An online tool to easily run stock assessment models, using SS3 and YFT 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; anneelise.nieblas@ird.fr.
[2] IFREMER- DOI, rue Jean Bertho, 97822 LE PORT CEDEX, La Réunion, France.
[3] IRD - UMR MARBEC 248, Av. Jean Monnet, 34200 Sète, 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 super-computers 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. 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. 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 illustrate the stock assessment work flow through the VRE, using the last stock assessment of yellowfin, provided by the IOTC, as an example. Currently, we are also able to replicate past stock assessments of swordfish and bigeye tuna, and theoretically, any species model using SS3 can be similarly replicated. 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.

KEYWORDS: Indian Ocean, large pelagic fish, yellowfin tuna, *Thunnus albacares*, scientific cloud, stock assessment, grid computing, online processing.

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 parametrization 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 parametrization, parallelization, and execution of various steps of the stock assessment and the visualization of the results of SS3 to users with varying levels of expertise.

We follow a similar approach as [Imzilen et al., 2017], 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) visualization services with an interactive interface to select model runs, and (iv) automated reporting to dynamically generate documents of results (Figure 1; please visit the current IOTC SS3 VRE home page for more information and to join).

The first part of the work consisted of testing the feasibility of reproducing past IOTC SS3 stock assessment models of tropical tunas and billfish (providedby 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.

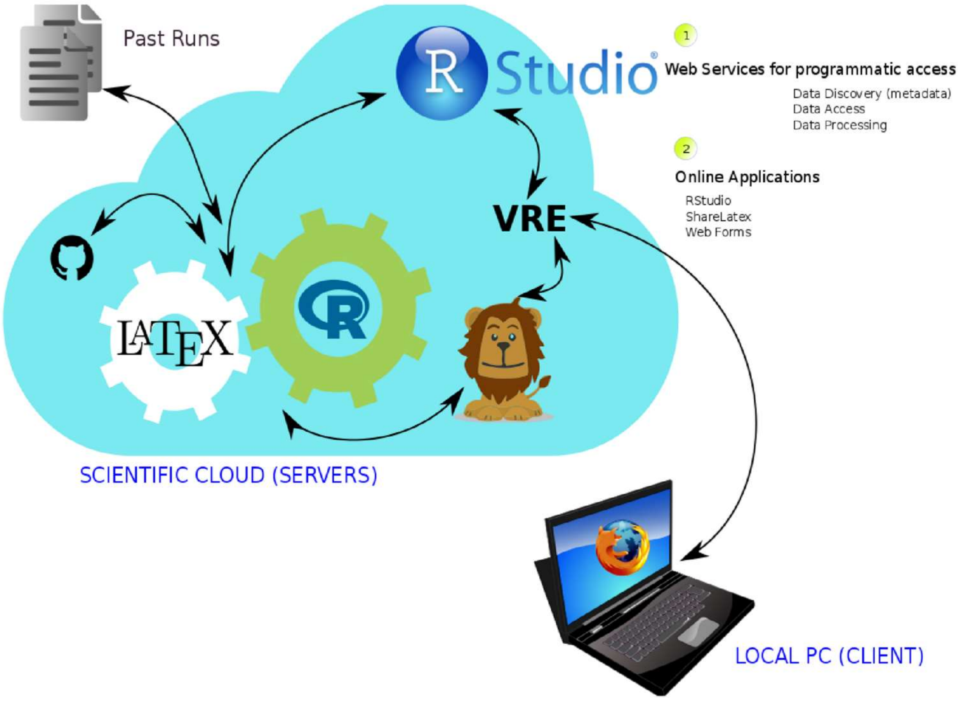


Figure 1: Overview of VRE services. The user (client) accesses the scientific cloud server of the VRE from their local PC. The VRE uses past model parameters and data inputs to inform and compile the SS3 model for the specified species. Web services are provided for programmatic access, such as data discovery, access, and processing. Online applications are available and include Web Forms to parametrize models without requiring advanced coding skills; Rstudio online, which can act similarly to a 'local' Rstudio to launch and process codes and investigate results, but gives access to cloud services; and Sharelatex for automatic reporting of the results.

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 (Candela et al., 2015). Currently, SS3 is available toresearchers with an NOAA VLab account (visit the NOAA VLab website to request access), 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 (pers. comm., R. Methot). While both versions of SS3 were successfully compiled, we currently use version 3.24 to replicate the model examples to which we have access. We plan to make the updated SS version 3.3 available as it becomes required by the user community.

Scenarios of use

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 15 seconds. 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). Models are run without the Hessian uncertainty calculations, as is common practice (pers. comm. Adam Langley, IOTC consultant). One run of each of these models on a personal computer varies considerably (Table 1). Based on these run times, we identified various scenarios of use, and calculated the CPU resources they require (Table 2).

Table 1: Minutes required for each of the SS3 models currently available as run (without Hessian uncertainty calculations) from a personal laptop or from the Rstudio online available on the VRE.

Model	Laptop	Rstudio online
simple	0.2	0.3
SWO	2.8	2.9
BET	23.4	27.8
YFT	0.4	0.5

Table 2: Scenarios of expected use of the VRE for the stock assessment, using YFT run time as an example. For one simulation of an IOTC YFT example code takes about 30 seconds to run on the Rstudio online infrastructure.

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).	(0.5*10000)/(3*24*60) = ~2 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.	(0.5*1000)/60 = ~8 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 work flow currently follows four main steps (Figure 2). The first step is to define the specifications and simulations desired for the model run(s). SS3 requires four files to run the model [Methot, 2013]: starter.ss, forecast.ss, control.ctl, and data.dat. We developed these four files into the R functions: writeStarter(), writeForecast(), writeControl(), and writeDat(), respectively, such that the parameters and data can be input simply and the files can be compiled automatically, saving time and avoiding error. A wrapper function, ss3.R, readsthe inputs from .csv input tables, within which input data tables are nested (see Appendix I for an example of the input files for YFT), and inserts these parameters and data into the write*() functions. The model .ss executable is then run on the four files that are produced. The ss3.R wrapper then uses the SS output() function of the r4ss R package [Taylor et al., 2011] to read the outputs of the model run. The results are given as list, which is then converted from the .Rdata format to a standardized netcdf format (Figure 3). Netcdf is a widely-used format that allows data from multiple runs to be archived to an open source Thredds data server (not yet established), such that assessment results can be accessed remotely and reproduced in any computing environment in the future.

