



**SCIENTIFIC COMMITTEE
FOURTEENTH REGULAR SESSION**

**Busan, Republic of Korea
8-16 August 2018**

Developments in the MULTIFAN-CL Software 2017-18

WCPFC-SC14-2018/ SA-IP-02

**Nick Davies¹, Dave Fournier², Yukio Takeuchi³, Fabrice Bouyé³
and John Hampton³**

¹ Te Takina Ltd, Whangarei, New Zealand

² Otter Research Ltd, North Saanich, British Columbia, Canada

³ Oceanic Fisheries Programme, The Pacific Community, Noumea, New Caledonia

Table of Contents

Executive summary	4
1 Introduction.....	5
2 Development overview	5
2.1 Team	5
2.2 Calendar.....	6
2.3 Collaboration and versioning	6
2.4 Compilation framework and Source code repository	6
2.4.1 Compilation framework	6
2.4.2 Compilation of Mac OS executable.....	7
2.4.3 Development version	7
2.4.4 Source code repository	7
2.5 Developer’s workshops	8
2.6 Benchmark testing 2017-18	8
2.6.1 Version 2.0.3.1.....	8
2.6.2 Version 2.0.4.0.....	8
2.7 Postings to website.....	9
2.7.1 18 January 2018 – version 2.0.3.1.....	9
2.8 Independent Peer Review of the 2011 bigeye tuna stock assessment.....	10
2.9 Tool development	10
2.9.1 R4MFCL.....	10
2.9.2 Testing framework	10
2.9.3 Viewer	12
2.9.4 Condor parallel processing facility	12
2.10 User’s guide	12
3 Training workshop.....	12
4 New features	13
4.1 Self-scaling Multinomial plus random effects (SSMULT_RE)	13
4.1.1 Rationale	13
4.1.2 Methods and Testing.....	13
4.1.2.1 Implementation with tuna models.....	14
4.1.2.2 Autoregressive-Autoregressive correlation estimation	15
4.2 Simulation mode	15
4.2.1 Generation of pseudo-observations	16
4.2.1.1 Simulating tagging data.....	16
4.2.2 Optimising simulation operations.....	17
4.2.3 Status and future work.....	17

4.3	Maturity at length	18
4.3.1	Rationale	18
4.3.2	Methods and Testing.....	18
4.4	Orthogonal-Polynomial recruitments	19
4.4.1	Rationale	19
4.4.2	Methods and Testing.....	20
4.5	Hessian diagnostic	21
5	Other enhancements and bug fixes	21
5.1	Shared effort deviates in the multi-sex model.....	21
5.2	Simulation mode reports.....	22
5.3	Autocorrelation in recruitment	22
5.4	Selection of minimizing method.....	22
5.5	Input/Output diagnostic of <i>.par</i> file	23
5.6	Bug fixes.....	23
5.6.1	Fix to pseudo-observed effort with missing observed effort.....	23
5.6.2	Fix to the selectivity pointer	23
5.6.3	Fix to the parameter list report <i>xinit.rpt</i>	24
5.6.4	Fix to likelihood components report	24
5.6.5	Fix to exploitable biomass section of <i>plot.rep</i>	24
5.6.6	Fix to simulation recruitment deviates	24
6	Application of new features	24
7	Future work	24
8	Discussion	25
9	References.....	26
10	Tables.....	28
11	Figures	33
12	Annex.....	47
12.1	Benchmark testing.....	47
12.2	Agenda for Training Workshop, March 2018	50
12.3	Agenda.....	50

EXECUTIVE SUMMARY

This paper summarises the developments made within the MULTIFAN-CL software project as carried out by the team at the Oceanic Fisheries Programme (OFP, The Pacific Community) from August 2017 to July 2018, and updates the report of Davies et al. (2017).

The progress made on implementing new features relate to improving its capability for stock assessments and its utility within the management strategy evaluation (MSE) framework being developed by OFP. These included:

-) Refining and testing the implementation of an innovative approach for fitting to size composition data, using the self-scaling Multinomial with random effects estimation (SSMULT_RE), and also the development of an autoregressive-autoregressive method (AU-AU) for the autocorrelation in the random effects; and, comprehensive testing using large tuna models.
-) Potentially improving model minimisation stability through a parameterisation using orthogonal polynomials for estimating the temporal-spatial recruitments, that can substantially reduce the total number parameters being estimated.
-) Reducing bias in assumptions for maturity at age by internally calculating this ogive given an input maturity at length vector and the model's growth estimates.
-) Making possible more detailed model fit diagnostics using the Hessian eigenvalues and eigenvectors.
-) Extending and improving the simulation mode for running MULTIFAN-CL as an operating model (OM) by:
 - o generating pseudo-observations of tagging data for both the estimation and projection model time periods, that include error due to reporting rate probabilities;
 - o optimising runtime and calculation performance; and,
 - o producing additional OM reports for MSY-related quantities and absolute biomass.

In addition, refinements were made to the features for: fitting multi-sex models; specifying the minimising method used in the fit, and, estimating autocorrelated recruitments.

The new features for the simulation mode were of immediate benefit for using MULTIFAN-CL within the management strategy evaluation (MSE) framework being developed at OFP in 2018 (Scott et al. 2018a, 2018b, 2018c). Takeuchi et al. (2018) details further testing of the recent developments in the multi-species/multi-sex functionality of MULTIFAN-CL. Another new feature used in the 2018 stock assessments was that for south Pacific albacore that included the internal conversion from maturity-at-length to maturity-at-age given the estimated growth curve (Tremblay-Boyer et al. 2018).

Ianelli et al. (2012) reported thirteen recommendations from an independent peer review panel specifically relating to MULTIFAN-CL. The rationale for these recommendations was to address the key areas of uncertainty for the tuna assessments reported to the Scientific Committee of the Western and Central Pacific Fisheries Commission (SC), through improving the biological description of population processes, better estimation of observation error, and the improved modelling of fishing mortality. Ten of the recommendations have now been implemented, one during 2017-18 (maturity at length), and three remain yet to be implemented. Rather, resources were directed to the new features listed above. The model uncertainty due to assumptions made for the relative importance amongst the various data types included in the integrated modelling approach has dominated most fisheries assessment models. The self-scaling properties of the SSMULT_RE approach offers a way to reduce this area of model uncertainty, thus it has been a priority. Also, a priority has been the support given to the developing MSE framework that seeks to use MULTIFAN-CL for Operating Model (OM) and Estimation Model (EM) candidates. It is therefore reasonable to maintain some flexibility in the priorities given to project tasks, and the three remaining recommendations of Ianelli et al. (2012) will be part of future project plans.

As in 2017, a 3-day training workshop for stock assessment analysts was held at Nouméa, during March 2018. This assisted in making the new features accessible to those undertaking the 2018 stock assessments. These workshops are now a routine project item that facilitates the rapid use of new features by users, and provides feedback on improvements required to MULTIFAN-CL.

In respect of future work, the key components of the MULTIFAN-CL project proposed for 2018-19 are itemised below:

Peer review recommendation (Ianeli et al. 2012)	Implementation
c. Long-term tag loss	Development
Other new features	Implementation
Self-scaling multinomial with random effects	Testing, draft report
Recruitment random effects estimation	Draft report
Constrain recruitment and effort deviates	Development
Tagging data informs growth estimation	Development
Outstanding testing of existing features	Development
Simulation mode – age-length pseudo-observations	Development
OM – additional sources of process error; Turing Tests	Development
EM – fits to only the projection pseudo-observations	Development
Report of the OM size compositions for projection period	Development

1 INTRODUCTION

MULTIFAN-CL is a statistical, age-structured, length-based model routinely used for stock assessments of tuna and other pelagic species. The model was originally developed by Dr Dave Fournier of Otter Research Ltd and Dr John Hampton (The Pacific Community) for application to south Pacific albacore tuna (Fournier et al. 1998).

The MULTIFAN-CL model is described in detail in the User’s Guide (Kleiber et al. 2018). It is typically fitted to total catch, catch rate, size-frequency and tagging data stratified by fishery, region and time period. For example, recent tropical tuna assessments (e.g. Tremblay-Boyer et al. 2017, McKechnie et al. 2017) encompass a time period of 1952 to 2016 in quarterly time steps, and model multiple separate fisheries occurring in 9 spatial regions specified. The main parameters estimated by the model include: initial numbers-at-age in each region (usually constrained by an equilibrium age-structure assumption), the number in age class 1 for each quarter in each region (the recruitment), growth parameters, natural mortality-at-age (if estimated), movement, selectivity-at-age by fishery (constrained by smoothing penalties or splines), catch (unless using the catch-conditioned catch equation), effort deviations (random variations in the effort-fishing mortality relationship) for each fishery, initial catchability, and catchability deviations (cumulative changes in catchability with time) for each fishery (if estimated). Parameters are estimated by fitting to a composite (integrated) likelihood comprised of the fits to the various data types, and penalized likelihood distributions for various parameters.

Each year the MULTIFAN-CL development team works to improve the model to accommodate changes in understanding of the fishery, to fix software errors, and to improve model features and usability. This document records changes made since August 2017 to the software and other components of the MULTIFAN-CL project both for the current release version (2.0.3.1), and the current unreleased development version, and updates the report for the previous period, 2016-17, (Davies et al. 2017).

2 DEVELOPMENT OVERVIEW

2.1 Team

The senior developer of MULTIFAN-CL is Dr Dave Fournier, of Otter Software in Canada. Assisting with programming is Nick Davies, with occasional programming carried out by Yukio Takeuchi and John Hampton

(SPC, New Caledonia). Other tasks include testing and debugging (ND, JH, and Fabrice Bouye (SPC)); documentation (ND); and planning and coordination (DF, ND, and JH). Related project software are developed or managed by FB (MULTIFAN-CL Viewer, Condor, GitHub, Jenkins), ND, and Robert Scott (R4MFCL, FLR4MFCL).

2.2 Calendar

August – November: Testing, planning and ongoing code development, Developer’s workshop

December – February: Testing and ongoing code development

March: Training and Developer’s workshops

April-July: Testing, ongoing code development and support for stock assessments

2.3 Collaboration and versioning

The repository and overall development are coordinated via the GitHub website on GitHub.com at <https://github.com/PacificCommunity/ofp-sam-mfcl> which is administered by Fabrice Bouye (fabriceb@spc.int) (section 2.4.4).

Problems with MULTIFAN-CL operation or compilation have been reported to the project management website so as to maintain a list of desired enhancements, and to allocate tasks among the project team. Some of the tasks identified during the previous reporting period (2016-17) have been addressed in the current period through the model developments completed in 2017-18. A master branch exists for the MULTIFAN-CL source code from which release versions are posted, and development branches (ongoing-dev, mac-dev) have been created for holding development versions of the source undergoing development and testing. A formal testing procedure has been designed before source code is merged from the branch to the trunk, and a manual for the testing of new compilations, standardizing the source code compilation procedure, and posting of executables is maintained.

2.4 Compilation framework and Source code repository

2.4.1 Compilation framework

A continuous integration facility allows for automatic nightly compilations of the MULTIFAN-CL source on the master branch. This automation is done using the software called Jenkins (<https://jenkins-ci.org/>): an Open Source continuous integration tool that comes bundled with a web server used for administration. This software is now installed on the Virtual Machine (VM) that is dedicated to MULTIFAN-CL development.

In this tool, we’ve added a custom scheduled task that automatically retrieves the MULTIFAN-CL source code out of the GitHub code repository (master branch); it also retrieves required libraries for the compilation. When done, our task compiles both debug and optimized versions of the software. We’ve also configured this task to produce code documentation out of the source code and to run some C++ code quality checking.

Doing a nightly compilation allows us to find out more quickly whether issues have been included in the source code repository without being solved by the developer. It also helps us identify issues in the makefile configurations that may prevent the compilation of MULTIFAN-CL on some more neutral environment (i.e.: on a machine that is different from the one of the developer’s). So far, our nightly builds have a 99% compilation success rate.

Steps have been taken during 2017-18 to extend this facility to support automated builds of the Windows (Visual Studio 2017) release executable. It is also intended to add to the Jenkins tool the running of automated tests using example fish model data, and, in the future, unit tests for the software.

A directory structure on the dedicated VM was used that is mirrored on all the developer’s platforms in respect of source code **Projects/**, associated libraries **libs/**, and **Testing/** directories. This ensures portability of source and makefiles among the developers and the automated build software.

2.4.2 Compilation of Mac OS executable

A Mac compilation development branch from ongoing-dev was created that contained MULTIFAN-CL code adapted for a Mac gcc-8.1 compilation. This also entailed a Mac compilation of the ADMB libraries from the source code in the ofp-sam-admb repository. Largely, this required inserting conditional Macros at various points in the code and developing a makefile specifically for the Mac compiler which differed substantially from that used for the Linux compilations. The various source code changes made for this compilation were tested for their compatibility of with the Linux (**gcc**) and Windows (Visual Studio 2017 **cl**) compilers. The conclusion was that the MULTIFAN-CL source code required for the Mac compilation was compatible with both the gcc and cl compilers. This retained the code's integrity as a single version for use with all three compilers. Consequently, this Mac development branch was merged into the primary branch of the MULTIFAN-CL repository.

Currently this compilation may be performed only using personal computers and has not yet been incorporated into the MULTIFAN-CL compilation framework at OFP. It is proposed to commission a Mac OS machine on the OFP network for this purpose.

2.4.3 Development version

Upon completing benchmark testing of a development version, the source code in the repository development branch is merged to the master branch and tagged with a release version number. At this point the development branch is created afresh for implementing any subsequent code developments. Another, point where a new development version is made is immediately preceding each developer's workshop at which large scale code changes are rapidly made. These are then added to the development branch after the workshop and following preliminary testing. Points during 2017-18 where development versions were created included:

-) Following the benchmark testing of version 2.0.3.1
-) Following the benchmark testing of version 2.0.4.0
-) Preceding the April 2018 developer's workshop

Developments currently implemented in the development version since January 2018 that have not yet been benchmark tested include:

-) Effective estimated sample size reports for robust normal likelihood
-) SSMULT_RE AU-AU improvements
-) Fix to orthogonal-polynomial recruitments for undertaking projections
-) A diagnostic check of the input/output of the **.par** file
-) Correction to the **xinit.rpt** report file
-) A model fit diagnostic of the Hessian using the singular value decomposition
-) An index of fishery-specific selectivities
-) Optimising simulation and operating model calculations and report generation
-) Additional simulation reports for Fmult and other quantities
-) Sharing effort_dev_coffs parameters for the multi-sex model
-) Extending the simulation mode for:
 - o Improvements to generating pseudo-observations for tagging data
 - o Generating pseudo-observations for the estimation model time periods
-) Correction for stochastic projection models with age_flags(182)==0
-) Maturity-at-length input with conversion to ages integrated within the model

2.4.4 Source code repository

The MFCL project is now hosted on GitHub.com at:

-) <https://github.com/PacificCommunity/ofp-sam-mfcl>

This site is only accessible to registered members of the OFP-SAM team. In order to better coordinate developments within components of the project, separate repositories were created for the:

-) User's Guide: <https://github.com/PacificCommunity/ofp-sam-mfcl-manual>
-) ADMB dependent library: <https://github.com/PacificCommunity/ofp-sam-admb>

A total of 33 source code commits were made to the development branch, including merges to the master branch preceding the release of version 2.0.3.1., and the commit of version 2.0.4.0. Substantial commits occurred following each of the developer’s workshops in November 2017 and April 2018.

2.5 Developer’s workshops

Two developer’s workshops were held at Nanaimo, British Columbia, Canada, firstly from 18 October – 3 November 2017, and secondly, 7 – 16 April 2018. The participants were the primary developer Dr Dave Fournier and Mr Nick Davies.

The main areas of development during the first workshop were:

- Self-scaling Multinomial M-estimator for composition data, including development of an auto-regressive – auto-regressive method for estimating an autocorrelation
- Estimating recruitments using orthogonal-polynomials
- Extending the simulation mode for generating pseudo-observations of tagging data
- Estimating autocorrelation in recruitments

The main areas of development during the second workshop were:

- Self-scaling Multinomial M-estimator for composition data, including development of an auto-regressive – auto-regressive method for estimating an autocorrelation
- A diagnostic check of the input/output of the *.par* file
- Correction to the *xinit.rpt* report file
- A model fit diagnostic of the Hessian using the singular value decomposition
- An index of fishery-specific selectivities
- Optimising simulation and operating model calculations
- Sharing *effort_dev_coffs* parameters for the multi-sex model
- Extending the simulation mode for generating pseudo-observations for the estimation model time periods

The specific details of these developments are provided in Sections 4 and 5.

2.6 Benchmark testing 2017-18

The benchmark testing framework is described in section 2.9.2, and the series of benchmark tests undertaken in 2017-18 are listed in Annex 12.1 with a brief description of the tests, and the features tested, is provided in this section.

2.6.1 Version 2.0.3.1

In August 2017, and preceding the first developer’s workshop, comprehensive benchmark testing was done of the development version versus the release version 2.0.2.2. Substantial developments and bug fixes had been made since the previous testing of the development version done in May 2017. The major changes in version 2.0.3.1 are listed in section 2.7.1.

Single evaluation tests of the Linux executables indicated identical calculations among the versions and compiler platforms. Differences in the *doitall* solutions among the versions for the Windows compilation platforms could be explained by differences in the minimisation “paths” taken to reach a solution by each version, which was attributed to the different compilers used (MinGW and Visual Studio 2017). Whereas, the Linux executables produced identical results.

2.6.2 Version 2.0.4.0

In February 2018, and preceding the second developer’s workshop, comprehensive benchmark testing was completed of the development version versus the release version 2.0.3.1. A list of the developments accumulated in, the now version 2.0.4.0, includes:

-) Options for function minimisers
-) SSMULT M-estimator without random effects for weight frequency data
-) SSMULT revision to the AR(1) penalty formulation

-) Revised scalars for SSMULT_RE parameters and the independent growth parameters
-) Correction to exploitable biomass report to plot.rep
-) Effective Sample Size calculation for robust-normal likelihood
-) Generation of simulation Pseudo-observations for tagging data
-) Estimating orthogonal-polynomial recruitments

Given that version 2.0.4.0 included options for minimisers, the flags were set for achieving comparability with version 2.0.3.1, and with the previous benchmark. To implement the standard limited memory Newton minimiser with nsteps = 7:

-) version 2.0.3.1: parest_flags(350) = 0 and parest_flags(192) = 0
-) version 2.0.4.0: parest_flags(350) = 1 and parest_flags(192) = 0

Data used for the 2016 and 2017 stock assessment models developed by OFP were added to the range of testing data sets used for the benchmark testing. Since these models implement many of the most recent features in MULTIFAN-CL, this improves the suite of features being tested for its integrity in the development version. The following test data were added:

-) SKJ2016
-) BET2017
-) YFT2017
-) SWO2017

Single evaluation tests of the Linux executables indicated identical calculations among versions, and differences in the doital solutions can be attributed to the development version performing better (improved minima) and with negligible differences among the platforms for this version. The differences among the versions may be attributed to revised parameter scaling, which results in different minimisation paths being taken. Moderate to large differences in the version 2.0.4.0 solutions occurred from those of version 2.0.3.1 for: STM2012, SWO2017 (especially for the windows platform); and, YFT2017 (only for the linux platform). A local minimum was found by the version 2.0.4.0 for the STM2012 test case (an improved objective function value) that produced large differences in model quantities. This was related to the test data and model configuration rather than the version 2.0.4.0 code.

While version 2.0.4.0 has been tested, it has not as yet been posted as the next release version.

