Western and Central Pacific Fisheries Commission

SCIENTIFIC COMMITTEE
FIFTEENTH REGULAR SESSION

Pohnpei, Federated States of Micronesia
12-20 August 2019

DEVELOPMENTS IN THE MULTIFAN-CL SOFTWARE 2018-19

WCPFC-SC15-2019/SA-IP-02

Nick Davies[3], Dave Fournier[2], Yukio Takeuchi[4], Fabrice Bouyé[1], and John Hampton[1]

Oceanic Fisheries Programme
The Pacific Community, Noumea, New Caledonia

[1] Oceanic Fisheries Programme, The Pacific Community, Noumea, New Caledonia

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

[3] Te Takina Ltd, Whangarei, New Zealand

[4] National Research Institute of Fisheries Science, Yokohama, Japan

Table of Contents

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, Noumea, New Caledonia) from August 2018 to July 2019, and updates the report of Davies et al. (2018).

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. Highlights during 2018-19 included:

- Features that improve model parameterisation by reducing the potential for correlations, e.g. the orthogonal-polynomial parameterisation of movement coefficients; and that enable users to diagnose the Hessian solution, to identify parameter correlations, and therefore to better identify and assess poorly defined models.
- The implementation of an innovative approach for fitting to size composition data, using the self-scaling Multinomial with random effects estimation (SSMULT_RE), is now close to completion, with an improved formulation of the M-estimator, akin to a likelihood. Simulation testing of the autoregressive-autoregressive method (AU-AU) for the autocorrelation in the random effects has proven positive, and has been applied successfully to a large tuna model.
- Promising developments of a proof-of-concept length-structured model, with a differentiable length-based growth transition matrix.
- The compilation framework has been consolidated with routine (daily) automated compilations of the repository "master" branch being undertaken on the three platforms: Linux, Windows, and Mac OS. The routine compilation procedure is now centralised at SPC-OFP, and relieves the developers of the responsibility for this component of the project. Maintaining the range of executables improves the utility of MULTIFAN-CL among users on any of the three operating systems.

In addition, a number of enhancements were made for modelling multiple species, sexes, or stocks that enable specific processes for: maturity-at-length, estimating the Richards growth curve, unique maximum ages, and optimised tagging calculations.

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. Developments towards these recommendations have been the basis of the MULTIFAN-CL workplan for the past 7 years. Eleven of the recommendations have now been implemented, one during 2018-19 (tagging data informs movement only), and two remain yet to be implemented. Rather, resources were directed to the new features listed above, with the SSMULT_RE feature being a priority. 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 taken priority. Also, other unforeseen tasks arose that were addressed: fishing in non-movement time periods; orthogonal-polynomial parameterization for movement; tagging data informs movement; diagnostics of model parameterisation and convergence; and, development of a length-structured model. Flexibility is therefore maintained in setting and reviewing the priorities of project tasks during the year. Many of the incomplete items from 2018-19 are to be rolled over into the work plan for 2019-20, and the two remaining recommendations of Ianelli et al. (2012) are also items for the 2019-20.

The fourth 3-day training workshop for stock assessment analysts was held at Nouméa, during April 2019. This assisted in making the new features accessible to scientists undertaking the 2019 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 2019-20 are itemised below:

Peer review recommendation (Ianelli 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
Length-structured model	Complete proof-of-concept
Recruitment environmental correlates	Development
Movement coefficient correlates	Development
Recruitment deviate penalties	Development
Tag release size compositions in multi-sex model	Development
Outstanding testing of existing features	Testing, draft report
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 tuna and billfish assessments (e.g. Tremblay-Boyer et al. 2018, Takeuchi et al. 2018) encompass long time periods, e.g. 1952 to 2017 in quarterly time steps, and model multiple separate fisheries occurring in up to 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 our understanding of the fishery, to fix software errors, and to improve model features and usability. This document records changes made since August 2018 to the software and other components of the MULTIFAN-CL project both for the current release version (2.0.5.1), and the current unreleased development version, and updates the report for the previous period, 2017-18, (Davies et al. 2018).

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-April: Training and Developer's workshops

May-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 2.4.7).

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 (2017-18) have been addressed in the current period through the model developments completed in 2018-19. 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 GitHub repository "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 a Linux Virtual Machine (VM) that is dedicated to MULTIFAN-CL development, and administers the compilations over the OFP network.

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).