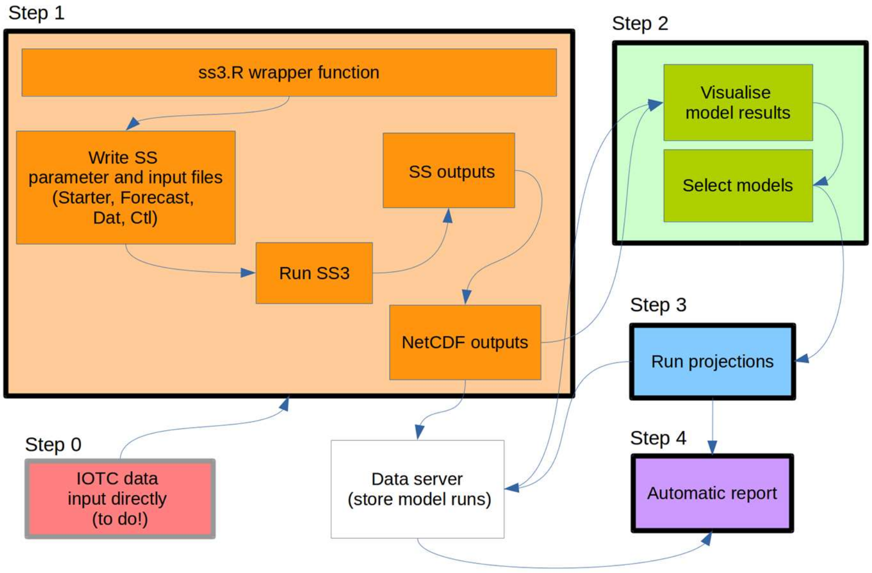


Figure 2: Work flow of the SS3 VRE outlined in four main steps. Step 1 consists of inputting data, parametrizing, and running the model. SS outputs are transformed to NetCDF and will be stored on a data server. Step 2 allows users to visualize the results of the model run(s) and select the best case model from which to make projections in Step 3. Projections will also be stored on the data server. Step 4 compiles the results of the work flow into an automatic report. Step 0, yet to be implemented, will automatically input new data directly from the IOTC database into the model at Step 1.

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 (Figure 3; see Appendix 2 for complete list of outputs).

SS model results are read by the SS output() function of the r4ss R package [Taylor et al., 2011], which gives a list of multiple data frames (e.g., outputs SS3). The dimensions of the data frames within this list are then defined, and variables are put into a netcdf file. The standardized netcdf format will then be saved on a Thredds data server. This step of the 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 parametrization 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 parametrize 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.

In Step 2, we package and display the results of the SS3 runs using a Shiny Proxy application within the VRE with output visualizations of each model run. This allows the user to select the best model(s) for retrospective analysis, and then projections (Step 3). The results of the retrospective analyses and theprojections will stored on the Thredds data server. Step 4 integrates the final results