2.7 Postings to website

There has been one posting of the MULTIFAN-CL release version to the website since August 2017.

2.7.1 18 January 2018 – version 2.0.3.1

The main changes relative to the previous version 2.0.2.2 included:

-) Visual Studio 2017 compilation of a Windows executable
-) Revised method for estimating the autocorrelation of BH-SRR deviates
-) Length-specific selectivity shared among sexes in a multi-sex model
-) Various combinations of aggregated/disaggregated fisheries data in a multi-species/sex model
-) Applying the sex ratio of predicted catch compositions to predicted size compositions in a multi-sex model
-) Including identifiers for sex in output reports in a multi-sex model
-) Improvements and developments in the SSMULT-RE and DM likelihood
-) Improving the code that applies tail compression to size composition data
-) Likelihood profiling in respect of model dependent variables, such as current biomass
-) Generating simulation tagging data

Some of the above changes have been described previously by Davies et al. (2017).

2.8 Independent Peer Review of the 2011 bigeye tuna stock assessment

An outcome of an independent peer review of the 2011 bigeye tuna stock assessment (Ianelli et al. 2012) was a set of recommendations for improvements and developments to the MULTIFAN-CL software. These aim not only to improve the software's application in the context of the bigeye assessment specifically, but also its stock assessment application more generally. These recommendations have formed the core of MULTIFAN-CL developments since the review, and an outline of the status in fulfilling these recommendations is provided.

Of the thirteen recommendations, 10 have been implemented and tested, and 3 remain yet to be developed. The recommendations that were identified to be undertaken for 2017-18 include (Table 1):

-) Maturity at length (recommendation "k")
-) Non-uniform size bins (recommendation "b")
-) Long-term and initial tag loss (recommendation "c")
-) Tags inform movement only (recommendation "d")

Development and testing has been completed for the maturity at length feature (see section 4.3). In 2017-18, following internal discussion, the only recommendation included in the workplan was that for: adding long-term and initial tag loss into MULTIFAN-CL, "c". Higher priority was given to: completing the development of self-scaling approaches for fitting size composition data (the self-scaling multinomial with random effects, SSMULT_RE, section 4.1); and, further implementations of the simulation mode (section 4.2). These other tasks have therefore taken priority over the implementation of the above three remaining features recommended by Ianelli et al. (2012). Indeed, the implementation of the recommendation for a multinomial distribution for size composition data has extended beyond the scope of that initially intended, with the innovative SSMULT_RE approach improving upon the simple Multinomial as suggested by the panel. However, it is proposed that work toward these remaining recommendations be resumed for 2018-19, and subsequent years (section 7, and Table 2):

2.9 Tool development

2.9.1 R4MFCL

The R scripts for working with MULTIFAN-CL, developed by OFP are maintained on a GitHub repository and have been partially updated to adapt to the recent MULTIFAN-CL release version file formats. These scripts are used to manipulate the input files, so that submitting model runs can be automated from R. Other scripts can be used to read in the output files, analyze the results, and generate plots and tables. Further refinements of these tools were undertaken as part of the 2018 stock assessments that consolidated new features to the utilities package.

2.9.2 Testing framework

The testing framework for MULTIFAN-CL compilations first developed in 2011-12, was applied during 2017-18. This framework ensures the repeatability and traceability of testing by streamlining the process for new source code developments through a system of model testing procedures and directories. The testing criterion is based upon pair-wise comparisons of model run results obtained using an existing MULTIFAN-CL compilation (usually the current release version) versus those from a development version compilation. Tests are undertaken over multiple processor platforms and architectures, with application to multiple input testing data sets, and with various options for the MULTIFAN-CL operation, viz. single or multiple model evaluations, or full doital model fits to convergence. This ensures a thorough integrity-check of model quantities and components of the objective function prior to the distribution of new versions.

Since March 2013 MULTIFAN-CL source code has undergone substantial developments, major ones including: multi-threading of the tagged population calculations, combined tagging observations for multiple species/sexes, multi-sex equilibrium yield calculation, time-variant selectivities for individual fisheries, adding a likelihood term for age-length composition data, approaches for fitting size composition data, and a simulation mode. These developments have been described (Davies et al. 2017) and more recent ones are described in Section 4.

Following the addition of these new features to the development version, regular testing of this versus the release version was undertaken to ensure the integrity of existing operations. Known as “benchmark tests” these are listed in Annex 12.1 and those undertaken in 2017-18 are described in section 2.6. The development version was last tested in February 2018 versus the release version 2.0.3.1, which defines the development version as the **benchmark** source code, version 2.0.4.0. Subsequent development versions were then tested relative to the benchmark to establish their integrity, after which they are defined as the new benchmark development version. The testing framework entails two levels of tests.

1. Establish the accepted development version

The first level of testing ensures the integrity of existing model features by undertaking tests using a range of single-species data: ALB2012, ALB2015, BET2011, BET2014, BET2017, YFT2011, SKJ2011, STM2012, SWO2013, SWO2017, YFT2014, YFT2017, SKJ2014 and SKJ2016; to conclude that single model evaluations and the fitted solutions are sufficiently close to regard the development version estimates as being essentially similar to the benchmark version. This indicates integrity of the development version for undertaking single-species model evaluations. Results are compared among the versions and operating systems, to confirm that the development and release versions produced identical solutions. When differences are found, which can be attributable to improvements in the development version, these are accepted.

Tests using multi-species data disaggregated among species are done which entails comparing the fitted solutions of the development version code versus those solutions obtained using the corresponding data for each species fitted individually. These tests concluded that the operations applying to each population in the disaggregated model have integrity and effectively emulate the solutions obtained when each population is modelled individually. Noting that species-specific fisheries data were supplied to the models in the test data examples used. Testing was not conducted using test data for which all fisheries data were aggregated among species (or sexes).

Similarly, tests are done for deterministic and stochastic projections with the pair-wise comparisons among versions and operating systems being made.

The benchmark tests undertaken during 2017-18 are described in section 2.6. All the benchmark tests concluded that the development version conserves the existing features and can either be advanced as the new release version, or accepted for the new benchmark development version.

2. Establishing integrity of new features

This second level of testing entails a detailed examination of new features. The inputs and model configuration are customized for the new features and the operation of the new algorithms are evaluated in respect of the original formulations. During 2017-18 this level of testing was done for the new features (see section 4), to ensure the correct calculations and the expected results.

Review of Testing Framework

In January 2016 the testing framework was reviewed by project members with the following agreed tasks for improvements:

- a) Tidy up the testing framework functions and utilities so as to be as automated as possible and more user-friendly with a view to including other team members in running the tests.
- b) Upgrade testing framework functions and utilities for applicability to both single-sex and multi-sex file formats, with portability over condor.
- c) Integrate the testing framework functions and utilities into the R4MFCL package and ensure compatibility with all assessment modelling applications.
- d) Create a Github repository for the testing framework functions, utilities, and testing data.
- e) Consolidate the R4MFCL Github repository with Rob Scott as the lead developer, and add access levels to Nick Davies as a support developer.
- f) Construct a suite of routine tests for the R4MFCL package to be run following each revision to the repository, and load the updated R4MFCL package to the testing framework.

- g) Construct a single routine MULTIFAN-CL test operation (e.g. single-evaluation of a fitted test model solution) to be conducted daily and directly from the Jenkins compilation utility that returns an exit status value, with an email report sent to the project developers.

Little action has been taken on these tasks during 2017-18 and is also unlikely in the remaining part of 2018. It is suggested that they be considered in the 2018-19 work plan for the MULTIFAN-CL project.

2.9.3 Viewer

A development version of the MULTIFAN-CL viewer that can display the results of a multi-species or multi-sex application was updated as new output was added to the report files. Development of this version is ongoing since the final output format of the multi-species/sex application is not yet complete. However, substantial improvements were made for displaying results of multi-sex models.

2.9.4 Condor parallel processing facility

The Condor (www.condor.wisc.edu) facility has been used routinely for managing multiple MULTIFAN-CL model runs on a grid currently numbering more than 40 computers; being windows or linux platforms. Support for 32-bit architecture has been discontinued since MULTIFAN-CL executables in this architecture have not been produced since version 1.1.5.9. This grid enables intensive model runs for: benchmark testing MULTIFAN-CL development versions; undertaking stock assessments that entail multiple model runs (e.g. sensitivity analyses and structural uncertainty analyses), and for management strategy evaluation. During 2017-18, additional Linux Virtual Machines were added to the grid to increase the number of model runs possible using the Linux development version executable.

2.10 User's guide

A revision the MULTIFAN-CL User's Guide (Kleiber et al. 2018) has been completed that documents the developments in version 2.0.3.1. Proposed future revisions include: incorporating the suggestions arising from the Training workshop (see section 3); the recent features added to version 2.0.4.0 (soon to be released on the website); and, those in the development version. The revised version will be posted on the website <http://www.multifan-cl.org/>.

3 TRAINING WORKSHOP

The third training workshop for users of MULTIFAN-CL at the Oceanic Fisheries Programme (Pacific Community) was held on 13 – 15 March 2018, at Nouméa, New Caledonia. The aim was to achieve proficiency with the new features in MULTIFAN-CL version 2.0.4.0, particularly with the approaches for fitting to size composition data, estimating orthogonal-polynomial recruitments, applying options for function minimisers, and generating simulation model pseudo-observations (see agenda in Annex 12.2). The workshop was delivered by Nick Davies and was well attended by five core stock assessment analysts.

The morning of the first day focussed primarily on setting up all participants with the MULTIFAN-CL compilation package on Linux Virtual Machines and installing the DDD debugger. This was essential for them to follow the Modules examples and participate in the debugging module. Some of the participants were able to complete the package installation themselves by following the instruction guide supplied to them before the workshop. Having this capability for compiling the development version has since proved to be of great benefit to the stock assessment analysts by increasing the efficiency in resolving problems during model development and in updating their copies of the development version with corrections or revisions. During the subsequent days all the agenda items were covered satisfactorily.

Participants were reasonably well-prepared for “hands-on” practical sessions using the examples for implementing the new features and for debugging the examples. Most found the methods easy to follow and were satisfied with their ability to locate simple causes for program interruption.

Several issues were raised during the workshop that warrant attention:

-) pseudo-observations of tagging data:
 - o make sure the sanity check is clear when the argument `-tag_seed` is supplied but the input `*.tag_sim` file is absent.

- the algorithm for generating pseudo-observations currently does not include error due to reporting rate of recaptures. This must be included by taking the estimates from the estimation model, and in assigning observed recaptures in respect of the binomial probability of being reported.
- Profile likelihood:
 - add to the Manual descriptions of the default operation for `age_flags(173, 174)`; i.e. 1 to `nyears` for `avg_bio`, and `nysr - 6` for `relative_depletion`.
 - possible modifications to be made to `scriptloop2` include setting an input variable for `parest_flags(346)`, revise the for loop and inputs required for the `relative_depletion` option and its implications on the value of `Mult` and `parest_flags(347)`, and setting an input variable for `age_flags(172)` which is missing.

It was also requested that directories for the detailed testing of new features be made available in order that stock assessment analysts can access the examples used, R scripts for diagnostics, and the testing reports.

The key features of the Github repositories for MULTIFAN-CL and the Manual were demonstrated, in particular how to post issues yet to be resolved by members of the project team.

4 NEW FEATURES

All new features implemented into MULTIFAN-CL source code have firstly been added to the development version. Once these features have been tested for their integrity, with no impacts on existing features, then the development version is merged into the release version of the code. The current release version is 2.0.4.0. Most of the developments described below are currently implemented in version 2.0.4.0, while other have been made to the development version since January 2018 (listed in section 2.4.3) and will be merged to the next release version upon the completion of benchmark testing.

4.1 Self-scaling Multinomial plus random effects (SSMULT_RE)

4.1.1 Rationale

Ianelli et al. (2012) made recommendation “j” to: “Add an option which allows the analyst to assume a multinomial likelihood for the compositional data in the first phases and only transition to the robust normal likelihood in the later phases.” This feature has been in development since 2014, it has since formed a significant component of the developments and was mostly completed during 2016-17. While the recommendation provided the initial impetus for a development to use the Multinomial likelihood, it has since been expanded to address the three known weaknesses with this method (Francis 2014):

- It assumes the variance is proportional to the sample size, which is usually violated because variances are often larger,
- It doesn't adequately account for positive correlations, and,
- It doesn't adequately account for process error such that effective sample sizes often underestimate the true error.

Size composition data are pivotal to the tuna stock assessments using MULTIFAN-CL, in particular for estimating growth and recruitments. Improvements that might offset these weaknesses would enable the correct use of these data in the integrated modelling approach, and this became the focus of this development. The lead developer, Dr Dave Fournier, formulated an innovative size composition minimizing function, being the Self-scaling Multinomial plus Random Effects (SSMULT_RE) method. Indeed, in developing this feature, the recommendation for a Multinomial distribution for size composition data (recommendation “j”) has been extended well beyond the scope of that initially intended, by researching a substantially better approach.

4.1.2 Methods and Testing

While the SSMULT_RE was described by Davies et al. (2016, 2017), a brief overview follows. The SSMULT_RE addresses the above-mentioned three weaknesses by modifying the Multinomial to have three attributes: self-scaling properties (it estimates the effective-sample sizes, ESS), maintaining overdispersion, and to estimate autocorrelated random effects. We refer to this as the self-scaling Multinomial with estimation of random effects (SSMULT_RE), being a form of M-estimator (Huber 2009). In addition to addressing the noted

deficiencies in the multinomial, the SSMULT_RE retains the key multinomial property of being able to deal with observed zero proportions in a completely natural way. Therefore, it does not require modification of the data (by adding an arbitrary constant) to remove observed zeros, which is required for the robust-normal or logistic-normal likelihoods.

4.1.2.1 Implementation with tuna models

At both of the Developer's Workshops during 2017-18 the SSMULT_RE was refined, with testing by simulation to assess its performance in estimating the key parameters: Effective Sample Size (ESS) and correlation in the residuals; and, with preliminary testing of its implementation using real stock assessment model data. Refinements included: a small fix made to the AR(1) process for estimating the autocorrelation, ρ , of the residuals, calculation and reporting of the ESS of the fits using the robust-normal likelihood, and extending the SSMULT method that excludes the estimation of random effects (SSMULT_noRE), for application to weight-frequency data. In between the workshops, it was tested in application to two complex and large tuna models: SKJ2016 and BET2017, (McKechnie et al. 2016 and McKechnie et al. 2017, respectively).

The BET2017 example includes both length and weight frequency data, therefore providing an interesting comparison among the methods for fitting these size composition data. The implementation of the following four alternative approaches was tested:

-) **RobNorm** – robust normal+constants size composition likelihood
-) **DM_noRE** – self-scaling Dirichlet-Multinomial WITHOUT random-effects; assumed Nmax = 100
-) **SSM_noRE** – self-Scaling Multinomial WITHOUT random effects estimation
-) **SSM_RE** – self-Scaling Multinomial WITH random effects estimation assuming the AR(1) process

The RobNorm model was included for the purpose of reference, as that being the current method employed for stock assessments. The grouping of the fisheries chosen for estimating the DM_noRE, SSM_noRE, and SSM_RE parameters was identical, and parameter estimation was de-activated for groups that lacked data.

For the DM_noRE method, the length data (LF) ESS are substantially lower than the observed values for all groups (Figure 1), however the weight frequency (WF) sample size multiplier parameters were estimated at the upper bound and the WF ESS rapidly asymptotes to the assumed Nmax value (Figure 2). It was found that model solutions were sensitive to this assumption. Simulation experiments with Nmax approaching infinity indicated that high numbers of zero observations results in the DM_noRE being somewhat sensitive to Nmax.

The SSM_noRE parameter estimates were not estimated at the bounds and the LF ESS were substantially lower than the observed for all groups (Figure 1). For some fishery groups the WF ESS closely approximates the observed values (groups 1, 3, 4, and 5), while for others they are substantially lower (Figure 2).

For the SSM_RE model, some of the LF data ρ parameter estimates are away from the upper bound, whereas for the WF data almost all the rho parameters are at the upper bound. ESS for LF data were lower than the observed values for half of the groups (Figure 1), but the WF ESS either closely approximates, or exceeds the observed values (Figure 2).

Visual comparisons of the quality of fit to the size data is difficult. However, it appears the fit to the LF data differs among the models, with some improvement in the fit for the PS_UNB fisheries in regions 3 and 8 for the DM_noRE and SSM_noRE models (Figure 3). For the SSM_RE model there appears to be a better fit to the modal peak in the LF data for some fisheries. There are noticeable improvements in the fit to the WF data from the LL_ALL fisheries for the SSM_noRE model, and in some cases for the DM_noRE model (Figure 4). However, for the SSM_RE model a worse fit to the WF data is obtained for some fisheries.

The DM_noRE, SSM_noRE and SSM_RE models all produce worse fits to the other data types (effort and tagging) compared to the RobNorm model, as indicated by higher values for these objective function terms, with those for the SSM_noRE being comparably worse (Table 3). This might be expected, given that the self-scaling properties of the alternative methods used to fit size composition data affects the relative "importance" among the data types in the integrated model fit. This affect is case-specific and produces different outcomes for the BET2017 and SKJ2016 data sets used to investigate the methods. Certainly, in the case of the BET2017 data for the DM_noRE method, the assumed value for Nmax affects the model solution, producing a

substantially different model solution in respect of absolute biomass compared to the RobNorm and SSM_noRE models (Table 3, Figure 5). Overall, the SSMULT_noRE and SSMULT_RE approach produce similar solutions in respect of model quantities to the RobNorm model. For the SKJ2016 example (having only LF size composition data), the relative similarity to the RobNorm method is also different among the self-scaling methods, being most similar for the DM_noRE approach (Figure 6).

Given the problems with using the AR(1) process in estimating correlation in the random effects and recent developments of an autoregressive-autoregressive (AU-AU) method (see section 4.1.2.2) these tests must be repeated so as to better evaluate the SSMULT_RE method.

4.1.2.2 Autoregressive-Autoregressive correlation estimation

Preceding the October 2017 workshop, progress was made in developing an autoregressive-autoregressive (AU-AU) method for estimating the autocorrelation, ρ , of the residuals in the SSMULT_RE. This is akin to "long memory" processes, i.e. processes for which the autocorrelation dies out more slowly than for an AR(1) process. It was demonstrated that for the general case of k -levels of an AU-AU correlation in the special case where each level has variance = 1, the λ 's (analogous to ρ coefficients) can be solved for using the Choleski decomposition of the inverse correlation matrix. This is a tractable calculation that can be readily done within a real stock assessment application, i.e. within each minimization step. The method enables one to get a stable parameterization of the autocorrelation matrix for an autoregressive process of any order and this is done using a cubic spline parameterisation. The attractive feature of the AU-AU process is that it deteriorates rapidly from level 1 to level k , which may be more realistic for fisheries composition data. A fault of the AR(1) process (that we have investigated to date) is that the single ρ is incapable of adequately describing this decay in the correlation with time, which is really high initially, but then weak as the fish age and grow. Consequently, the single ρ estimate obtained using the AR(1) process seems to always hit the upper bound value close to 1.

Further progress was made on the AU-AU development since October 2017 with a full simulation model developed in ADMB. The simple simulation analyzer indicated the AU-AU method was largely unbiased in estimating the autocorrelation in multinomial data. A simple fish model simulator and estimator model was then developed that included an AU-AU process in the simulation data, and in the random effects component of the SSMULT_RE estimator. Preliminary simulations indicated relatively unbiased estimates for the autocorrelation and effective sample sizes.

