the results of these changes on the graphical interface as well as through the automated reporting service. A collaborative environment such as the VRE uses simple tools to enable the storage and access of the data and source codes necessary to replicate past results or to try new parameterizations of the model. Increasing access to this complex model will bring more transparency and collaboration within working groups by providing "non-modelers" with a possibility to test hypotheses for the stock assessment. This will also increase the number of users of various levels of expertise: from experts, to managers, to even wider audiences with the potential applications of these tools to serious games.

INTRODUCTION

Several different types of stock assessment models are used to provide scientific advice to managers about exploited populations. Stock Synthesis 3 (SS3) is a statistical catch-at-age model that is used widely (Methot and Wetzel, 2013), including assessments for several stocks under the management of the Indian Ocean Tuna Commission (IOTC). SS3 is flexible in terms of data inputs and complexity, making it possible to run with data-poor stocks. It can use a diverse array of fishery and survey data, including both age and size structure of the population.

SS3 is based on ADMB C++ software that maximizes the goodness-of-fit of a set of parameter values, and then calculates the variance of these parameters using inverse Hessian and MCMC methods. This software is complex and advanced technical skills are required to develop the models. As such, the developers of SS3 have provided a Graphical User Interface (GUI) to aid in the set up and parameterization of complex assessment models (Methot, 2017). However, the production of outputs can still be time-intensive and complex when thousands of simulations are needed to include the multiple sources of uncertainty in the assessment results. Interactions with the results also tend to necessitate skilled language programming.

As few stock assessment participants have the specific technical skills required to reproduce these outputs, our aim is execute the entire IOTC SS3 stock assessment work flow online, using a Virtual Research Environment (VRE; Candela et al. 2013) on the H2020 BlueBridge infrastructure (European Union grant agreement No 675680). In collaboration with the IOTC and the FAO, IRD and IFREMER would like to develop this VRE to facilitate the parameterization, parallelization, and execution of various steps and the visualization of the results of SS3 to users with varying levels of expertise. We follow a similar approach as Imzilen et al. (2016), who developed a VRE based on Virtual Population Analysis of Atlantic bluefin tuna (*Thunnus thynnus*) used in the stock assessment work flow of the International Commission for the Conservation of Atlantic Tunas. The SS3 VRE will provide various collaborative web services, including: (i) a workspace to share documents, codes or data, (ii) webpages or an RStudio server to process data and codes online, (iii) visualisation services with an interactive interface to select model runs, and (iv) automated reporting to dynamically generate documents of results.

The first part of the work consists of testing the feasibility of reproducing past IOTC SS3 stock assessment models of tropical tunas and billfish (provided by the IOTC and related consultants) on the BlueBridge infrastructure. We then repackage the SS3 codes so that they can be parametrized, executed and edited online from a simple web page, with standardized data outputs. A collaborative environment such as the VRE uses simple interfaces to facilitate the storage and access of the data and source codes necessary to replicate past results or to explore new parametrizations of the model. As the VRE is a work in progress, we will present an offline R Shiny application to showcase the kind of automatic outputs that can be visualized. We will present an example of an automatic report created from an automated SS3 run. We encourage suggestions from the group on the parameters that are tested and changed most frequently during working groups, and the specific outputs that the group would like to visualize to investigate the model.

METHODS

SS3 model codes (Linux versions 3.24 and 3.3) were provided by the NOAA/SS3 team and were successfully compiled on the Linux-based Rstudio online of the BlueBridge infrastructure. A single run of the “simple” stock assessment model example (provided by the SS3 team and accessed from NOAA's SS3 virtual laboratory) was executed in 2.5 minutes. Currently, SS3 is available to researchers with an NOAA VLab account, but we have confirmed that it is acceptable to the developers and maintainers of SS3 that we make the software available to users in the format of the VRE.