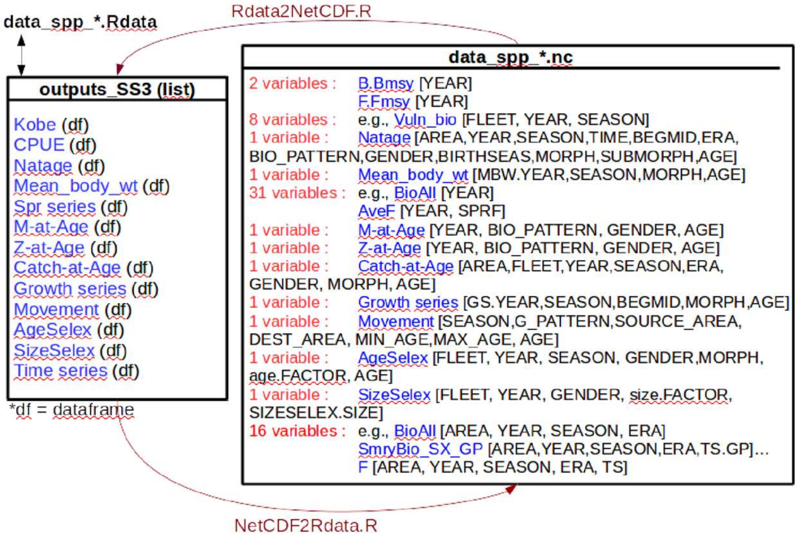

data_spp_*.Rdata
Rdata2NetCDF.R
outputs_SS3 (list)
Kobe (df)
CPUE (df)
Natage (df)
Mean_body_wt (df)
Spr series (df)
M-at-Age (df)
Z-at-Age (df)
Catch-at-Age (df)
Growth series (df)
Movement (df)
AgeSelex (df)
SizeSelex (df)
Time series (df)
*df = dataframe
data_spp_*.nc
2 variables : B.Bmsy [YEAR]
F.Fmsy [YEAR]
8 variables : e.g., Vuln_bio [FLEET, YEAR, SEASON]
1 variable : Natage [AREA,YEAR,SEASON,TIME,BEGMID,ERA, BIO_PATTERN,GENDER,BIRTHSEAS,MORPH,SUBMORPH,AGE]
1 variable : Mean_body_wt [MBW.YEAR,SEASON,MORPH,AGE]
31 variables : e.g., BioAll [YEAR]
AveF [YEAR, SPRF]
1 variable : M-at-Age [YEAR, BIO_PATTERN, GENDER, AGE]
1 variable : Z-at-Age [YEAR, BIO_PATTERN, GENDER, AGE]
1 variable : Catch-at-Age [AREA,FLEET,YEAR,SEASON,ERA, GENDER, MORPH, AGE]
1 variable : Growth series [GS.YEAR,SEASON,BEGMID,MORPH,AGE]
1 variable : Movement [SEASON,G_PATTERN,SOURCE_AREA, DEST_AREA, MIN_AGE,MAX_AGE, AGE]
1 variable : AgeSelex [FLEET, YEAR, SEASON, GENDER,MORPH, age.FACTOR, AGE]
1 variable : SizeSelex [FLEET, YEAR, GENDER, size.FACTOR, SIZESELEX.SIZE]
16 variables : e.g., BioAll [AREA, YEAR, SEASON, ERA]
SmryBio_SX_GP [AREA,YEAR,SEASON,ERA,TS.GP]...
F [AREA, YEAR, SEASON, ERA, TS]
NetCDF2Rdata.R

Figure 3: Examples of the SS output of the model run that are converted to a standardized netcdf format.

of the work flow into an automatically generated report of the stock assessment, using R and Latex. This automated report will allow users to quickly update the stock assessment and incorporate the scientific advice given during the meeting. ShareLatex will be available on the VRE to compile both the R and Latex codes on the same R server, thus ensuring that codes are compiled in the same environment, facilitating a collaborative editing process of the final document ("Google doc-like" collaborative editing).

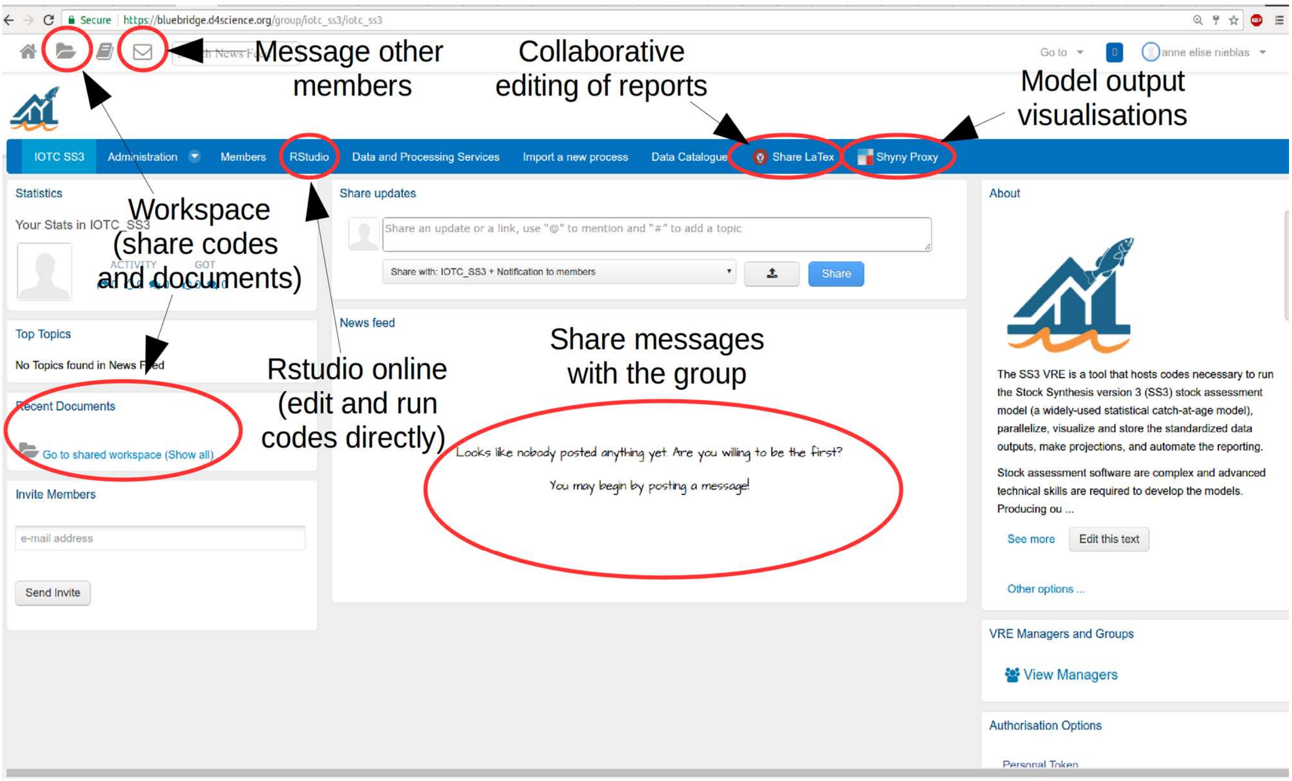
Efforts to comply with standards

Efforts have been made to make this work as generic as possible so that as much of the work flow can be useful for other cases (e.g., species, models). The R codes are available through Web Services. Metadata will be provided with generic formatting and outputs (figures and tables) that will describe work flows and results to allow data and process discovery and replication of the assessment. Furthermore, the SS outputs are saved in the widely-used NetCDF data format, allowing them to be used in any computing environment. Furthermore, these NetCDF data will be stored on a dedicated server (e.g., Thredds), with online data access.

RESULTS AND DISCUSSION

Online collaborative environment

The BlueBridge project enables an online collaborative environment by providing the infrastructure necessary to parametrize, visualize, and access a work flow on a VRE such as that described above (Figure 4). This environment will be available to a list of members who can share documents, messages, data and codes in both a public and private space. An Rstudio server will be incorporated



Message other members

Collaborative editing of reports

Model output visualisations

Workspace (share codes and documents)

Rstudio online (edit and run codes directly)

Share messages with the group

Looks like nobody posted anything yet. Are you willing to be the first?