The code used for the ADMB simple fish model simulator has been integrated into the MULTIFAN-CL code with implementation for length and weight composition data. It has been implemented in application using the BET2017 data, over a range of 5 to 7 AU-AU levels, and preliminary ρ estimates appear plausible and do not hit the parameter bounds.

In summary, the AU-AU method for estimating the autocorrelation, ρ , of the residuals in the SSMULT_RE appears valid and its implementation in MULTIFAN-CL appears successful. Preliminary fits to length and weight composition data appear plausible. It remains to be tested further with application to other tuna data.

4.2 Simulation mode

To implement full simulation capability in MULTIFAN-CL, a simulation mode has been implemented such that an operating model (OM) is used for projections to produce pseudo-observed data with specified observation and process error as might be used in management strategy evaluations (MSE). This has been described previously (Davies et al. 2017), but is described here again briefly, together with the further developments added in 2017-18. The algorithm for producing OM predictions for the projection period, is as follows:

- As normal for undertaking a projection, input data for the projection time periods (e.g. in the *.frq* or *.tag_sim* files) is supplied.
- Specific flags are set that activate the feature specify the observation error for generating simulation data for either or both the estimation or projection time periods derived from the OM quantities in each period.
- The OM simulations with projection are run and the various output reports are generated.

Essentially, the algorithm for generating pseudo-observations is activated by **parest_flags(241)** with the data to be generated specified by entries made in the projection section of the fisheries data in the **.frq* file and for tagging data in the *.tag_sim* file.

4.2.1 Generation of pseudo-observations

While this feature enables simulation data to be generated for the projection time periods, in 2017-18 it was extended to allow observation error to be applied to the OM predictions of all the data types for the estimation time periods. This was relatively straight-forward for the catch, effort and size composition data types, but entailed a more complex procedure for the tagging data type, i.e. for real tagging events which took place during the estimation model time periods.

4.2.1.1 *Simulating tagging data*

Tagging data provide an influential observation type for the integrated model fit in respect of estimating absolute abundance and exploitation levels, and are therefore important in simulation analyses using OMs. Davies et al. (2017) describe the algorithm and initial stages of implementing this feature, and in 2017-18 it was extended to produce pseudo-observed tagging data for both the projection and estimation model time periods. Broadly, the algorithm for generating simulation tag recapture pseudo-observations was as follows:

-) Assign the release sample to an initial tag release cohort of numbers at age; this is based upon the predicted catch-at-age for the fishing incident associated with the simulation release event.
-) Perform catch calculations upon the simulation tagged population. These calculations are based upon the existing code for tag groups within the estimation period to ensure consistency. This generates the predicted tag recaptures for the simulation tag release groups.

The extended feature entails two components:

-) Simulated pseudo-observations for real tagging events that occurred during the estimation model time periods (**sim_realtag**)
-) Simulated pseudo-observations for virtual tagging events to occur during the projection model time periods (**sim_tag**)

For the **sim_realtag**, the tag release samples are as occurred during the real tagging events that were used for the estimation model fit; while for the **sim_tag** the tag release samples are virtual samples and are specified by the user. Regarding the estimation time periods, it was decided to use the structure of the historical data as it occurred, i.e. the same tag release numbers by size, and to assign observation error in the corresponding data that goes into the respective likelihood terms. For the **sim_realtag** case, while the method for calculating the tag releases at age was different, the algorithm for generating predicted recaptures at length was similar to that of the **sim_tag** case.

In MULTIFAN-CL observations from real tagging events are fitted in respect of age, and not in respect of their length at release. However, the observations are input in respect of their lengths. Therefore, the ages and lengths are decoupled and so it is not a simple matter of reporting the expected values for the model tag recaptures, since these are indexed only in respect of age. For projection **sim_tag** data this is managed by a conversion of ages to lengths. While the **sim_realtag** algorithm is somewhat analogous to that used for the projection tagging events, it differs in: retaining a link to the observed release length of each simulated tag recapture; and, accommodates the recaptures made for events released close to the end of the estimation period which are predicted to occur into the projection time periods.

The approach developed in 2016-17 was also reviewed to ensure the fundamental principles were sound, and some revisions were made to the algorithm for creating the multinomial distributions for the probabilities of recapture. An exception was noted for the case of pooled tags, i.e. tagged fish that are aggregated into a “pooled” tag group after a user-specified period at liberty. Given that the aim is to generate pseudo-observations that resemble real observations, simulated data for a pooled tag group would be inconsistent because these do not exist in the real observations (where all observed recaptures relate to a specific release event). A sanity check was therefore added such that **age_flags(96)** must be set very high to avoid simulation tagged fish from entering the pooled tag group.

A key aspect of this feature developed in 2017-18 was to include error in respect of the fishery-specific reporting rates. For estimation period tagging events (`sim_realtag`) the estimated reporting rates are available, while for the projection events (`sim_tag`) the reporting rates must be assumed. In both cases, it is implemented by “rolling” into the simulation multinomial probability of a tag recapture (so that an extra binomial simulation is not needed). In the `sim_tag` case a matrix of simulation reporting rates specific to fishery and tag-event in the projections is specified at the bottom of the input file `.tag_sim`. This enables total flexibility in the reporting rates to be assumed in the simulation projections.

The `sim_realtag` feature simulates pseudo-observations based upon the operating model predictions of the probabilities of recapture given the actual release samples. In testing the feature, valid comparisons can therefore be made between the operating model predicted total recaptures for each tagging event, and the total pseudo-observed recaptures over a number of simulations (Figure 7, Figure 8). Immediately evident is the variability in the pseudo-observations attributable to the multinomial probabilities of recapture. The correspondence between the operating model predicted total recaptures and the range, and mean, of the simulated total recaptures is very close for all tagging events. For several events the range of simulated total recaptures is a little higher than the operating model predictions, and this is probably because the simulation recaptures include those occurring after the final time period of the estimation model, i.e. during the projection time periods. Overall, the result indicates the method is operating as intended.

4.2.2 Optimising simulation operations

OM simulations may be computationally intensive, particularly when implemented as part of MSE using a range of OMs, harvest strategies and over a large number of simulations. Opportunities to optimise these processes were explored in respect of the OM simulation procedures and the reports being generated.

Preceding the simulation loop, the OM calculation was optimized to exclude the initial function evaluation and derivative computations, and to simply undertake the dynamic model calculations. These calculations will otherwise be done by default during an `feval=0` as part of the minimisation procedure that precedes the stochastic simulation loop. This will be of particular benefit for the MSE framework where each evaluation entails a single simulation that includes the preceding `feval=0` run. Note that within the simulation algorithm these calculations are also not performed (is not called within the `minimising_routine` and `gradient_switch=0`).

Usually the binary file `*.hes` from a model solution may typically be very large and porting it over networks or the internet to remote hosts may be difficult. An improvement was made that removes the requirement for it to be available when generating the inputs for stochastic simulation projections. This entailed certain processes being conditional upon `parest_flags(237)==1` i.e. when error in terminal numbers-at-age is to be assigned. Generating stochastic inputs to include variability in terminal numbers at age, activated by `parest_flags(237)==1`, does require the `*.hes` to be present in the working directory. This change also applied to the usually large files: `*.dep` and `*.dp2`. Care was taken to ensure this optimisation had no impact on calculating derivatives for the dependent variables as is needed for obtaining the variances of a model solution, i.e. when not undertaking simulations, that do require these files to be present.

By default, reports of simulation model quantities are generated ("`other_projected_stuff`", "`projected_randomized_catches`", "`projected_randomized_catch_at_age`"), which over a large number of simulations may comprise a large amount of data. When simulations are run on remote hosts, this can be prohibitive for data transfer over the internet or within networks. A `parest_flags(191)` was assigned to activate these detailed reports to be produced, otherwise the default is for no reports to be produced. This flag is compatible with the operation of the other report control `parest_flags(186-190)`.

4.2.3 Status and future work

In summary, the status of implementing the simulation mode feature is presented in Table 4. Currently, pseudo-observations can be generated for all data types and for the estimation and projection time periods, except for age-length data. Process error is currently operating only for stochastic recruitments. The next stage for this feature is to extend process error for: autocorrelation in recruitments, overdispersion in the probability of tag recaptures, selectivity deviates, and to complete the testing for process error in the effort deviates and

terminal numbers at age. These components will most likely improve the “reality” of pseudo-observations generated for input to MSE studies.

4.3 Maturity at length

4.3.1 Rationale

The impetus for this feature was from a recommendation “k” of the BET Peer Review Panel (see section 2.8):

When the available maturity data are based on length, this should be input to the model rather than maturity-at-age. Converting to ages should be done within the model. Presently, the maturity-at-age is based on a fixed age-length relationship.

This approach avoids potential for bias in the calculation of spawning biomass when the growth function is estimated within the model fit, and the resulting solution differs from that used to externally to convert maturity at length to maturity at age for the assumed and “fixed” ogive subsequently input to the model. This feature is defined here as **mat_len**.

4.3.2 Methods and Testing

The **mat_len** feature entails input of the vector of maturity at length to MULTIFAN-CL that is derived from the "spawning potential" at length calculated using the length-specific vectors for:

-) Maturity ogive from ovary histology
-) Fecundity
-) Spawning fraction
-) Proportion females

The maturity at length ogive is the normalised product from these vectors calculated external to MULTIFAN-CL and is input from the ***.ini** file. This ogive is then converted to a vector in respect of ages for application in the age-structured model calculations, which is done using the model's growth function (either estimated or fixed) for the mean length-at-age and standard deviation. The conversion was done by two possible approaches using cubic spline functions:

-) Simple spline – simply evaluates the cubic spline function at the mean length at age as calculated from the von Bertalanffy curve (**simp_spln**)
-) Weighted spline – evaluates the cubic spline function at the mean length at age as calculated from the von Bertalanffy curve, and assumes that the length at age is normally distributed around the mean length; a rough integration over the length distribution is performed with steps equal to 0.5*standard deviation (**wtd_spln**)

The resulting maturity at age ogive is then used for the spawning biomass calculations within the model. Noting that, with estimated growth varying with each model evaluation during the minimisation procedure towards a solution, the maturity at age will vary with each evaluation. Whereas the original method preceding this feature maintains a constant maturity at age throughout the minimisation procedure.

Code development of **mat_len** required a differentiable **max()** function required for normalising the maturity at age ogive calculated using the weighted spline method. A formulation was developed and added to the code which produced correct derivatives. Activation of the **mat_len** feature is achieved by setting **age_flags(188)**.

Testing the **mat_len** feature entailed comparing the following two methods:

-) The maturity at age input to the BET2017 model as calculated externally based upon an assumed growth curve (the previous method).
-) The maturity at age derived using the **mat_len** feature that converts from the input maturity at length ogive using the growth curve as estimated for the BET2017 model solution.

The maturity at length ogive input for the BET2017 example model is shown in Figure 9. Starting from the BET2017 diagnostic case model solution (**'refcase'**) **.par** file, minimisations were performed that implemented the **mat_len** feature using both the approaches: **simp_spln** and **wtd_spln**. In both cases the

minimisation was rapid, achieving convergence within 16 – 17 evaluations. The objective function and gradients were improved relative to that of a single evaluation indicating the BH-SRR was re-fitted to a new solution.

The maturity at age ogives differed among the models, with that of the refcase having a declining right-hand limb, while those of the mat_len feature being reasonably non-decreasing, and overall having a higher relative age at maturity (Figure 10). The expected effect of these differences in respect of this example is that at low fishing mortality states, models using the mat_len feature will have relatively higher spawning biomass; while, at higher fishing mortality states, these models will have relatively lower spawning biomass.

This was evident as the most notable effect, that while absolute values of total biomass were unaffected, absolute values of spawning biomass for the matlen models were lower. Relative to the refcase, absolute values of spawning biomass, dynamic and equilibrium, were lower by 10 to 18%, simply due to the increased average age at maturity for the simp_spln and wtd_spln models (Figure 11). Hence, the mat_len feature influences only the BH-SRR fit and estimates of stock status relating to spawning biomass.

The second effect of note is the decline in spawning biomass status relative to the unfished state, e.g. SBcurr.SBcurrF0, which has declined by around 7 – 9%. This is due to the difference in the right-hand limbs of the maturity at age ogives, where SB_F0 will be lower for the refcase model, and consequently the estimated stock status will be higher relative to the matlen models. Note however, that the estimated stock status relative to SB_MSY were almost identical among the models, since at this MSY level, the fishing mortality is relatively high and so the difference in the equilibrium and absolute spawning biomasses will be similar among the methods.

There was almost negligible effect due to the matlen feature on the equilibrium yield estimates, with MSY being nearly identical, with recruitment estimates also being essentially identical. The higher maturity at age of the matlen models results in a lower spawning biomass per recruit, and because recruitment is unchanged, this is illustrated by all points on BH-SRR being shifted to the left, at a lower spawning biomass (Figure 12). Given the same absolute recruitment, therefore MSY is essentially unaffected.

There are a couple of reasons for the differences between the BET2017 refcase and the mat_len maturity at age ogives:

-) The input maturity at length ogive may not be the same as used for the BET2017 refcase
-) The method used for BET2017 refcase was iterative and the growth curve used for calculating the refcase maturity at age ogive may not be the same as that in the final refcase model that was used for the mat_len models.

Ultimately, the main cause of the difference is in the growth curves used by the two approaches, and the effects of this highlights the improvement achieved by implementing the mat_len feature that removes this potential source of error.

4.4 Orthogonal-Polynomial recruitments

4.4.1 Rationale

There is a tendency for some complex and large fish models to estimate local minima, such that alternative minimization methods (e.g. quasi-Newton or limited memory Newton) may follow different paths, and hence find different maximum likelihood estimates (MLE). For a “well-behaved model”, the MLE should be approximately the same irrespective of the minimizer type used. This has been demonstrated for some examples, with almost identical MLEs being obtained from three alternative minimisers. However, in an attempt to address the problem for complex models, a method has been developed that may make such models more robust by reducing the number of important parameters. This method uses orthogonal polynomials for estimating the recruitments in each region and time period (**orth-poly**). Although, the method was implemented some years ago in proof-of-concept into the MULTIFAN-CL code, it had not undergone testing, nor had it been implemented in any assessment models. During 2017-18, the method was reviewed, improved and fully tested, with the capability of it being extended for the multi-species/sex cases.

4.4.2 Methods and Testing

The feature is an alternative to the “standard” recruitment parameterisation that estimates a mean and the deviates for temporal and spatial variability. Rather, this feature uses a linear hierarchical model for the polynomials of the four possible levels: year, season, region, and season-region interactions, and calculates the Gram-Schmidt orthogonalisation basis matrix.

$$R_{y,r,s} = P_y + P_r + P_s + P_{r,s}$$

Where $R_{y,r,s}$ is the recruitment in year y , region r , and season s , and P_i is the polynomial for the i^{th} level. The activation and degrees of the polynomial in each level is set using `parest_flags`. For the multi-sex case the orthogonal polynomial parameters are shared among the sexes, which invokes a 50:50% sex ratio in the recruitments implicitly.

The design for testing orthogonal polynomial recruitment parameterisation entailed three stages:

-) Detailed and stepwise implementation using the SKJ2016 data
-) Implementation using the YFT2017 and BET2017 data
-) Implementation within a projection model using the SKJ2016 data for deterministic and stochastic projections.

Specific results for only the first stage are presented here as a demonstration of the feature. Results for the other stages were positive, indicating the feature has been successfully implemented and operates as intended. Comparisons were made relative to a model using the “standard” method with independent recruitments being estimated, denoted as Refcase.

The relative effects of increasing the polynomial levels upon the complexity of estimated recruitments showed that, while for the low number of levels the complexity is substantially reduced, when all four levels are included, (with the assumed number of degrees = 20), there is a close approximation in the complexity obtained for the Refcase model (Figure 13).

A slightly longer minimisation time was required to fit the orth-poly method, around 17.5 hrs versus around 12 hrs for SKJ2016, and for YFT2017 a substantially longer time than for the refcase fit.

In the cases tested, around 700 to 1800 fewer parameters were estimated, and the AIC indicated an improved model (i.e. lower complexity with a comparably "good" fit). In the SKJ2016 case this represented a **62% reduction** in the recruitment parameters.

In all cases a change was evident in the relative importance among the data types in the integrated fit when using the orthogonal polynomial recruitment parameterisation. This may be due to the limited “flexibility” of the parameterisation in fitting the size composition data, which causes a worsening in the fit to this data type. This effect was most pronounced for the YFT2017 model, where a somewhat worse fit to the size composition and effort data is offset by an improved fit to the tagging data, resulting in lower average recruitments, more rapid growth for young age classes, and notable differences in management-related quantities. This effect was less evident for the SKJ2016 model, where a comparable solution was found (Figure 14).

It is often the case that when low relative weight is assumed for the BH-SRR penalty term, a lack of normality in the BH-SRR fit results. Consequently, when using the BH-SRR deviates in stochastic projections this results in notable differences in the mean of stochastic projections from that of the deterministic projection. The orthogonal polynomial parameterisation may worsen this effect when a very low BH-SRR penalty is assumed due to its "inertia" in estimating the recruitments. This effect might be offset by increasing the polynomial degrees.

In the case where low relative weight on the BH-SRR penalty term was assumed (i.e. YFT2017, $wt = 0.01$), instability in the minimisation was found when using the orthogonal polynomial parameterisation. While a reasonable solution for the dynamic model quantities and convergence was obtained, an implausible fit to the BH-SRR function occurred. It was remedied by resuming the fit starting from a reasonable value for beta. One should therefore use caution when assuming a very low weight on the BH-SRR term together with the orthogonal polynomial parameterisation for recruitments.

4.5 Hessian diagnostic

During the 2018 assessments, the need for a MULTIFAN-CL model diagnostic of a non-positive definite Hessian solution was identified. Existing in MULTIFAN-CL were the calculations for the covariance matrix being the product of the inverse Hessian and gradients of the dependent variables. This is done because the inverse of the Hessian matrix approximates the variance/covariance matrix of parameter estimates. Then the Choleski decomposition of the covariance matrix is calculated, and where the determinant has a near-zero value, this indicates a non-positive definite; i.e. the parameters are strongly correlated. If this calculation fails, it can be used as a diagnostic for the matrix not being positive definite. An alternative was added for a new option to use the singular value decomposition (SVD) to calculate the eigenvalues and eigenvectors of the Hessian. For really difficult ill-conditioned symmetric matrices the SVD is more stable. The feature uses the optimized routines in the OpenBLAS dependent libraries and is consequently more efficient than the ADMB decomposition routines, as well as identifying non-positive eigenvalues. This new feature is activated using `parest_flags(145)=9` and generates a report **.svd_report* with the first line being the eigenvalues and subsequent lines (npars) being the matrix of the parameter-specific eigenvectors.

The “rule of thumb” diagnostics that an analyst can use to identify a non-positive definite Hessian solution and diagnose the influential parameters are:

-) The eigenvalues are in descending order. Check for 0 or negative values that indicate a non-positive definite Hessian, or a parameter that is never used.
-) Take the top highest eigenvectors and produce plots for each vector that illustrate the most influential parameters, and sort for the index that links it to the parameter names.
-) Do this for all the highest rows and unique parameter names over all the rows, this identifies the most influential parameters in the fit.

The minimum, mean and maximum eigenvalues for a test example (SKJ2016) were: 2.05617e-11, 46.7, and 1068.7, respectively, indicating a positive-definite solution as there were no zero values. The values for the first row of the eigenvector matrix indicate the most influential parameters that were identified as being the movement and effort deviate coefficients (Figure 15).