During 2018-19 this facility was extended to support automated builds of the Windows (Visual Studio 2019) release executable. The Windows10 VM used for undertaking the benchmark testing framework (see

2.6) provided the platform for undertaking the routine compilation administered by Jenkins. Therefore, a Windows automated compilation using VS2019 is now up and running on the Windows10 VM. As compiling under Windows with CL is slightly different than compiling under Linux or macOS with GCC, the Windows compilation task runs through the different Jenkins batch script instead of pre-defined ANT tasks as on Linux and MacOS. It was also extended to support automated build of the Mac OS executable (see 2.4.4).

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 dependent libraries

For compilation of the dependent **OpenBLAS** library, the "dynamic architecture feature" was included to the routine compilations that builds several kernels for various processor types, and allows selection if the appropriate kernel at run-time. This may avoid the case where a MULTIFAN-CL executable that was compiled with OpenBLAS on a platform having a very recent processor, subsequently failing upon execution if functions calling the OpenBLAS library are attempted on platforms having relatively older processors. This compilation method results in a substantial increase (22 MB) in the executable size. However, it was noted that OpenBLAS libraries are important for the calculations used for the eigenvalues and eigenvectors of the Hessian and also aspects of the SSMULT_RE.

In order for the MULTIFAN-CL project to be completely portable, three shell scripts were prepared that automate the compilation of all the dependent libraries, before compiling MULTIFAN-CL. These scripts apply different options for OpenBLAS, QD and compilation flags for MULTIFAN-CL. The script "build_openblas4mfcl.sh" builds 3 options of this library: "default", "generic", and "dynamic", where the "dynamic architecture feature" builds several kernels for various processor types, and allows them to be selected at run-time. Similarly, the script "build_qd4mfcl.sh" builds 4 options of the QD library: "default", "O3", "O3fma", and "native". Given the various combinations of compilation options among the dependent libraries, ADMB and MULTIFAN-CL, compilations of 25 different executables may be produced. For a single option, it compiles in total: 49 minutes 5 seconds. This facilitates the portability of the entire MULTIFAN-CL compilation project including the dependent libraries, such that the complete project may be constructed and compiled with one step.

It is now possible to include in the automated compilation administered by Jenkins, compilation of the dependent libraries QD and OpenBLAS. With the exception of a couple of manual steps required to configure particular options, the integrated compilation of the entire project is now undertaken within the Jenkins routine compilation procedure.

2.4.3 Compilation of Linux executable

The Linux version used for compilations to date is Ubuntu 16.04, with the gcc compiler version 5.4.0 (denoted **Exe_A**). A trial compilation was done compiled from same source code with same makefile using Ubuntu version is 18.04 with gcc 7.3.0 (denoted **Exe_B**). Performance tests of the two executables of an identical operation (100 function evaluations) with each indicated the Exe_B was about 10% faster.

The first reason for the improvement is the advancement of general optimization techniques between gcc 5.4.0 and gcc 7.3.0. The second possible reason may be due to the version of glibc (C runtime library) installed on the platform used to compile the executable. The version of glibc installed with ubuntu16.04 is glibc 2.23, while the version installed by ubuntu18.04 is glibc 2.27. The glibc installed by ubuntu 18.04 provides better (faster) Math functions.

2.4.4 Compilation of Mac OS executable

In 2017-18, the MULTIFAN-CL source code was adapted for compatibility with the Linux (**gcc**), Windows (Visual Studio 2019 **cl**), and Mac OS (Homebrew **gcc-x**) compilers. This retained the code's integrity as a single version for use with all three compilers.

During 2018-19, it was demonstrated that routine Mac OS compilations of the “master” and “development” branches was possible, and showed that it could be readily integrated into the MULTIFAN-CL project. This has been valuable and beneficial to the project because a Mac OS executable may be added to the release portfolio, and there are a number of potential users of MULTIFAN-CL who work on Mac OS platforms.

In February 2019, a dedicated Mac OS mini-computer was added to the project’s array for undertaking routine compilations. It was tested and successfully produced Mac OS executables of the MULTIFAN-CL version 2.0.5.1. Subsequently, the Mac OS compilation was added to the routine automated Jenkins compilations undertaken daily (section 2.4.1).

Prior to including the Mac OS executable in benchmark testing, a pair-wise comparison of a model evaluation was made between the Mac OS executable of version 2.0.5.1 and that of the Linux executable, using an example tuna model. With only negligible differences at mostly the 14th significant digit, the Mac OS executable produced a nearly identical result. These minor differences may result in a slightly different minimisation path among the two executables, which in some complex models might result in different converged solutions. It was concluded that the Mac OS compilation is of similar standard to the Linux and VS2017 executables, and should be included in the next set of benchmark tests of the development version, and included when posted for release.