Scenarios of use

Additionally, we tested several previous stock assessment models, provided by the IOTC and their consultants, including past assessments of swordfish (SWO, *Xiphias gladius*), bigeye tuna (BET; *Thunnus obesus*), and yellowfin tuna (YFT; *Thunnus albacares*). One run of each of these models on a personal computer takes about 1.5 hours with the Hessian (uncertainty calculations), or 20 minutes without the Hessian. On the Bluebridge infrastructure Rstudio online, these models take about 4.3 hours (with Hessian) and 1.1 hours (without Hessian). Models are commonly run without the Hessian (*pers. comm*. Adam Langley, IOTC consultant). Based on these run times, we identified various scenarios of use, and calculated the CPU resources they require (Table 1).

Table 1 : Scenarios of expected use of the VRE for the stock assessment. For one simulation of an IOTC example code on laptop: ~1.5 hours with uncertainty calculations, ~20 min without (complies with consultant’s methodology). For one simulation of an IOTC example on Rstudio online: 4.3 hours with uncertainty calculations, ~1.1 hours without.

ID	Summary	CPUs required:
Scenario 1	A consultant developing a model and running sensitivity analyses before the stock assessment (10,000 iterations). Results required within 1 day for a total of 3 days (i.e., allowing for 3 major modifications to the model).	(20*10000)/(3*24*60) = ~47 CPUs
Scenario 2	A consultant making modifications on the model during the stock assessment (1,000 iterations). Results required in 1 hour for 1 day of the meeting.	(20*1000)/60 = ~334 CPUs
Scenario 3	Meeting participants individually exploring parameters	
Scenario 3a	One simulation per user, approximately 30 users. Available for the full duration of the meeting, e.g. 5 days.	30 CPUs available throughout meeting
Scenario 3b	Each user allowed 10 iterations. Results required immediately (i.e., duration of single run). Schedule could be specified for a period of time within a single day.	30*10 = 300 CPUs

Work flow

The four files that are required to define and run the SS3 model (Methot, 2013): starter.ss, forecast.ss, control.ctl, and data.dat were developed into R functions: writeStarter.R, writeForecast.R, writeControl.R, and writeDat.R, respectively. These functions automatically incorporate inputs from different species models. We used the 'simple' example, and past stock assessment codes for SWO, BET, and YFT to test the functions. A wrapper function, ss3.R, reads the inputs from a .csv table, within which input data tables are nested (see Appendix I for an example of the input files for SWO), and inserts these into the 'write' functions. The model executable is then run on the four files that are produced. The ss3.R function then calls the SS_output function of the r4ss R package (Taylor et al., 2011) to read the outputs of the run. The results are given as list, which is then converted from the .Rdata format to a standardized netcdf format. Netcdf is a widely-used format that allows data from multiple runs to be archived to an open source Thredds data server, such that assessment results can be accessed remotely and reproduced in the future.

Based on expert advice (*pers. comm.* Dan Fu and Sylvain Bonhommeau), we currently restructure the output to include information necessary to create the Kobe stock status plot (i.e., B.Bmsy, and F.Fmsy), time series of biomass in absolute numbers and relative to B0, recruitment variations, catch per unit effort, catch-at-age, movement data, and information on selectivity (see Appendix 2 for complete list of outputs).

This work flow can be executed by running these functions in the R environment through the BlueBridge Rstudio online infrastructure, or from Web Forms to allow users to focus on parameterization of the runs rather than interaction with the programming language. The single-run structure will be converted to a web processing service to enable individual users to parameterize aspects of the model. This WPS version will be parallelized within the FAO supercomputing resources, allowing multiple iterations to be executed simultaneously, and thus incorporating uncertainty.

We will package and display the results of the SS3 runs using an Rshiny application within the VRE with output visualizations of each model run to select the best models for retrospective analysis, and then projections. Final results will be integrated into an automatically generated report of the stock assessment, using R Markdown. This automated report will allow users to quickly update the stock assessment and incorporate the scientific advice given during the meeting.