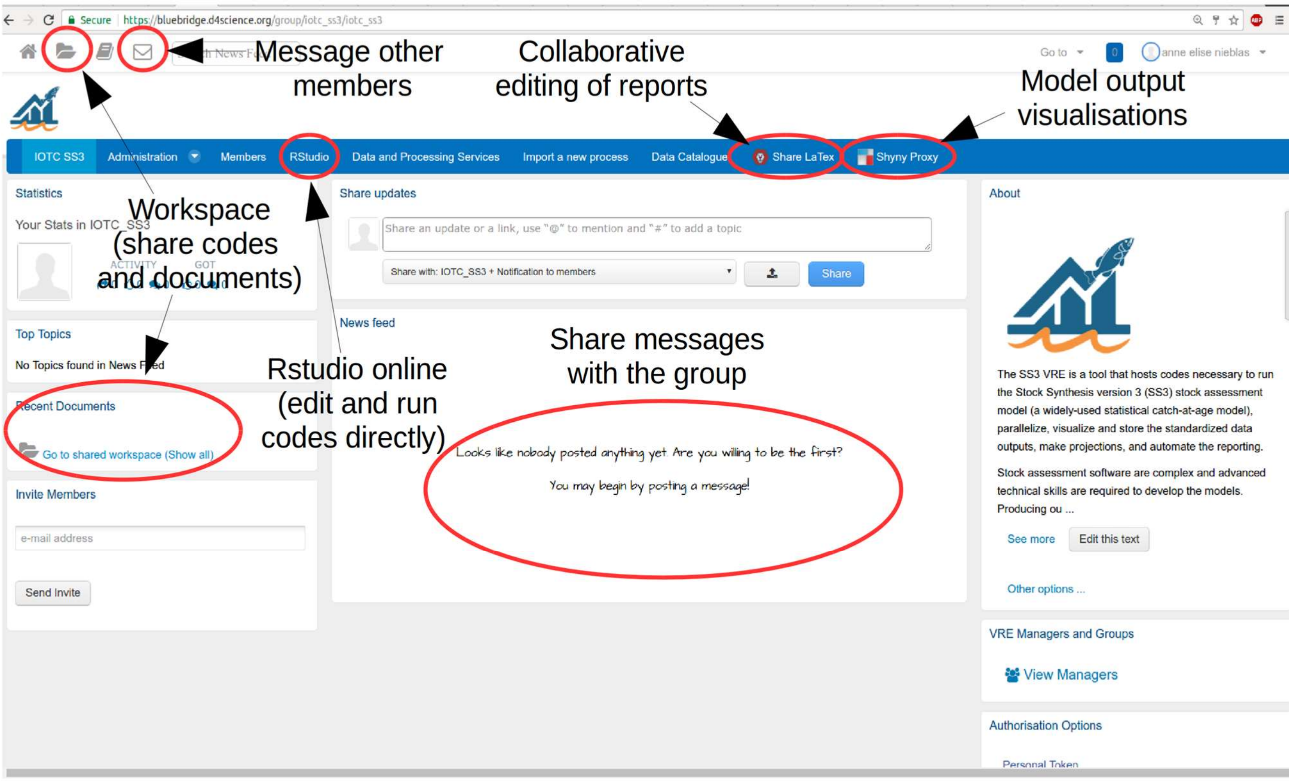
You may begin by posting a message!

The SS3 VRE is a tool that hosts codes necessary to run the Stock Synthesis version 3 (SS3) stock assessment model (a widely-used statistical catch-at-age model), parallelize, visualize and store the standardized data outputs, make projections, and automate the reporting.

Stock assessment software are complex and advanced technical skills are required to develop the models. Producing ou ...

into the environment that acts exactly as a desktop application, with a private work space 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, such that users can focus on parametrization of the model instead of programming (not yet implemented).

At the WPM08, we will present examples of the work flow by executing a single run of the last YFT model (run time of 30 seconds) by launching the code through the Rstudio online application (Figure 5). We will show examples of an R Shiny output visualization (Figure 6), as well as an example of the generation of an automatic report using the collaborative ShareLateX option on the VRE (Figure 7).