5 OTHER ENHANCEMENTS AND BUG FIXES

5.1 Shared effort deviates in the multi-sex model

During the development of a south-west Pacific swordfish stock assessment model (Takeuchi et al. 2017), an enhancement to the feature for a sex-disaggregated model was identified. For instances where no sex-disaggregated data are available for fitting a multi-sex model, the model is not informed in respect of the region-specific sex-ratios. In this case, it may be desirable to constrain the sex-specific effort deviates to be similar or shared among the sexes. This may avoid implausible estimates in preventing the model from compensating for the shared average catchability by having strongly positive or negative effort deviate coefficients (`effort_dev_coffs`).

A special case was made in the code that transfers the `effort_dev_coffs` to and from the parameter x-vector during minimization. It was extended to allow for grouping flags such that the respective fisheries corresponding to the two sexes could be grouped to share the same parameters. The `fish_flags(79)` were assigned to this task. A proof-of-concept test using the SWO multi-sex model was positive (see Takeuchi et al. (2018) for more information).

5.2 Simulation mode reports

Additional reports were generated when running MULTIFAN-CL in simulation mode as may be required for particular performance indicators relating to projection model status in respect of F_{MSY} or the unfished spawning biomass. Following the yield calculation for each projection simulation, a file is generated (**Fmults.txt**) containing the fishing mortality multiplier (Fmult) and several other MSY-related quantities. Also, report files for the fished and unfished spawning biomass in each time period are generated (**projected_spawning_biomass** and **projected_spawning_biomass_noeff**, respectively), having a similar format to the reports containing population numbers at age (e.g. **projected_numbers_at_age**).

5.3 Autocorrelation in recruitment

In 2016-17 an implementation was developed for estimating auto-correlation in estimated recruitments (Davies et al. 2017). In 2017-18 revisions were made to the feature such that: the autocorrelation, ρ , of the log-normal recruitments residuals from the BH-SRR is estimated as a free parameter; and, a correction was made to the calculation for the AR(1) process penalty term. The effect of the revisions was tested in a comparison of models fitted with:

-) no estimation of autocorrelation in BH-SRR deviates, only the moment estimate is calculated from the estimated deviates.
-) estimation of ρ for the BH-SRR deviates as a free parameter, and with the AR(1) penalty replacing the log-normal assumption for the BH-SRR relationship.

This produced minimal effect on the SKJ2016 derived quantities (<0.25%) and minimal difference to moment estimate of ρ . The tests over three tuna model examples highlighted several considerations:

-) The absolute value of the moment estimate was low, with the highest being around 0.3 for YFT2017, but the estimated ρ for that case was low (0.006).
-) Realistically there is little effective difference in terms of an autocorrelation between $\rho = 0$ or 0.3, and therefore little is gained in improving the model fit by estimating the additional parameter having such a low value.
-) It is possible that adding the AR(1) penalty (in some cases quite large ~ 43.2) to the integrated fit, may have altered the relative influence of particular data types on the final solution. Therefore, given the minimal effective change in the moment estimate for ρ , the differences among the models may not be purely due to estimating the additional single parameter.

These should be taken into account when investigating autocorrelation in recruitments using this feature.

5.4 Selection of minimizing method

A selection of minimizing methods has historically been employed in MULTIFAN-CL, comprising variations on the quasi-Newton (QN) approach, with the differences being in the implementation. At each i^{th} minimization step, the calculation has the step vector s_i and the gradients $g_{\{i+1\}}$ and g_i . The difference $g_{\{i+1\}} - g_i$ indicates information about the Hessian in the s_i direction. The Hessian approximation is updated using this information. While for the limited memory Newton (LMN) approach, the more recent g_i and s_i are saved, and used to update the search direction, instead of calculating a new Hessian approximation at each step. This may improve minimization efficiency, depending upon the specific case of the integrated model fit, such as when *npars* is large because less memory is stored and quicker minimisation steps are taken. For this approach, the memory held between minimisation steps is specified by defining the number of steps saved, which affects the “precision” of the minimization.

These methods have been implemented in the code historically, but in 2017-18 the facility was added for the analyst to select among the methods specifically using settings for `parest_flags(351)`. Also, a variation of the LMN was created that employs quad-double precision (16 byte floating point arithmetic) for the update

routine, thus saving more steps without running into numerical instability, and also most likely increasing efficiency.

In testing this feature, a comparison was made among the following applications of the various minimizing methods using the YFT2017 example:

-) **LMN_7**: limited memory Newton minimizer, standard double precision, nsteps = 7 (as for the YFT2017 assessment diagnostic case)
-) **LMN_400**: limited memory Newton minimizer, standard double precision, nsteps = 400
-) **LMN2_400**: limited memory Newton minimizer, quad-double precision, nsteps = 400
-) **QNEWT**: quasi Newton minimiser

For this example, the direction of minimization is such that a larger (negative) value indicates a better fit. Increasing nsteps for the LMN from 7 to 400 improved the fit (Table 5), however the double precision LMN did not substantially improve on the standard precision method. Certainly, increasing the number of nsteps for the LMN method significantly improved minimization efficiency, while the efficiency of the LMN_400 and QNEWT methods was comparable (Table 5). Overall, the quasi Newton method obtained the best solution, with a slightly better objective function value than the LMN_400 by 0.349 (Table 5). Despite these differences in the objective function values, the relative differences in the reference point model quantities were minimal (<0.6%).

5.5 Input/Output diagnostic of *.par* file

For a complex and large model, the *.par* file may become unwieldy for debugging the input and output, especially when adding parameters for new features. A debugging tool was added for code developers to assess the impacts on the parameter sequence in the *.par* file. The `parest_flags(195)` was assigned for activating this option to read/write “`check_numbers`” in the file. The default condition for the flag setting allows the file to be free of `check_numbers` which is more convenient for external utilities that read/write the *.par*.

5.6 Bug fixes

5.6.1 Fix to pseudo-observed effort with missing observed effort

An existing component of the simulation mode feature is to generate pseudo-observations of effort for the estimation model time periods without error. These may be generated from the model predicted effort that produced the observed catch, with catch errors. This is modelled in two ways depending upon the real observed effort input to the model estimation procedure:

-) Observed effort is available and input: $\text{predicted effort} = \text{effort} + \text{estimated effort_deviate}$ (where effort is the observed value input)
-) Effort is missing and a “-1” is input: $\text{predicted effort} = \text{effort} + \text{estimated effort_deviate}$ (where effort is a proxy value generated internally in the model)

In both cases the $\text{predicted effort} = \text{effort} + \text{effort_deviate}$. An error in the pseudo-observations report was found for time periods having missing effort, such that the pseudo-observation of predicted effort was always the same as the previous incident. This was because this was the proxy value assumed internally in MULTIFAN-CL. The correction was made to include the estimated `effort_deviates` in the reported model predictions for both instances of effort being observed and missing.

5.6.2 Fix to the selectivity pointer

Fishery-specific selectivities may be estimated that are time-variant, including both time-blocks and seasonality. A problem was identified with the seasonal selectivities that were incorrectly indexing the calendar months with respect to model time periods. The correction entailed substantial revisions to make the pointer assignment robust to all possible model configurations for: the first month of the first model year; the number of seasons specified within a calendar year; and, the number of recruitments in a calendar year. It was tested positive using a complex example (ALB2012) in proof-of-concept.

In addition to this correction, a table was generated that indexes the selectivity at age estimates reported in the *plot.rep* file in respect of the fisheries and the strata assumed for the time-variant selectivities if these are estimated. This may avoid potential confusion when documenting the estimates.

5.6.3 Fix to the parameter list report *xinit.rpt*

The report file *xinit.rpt* lists the independent parameters being estimated that make up the x-vector. A simple syntax error produced an omission in the report for the recruitment parameters, such that those for the standard parameterisation (mean + deviates) were no longer being reported. This occurred as part of developing the orthogonal-polynomial recruitments feature. Also, a small correction was needed to ensure no blank spaces are inserted in the parameter labels in this file. This ensures the input of the file to external utilities is robust.

5.6.4 Fix to likelihood components report

The report file *test_plot_output* contains details of the component terms of the total objective function calculated in the integrated model fit. During the objective function calculation, each component term is assigned to members of the class *ppstf*. Values for the total catch likelihood term (*tot_catch_like_by_realization*) are assigned from the normal fishing mortality run, when *fish_flags(55)=0*. However, when the zero-fishing run is performed for the unfished population calculations, i.e. *fish_flags(55)=1*, the values in *tot_catch_like_by_realization* from the run with normal fishing mortalities were re-assigned by those for the no-fishing case, and consequently the reported values were incorrect (being very large). This assignment was corrected.

5.6.5 Fix to exploitable biomass section of *plot.rep*

The report of exploitable biomass to the *plot.rep* was conditional upon the pointers for the estimated fishery selectivities. For the case where *fish_flags(71)=0*, i.e. there is only one time-block for the selectivity, the pointer was incorrectly set, such that instead of indexing the model years (i.e. each time period), it was indexing calendar years. This was corrected and the report was validated.

5.6.6 Fix to simulation recruitment deviates

For cases where the BH-SRR is fitted to recruitments estimated for each time period, e.g. quarterly, rather than the annualized recruitments (aggregated), an error was found in the application of the simulation pointer for the randomised BH-SRR deviates. For simulation projections, the pointer is input in the *simyears* file. However, in all cases the pointers were referenced to *calyr* – the calendar year, even for the case where the pointer related to each time period (e.g. year-quarters). This produced a matrix bound exception. The correction was made and tested positive using a simulation projection model having a BH-SRR fitted to recruitments for each time period.

6 APPLICATION OF NEW FEATURES

The primary new feature described in sections 4 and 5 employed in 2018 was that for the simulation mode being employed for the generation of pseudo-observations and implementation in the MSE framework by Scott et al. (2018a, 2018b, 2018c, 2018d).

Other features applied to the albacore tuna assessment and to a further development of the swordfish sex disaggregated model include:

-) Shared effort deviates among sexes (Takeuchi, Y. 2018)
-) Maturity at length (south Pacific albacore tuna, Tremblay-Boyer et al. 2018)

7 FUTURE WORK

The future work plan for the development of new features in MULTIFAN-CL is presented in Table 2, with those having high priority being undertaken in 2018-19, while others may be addressed in subsequent years. The primary focus is on the implementation of the simulation mode for its application to MSE. While some tasks have been completed during 2018, those remaining include: improving the efficiency for using MULTIFAN-CL as an estimation model (EM), and achieving realistic pseudo-observations by incorporating forms of process error such as autocorrelation in recruitments, overdispersion in the probability of tag recaptures, and random

selectivity deviates. Other developments relating to the improving stock assessment model include: informing growth estimation with tagging data, completing the SSMULT_RE (in particular testing the AU-AU method for estimating the autocorrelation, ρ , of the residuals with application to other tuna data), constraining recruitment and effort deviates, and allowing for long-term and initial tag-loss in tagging data.

Other items for developments and improvements that arose during the 2017-18 period include:

-) Commissioning a Mac OS machine on the OFP network for compiling Mac executables of release versions
-) Apportioning tagging data aggregated among sexes that avoids the implicit assumption that the age distribution by sex of tagged fish reflects that of the sex-specific regional population at age.

Of note, are to resume work on the three remaining recommendations of Ianelli et al. (2012) yet to be implemented in subsequent years:

-) Non-uniform size bins (recommendation "b")
-) Long-term tag loss (recommendation "c")
-) Tags inform movement (recommendation "d")

Tasks of immediate priority and relating to the new features recently added (Section 4) are discussed in the following section.

8 DISCUSSION

The focus in 2017-18 for implementing new features in MULTIFAN-CL was to improve its capability for stock assessment models and its utility within the MSE framework being developed by OFP. In respect of stock assessment model development, the aim was to reduce sources of bias due to: assumptions in the statistical model and the maturity at age ogive; and, excessive model complexity. A real advance was made in the self-scaling method, SSMULT_RE, with the development of an autoregressive-autoregressive method (AU-AU) for the autocorrelation in the random effects. This method appears to better estimate the autocorrelation and preliminary tests using complex examples suggest it overcomes the problems found with the AR(1) process, where autocorrelation was consistently estimated at the upper bounds. This was the last “stumbling block” in developing the self-scaling method. Substantial reductions in the total number of parameters to be estimated is now possible by using the orthogonal polynomials parameterization for estimating the recruitments. This offers a means for exploring reduced model complexity to avoid minimisation stability caused by an overly complex model. Internally calculating the maturity at age ogive given an input maturity at length vector and the model’s growth estimates serves to avoid bias caused by assuming an ogive based upon a fixed growth curve external of the model. Finally, the detailed model fit diagnostics now possible using the Hessian eigenvalues and eigenvectors offer the analyst a ready and powerful tool for identifying influential parameters and the causes of non-positive definite Hessian estimates. Collectively, these features offer real improvements for the development of stock assessment models using MULTIFAN-CL.

In respect of its utility within the MSE framework, the most significant new feature added to the simulation mode for running MULTIFAN-CL as an OM was the capability for generating pseudo-observations of tagging data in both the estimation and projection model time periods. This required discrete and complex algorithms for calculating the multinomial probabilities of recapture, including the estimated and assumed fishery- and tagging event-specific reporting rate probabilities. This offers a means for including tagging data as a component of the management strategies being evaluated. Another improvement was to optimize the performance of MULTIFAN-CL as an OM; this being essential given the computational overhead of a large scale MSE framework.

This rapid and extensive development has impacted on the project’s support role during 2017-18, as indicated by the delay in posting the next release version 2.0.4.0 (benchmark tested in February 2018), the long period since the last benchmark test of the development version and posting an updated User’s Guide. Regular benchmark testing and testing immediately following new developments is certainly best practice, which is always intended within the project, however, the momentum of the developments in 2017-18 has prevented it. It is now an immediate and important priority task.

A priority for future work in 2018-19 is to continue to develop the utility of MULTIFAN-CL for the MSE framework. Most important is to improve the “reality” of the pseudo-observations being generated from an OM using MULTIFAN-CL. This involves including sources of process error described in section 4.2.3 and listed in Table 4. Support will then be provided for the subsequent Turing Tests to be done that assesses the “realism” of the pseudo-observations, with the necessary refinements subsequently being made to the code. Another development needed is to make it feasible for MULTIFAN-CL to be used as an EM in an MSE, by reducing the computational overhead of the model fit to simulated data. An avenue of exploration for this is to fit only the parameters needed for the projection model time periods, while keep fixed those already obtained for the estimation model, and fixed at the initial fitted values.

A second priority for future work in 2018-19 is to consolidate the work completed on the self-scaling methods for fitting composition data - in particular the SSMULT_RE, that now includes the AU-AU method for autocorrelation estimation in the random effects. It is now possible to complete the testing of this method using a range of complex tuna models, and to prepare for submission and peer review, a draft paper that introduces this size-composition M-estimator. Doing this consolidates a large volume of work done over the past 3-4 years on an innovative and significant development in fisheries population modelling.

9 REFERENCES

Davies, N., Fournier, D., Takeuchi, Y., Bouyé, F., and Hampton, J. 2017. Developments in the MULTIFAN-CL software 2016-17. WCPFC-SC13-2017/SA -IP-05

Fournier, D.A., Hampton, J., and Sibert, J.R. 1998. MULTIFAN-CL: a length-based, age-structured model for fisheries stock assessment, with application to South Pacific albacore, *Thunnus alalunga*. *Can. J. Fish. Aquat. Sci.* **55**:2105-2116

Francis, R.I.C.C. 2014. Replacing the multinomial in stock assessment models: a first step. *Fisheries Research* **151**: 70-84

Huber, P.J. 2009. Robust statistics. 2nd edition/Huber, P.J., Ronchetti, E.M., John Wiley & Sons, Inc., Hoboken, New Jersey

Hrafnelsson, B. and Stefansson, G. 2004. A model for categorical length data from groundfish surveys. *Can. J. Fish. Aquat. Sci.* **61**:1135-1142

Ianelli, J., Maunder, M., and Punt, A. 2012. Independent review of 2011 WCPO bigeye tuna assessment. WCPFC-SC8-SA-WP-01

Kleiber, P., Fournier, D., Hampton, J., Davies, N., Bouyé, F., and Hoyle, S. 2018. MULTIFAN-CL User's Guide. <http://www.multifan-cl.org/>

McKechnie, S., Hampton, J., Pilling, G. and Davies, N. 2016. Assessment of skipjack tuna in the western and central Pacific Ocean. WCPFC-SC12-2016/SA-WP-04. Bali, Indonesia 3-11 August 2016.

McKechnie, S., Pilling, G. and Hampton, J. 2017. Assessment of bigeye tuna in the WCPO. WCPFC-SC13-2017/SA-WP-3, Rarotonga, Cook Islands 9-17 August 2017.

Methot, R. 2013. User Manual for Stock Synthesis – model version 3.24s. http://nft.nefsc.noaa.gov/Stock_Synthesis_3.htm

Punt, A.E. 2015 in press. Some insights into data weighting in integrated stock assessments. *Fish.Res.* (2015) <http://dx.doi.org/10.1016/j.fishres.2015.12.006>

Scott, F., Scott, R., Davies, N., Pilling, G. and Hampton, J. 2018a. Technical developments in the MSE modelling framework. WCPFC-SC14-2018/MI-IP-02, Busan, Republic of Korea, 8–16 August 2018.

Scott, R., Scott, F., Pilling, G., Hampton, J. and Davies, N. 2018b. Selecting and Conditioning the Operating Models for WCPO Skipjack. WCPFC-SC14-2018/MI-WP-03, Busan, Republic of Korea, 8–16 August 2018.

Scott, F., Scott, R., Davies, N., Pilling, G. and Hampton, J. 2018c. Performance indicators for comparing management procedures using the MSE modelling framework. WCPFC-SC14-2018/MI-WP-04, Busan, Republic of Korea, 8–16 August 2018.

Scott, R., Scott, F., Davies, N., Pilling, G. and Hampton, J. 2018d. Generating pseudo data in MULTIFAN-CL. WCPFC-SC14-2018/MI-IP-03, Busan, Republic of Korea, 8–16 August 2018.

Takeuchi, Y. et al. 2018. Testing MULTIFAN-CL functionality for multispecies/multisex assessments, using SW Pacific swordfish. WCPFC-SC14-2018/SA-IP-10, Busan, Republic of Korea, 8–16 August 2018.

Tremblay-Boyer, L., McKechnie, S., Pilling, G. and Hampton, J. 2017. Assessment of yellowfin tuna in the WCPO. WCPFC-SC13-2017/SA-WP-05, Rarotonga, Cook Islands 9-17 August 2017.

Tremblay-Boyer, L., Hampton, J., and Pilling, G. 2018. Stock assessment of south Pacific albacore tuna in the WCPO. WCPFC-SC14-2018/SA-WP-05, Busan, Republic of Korea, 8–16 August 2018.

10 TABLES

Table 1. New features added to MULTIFAN-CL with respect to their state of completion as of August 2018.