Online collaborative environment

The BlueBridge project enables an online collaborative environment by providing the infrastructure necessary to parameterize, visualize, and access a work flow such as this. This environment will be available to a list of members who can share documents, messages, data and codes in both a public and private environment. An Rstudio server will be incorporated into the enviroment that acts exactly as a desktop application, with a private workspace for each member but that is configured to ensure that codes compile correctly. For users without experience in R, we will package these codes through a Web Processing Service, which allows the codes to run directly from a web page in the VRE, and users can focus on parameterization of the model instead of programming.

RESULTS AND DISCUSSION

At the WPB15, we will present examples of the work flow by executing a single run. We will show examples of the R Shiny application output displays, as well as an example of the generation of an automatic report. The end goal of the SS3 VRE is to allow participants of stock assessments to use the VRE interactively at working groups in order to explore input parameters and results, to store and replicate past results, to give more transparency to the decision-making process, and to enhance collaboration within working groups. Improving the ease of use of this complex model will bring more transparency and collaboration within working groups by providing "non-modelers" with a possibility to test hypotheses and explore uncertainty for stock assessment. Technical performance, document production, and harmonization of content are expected to be enhanced due to this increased participation. We hope to show the potential of this environment to foster collaboration and incorporation of scientific advice within working groups. We particularly encourage feedback on these tools and their application from the community of users to improve their utility in the future.

APPENDICES

Appendix 1 : An example of the input table, using SWO's 2015 stock assessment inputs.

STARTER.SS		
variables	SWO	
version	3.21	
filename	../test_SWO/starter.ss	
comments	starter.ss made using writeSS324Starter.R	
model	SWO	
init_vals		0
display_deets		1
age_str_rep		1
checkup		0
param_trace		0
cum_report		1
full_priors		0
soft_bounds		1
num_data_out		1
turn_off_est		10
MCMCburn		10
MCMCthin		2
jitter		0
sdrep_start		-1
sdrep_end		-2
n_sd_yrs		0
sd_yr_vector	#	
final_conv	0.0001	
retro_yr		0
minage_sumbio		1
dep_basis		1
frac_depden	0.25	
SPR_rep_basis		4
F_rep_units		1
F4_age_range		
F_rep_basis		0
endfile_val		999

FORECAST.SS		
variable	SWO	
version	3.24	
comments	forecast.ss made using writeSS324Forecast.R	
filename	../test_SWO/forecast.ss	
benchmarks		1
MSY		2
first_yr_avg_recF		
end_yr_avg_recF		
F_mult		
SPR_target	0.4	
biomass_target	0.4	
bmark_yrs	rep(0,6)	

bmark_relF_basis		2
forecast		2
n_forecast_yrs		10
F_scalar	0.2	
Fcast_yrs	c(2015,2015,0,0)	
control_rule		0
control_rule_uplim	0.4	
control_rule_lowlim	0.1	
control_rule_buff	0.75	
n_fcast_loops		3
first_fcast_loop		3
fcast_loop_ctl3		0
fcast_loop_ctl4		0
fcast_loop_ctl5		0
first_yr_capsall		2500
imp_err	0.	
rebuilder		0
rebuilder_Ydecl		1999
rebuilder_Yinit		-1
fleet_relF		1
fcast_catch_basis		2
fishery_names	FISHERY1 FISHERY2 FISHERY3	
fishseas_F		1
max_total_catch_fleet	rbind(rep(-1,4),rep(-1,4),rep(-1,4))	
max_total_catch_area	rep(-1,4)	
fleet_assignment	rbind(rep(0,4),rep(0,4),rep(0,4))	
allocation_fractions		
n_forecast_catch		0
Fcast_basis		2
endfile_val		999