Share an update or a link, use "@" to mention and "#" to add a topic

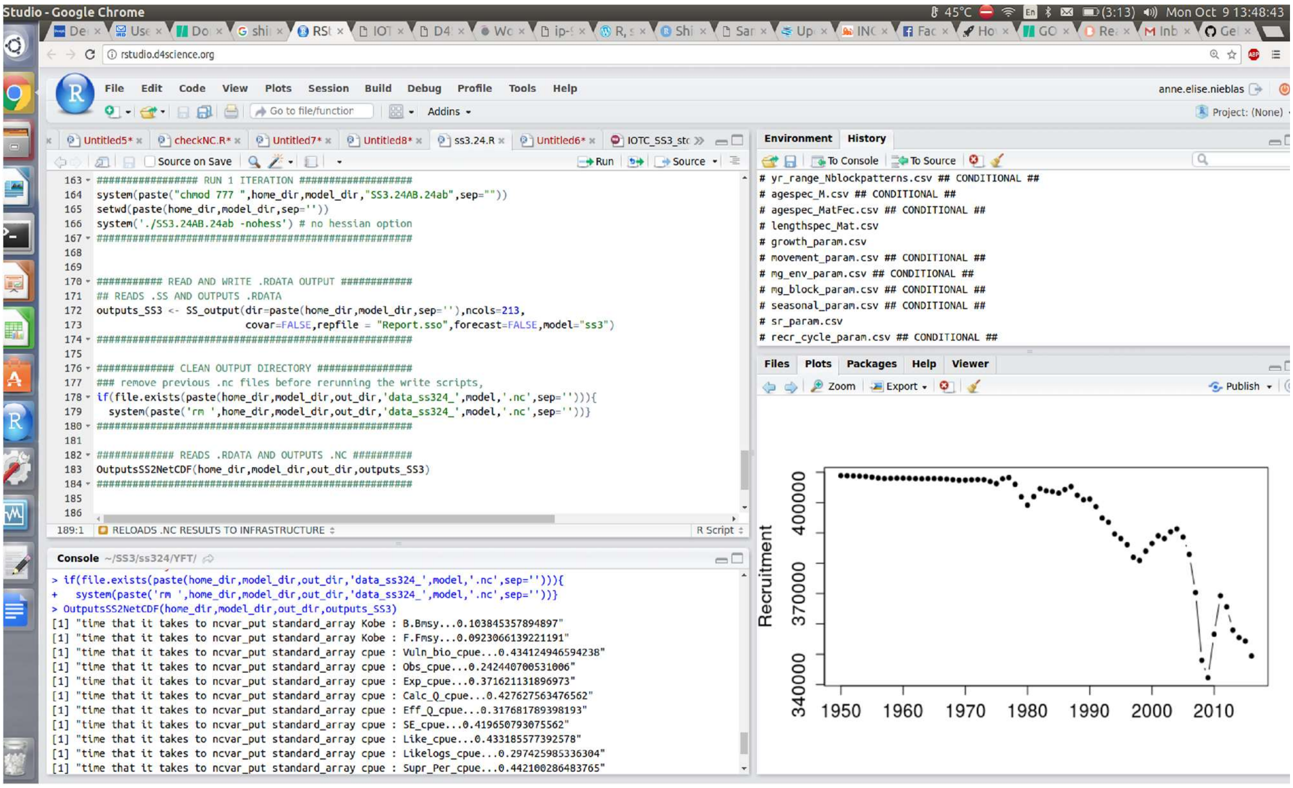
Looks like nobody posted anything yet. Are you willing to be the first?

You may begin by posting a message!

The SS3 VRE is a tool that hosts codes necessary to run the Stock Synthesis version 3 (SS3) stock assessment model (a widely-used statistical catch-at-age model), parallelize, visualize and store the standardized data outputs, make projections, and automate the reporting.

Stock assessment software are complex and advanced technical skills are required to develop the models. Producing ou ...

Figure 4: A screenshot of the IOTC SS3 VRE display. From this space, users can share codes, documents, and messages, can edit or run codes directly from the Rstudio online application, can edit automatic reports through the ShareLaTex application, and can visualize model outputs through the ShinyProxy application.

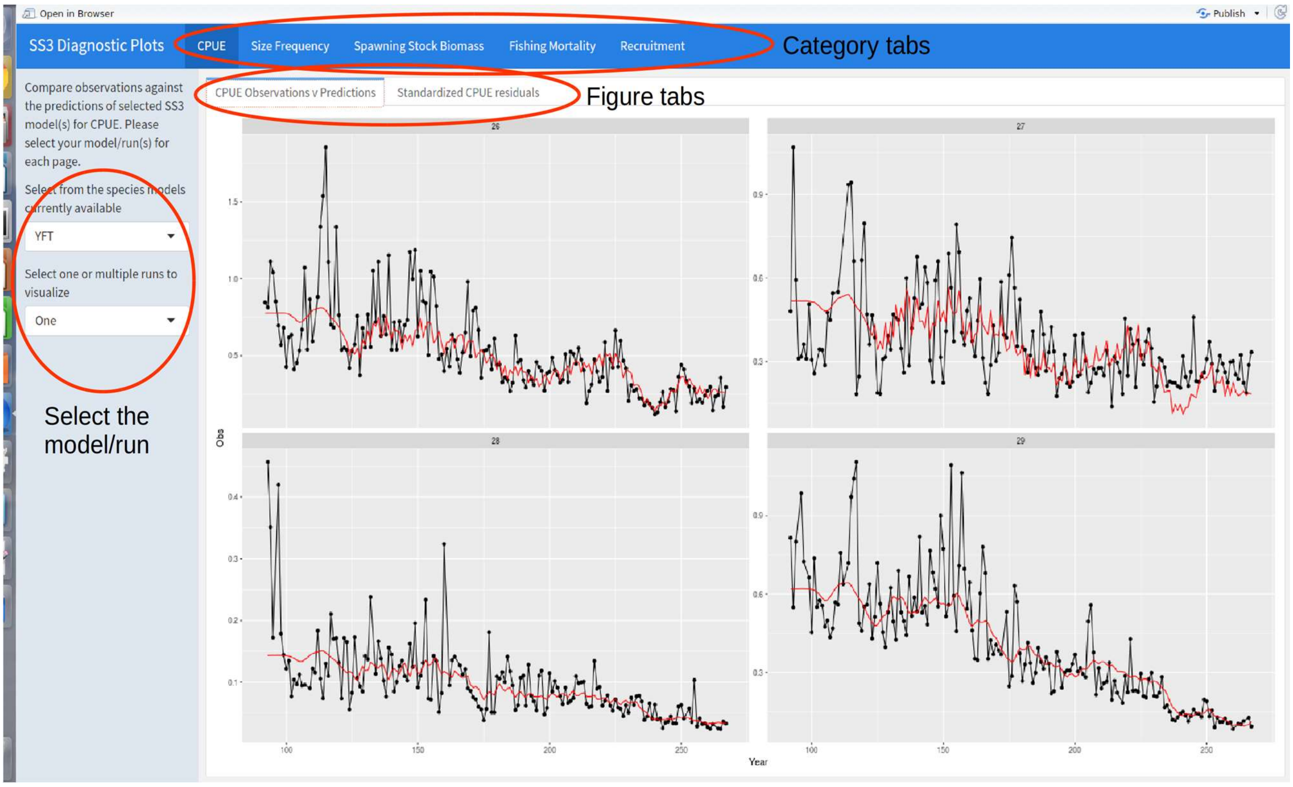


```
################# RUN 1 ITERATION ###################
system(paste("chmod 777 ",home_dir,model_dir,"SS3.24AB.24ab",sep=""))
setwd(paste(home_dir,model_dir,sep=''))
system('./SS3.24AB.24ab -nohess') # no hessian option
#####################################################


########### READ AND WRITE .RDATA OUTPUT ############
## READS .SS AND OUTPUTS .RDATA
outputs_SS3 <- SS_output(dir=paste(home_dir,model_dir,sep=''),ncols=213,
                         covar=FALSE,repfile = "Report.sso",forecast=FALSE,model="ss3")
#####################################################

############# CLEAN OUTPUT DIRECTORY ################
### remove previous .nc files before rerunning the write scripts,
if(file.exists(paste(home_dir,model_dir,out_dir,'data_ss324_',model,'.nc',sep=''))){
  system(paste('rm ',home_dir,model_dir,out_dir,'data_ss324_',model,'.nc',sep=''))}
#####################################################

############# READS .RDATA AND OUTPUTS .NC ##########
OutputsSS2NetCDF(home_dir,model_dir,out_dir,outputs_SS3)
#####################################################
```

```
> if(file.exists(paste(home_dir,model_dir,out_dir,'data_ss324_',model,'.nc',sep=''))){
+   system(paste('rm ',home_dir,model_dir,out_dir,'data_ss324_',model,'.nc',sep=''))}
> OutputsSS2NetCDF(home_dir,model_dir,out_dir,outputs_SS3)
[1] "time that it takes to ncvar_put standard_array Kobe : B.Bmsy...0.103845357894897"
[1] "time that it takes to ncvar_put standard_array Kobe : F.Fmsy...0.0923066139221191"
[1] "time that it takes to ncvar_put standard_array cpue : Vuln_bio_cpue...0.434124946594238"
[1] "time that it takes to ncvar_put standard_array cpue : Obs_cpue...0.242440700531006"
[1] "time that it takes to ncvar_put standard_array cpue : Exp_cpue...0.371621131896973"
[1] "time that it takes to ncvar_put standard_array cpue : Calc_Q_cpue...0.427627563476562"
[1] "time that it takes to ncvar_put standard_array cpue : Eff_Q_cpue...0.317681789398193"
[1] "time that it takes to ncvar_put standard_array cpue : SE_cpue...0.419650793075562"
[1] "time that it takes to ncvar_put standard_array cpue : Like_cpue...0.433185577392578"
[1] "time that it takes to ncvar_put standard_array cpue : Likelogs_cpue...0.297425985336304"
[1] "time that it takes to ncvar_put standard_array cpue : Supr_Per_cpue...0.442100286483765"
```

```
# yr_range_Nblockpatterns.csv ## CONDITIONAL ##
# agespec_M.csv ## CONDITIONAL ##
# agespec_MatFec.csv ## CONDITIONAL ##
# lengthspec_Mat.csv
# growth_param.csv
# movement_param.csv ## CONDITIONAL ##
# mg_env_param.csv ## CONDITIONAL ##
# mg_block_param.csv ## CONDITIONAL ##
# seasonal_param.csv ## CONDITIONAL ##
# sr_param.csv
# recr_cycle_param.csv ## CONDITIONAL ##
```