2.4.5 Visual Studio 2019 Windows compilation

The method for the VS2017 compilation of the dependent ADMB library developed in 2017-18 was somewhat piecemeal, entailing separate and sequential compilations of each sub-makefile, and manually installing each library. A “master” makefile was therefore drafted that undertakes the directory “clean” rule, and compilation of all five sub-makefiles from a single command, thus integrating the compilation into a single step.

During the benchmark testing, it appeared that the fitted solutions of the Windows MULTIFAN-CL executable depended upon the host platform upon which the run was conducted, indicating that the floating point calculations may depend upon the host machine's maths co-processor, and that this may differ among machines or Windows operating systems. This presented potentially negative implications for the reproducibility of model results, which is important for maintaining consistency in the management advice based upon the model quantities. This behaviour was attributable to the existence of so-called FMA (Fused Multiply-Add) features implemented in Intel CPUs. FMA performs a calculation like x*y+z in one step, without rounding at intermediate steps (x*y and (x*y)+z), so that it can avoid rounding errors. FMA is one of floating point contraction which is disabled if the floating point compilation option **/fp:precise** is used. It is generally recommended that the **/fp:fast** option be used, as it is more similar to that of the Linux compiler. The VS2019 Windows executable was therefore compiled with the **/fp:fast** option and found to produce almost identical results to the Linux executable. This ensures the floating point calculations may be consistent with the Linux executable, thus avoiding conflicting results among operations performed on the different platforms. The VS2019 makefile used for the routine compilations was therefore modified accordingly.

Debugging the compilation and run-time errors of Windows executables has always proved to be difficult. A Windows10 VM was constructed on the lead developer’s mainframe, and Visual Studio 2019 was installed. The MULTIFAN-CL development version project, including all compilation directories, was copied onto the VM, and compilations were successfully completed. This facility enables the debug of problems that are specific to the Windows executable.

2.4.6 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 2018-19 where development versions were created included:

- Following the benchmark testing of version 2.0.5.1
- Following the benchmark testing of version 2.0.6.0
- Preceding the April 2019 developer's workshop

Developments currently implemented in the development version since January 2019 that have not yet been benchmark tested include:

- Tagging data conditioned in respect of recaptures only – improvements completed
- Reduced memory requirements for tagging calculations
- Hessian diagnostic – improvements completed
- Fix to multi-species single region model
- Fix to un-initialised variable
- Fix to Cobb-Douglas with zero-fishing
- Fix to Cobb-Douglas parameter placement
- Fix to composition tail-compression
- Fix to SSMULT_RE with missing composition data
- Fix to set_value() routines

2.4.7 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 47 source code commits were made to the master and development branched, including merges to the master branch preceding the release of version 2.0.5.1., and the commit of version 2.0.6.0. Substantial commits occurred following each of the developer's workshops in November 2018 and May 2019.

The branches of the repository are managed such that following benchmark testing, the development version held in either of the "mac-dev" or "ongoing-dev" branches, is then merged to the "master" branch. This creates a clear node in the "master" branch tagged as being the next release version. At that point a new development version is created in one of the "mac-dev" or "ongoing-dev" branches for undertaking the next phase of developments. This approach was followed for each of the versions 2.0.4.0, 2.0.5.1, and 2.0.6.0, with the current development version being maintained in the "mac-dev" branch.

2.5 Developer's workshops

Two developer's workshops were held at Nanaimo, British Columbia, Canada, firstly from 24 October – 6 November 2018, and secondly, 23 April – 6 May 2019. The participants were the primary developer Dr Dave Fournier and Mr Nick Davies.

The main items during the first workshop were:

- Recruitment random effects estimation
- Exploration of possible over-parameterisation of the BET2018 model
- A new diagnostic of parameter correlation
- Revised parameterization for movement diffusion coefficients
- SSMULT-RE with AU-AU autocorrelation
- Estimator model fits only projection parameters
- Length-structured model

The items during the second workshop were:

- Tagging data conditioned in respect of recaptures only
- Length-structured model – proof-of-concept

- VS2019 compilation of Windows executable
- Correction to SSMULT_RE with missing composition data

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

2.6 Benchmark testing 2018-19

The benchmark testing framework is described in section 2.9.2, and the series of benchmark tests undertaken in 2018-19 are listed in Annex 12.1. A brief description of the tests, and the features tested, is provided in this section.