	SWO.CTL	
Variable	SWO	
version	3.21e	
model	SWO	
n_growthpatterns		1
n_submorphs		1
submorph_growthvar		
morph_dist		
n_seasons_peryear		1
n_areas		4
n_recrassign		4
recr_inter		0
recr_assign_tab	read.csv('../input/recr_assign_tab_SWO.csv',sep=',',header=F)	
n_move_defs		8
age_firstmove	0.6	
movement_def	read.csv('../input/movement_def_SWO.csv',sep=',',header=F)	

n_block_patterns	0
n_blocks_per_pattern	
yr_range_Nblockpatterns	
frac_female	0.5
natM_opt	0
n_brk_pts	
age_brk_pts	
Lorenzen_ref_age	
agespec_M	
growth_mod	1
growth_amin	0.01
growth_amax	999
agespec_K_amin	
agespec_K_amax	
sd_add_to_laa	0
cv_patt	0
mat_opt	3
agespec_MatFec	read.csv('../input/agespec_MatFec_SWO.csv',sep=',',header=F)
lengthspec_Mat	
first_mat_age	1
fec_opt	1
hermaph_opt	0
hermaph_seas	
include_males	
offset_method	1
time_var_adj	1
growth_param	read.csv('../input/growth_param_SWO.csv',sep=',',header=F)
movement_param	read.csv('../input/movement_param_SWO.csv',sep=',',header=F)
mg_env	0
mg_env_param	
mg_block	0
mg_block_param	
param_seasonality	rep(0,10)
seasonal_param	
mg_ann_dev_phase	4
sr_fun	3
sr_param	read.csv('../input/sr_param_SWO.csv',sep=',',header=F)
sr_env_link	0
sr_env_target	0
do_recr_dev	1
recr_dev_begin_yr	1950
recr_dev_end_yr	2013
recr_dev_phase	6
adv_opt	1
recr_dev_early_start	0
recr_dev_early_phase	-5
forecast_recr_phase	5
lambda_fcast_recr	1

last_yr_nobias		1970
first_yr_nobias		1971
last_yr_fullbias		2001
first_rec_nobias		2002
max_bias_adj		1
recr_cycle_period		0
min_recr_dev		-6
max_recr_dev		6
n_recr_devs		0
n_recr_cycles		
recr_cycle_param		
recr_dev		
f_ballpark	0.2	
f_ballpark_yr		2003
f_method		3
max_f		4
f_start_val		
f_phase		
n_f_inputs		
n_tuning		2
f_param		
initF_param	read.csv('../input/initF_param_SWO.csv',sep=',',header=F)	
Q_tab	read.csv('../input/q_tab_SWO.csv',sep=',',header=F)	
Q_param	read.csv('../input/q_param_SWO.csv',sep=',',header=F)	
size_select_tab	read.csv('../input/size_select_tab_SWO.csv',sep=',',header=F)	
age_select_tab	read.csv('../input/age_select_tab_SWO.csv',sep=',',header=F)	
size_select_param	read.csv('../input/size_select_param_SWO.csv',sep=',',header=F)	
age_select_param		
do_tag		0
tag_param		
tag_param_rep		
tag_param_decay		
var_adj_factor		1
var_adj_tab	read.csv('../input/var_adj_tab_SWO.csv',sep=',',header=F)	
max_lambda_phase		4
sd_offset		1
n_changes_lambda		44
like_comp_tab	read.csv('../input/like_comp_tab_SWO.csv',sep=',',header=F)	
read_specs		0
var_control		
selex_std_bin		
growth_std_bin		
NatAge_std_bin		
endfile_val		999