Figure 5: The RStudio online application of the VRE functions exactly as a local Rstudio, except that it is accessible within the browser, and the environment is already configured to run the functions that are shared in the VRE, facilitating the use of the shared online code.



Compare observations against the predictions of selected SS3 model(s) for CPUE. Please select your model/run(s) for each page.

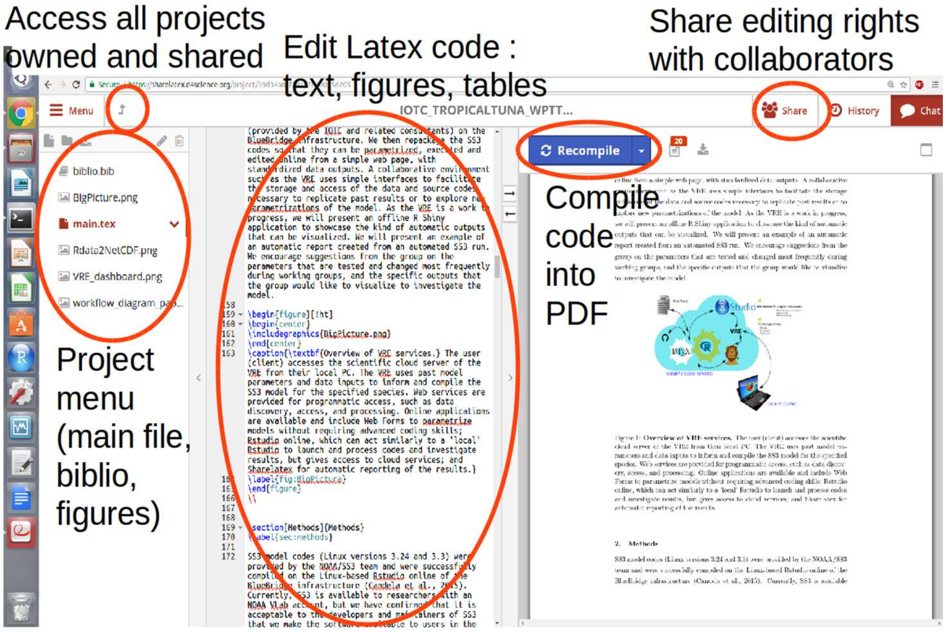
Select from the species models currently available

YFT

Select one or multiple runs to visualize

One

Figure 6: The R Shiny allows users to visualize the results of current or past model runs, allowing runs to be overlaid to compare results. The Shiny for the IOTC SS3 VRE includes diagnostic plots aimed at comparing between observations and predictions of the model(s). The plots currently included are CPUE observations versus predictions, standardized residuals of CPUE, size frequency plots of aggregated length frequency, and mean length and weight distributions, spawning stock biomass, fishing mortality, and recruitment variations.