Peer review recommendations		
Task	Description	Status of completion
k. Maturity-at-length	When maturity data are based on length, converting to ages should be done within the model. Presently, the maturity-at-age is based on a fixed age-length relationship.	Development 100%; Testing 100%
b. Non-uniform size bins	Allow the length bins to be of different widths. One might, for example, want many narrow length bins for the smaller lengths, but fewer but wider length bins for the larger lengths.	Development 0%
c. Long-term tag loss	Allow for long-term and initial tag-loss. Currently initial tag-loss is implemented by reducing the number of animals tagged when inputting data to the model and no account can be taken of long-term tag-loss.	Development 0%
d. Tags inform movement	Include an option which allows the tagging data to inform movement only rather than movement and mortality	Development 0%
Other developments		
Task	Description	Status of completion
Orthogonal (structured) recruitment estimates	Add recruitment parameterization using orthogonal-polynomials.	Development 100%; Testing 100%
Simulation mode	Output of simulation operating model pseudo-observations for use in simulation model testing and as an Operating Model for Management Strategy Evaluations	Development 90%; Testing 100%
Stochastic projections	Output report of stochastic MSY quantities calculated from projection period	Development 100%; Testing 100%
Self-scaling multinomial with random effects	Add a multinomial likelihood for composition data that is modified to be self-scaling and estimates auto-correlated random effects.	Development 100%; Testing 60%
Shared eff_devs	Shared catchability/effort deviates among sexes, as it does for the average catchability. To prevent the model from compensating for the shared average catchability by having strongly positive or negative effort devs for the 2 sexes for a fishery.	Development 100%; Testing 100%
Report comments	Add comment descriptions of the selectivity parameter configurations in the output <i>.par</i> and <i>plot.rep</i> reports	Development 100%; Testing 100%
Generate stochastic inputs	Remove the requirement for the *.hes file when generating stochastic simulation inputs in the steps having <code>parest_flags(145) = 7</code> and <code>8</code>	Development 100%; Testing 100%
Optimising simulation OM	Optimize the stochastic projection model (OM) calculations to exclude the initial function evaluation, and derivative computations, to undertake simply the dynamic model calculations.	Development 100%; Testing 100%
Initial biomass approximates the equilibrium	Abundance in the first model period is equivalent to that estimated using the	Development 90%; Testing 20%

unexploited biomass	estimated BH-SRR in an unexploited equilibrium state	
Region-specific yield estimation	Calculate the equilibrium productivity and biomass from each region in a spatially disaggregated model	Development 100%; Testing 20%
Exclude reporting rates from tag predictions during mixing period	Do not to account for the estimated tag reporting rate when solving the Newton-Raphson catch algorithm for tags recaptured during the mixing period	Development 100%; Testing 20%

Table 2. New features to be added to MULTIFAN-CL in 2018-19 and those for which implementation and testing is to be completed, and in subsequent years.

Peer review recommendations		
Task	Description	Implementation
c. Long-term and initial tag loss	Allow for long-term and initial tag-loss. Currently initial tag-loss is implemented by reducing the number of animals tagged when inputting data to the model and no account can be taken of long-term tag-loss.	2018-19
Other developments		
Task	Description	Implementation
Recruitment random effects	Estimate the autocorrelation and variance of recruitment deviates as random effects rather than as independent parameters. Attempt to extend this functionality also to effort deviates.	2018-19
Constrain deviates	Apply constraints on the recruitment deviates such that the $\bar{x} = 0$. Apply also to the effort deviates for fisheries with missing effort data for the complete time series.	2018-19
Tags inform growth	Development of a feature that incorporates size data from tag recaptures to inform growth estimation.	2018-19
Outstanding testing of existing features	- Catch-conditioned estimation of fishing mortality - Reinstate the original estimation of length-based selectivities as activated by fish_flags(i,26)=3 that integrates the distribution of lengths-at-age in calculating the selectivity-at-age. (single species only)	2018-19
Self-scaling multinomial with random effects (SSMULT_RE) and Dirichlet-Multinomial	Consolidate the self-scaling methods for fitting composition data - in particular the AU-AU method for autocorrelation estimation in the SSMULT_RE. Support the submission and peer review of the draft paper that introduces the size-composition M-estimator; and application to a stock assessment example.	Complete testing 2018-19
Simulation mode	Output of simulation operating model pseudo-observations for use in simulation model testing and as an Operating Model for Management Strategy Evaluations: - implement pseudo-observations for tagging and age-length data;	2018-19
Stochastic projection functionality	- implement process error in future recruitments with application of the derived autocorrelation coefficient in historical recruitment estimates - fix a bug in generating inputs for stochasticity in N_term,age and eff_devs	2018-19
Turing test	Ensure the quality of pseudo-observations to be made more realistic by: - Including the sel_dev_coffs and eff_devs estimates in applying process error in projection size compositions and effort - Including over-dispersion error in tagging data	2018-19
EM fit only projection pars	For the "assessment" estimation model (EM) embedded in a management procedure, only fit parameters relating to the "new" data provided for the projection time periods, i.e. for effort devs,	2018-19

	catchability devs, recruitment devs, while holding all other parameters fixed at the initial values.	
OM size comps	Generate a report of the OM size compositions for projection period without error at the end of the projection period as required for deriving economics-based indicators.	2018-19
Windows executable	Improve the static build using Visual Studio 2017 to reduce the number of dependent .dll files required to be ported with the executable.	2018-19
Cobb-Douglas scalar	A biomass-related effect on catchability implying hyper-stability in catch rates.	Complete testing 2019-20
Initial biomass approximates the equilibrium unexploited biomass	Calculate the equilibrium productivity and biomass from each region in a spatially disaggregated model	Complete testing 2019-20
Exclude reporting rates from tag predictions during mixing period	Do not to account for the estimated tag reporting rate when solving the Newton-Raphson catch algorithm for tags recaptured during the mixing period	Complete testing 2019-20
Region-specific yield estimation	Calculate the equilibrium productivity and biomass from each region in a spatially disaggregated model	Complete testing 2019-20

Table 3. Comparison among self-scaling model quantities for the BET2017 example, with the % difference between the DM_noRE and RobNorm models (%diff_DM_noRE_RobNorm), and between SSM_noRE and RobNorm models (%diff_SSM_noRE_RobNorm), and between SSM_RE and RobNorm models (%diff_SSM_RE_RobNorm).

Model quantity	RobNorm	DM_noRE	SSM_noRE	SSM_RE	%diff_DM_noRE_RobNorm	%diff_SSM_noRE_RobNorm	%diff_SSM_RE_RobNorm
MSY	38390	60530	36090	38500	57.67	-5.99	0.29
Ccurr.MSY	4.262	2.807	4.427	4.203	-34.13	3.88	-1.37
Fmsy	0.049	0.047	0.050	0.051	-5.42	1.10	3.90
Fmult	1.260	2.461	1.107	1.258	95.32	-12.14	-0.16
Fcurr.Fmsy	0.794	0.406	0.903	0.795	-48.80	13.82	0.16
B0	2415000	4127000	2292000	2517000	70.89	-5.09	4.22
Bmsy	778700	1298000	724300	751800	66.69	-6.99	-3.45
Bcurr	1051942	2349279	943701	1080590	123.33	-10.29	2.72
SB0	1704000	2910000	1634000	1795000	70.77	-4.11	5.34
SBmsy	455700	769200	428100	445300	68.80	-6.06	-2.28
SBcurr	596639	1486176	541804	640740	149.09	-9.19	7.39
Bcurr.Bmsy	1.351	1.810	1.303	1.437	33.98	-3.55	6.40
SBcurr.SBmsy	1.309	1.932	1.266	1.439	47.57	-3.34	9.90
SBcurr.SBcurrF0	0.325	0.572	0.291	0.338	76.19	-10.31	4.11
SBlatest.SBlatestF0	0.361	0.592	0.329	0.366	64.06	-8.81	1.53
obj_bhsteep	0.666	0.590	0.950	0.429	-11.34	42.67	-35.55
obj_effdev	2558.312	3568.815	4774.464	2971.958	39.50	86.63	16.17
obj_catdev	178.024	202.724	211.918	181.637	13.87	19.04	2.03
obj_lencomp	-173539.167	104696.883	23132.805	1197.042	-160.33	-113.33	-100.69
obj_wtcomp	-921151.942	313238.573	20046.861	-105673.633			
obj_tagdata	4805.794	5061.678	5433.889	5596.110	5.32	13.07	16.45
obj_agelngdata	0.000	0.000	0.000	0.000			
Obj	-1086596.426	218461.027	54225.784	-95348.976	120.11	104.99	91.22
gradient	0.0095974	0.0096612	0.0085484	0.0090625	0.66	-10.93	-5.57
Lmin	22.446	22.444	20.037	19.943	-0.01	-10.73	-11.15
Lmax	151.510	151.510	151.510	151.510	0.00	0.00	0.00
K	0.073	0.073	0.073	0.073	0.00	0.00	0.00

Table 4. Status of developments for the simulation mode feature.

	Estimation Period		Projection period	
Observation error		Report		Report
Size composition	Multinomial	test_lw_sim	Multinomial	test_lw_sim
Catch and Effort	Log-normal	effort_sim catch_sim	Log-normal	effort_sim catch_sim
Tagging	Multinomial	report.realtag_#	Multinomial	report.simtag_#
Age-length	<i>Pending</i>	<i>Pending</i>	<i>Pending</i>	<i>Pending</i>
Process error				
Recruitment	-	-	Stochastic: - Absolute - BH-SRR deviates <i>(pending – include autocorrelation)</i>	plot.rep
Tagging	-	-	<i>(pending – include overdispersion)</i>	-
Effort deviates	-	-	Stochastic <i>pending testing</i>	.par
Selectivity deviates	-	-	Stochastic <i>pending</i>	.par
N _{age} in terminal period	-	-	Stochastic <i>pending testing</i>	plot.rep

Table 5. Comparison among YFT2017 model solutions and minimiser efficiency for alternatives of the quasi Newton and limited memory Newton minimisers.

Minimiser	Objective function	Convergence time
LMN_7	1202570.01603299	50 hrs 17 mins
LMN_400	1202570.28171878	14 hrs 54 mins
LMN_p2_400	1202570.20790068	14 hrs 57 mins
QNewt	1202570.63109774	13 hrs 00 mins

11 FIGURES

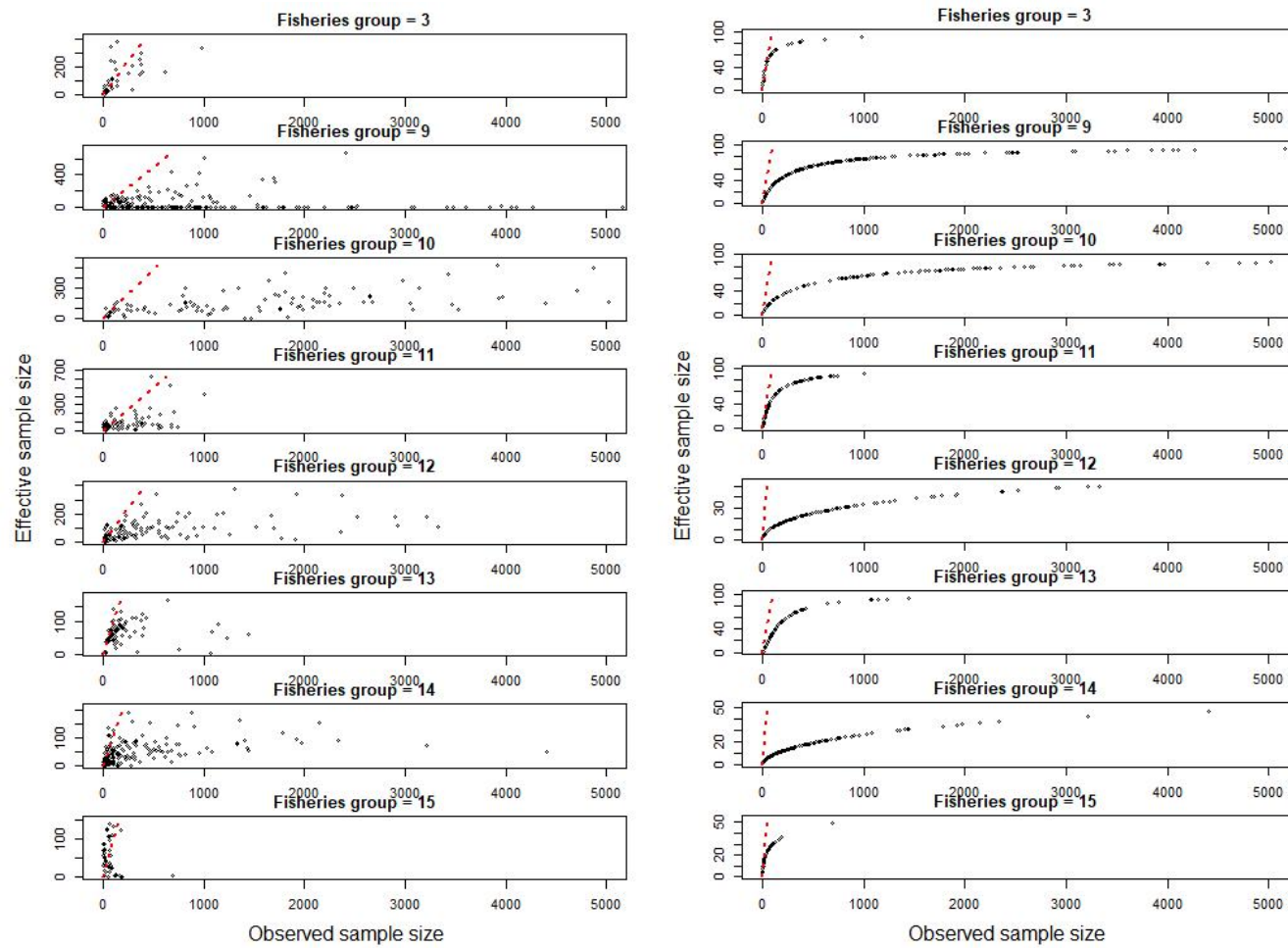


Figure 1. The estimated effective sample sizes for the RobNorm (left panels) and DM_noRE (right panels) models for the BET2017 example, versus the observed (actual) sample sizes in respect of each fishery grouping for length frequency samples. The dashed red straight line is the 1:1 ratio, and each circle represents an individual sample.

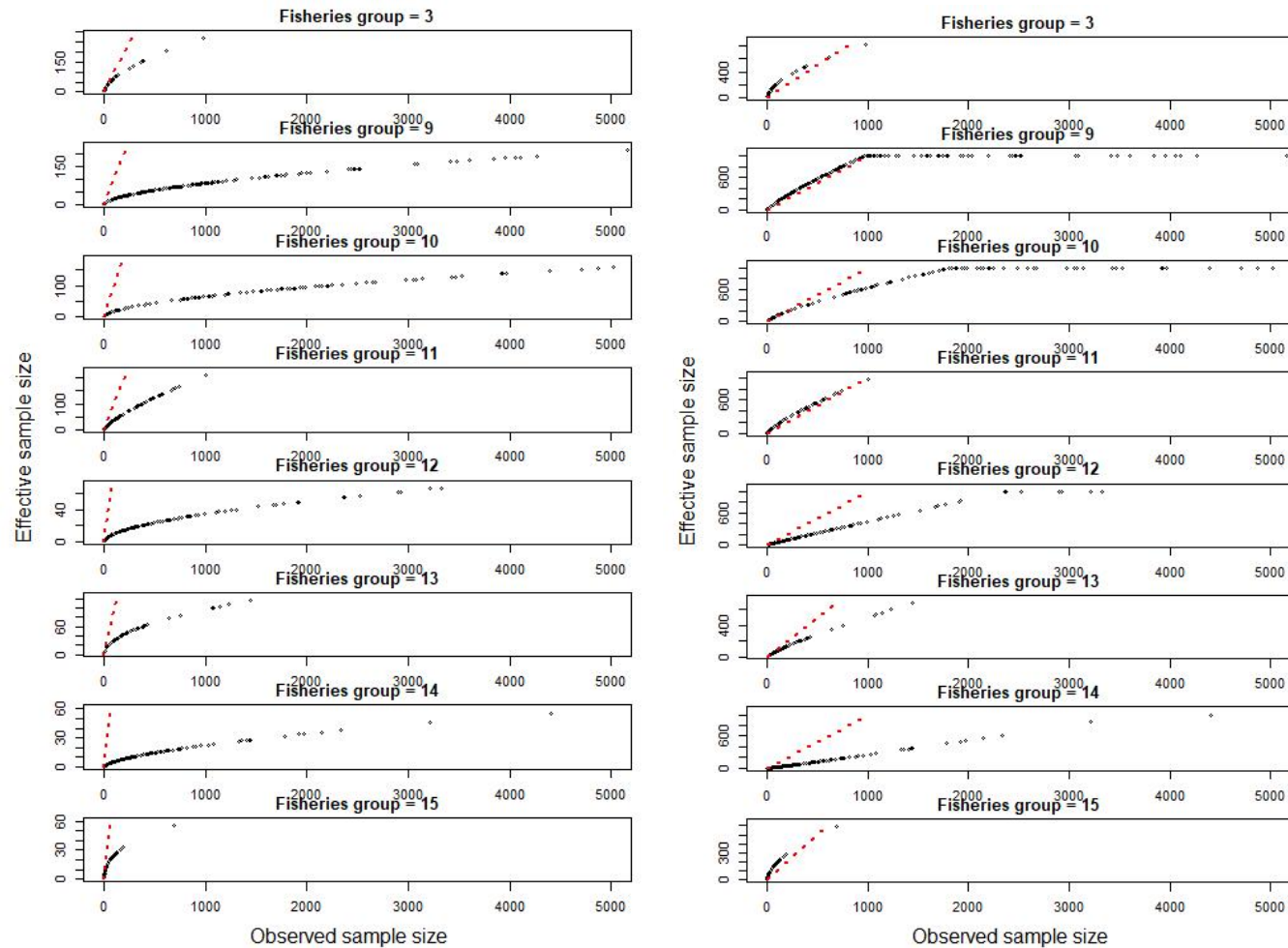


Figure 1 cont. The estimated effective sample sizes for the SSMULT_noRE (left panels) and SSMULT_RE (right panels) models for the BET2017 example, versus the observed (actual) sample sizes in respect of each fishery grouping for length frequency samples. The dashed red straight line is the 1:1 ratio, and each circle represents an individual sample.