SWO.DAT		
Variable	SWO	
model	SWO	
version	3.24	
start_yr		1950
end_yr		2015
n_seasons_peryear		1
n_months_perseason		12
spawning_season		1
n_fleets		12
n_surveys		10
n_areas		4
fishsurvey_names	GI_NE%LL_NE%GI_NW%LL_NW%GI_SE%LL_SE %ALGI_SW%EUEL_SW%ISEL_SW%JPLL_SW %TWFL_SW%TWLL_SW%UJPLL_NW%UJPLL_NE %UJPLL_SW%UJPLL_SE%UTWLL_NW%UTWLL_NE %UTWLL_SW%UTWLL_SE%UPOR_SW%UESP_SW	
sample_timing	rep(0.5,22)	
fleet_area	c(2,2,1,1,4,4,rep(3,6),1,2,3,4,1,2,3,4,3,3)	
catch_units	rep(1,12)	
se_logcatch	rep(0.01,12)	
n_genders		2
n_ages		30
init_equil_catch	rep(0,12)	
n_catch_obs		66
catch=catch.csv	read.csv('../input/catch_SWO.csv',sep=',',header=F)	
n_cpue_obs		285
cpue_units=cpue_units.csv	read.csv('../input/cpue_units_SWO.csv',sep=',',header=F)	
cpue=cpue.csv	read.csv('../input/cpue_SWO.csv',sep=',',header=F)	
n_fleets_w_discards		0
n_discard_obs		0
discard_units=discard_units.csv		
discards=discards.csv		
n_mnbodywt_obs		0
df_mnbodywt		30
mnbodywt=mnbodywt.csv		
lengthbin_method		1
binwidth		
pop_minsize		
pop_maxsize		
n_popbins		
lowedge_popbin		
comp_tail		0
add_to_comp	0.01	
bin_to_combine_genders		0
n_lengthbins		25
lowedge_lenbin	c(15,45,54,63,72,81,90,99,108,117,126,135,144,153,162,171,180,189,198,207,216,225,234,243,252)	
n_length_obs		274
length_comp=length_comp.csv	read.csv('../input/length_comp_SWO.csv',sep=',',header=F)	
n_agebins		0

n_age_obs	0
Lbin_method	0
agebin_combine_genders	0
age_comp=age_comp.csv	
n_mnsizeage_obs	0
mn_size_at_age=mn_size_at_age.csv	
n_envvar	0
n_env_obs	0
env_data=env_data.csv	
n_sizefreq_methods	0
nbins_per_method	
sizefreq_units	
sizefreq_scale	
sizefreq_mincomp	
n_sizefreq_obs	
lowedge_sizefreq_bins	
sizefreq=sizefreq.csv	
do_tags	0
n_tag_groups	
n_recap_events	
mix_period	
max_tracking	
release=release.csv	
recapture=recapture.csv	
do_morphcomp	0
n_stockcomp	
n_stocks	
mincomp	
stockcomp=stockcomp.csv	
endfile value	999

Appendix 2: Outputs variables and their respective dimensions that are currently transformed into netcdf from the .Rdata as generated by the SS_output function of the r4ss R package.

Variable	Dimensions
	KOBE
B.Bmsy	YEAR
F.Fmsy	YEAR
CPUE	
Vuln_bio	FLEET, YEAR, SEASON
Obs	FLEET, YEAR, SEASON
Exp	FLEET, YEAR, SEASON
Calc_Q	FLEET, YEAR, SEASON
Eff_Q	FLEET, YEAR, SEASON
SE	FLEET, YEAR, SEASON
Dev	FLEET, YEAR, SEASON
Like	FLEET, YEAR, SEASON
Likelogs	FLEET, YEAR, SEASON
Supr_Per	FLEET, YEAR, SEASON
	NATAGE
natage	AREA, YEAR, SEASON, TIME, BEGMID, ERA, BIO_PATTERN, GENDER, BIRTHSEAS, MORPH, SUBMORPH, AGE
	MEAN_BODY_WT
meanbodywt	MBW.YEAR, SEASON, MORPH,AGE
	SPR SERIES
Bio_all_spr	YEAR
Bio_Smry_spr	YEAR
SPBzero	YEAR
SPBfished	YEAR
SPBfished /R	YEAR
SPR	YEAR
SPR_std	YEAR
Y/R	YEAR
GenTime	YEAR
F_std	YEAR
Num_Smry	YEAR
MnAge_Smry	YEAR
Enc_Catch	YEAR
Dead_Catch	YEAR
Retain_Catch	YEAR
Enc_Catch_B	YEAR
Dead_Catch_B	YEAR
Retain_Catch_B	YEAR
Enc_Catch_N	YEAR
Dead_Catch_N	YEAR
Retain_Catch_N	YEAR
MnAge_Catch	YEAR
SPB	YEAR
Recruits	YEAR
Tot_Exploit	YEAR
More_F(by_Morph)	YEAR
sum_Apical_F	YEAR
F=Z-M	YEAR
spr	YEAR
aveF	YEAR, SPRF