Access all projects owned and shared
Edit Latex code : text, figures, tables
Share editing rights with collaborators
Menu
IOTC_TROPICALTUNA_WPTT...
Share
History
Chat
Recompile
biblio.bib
BigPicture.png
main.tex
Rdata2NetCDF.png
VRE_dashboard.png
workflow_diagram_pa...
Project menu (main file, biblio, figures)
Compile code into PDF
Methods

```
(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.

\begin{figure}[!ht]
\begin{center}
\includegraphics{BigPicture.png}
\end{center}
\caption{\textbf{Overview of VRE services.} The user
(client) accesses the scientific cloud server of the
VRE from their local PC. The VRE uses past model
parameters and data inputs to inform and compile the
SS3 model for the specified species. Web services are
provided for programmatic access, such as data
discovery, access, and processing. Online applications
are available and include Web Forms to parametrize
models without requiring advanced coding skills;
Rstudio online, which can act similarly to a 'local'
Rstudio to launch and process codes and investigate
results, but gives access to cloud services; and
Sharelatex for automatic reporting of the results.}
\label{fig:BigPicture}
\end{figure}
\\


\section[Methods]{Methods}
\label{sec: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 (Candela et al., 2015).
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
```

Figure 7: Sharelatex gives users the options to automatically generate reports/figures and collaborate editing of the text with other VRE members. The user accesses the Sharelatex option from the VRE (see Fig 4), and has options to either create their own project, or edit projects that are shared with them. The Sharelatex dashboard is divided into three main panels: the project menu (left), which includes the files necessary to compile the document (e.g., the *.tex file where the latex/knitr code is written, the bibliography file (*.bib), and the figures to insert), the console (middle) where the code can be edited, and the compiled pdf (right).

Benefits of the VRE

The VRE provides an online collaborative environment that enables individual users access to the computing resources and infrastructure of the BlueBridge project. This means that runs can be executed on servers and cloud infrastructure with no need for local computing power. This thereby allows more runs to be performed in less time (with parallelization), enabling more uncertainty calculations to be included in the stock assessment. With the decreased time necessary to compute numerous runs, major changes to the parametrization of the model can be performed in the context of the working group.

The VRE provides a collaborative environment for members, with a workspace to share documents, codes, and messages. Source codes are now accessible that were not before. Programming on the VRE is also practical for users as the computing environment is already compiled for all users and no installation is necessary. For those users without advanced programming skills, the VRE will provide a graphical user interface (via a Web Processing Service), such that models can be easily parametrized, increasing the number of potential users and participation in model development. Finally, the VRE ensures a backup of the selected runs (with detailed metadata describing the workflow, specifications, results and data) stored on its Thredds server, enabling reproducibility, and data and process discovery. Furthermore, model outputs can be browsed and visualized with graphical interfaces (Shiny). Reports can be at least partially automated with plots automatically generated with knitr. The text of the report can be edited collaboratively with Sharelatex.

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 the complex SS3 model will bringmore 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.

ACKNOWLEDGEMENTS

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

REFERENCES

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

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

R. Methot. User Manual for Stock Synthesis: Model Version 3.24s., 2013.

R. Methot. User Manual for Stock Synthesis: Model Version 3.30., 2017.

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

I. Taylor, I. Stewart, A. Hicks, T. Garrison, A. Punt, J. Wallace, and C. Wetzel. r4ss: R code for Stock Synthesis. R package version 1.16.. (English), 2011.

APPENDICES

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

	STARTER.SS	
variables	YFT	
version	3.21e	
filename	../test_YFT/starter.ss	
comments	starter.ss made using starterSS324.R	
model	YFT	
init_vals		1
display_deets		0
age_str_rep		1
checkup		0
param_trace		0
cum_report		1
full_priors		1
soft_bounds		1
num_data_out		3
turn_off_est		7
MCMCburn		10
MCMCthin		1
jitter		0
sdrep_start		-1
sdrep_end		-2
n_sd_yrs		0
sd_yr_vector	#	
final_conv	0.075	
retro_yr		0
minage_sumbio		1
dep_basis		1
frac_depden		1
SPR_rep_basis		2
F_rep_units		4
F4_age_range	1 28	
F_rep_basis		2
endfile_val		999
	FORECAST.SS	
variable	YFT	
version	3.20b	
comments	forecast.ss made using writeSS324Forecast.R	
filename	../test_YFT/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	c(0,0,-7,0,-7,0)	
bmark_relF_basis		1

forecast		2
n_forecast_yrs		4
F_scalar		1
Fcast_yrs	c(-7,0,-7,0)	
control_rule		2
control_rule_uplim	0.01	
control_rule_lowlim	0.001	
control_rule_buff		1
n_fcast_loops		3
first_fcast_loop		3
fcast_loop_ctl3		0
fcast_loop_ctl4		0
fcast_loop_ctl5		0
first_yr_capsall		290
imp_err		0
rebuilder		0
rebuilder_Ydecl		-1
rebuilder_Yinit		-1
fleet_relF		1
fcast_catch_basis		2
fishery_names	FISHERY1	
fishseas_F		1
max_total_catch_fleet	rep(-1,25)	
max_total_catch_area	rep(-1,4)	
fleet_assignment	rep(0,25)	
allocation_fractions		
n_forecast_catch		0
Fcast_basis		2
endfile_val		999