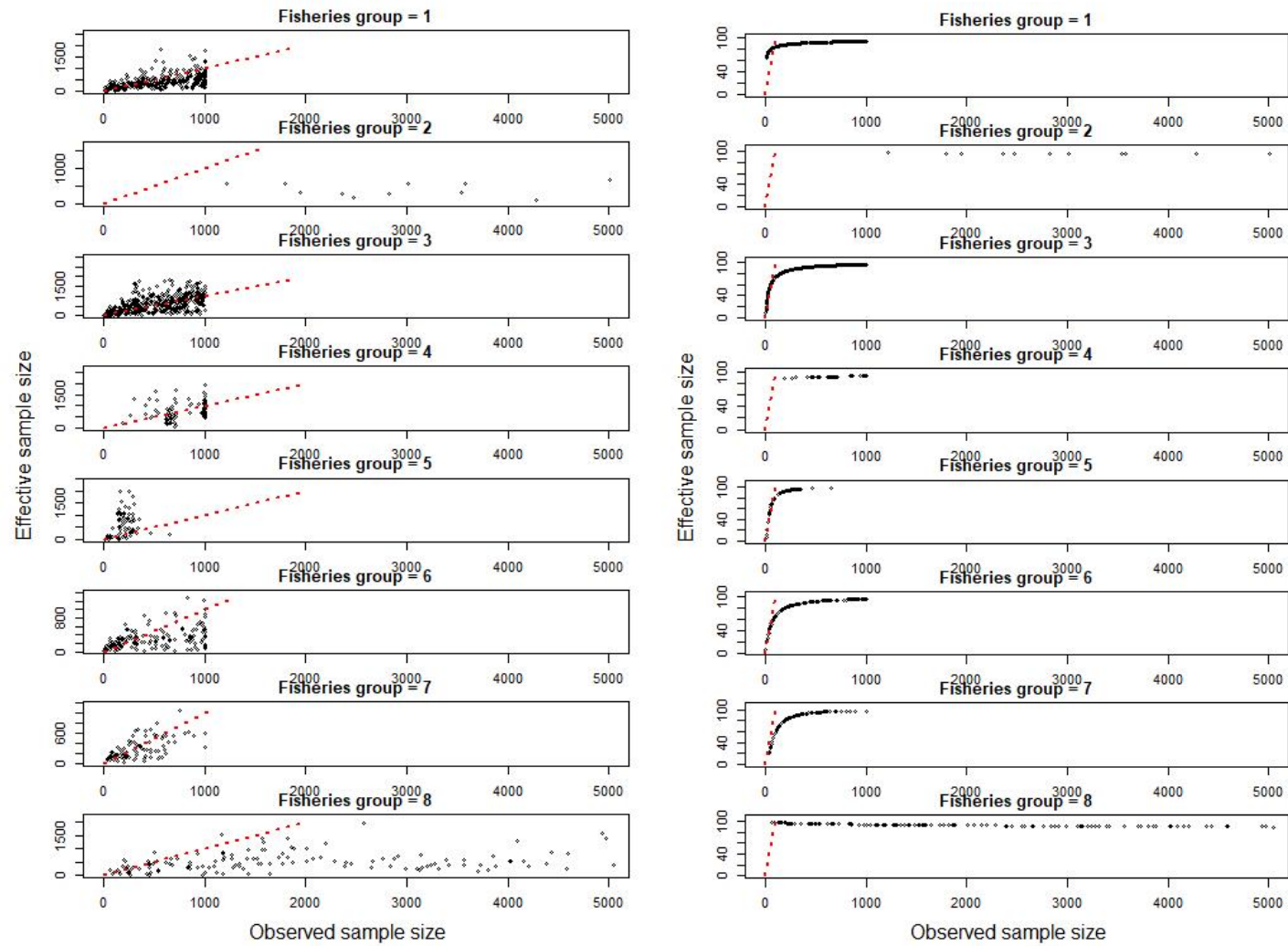


Figure 2. The estimated effective sample sizes for the Robust Normal (left panels) and DM_noRE (right panels) models for the BET2017 example, versus the observed (actual) sample sizes in respect of each fishery grouping for weight frequency samples. The dashed red straight line is the 1:1 ratio, and each circle represents an individual sample.

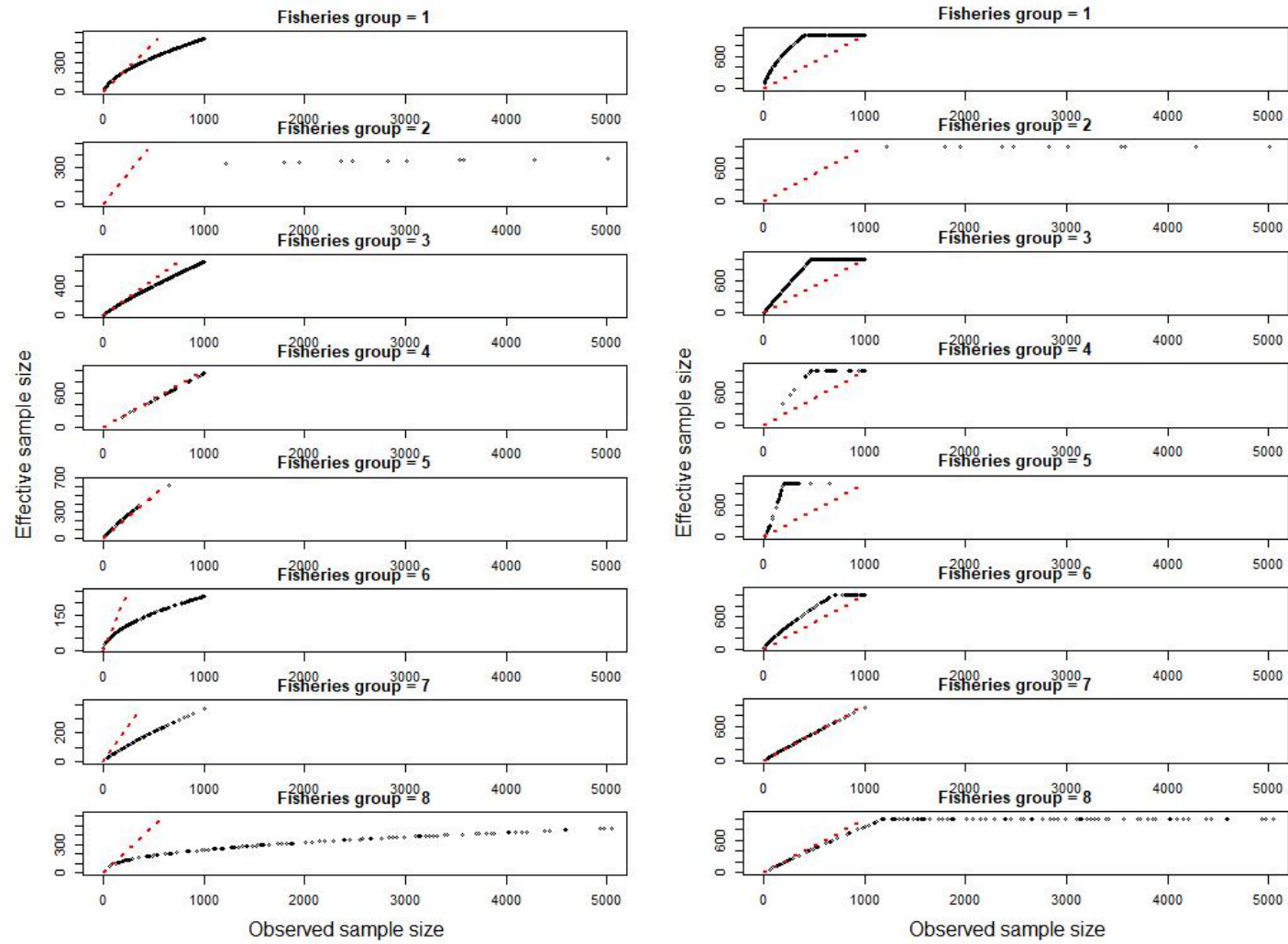


Figure 2 cont. The estimated effective sample sizes for the SSMULT_noRE (left panels) and SSMULT_RE (right panels) models for the BET2017 example, versus the observed (actual) sample sizes in respect of each fishery grouping for weight frequency samples. The dashed red straight line is the 1:1 ratio, and each circle represents an individual sample.

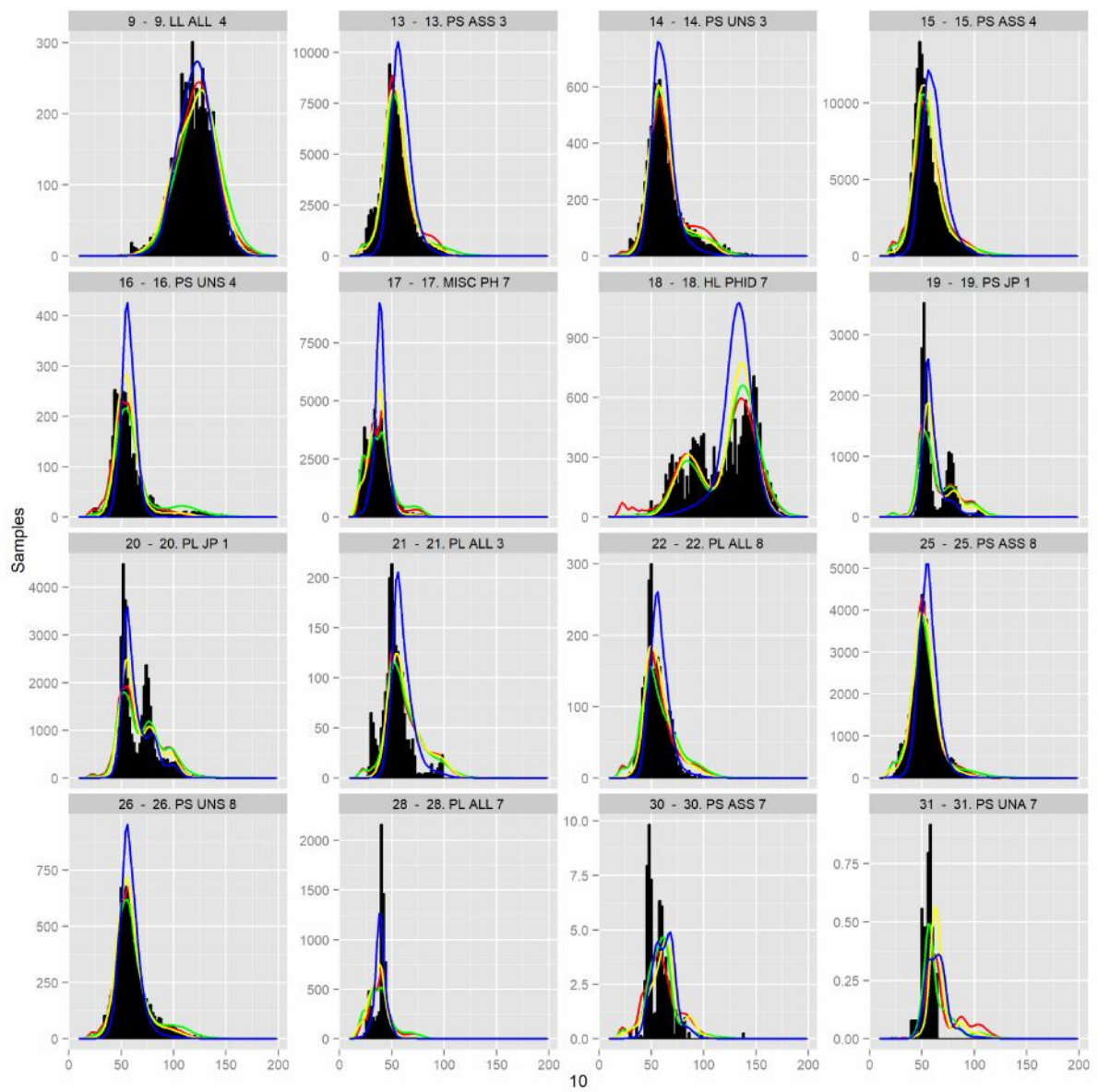
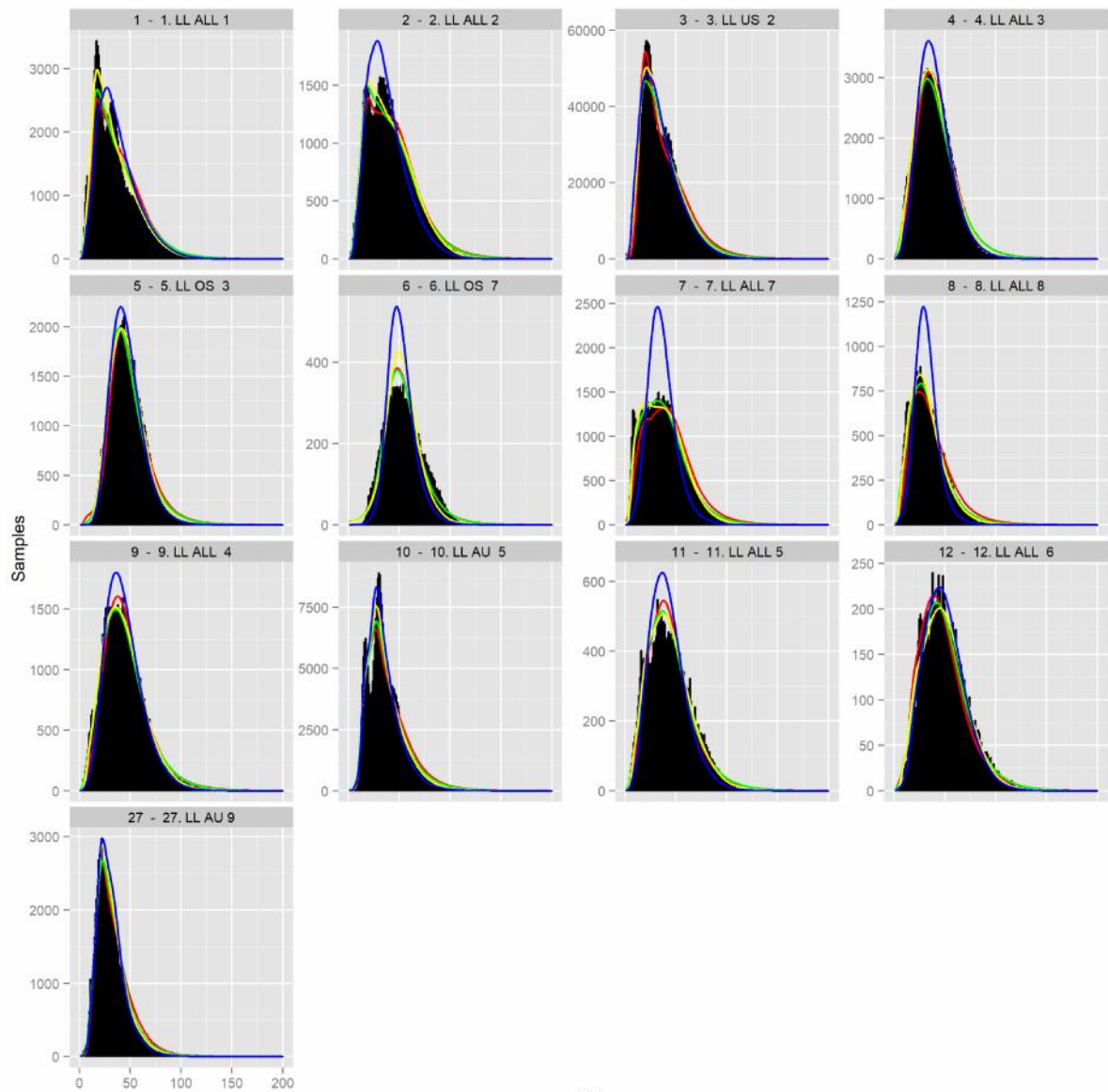


Figure 3. The fit to the aggregated observed length frequency data from each fishery (histograms in the individual panels) for the RobNorm (red lines), DM_noRE (green lines), SSM_noRE (yellow lines), and SSM_RE (blue lines) models.



10

Figure 4. The fit to the aggregated observed weight frequency data from each fishery (histograms in the individual panels) for the RobNorm (red lines), DM_noRE (green lines), SSM_noRE (yellow lines), and SSM_RE (blue lines) models.

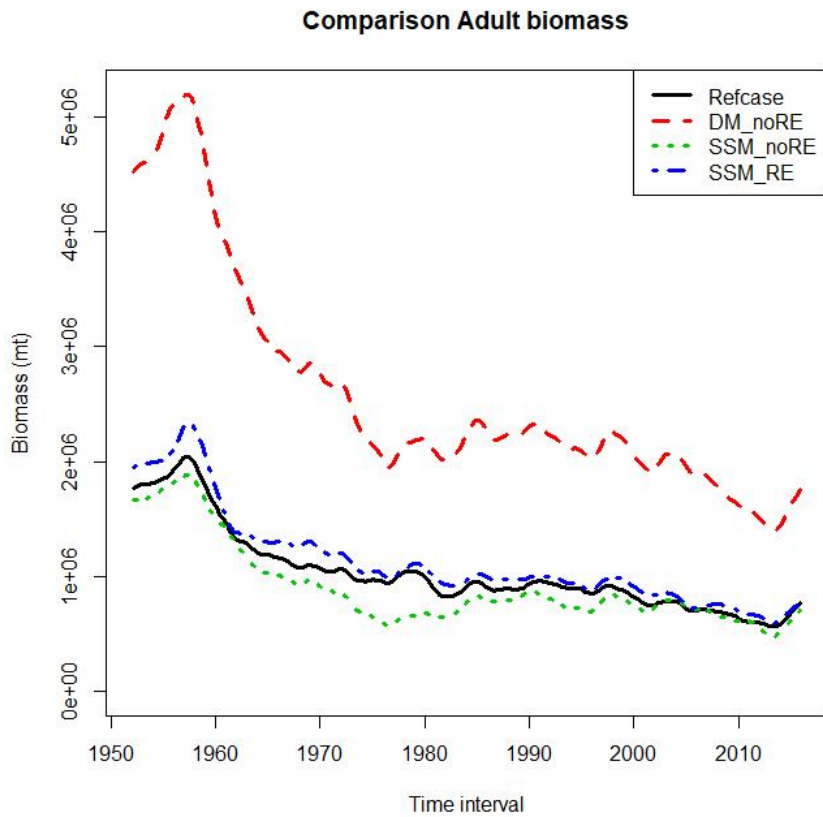


Figure 5. Comparison of BET2017 spawning biomass (t) for the RobNorm (black line), DM_noRE (red lines), SSM_noRE (green line), and SSM_RE (blue line) models.

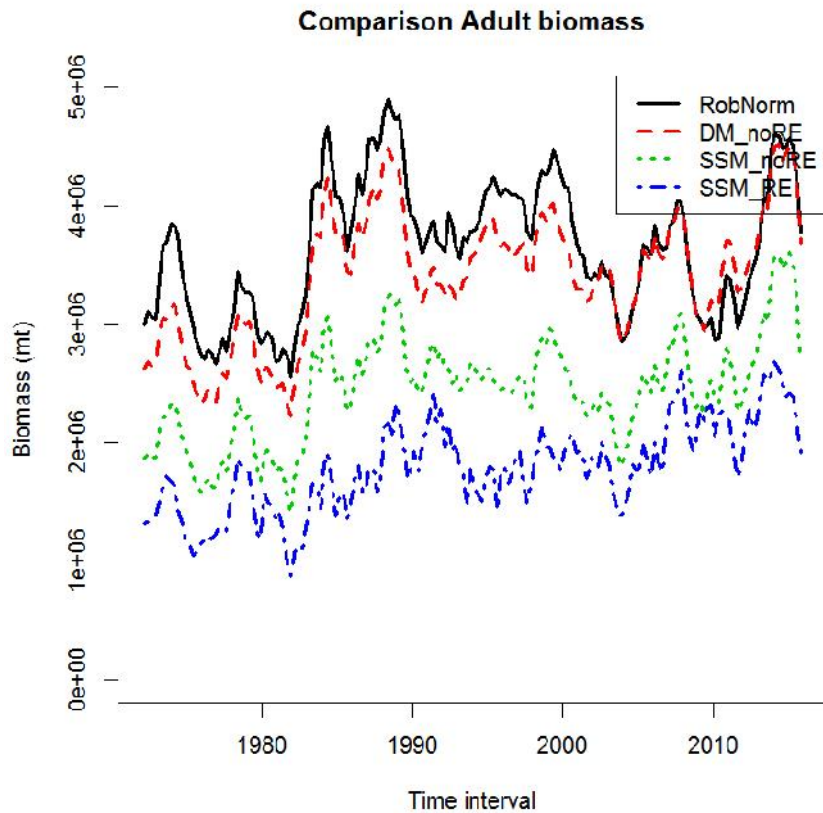


Figure 6. Comparison of SKJ2016 spawning biomass (t) for the RobNorm (black line), DM_noRE (red lines), SSM_noRE (green line), and SSM_RE (blue line) models.