maxF	YEAR, SPRF
M-AT-AGE	
M-at-age	YEAR, BIO_PATTERN, GENDER, AGE
Z-AT-AGE	
Z-at-age	YEAR, BIO_PATTERN, GENDER, AGE
CATCH-AT-AGE	
catage	AREA, FLEET, YEAR, SEASON, ERA, GENDER, MORPH, AGE
GROWTH SERIES	
growthseries	GS.YEAR, SEASON, BEGMID, MORPH, AGE
MOVEMENT	
movement	SEASON, G_PATTERN, SOURCE_AREA, DEST_AREA, MIN_AGE, MAX_AGE, AGE
AGESELEX	
ageselex	FLEET, YEAR, SEASON, GENDER, MORPH, age.FACTOR, AGE
SIZESELEX	
sizeselex	FLEET, YEAR, GENDER, size.FACTOR, SIZESELEX.SIZE
TIME SERIES	
Bio_all_ts	AREA, YEAR, SEASON, ERA
Bio_smry_ts	AREA, YEAR, SEASON, ERA
SpawnBio	AREA, YEAR, SEASON, ERA
Recruit_0	AREA, YEAR, SEASON, ERA
Spbio_GP	AREA, YEAR, SEASON, ERA
SPB_vir_LH	AREA, YEAR, SEASON, ERA
SmryBio_SX_GP<-	AREA, YEAR, SEASON, ERA, TS.GP
SmryNum_SX_GP<-	AREA, YEAR, SEASON, ERA, TS.GP
sel_B_ts	AREA, YEAR, SEASON, ERA, TS
dead_B_ts	AREA, YEAR, SEASON, ERA, TS
retain_B_ts	AREA, YEAR, SEASON, ERA, TS
sel_N_ts	AREA, YEAR, SEASON, ERA, TS
dead_N_ts	AREA, YEAR, SEASON, ERA, TS
retain_N_ts	AREA, YEAR, SEASON, ERA, TS
obs_cat_ts	AREA, YEAR, SEASON, ERA, TS
F_ts	AREA, YEAR, SEASON, ERA, TS

ACKNOWLEDGEMENTS

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the BlueBRIDGE project (Grant agreement No 675680).

REFERENCES

Candela, L., Castelli, D. & Pagano, P., (2013). Virtual Research Environments: An Overview and a Research Agenda. Data Science Journal. 12, pp.GRDI75–GRDI81. DOI: http://doi.org/10.2481/dsj.GRDI-013

R.D. Methot Jr., User Manual for Stock Synthesis: Model Version 3.24s (2013) 152 pp.

R.D. Methot Jr., User Manual for Stock Synthesis: Model Version 3.30 (2017) 181 pp.

R.D. Methot Jr., C.R. Wetzel, Stock synthesis: A biological and statistical framework for fish stock assessment and fishery management, Fisheries Research 142 (2013) 86– 99.

T. Imzilen, S. Bonhommeau, T. Rouyer, L. T. Kell, J. Barde, Online collaborative environment to run the eastern bluefin tuna stock assessment workflow, Collect. Vol. Sci. Pap. ICCAT, 73(7) (2017) 2528-2534.

Taylor, I.G., Stewart, I.J., Hicks, A., Garrison, T.M., Punt, A.E., Wallace, J.R., Wetzel, C.R., r4ss: R code for Stock Synthesis. R package version 1.16. (2011) http://R-Forge.R-project.org/projects/r4ss/.