YFT.CTL

Variable	YFT	
version	3.21e	
model	YFT	
n_growthpatterns		1
n_submorphs		1
submorph_growthvar		
morph_dist		
n_seasons_peryear		1
n_areas		4
n_recrassign		2
recr_inter		0
recr_assign_tab	read.csv('../input/recr_assign_tab_YFT.csv',sep=',',header=F)	
n_move_defs		10
age_firstmove		2
movement_def	read.csv('../input/movement_def_YFT.csv',sep=',',header=F)	
n_block_patterns		0
n_blocks_per_pattern		
yr_range_Nblockpatterns		

frac_female	0.5
natM_opt	3
n_brk_pts	
age_brk_pts	
Lorenzen_ref_age	
agespec_M	read.csv('../input/agespec_M_YFT.csv',sep=',',header=F)
growth_mod	3
growth_amin	1
growth_amax	28
agespec_K_amin	12
agespec_K_amax	seq(2,13,by=1)
sd_add_to_laa	0
cv_patt	0
mat_opt	3
agespec_MatFec	read.csv('../input/agespec_MatFec_YFT.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_YFT.csv',sep=',',header=F)
movement_param	read.csv('../input/movement_param_YFT.csv',sep=',',header=F)
mg_env	1
mg_env_param	read.csv('../input/mg_env_param_YFT.csv',sep=',',header=F)
mg_block	0
mg_block_param	
param_seasonality	rep(0,10)
seasonal_param	
mg_ann_dev_phase	7
sr_fun	3
sr_param	read.csv('../input/sr_param_YFT.csv',sep=',',header=F)
sr_env_link	0
sr_env_target	0
do_recr_dev	1
recr_dev_begin_yr	101
recr_dev_end_yr	272
recr_dev_phase	3
adv_opt	0
recr_dev_early_start	
recr_dev_early_phase	
forecast_recr_phase	
lambda_fcast_recr	
last_yr_nobias	
first_yr_nobias	
last_yr_fullbias	

first_rec_nobias	
max_bias_adj	
recr_cycle_period	
min_recr_dev	
max_recr_dev	
n_recr_devs	
n_recr_cycles	
recr_cycle_param	
recr_dev	
f_ballpark	0.1
f_ballpark_yr	220
f_method	3
max_f	2.9
f_start_val	
f_phase	
n_f_inputs	
n_tuning	4
f_param	
initF_param	read.csv('../input/initF_param_YFT.csv',sep=',',header=F)
Q_tab	read.csv('../input/q_tab_YFT.csv',sep=',',header=F)
Q_param	
size_select_tab	read.csv('../input/size_select_tab_YFT.csv',sep=',',header=F)
age_select_tab	read.csv('../input/age_select_tab_YFT.csv',sep=',',header=F)
size_select_param	
age_select_param	read.csv('../input/age_select_param_YFT.csv',sep=',',header=F)
do_tag	1
tag_param	read.csv('../input/tag_param_YFT.csv',sep=',',header=F)
tag_param_rep	read.csv('../input/tag_param_rep_YFT.csv',sep=',',header=F)
tag_param_decay	read.csv('../input/tag_param_decay_YFT.csv',sep=',',header=F)
var_adj_factor	1
var_adj_tab	read.csv('../input/var_adj_tab_YFT.csv',sep=',',header=F)
max_lambda_phase	1
sd_offset	1
n_changes_lambda	0
like_comp_tab	
read_specs	0
var_control	
selex_std_bin	
growth_std_bin	
NatAge_std_bin	
endfile_val	999

YFT.DAT

Variable	YFT
model	YFT
version	3.24

start_yr	13
end_yr	276
n_seasons_peryear	1
n_months_perseason	3
spawning_season	1
n_fleets	25
n_surveys	4
n_areas	4
fishsurvey_names	FISHERY1%FISHERY2%FISHERY3%FISHERY4%FISHERY5%FISHERY6%FISHERY7%FISHERY8%FISHERY9%FISHERY10%FISHERY11%FISHERY12%FISHERY13%FISHERY14%FISHERY15%FISHERY16%FISHERY17%FISHERY18%FISHERY19%FISHERY20%FISHERY21%FISHERY22%FISHERY23%FISHERY24%FISHERY25%SURVEY1%SURVEY2%SURVEY3%SURVEY4
sample_timing	rep(0.5,29)
fleet_area	c(rep(1,9),2,3,rep(4,4),rep(2,3),4,4,rep(1,4),4,1,2,3,4)
catch_units	rep(1,25)
se_logcatch	rep(0.01,25)
n_genders	1
n_ages	28
init_equil_catch	rep(0,25)
n_catch_obs	264
catch=catch.csv	read.csv('../input/catch_YFT.csv',sep=',',header=F)
n_cpue_obs	683
cpue_units=cpue_units.csv	read.csv('../input/cpue_units_YFT.csv',sep=',',header=F)
cpue=cpue.csv	read.csv('../input/cpue_YFT.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	2
binwidth	2
pop_minsize	10
pop_maxsize	198
n_popbins	
lowedge_popbin	
comp_tail	1.00E-005
add_to_comp	1.00E-007
bin_to_combine_genders	0
n_lengthbins	95
lowedge_lenbin	seq(10,198,by=2)
n_length_obs	1857
length_comp=length_comp.csv	read.csv('../input/length_comp_YFT.csv',sep=',',header=F)
n_agebins	0

lowedge_agebins		
n_ageerr_defs		0
age_matrix=age_matrix.csv		
n_age_obs		0
Lbin_method		1
agebin_combine_genders		1
age_comp=age_comp.csv		
n_mnsizeage_obs		0
mn_size_at_age=mn_size_at_age.csv		
n_envvar		7
n_env_obs		980
env_data=env_data.csv	read.csv('../input/env_data_YFT.csv',sep=',',header=F)	
n_sizefreq_methods		0
nbins_per_method		
sizefreq_units		
sizefreq_scale		
sizefreq_mincomp		
n_sizefreq_obs		
lowedge_sizefreq_bins		
sizefreq=sizefreq.csv		
do_tags		1
n_tag_groups		131
n_recap_events		1485
mix_period		3
max_tracking		16
release=release.csv	read.csv('../input/release_YFT.csv',sep=',',header=F)	
recapture=recapture.csv	read.csv('../input/recapture_YFT.csv',sep=',',header=F)	
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