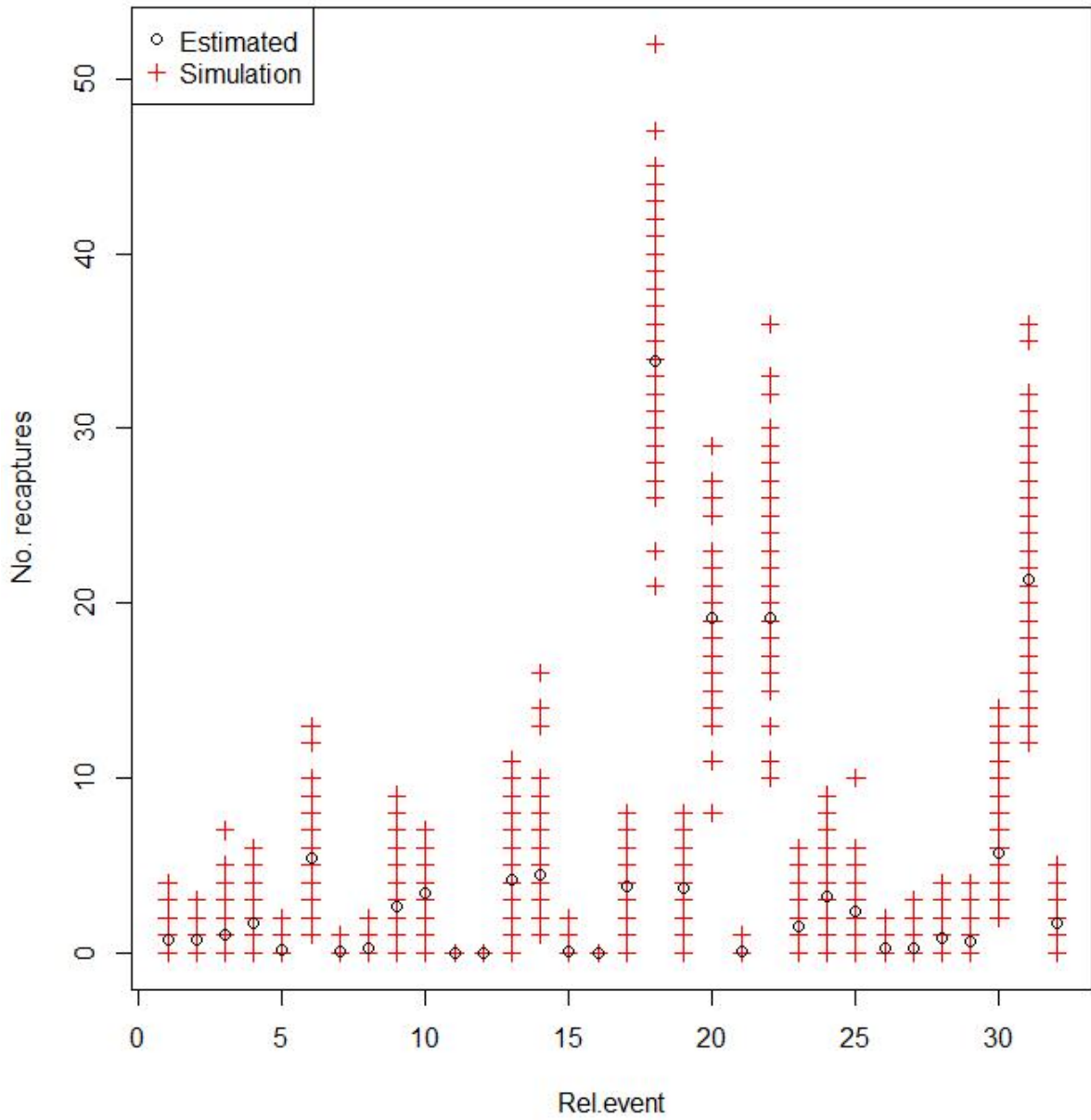


Figure 7. Total recaptures of tagging events that occurred during the estimation period as predicted by the operating model (Estimated) and as simulated from 100 simulations (Simulation) for the ALB2015 example.

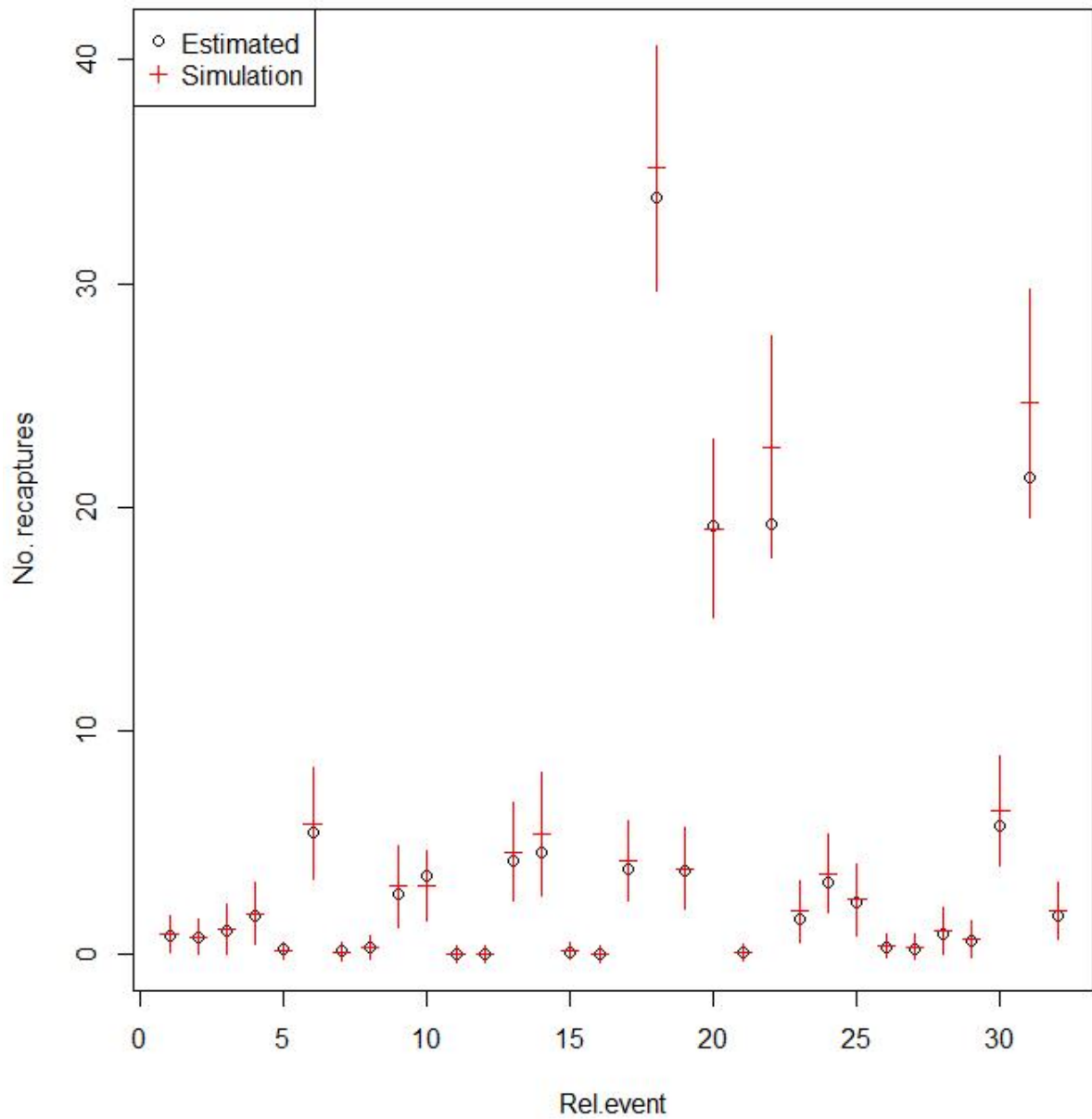


Figure 8. Total recaptures of tagging events that occurred during the estimation period as predicted by the operating model (Estimated) and the mean (± 1 stdev) of pseudo-observed tag recaptures from 100 simulations (Simulation) for the ALB2015 example.

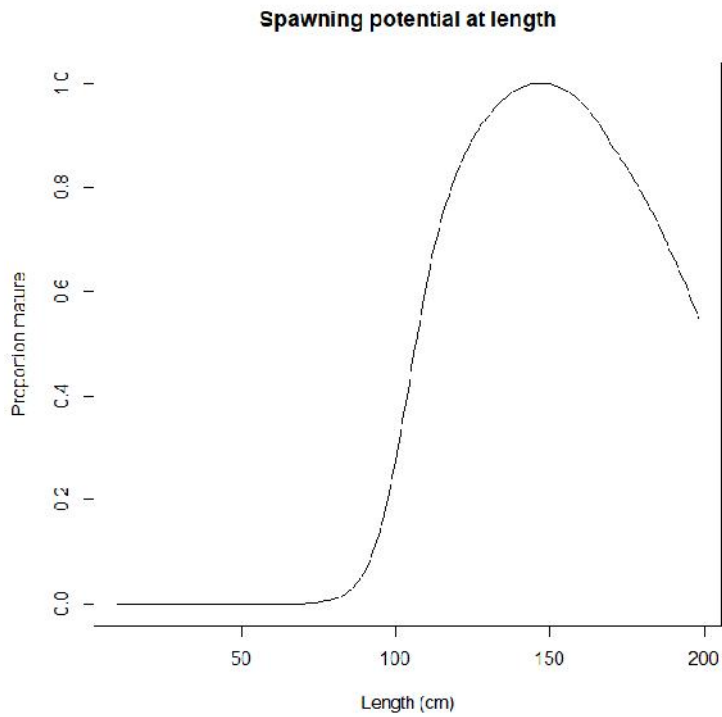


Figure 9. Maturity at length ogive for BET2017 as used for deriving maturity at age for the refcase, and as input to MULTIFAN-CL for testing the mat_len feature to convert to maturity at age internally.

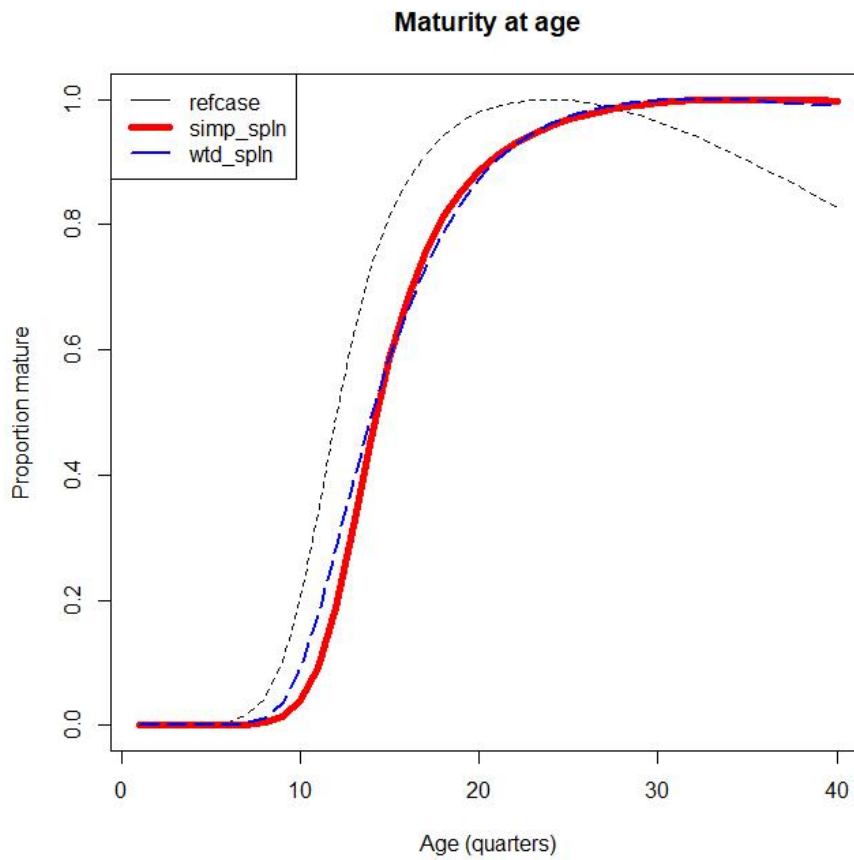


Figure 10. Comparison of maturity at age estimates for the BET2017 model (refcase), and those obtained using the mat_len feature to convert to maturity at age internally.

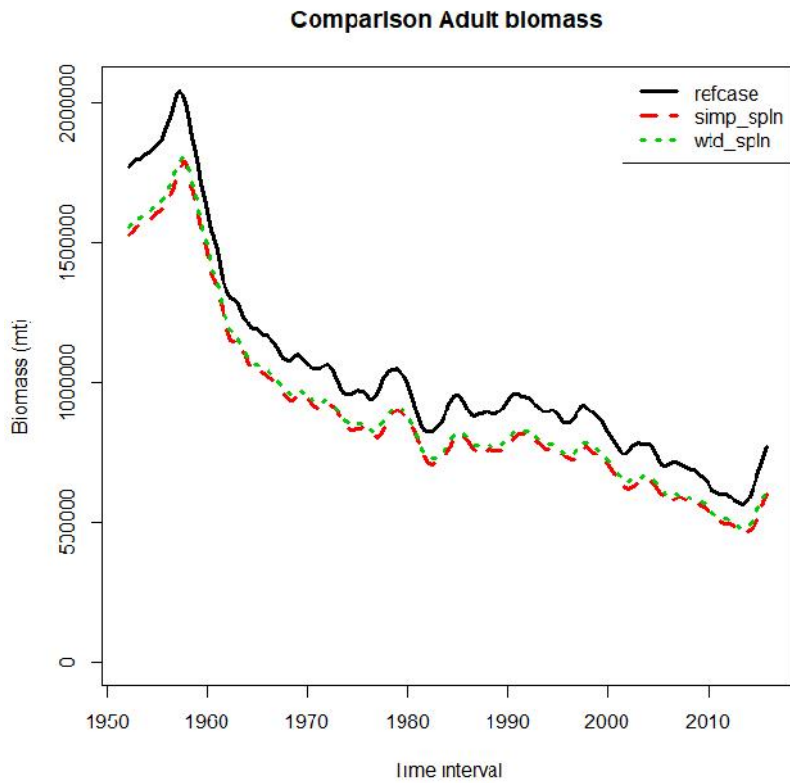


Figure 11. Comparison of BET2017 adult absolute biomass estimates for the BET2017 model (refcase), and those obtained using the mat_len feature to convert to maturity at age internally.

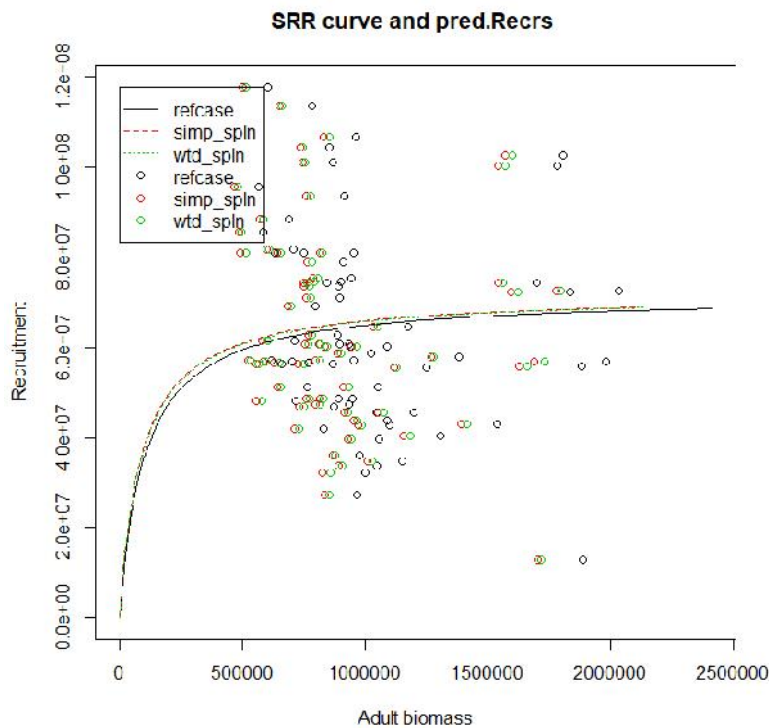


Figure 12. Comparison of spawning biomass (t) and estimated BH-SRR deviates for the BET2017 model (refcase), and those obtained using the mat_len feature to convert to maturity at age internally.

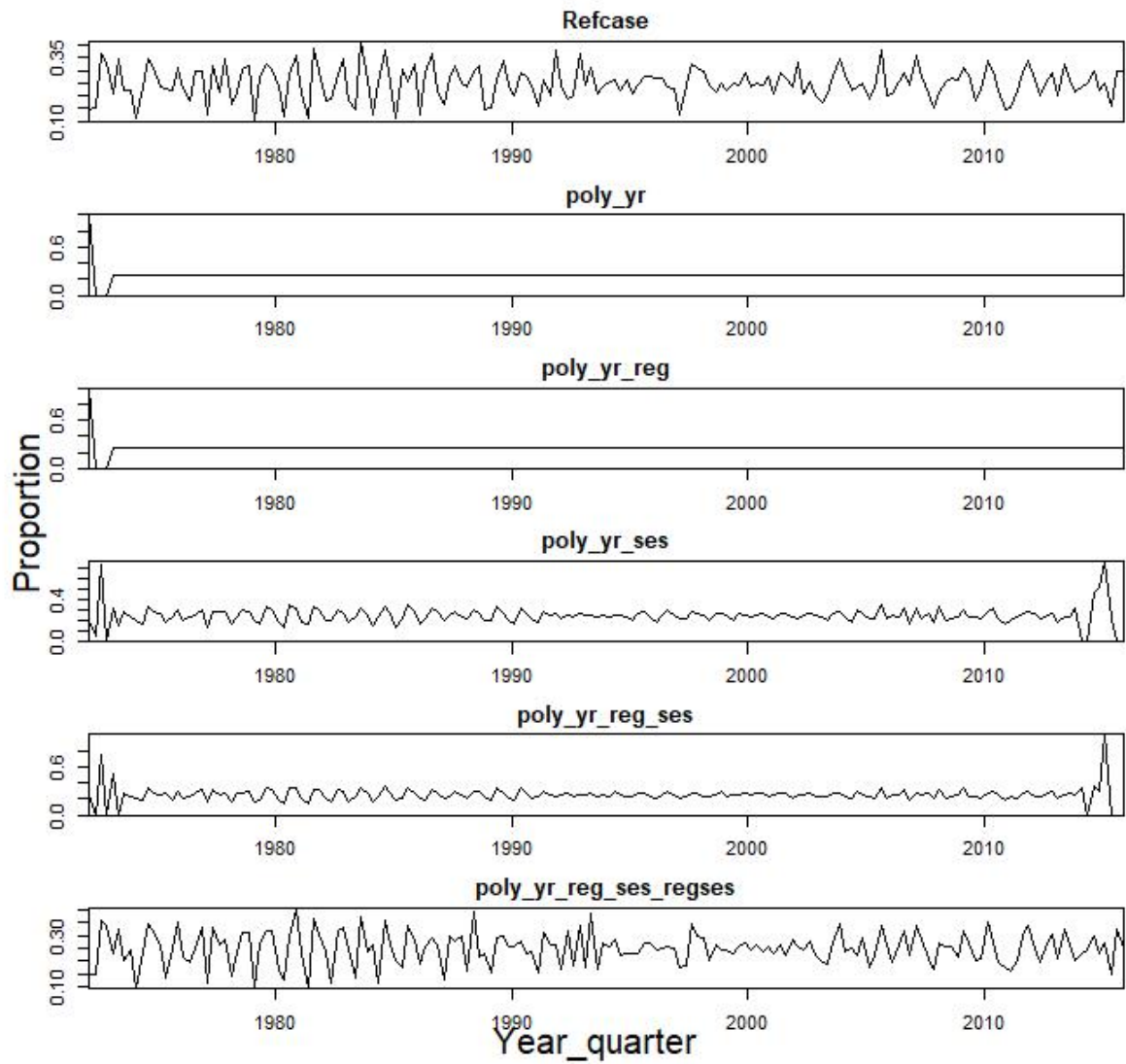


Figure 13. Proportion of annual (calendar year) recruitment in each season estimated using the “standard” mean+deviate recruitment parameterisation, model Refcase, and for each of the five options for orthogonal-polynomial recruitment parameterisation levels, where poly_yr_reg_ses_regeses includes all four levels.

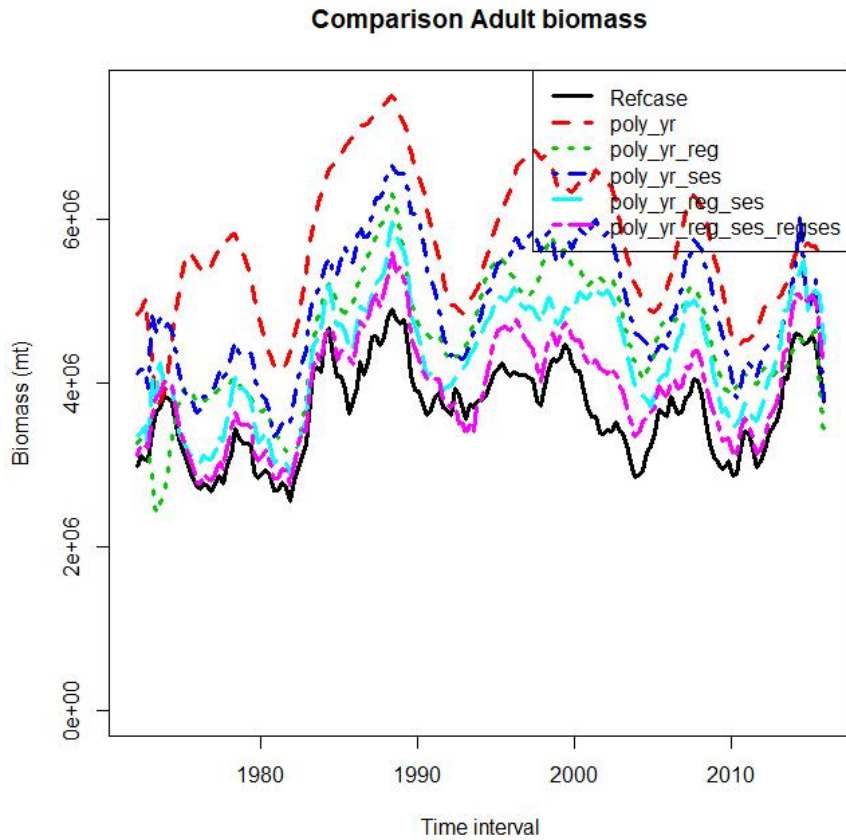


Figure 14. Comparison of spawning biomass (t) estimated using the “standard” mean+deviate recruitment parameterisation, (model Refcase, black line), and for each of the five options for orthogonal-polynomial recruitment parameterisation levels, where poly_yr_reg_ses_reises includes all four levels (dashed purple line).

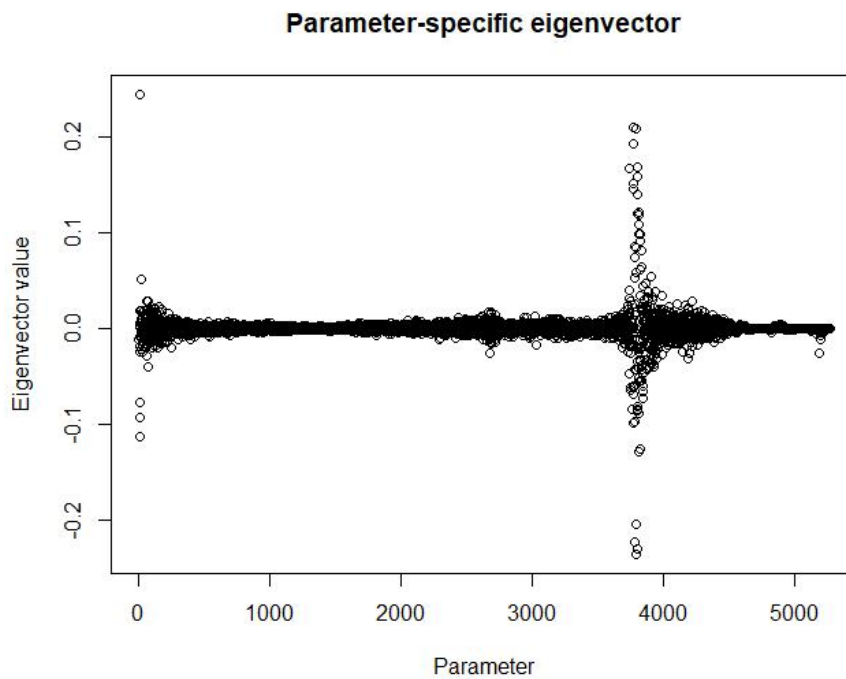


Figure 15. Values for the first row of the eigenvector matrix for the SKJ2016 model solution indicating the most influential parameters.

12 ANNEX

12.1 Benchmark testing

Summary of benchmark tests of the development version using single species data undertaken August 2013 to July 2018.

Compilation name	Compilation folder & Exec.name	Compilation date	Test folder	Test date	Test configurations	Results folder	Key results
Benchmark testing 16 September 2014							
MTHRD_2014_05_13	multi_spp_2014_05_13; exec_txf/mfclo64_2014_05_13	May.2014	2014-09-12_fit_bet	16/09/2014	af145 = -3; penwt=10.0**af145	2014-09-12_fit_bet	penwt=0.001; Obj.=1068534.244292082731; MSY=3.047e+06; B0=2.217e+08
MTHRD_2014_05_17	multi_spp_2014_05_17; exec_txf/mfclo64_2014_05_17	Sep.2014	2014-09-12_fit_bet	16/09/2014	af145 = -3; penwt=10.0**af145	2014-09-12_fit_bet	penwt=0.001; Obj.=1068556.974073923426; MSY=3.422e+06; B0=2.514e+08
Benchmark and Release version testing 22 September 2014							
MTHRD_2014_05_17	multi_spp_2014_05_17; exec_txf/mfclo64_2014_05_17	Sep.2014	2014-09-22_fit_bet	22/09/2014	af145 = 1; pen_wt=af145	2014-09-22_fit_bet	pen_wt=1; Obj.=1068477.068569175899; MSY=1.894e+04; B0=1.410e+06
Release_vsn.1.1.5.8	C:\Nick\MFCL\2011-10-07\mfc\; C:\Nick\I_Assessments\MFCL\2014_09_12	Sep.2014	2014-09-22_fit_bet	22/09/2014	af145 = 1; pen_wt=af145	2014-09-22_fit_bet	pen_wt=1; Obj.=1068457.480964803835; MSY=1.913e+04; B0=1.420e+06
Development version and Release version testing 20 October 2014							
MTHRD_2014_09_16	multi_spp_2014_06_19; exec_txf/mfclo64_2014_06_19	8 Oct.2014	2014-09-22_fit_bet	20/10/2014	af145 = 1; pen_wt=af145	2014-09-22_fit_bet	pen_wt=1; Obj.=1068477.068569175899; MSY=1.894e+04; B0=1.410e+06
Release_vsn.1.1.5.8	C:\Nick\MFCL\2011-10-07\mfc\; C:\Nick\I_Assessments\MFCL\2014_09_12	Sep.2014	2014-09-22_fit_bet	20/10/2014	af145 = 1; pen_wt=af145	2014-09-22_fit_bet	pen_wt=1; Obj.=1068457.480964803835; MSY=1.913e+04; B0=1.420e+06
Benchmark and Development version testing 6 November 2014							
MTHRD_2014_05_17	multi_spp_2014_05_17; exec_txf/mfclo64_2014_05_17	Sep.2014	2014-11-06_fit_bet	6/11/2014	af145 = 1; pen_wt=af145	2014-11-06_fit_bet	pen_wt=1; Obj.=1068474.325769255171; MSY=1.884e+04; B0=1.397e+06
MTHRD_2014_09_16	multi_spp_2014_06_19; exec_txf/mfclo64_2014_06_19	5 Nov.2014	2014-11-06_fit_bet	6/11/2014	af145 = 1; pen_wt=af145	2014-11-06_fit_bet	pen_wt=1; Obj.=1068511.045675329166; MSY=1.899e+04; B0=1.405e+06
Benchmark and Development version testing 19-27 February 2015							
MTHRD_2014_09_16	multi_spp_2014_06_19; exec_txf/mfclo64_2014_06_19	5 Nov.2014	2015-02-27_fit_bet/ bnchmrk_bet2011_fit	27/02/2015	af145 = 1; pen_wt=af145	2015-02-27_fit_bet	pen_wt=1; Obj.=1068511.045675329166; MSY=1.899e+04; B0=1.405e+06
MTHRD_2014_11_07	multi_spp_2014_11_07; exec_txf/mfclo64_2014_11_07	26 Feb.2015	2015-02-27_fit_bet/ devvsn_bet2011_fit	27/02/2015	af145 = 1; pen_wt=af145	2015-02-27_fit_bet	pen_wt=1; Obj.=1068553.274013987277; MSY=1.854e+04; B0=1.386e+06

Benchmark and Development version testing 16 November - 2 December 2015							
MTHRD_2014_11_07	multi_spp_2014_11_07; exec_txf/mfclo64_2014_11_07	26 Feb.2015	2015-11-25_fit_bet/ bnchmrk_bet2011_fit	25/11/2015	af145 = 1; pen_wt=af145	2015-11-25_fit_bet	pen_wt=1; Obj.=1068553.274013987277; MSY=1.854e+04; B0=1.386e+06
MTHRD_2015_09_15	multi_spp_2015_09_15; exec_txf/mfclo64_2015_09_15	3 Nov.2015	2015-11-25_fit_bet/ devvsn_bet2011_fit	25/11/2015	af145 = 1; pen_wt=af145	2015-11-25_fit_bet	pen_wt=1; Obj.=1.06765392664784e+06; MSY=1.775e+04; B0=1.074e+06
Benchmark and Development version testing 17 February 2016							
MTHRD_2015_09_15	multi_spp_2014_11_07; exec_txf/mfclo64_2014_11_07	26 Feb.2015	2016-02-17_fit_skj2014/ release_skj2014_fit	18/02/2016	af145 = -1; pen_wt=10^af145	2016-02-17_fit_skj2014	pen_wt=0.1; Obj.=170692.859; MSY=4.047e+005; B0=6.585e+006
MTHRD_2015_09_15	multi_spp_2015_09_15; exec_txf/mfclo64_2015_09_15	2 Feb.2016	2016-02-17_fit_skj2014/ devvsn_skj2014_fit	18/02/2016	af145 = -1; pen_wt=10^af145	2016-02-17_fit_skj2014	pen_wt=0.1; Obj.=1195760.641; MSY= 1.467e+05; B0=4.319e+06
Benchmark and Development version testing 20 June 2016							
MTHRD_2015_09_15	multi_spp_2015_09_15; exec_txf/mfclo64_2015_09_15	2 Feb.2016	2016-06-10_fit_yft2014/ bnchmrk_yft2014_fit	17/06/2016	af145 = -2; pen_wt=10^af145	2016-06-10_fit_yft2014	pen_wt=0.01; Obj.=1195760.515; MSY= 1.469e+05; B0=4.322e+06
MTHRD_2015_09_15	multi_spp_2015_09_15; exec_txf/mfclo64_2015_09_15	21 Jun.2016	2016-06-10_fit_yft2014/ devvsn_yft2014_fit	17/06/2016	af145 = -2; pen_wt=10^af145	2016-06-10_fit_yft2014	pen_wt=0.01; Obj.=1195760.641; MSY= 1.467e+05; B0=4.319e+06
Benchmark and Development version testing 16 August 2016							
MTHRD_2015_09_15	multi_spp_2015_09_15; exec_txf/mfclo64_2015_09_15	21 Jun.2016	2016-08-16_fit_skj2014/2016-08- 16_fit_skj2014_bnchmrk	16/08/2016	af145 = -1; pen_wt=10^af145	2016-08-16_fit_skj2014	pen_wt=0.1; Obj.=170249.547; MSY= 4.046e+05; B0=6.562e+06
MTHRD_2016_04_15	multi_spp_2016_04_15; exec_txf/mfclo64_2016_04_15	15 Aug.2016	2016-08-16_fit_skj2014/2016-08- 16_fit_skj2014_devvsn2	16/08/2016	af145 = -1; pen_wt=10^af145	2016-08-16_fit_skj2014	pen_wt=0.1; Obj.=170214.267; MSY= 3.982e+05; B0=6.405e+06
Benchmark and Development version testing 7 November 2016 - devvsn4							
MTHRD_2015_09_15	multi_spp_2015_09_15; exec_txf/mfclo64_2015_09_15	21 Jun.2016	2016-11-07_fit_skj2014/2016-11- 07_fit_skj2014_bnchmrk	7/11/2016	af145 = -1; pen_wt=10^af145	2016-11-07_fit_skj2014	pen_wt=0.1; Obj.=170249.547; MSY= 4.046e+05; B0=6.562e+06
MTHRD_2016_10_18	multi_spp_2016_10_18; exec_txf/mfclo64_2016_10_18	7 Nov.2016	2016-11-07_fit_skj2014/2016-11- 07_fit_skj2014_devvsn4	7/11/2016	af145 = -1; pen_wt=10^af145	2016-11-07_fit_skj2014	pen_wt=0.01; Obj.=170214.267; MSY= 3.982e+05; B0=6.405e+06
Benchmark and Development version testing 29 March 2017							
MTHRD_2016_10_18	multi_spp_2016_10_18; exec_txf/mfclo64_2016_10_18	7 Nov.2016	2017-03- 29_fit_yft2014/vsn2.0.2.1_yft20 14_fit	29/03/2017	af145 = -2; pen_wt=10^af145	2017-03-29_fit_yft2014	pen_wt=0.01; Obj.=1195611.492; MSY= 592400; B0=4460000
MTHRD_2016_10_18	multi_spp_2016_10_18; exec_txf/mfclo64_2016_10_18	28 Mar.2017	2017-03- 29_fit_yft2014/vsn2.0.2.2_yft20 14_fit	29/03/2017	af145 = -2; pen_wt=10^af145	2017-03-29_fit_yft2014	pen_wt=0.01; Obj.=1195578.322; MSY= 583200; B0=4374000
Benchmark (vsn.2.0.2.2) and Development version (devvsn7) testing 17 May 2017							
MTHRD_2016_10_18	multi_spp_2016_10_18; exec_txf/mfclo64_2016_10_18	4-May-17	2017-05- 17_fit_yft2014/vsn2.0.2.2_yft20 14_fit	17/05/2017	af145 = -2; pen_wt=10^af145	2017-05-17_fit_yft2014	pen_wt=0.01; Obj.=1195578.322; MSY= 583200; B0=4374000
MTHRD_2017_05_02	..\Assessments\MFCL\2017_05_02_devvsn7	17-May-17	2017-05- 17_fit_yft2014/devvsn7_yft2014 _fit	17/05/2017	af145 = -2; pen_wt=10^af145	2017-05-17_fit_yft2014	pen_wt=0.01; Obj.=1195578.322; MSY= 583200; B0=4374000

Benchmark (vsn.2.0.2.2) and Development version (devvsn7) testing 6 June 2017						
MTHRD_2016_10_18	multi_spp_2016_10_18; exec_txf/mfcl64_2016_10_18	4-May-17	2017-05-17_fit_yft2014/vsn2.0.2.2_yft2014_fit	6/6/2017	af145 = -2; pen_wt=10^af145	2017-06-06_fit_yft2014 pen_wt=0.01; Obj.=1195578.322; MSY=583200; B0=4374000
MTHRD_2017_05_02	multi_spp_2017_05_02; ..\Assessments\MFCL\2017_05_02_devvs n7	2-Jun-17	2017-05-17_fit_yft2014/devvsn7_yft2014_fit	6/6/2017	af145 = -2; pen_wt=10^af145	2017-06-06_fit_yft2014 pen_wt=0.01; Obj.=1195578.322; MSY=583200; B0=4374000
Benchmark (vsn.2.0.3.1) and Development version (devvsn9) testing 23 Jan 2018						
MTHRD_2017_05_02	multi_spp_2017_05_02; ..\Assessments\MFCL\2017_05_02_devvs n7	8-Aug-17	2018-01-23_fit_yft2017/vsn2.0.3.1_yft2017_fit	7/2/2018	af145 = -2; pen_wt=10^af145	2018-01-23_fit_yft2017 pen_wt=0.01; Obj.=1202382.67; MSY=175800; B0=4549000
MTHRD_2017_10_20	multi_spp_2017_10_20; ..\Assessments\MFCL\2017_10_20_devvs n9	19-Jan-18	2018-01-23_fit_yft2017/devvsn_yft2017_fit	6/2/2018	af145 = -2; pen_wt=10^af145	2018-01-23_fit_yft2017 pen_wt=0.01; Obj.=1202570.63; MSY=189500; B0=5142000

12.2 Agenda for Training Workshop, March 2018

MULTIFAN-CL training workshop – 13 to 15 March 2018, OFP

Venue: ICT room, SPC campus

Timetable: 8:30 am to 4:30 pm

Aim: To achieve proficiency with the new features in MULTIFAN-CL version 2.0.4.0, particularly with:

-) fitting size-composition data
-) orthogonal-Polynomial recruitments
-) options for minimisers
-) pseudo-observations of tagging data, and
-) using the Linux debugger

Presenters: Nick Davies

12.3 Agenda

1. Workshop outline
 -) Aim
 -) Scope of material and course structure
 -) Participant's requirements
2. Features in version 2.0.3.1

Features developed during the 2017 assessments, and partly implemented. An introduction is provided to each feature, followed by an application using an example model.

 -) Likelihood profiling in respect of model biomass estimates
 -) Alterations to the estimation of recruitment autocorrelation
 -) Improvements to multi-sex models: length-specific selectivity, and greater dis-aggregation of fisheries data possible
3. Features in version 2.0.4.0
 -) Orthogonal-Polynomial recruitments - alternative method for estimating temporal-spatial recruitments using polynomials that requires fewer parameters
 -) Size-composition likelihoods:
 - o The self-scaling multinomial M-estimator without random effects (SSMULT_noRE)
 - o Reports for estimated effective sample sizes – SSMULT, Dirichlet-Multinomial, and robust normal likelihood
 -) Options for minimisers
 - o limited memory Newton minimiser with either standard or increased precision, and
 - o quasi Newton minimiser
 -) Pseudo-observations of tagging data
 - o Description of current implementation
 - o Flag settings for activation and outputs generated
 - o Application using the ALB2015 example
 -) Various changes in version 2.0.4.0:
 - o new version 1055 of .par with format changes
 - o new sections for orthogonal-polynomial recruitments
 - o modified scalars for fish_pars(14, 15) and age_pars(3)
4. Debugging

-) Using `ddd` with the MULTIFAN-CL debug compilation executable **mfclsdbg64**, with `backtrace` to locate problem using the tools `call mp()`
 -) Low-level: detection of operation failures due to input errors; e.g. incompatible flags
 -) High-level: functioning of certain features that collapse with a problem; e.g. arithmetic exception
 -) Getting assistance with resolving the bug – what to prepare
5. Parked issues