2.6.1 Version 2.0.5.1

In August 2018, and preceding the first developer's workshop, comprehensive benchmark testing was done of the development version versus the release version 2.0.4.0. Substantial developments and bug fixes had been made since the previous testing of the development version done in January 2018. The major changes in version 2.0.5.1 are listed in section 2.7.2. Some minor coding issues were detected during testing relating to adjustments needed for compatibility with the VS2017 Windows compilation in respect of new developments (pseudo-observations of tagging data).

Single evaluation tests, with or without gradient calculations and a minimisation step, indicated no differences among the versions or compilation platforms. Differences in the doitall solutions among the platforms were mostly minor among the versions, and therefore no effects were due to the code changes made to the development version, but were rather due to differences among the compilation platforms. The difference between the Windows solutions obtained by the two versions was attributable to the floating point calculations of each, that result in different minimisation paths being taken. This was explained by differences in the floating point compilation options used among the compilers, and was addressed for the VS2017 compiler (see 2.4.5).

2.6.2 Version 2.0.6.0

In March 2019, and preceding the second developer's workshop, comprehensive benchmark testing was completed of the development version versus the release version 2.0.5.1. A list of the developments accumulated in the development version (now version 2.0.6.0), includes:

- **Allow fishing in non-movement time periods** – relaxed the requirement for movement in each fishing time period. Includes backward compatibility with solution ***.par*** files of earlier versions using parest_flags(357)=1 for retrospective capability which results in format changes that juxtapose move_flags(ir,ip) became move_flags(ir,ip+1), and the Dad matrices.
- **Parameterisation diagnostic** – a heuristic of the runtime correlation structure among the estimated parameter values activated by parest_flags(145) = 10 that implements routine correlation_report() to calculate the SVD eigenvalues and eigenvectors. The parest_flags(196) value determines the interval among evaluations for storing the estimated parameter values to be used in this diagnostic.
- **Movement parameterisation** – an alternative orthogonal-polynomial parameterisation of movement diffusion coefficients may be employed using age_flags(184) that may reduce correlations among the coefficients.
- **Maturity-at-length extended to multi-species** – where percent mature is input in respect of length rather than age, this was extended to the multi-species case.
- **Richards growth curve variance calculation** – was added to the calculation of stdev(mean length-at-age), and extended to the multi-species case.
- **Multi-species growth curve variance calculation** – correction was made to the calculation of stdev(mean length-at-age) in respect of each species.
- **Multi-species maximum age** – unique and different maximum age may be assigned for each species in a multi-species model.
- MSE estimator model – optimisation to fit only projection period params - first_unfixed_year, last_real_year (in progress)

- Optimised tagging calculations for multi-species case such that they are only done specific to the regions occupied by each species
- Correction for the case if(!age_flags(190) || age_flags(195)>0) - recruitment options for deterministic projections required the BH-SRR predictions for the recruitments with the regional distributions specified
- Revised formatting of pseudo-observations of size composition data
- Changes to scaling for some parameters (e.g. age_pars(4), mult_spp age_pars(3))
- New .par version number = 1059 with format changes for movement diffusion coefficients
- Corrections for the cases of no tagging data supplied and single region model with no movement
- Correction to allow very large fishery data input in .frq file
- Correction to set_value routines to ensure correct routines are used relative to formal argument types
- Added error check for missing stochastic simulation input files if nsim>0

Single evaluation tests for single-species data, with or without gradient calculations and a minimisation step, indicated no differences among the versions or compilation platforms. Single evaluation tests for multi-species and multi-sex data, produced identical model quantities among the versions, however the objective function terms differed by a small amount among versions. This was attributed to corrections made in version 2.0.6.0 that affect the tagging likelihood by a moderate amount for multi-species models.

Doitall fits indicated differences among the versions that can be attributed to the code changes in version 2.0.6.0:

- movement calculations were altered to allow fishing in non-movement periods
- transformation of the diffusion coefficients during input/output
- corrections to the stdev(mean length at age) calculation for multi-species

These will slightly alter the calculations between iterations causing different minimisation paths being taken that affect the solution obtained by devvsn11, with this being most apparent for the more complex or unstable models, and the multi-species models.

The differences in doitall fitted solutions among the platforms but within versions can be attributed to the floating point calculations of each, that result in different minimisation paths being taken.

While version 2.0.6.0 has been tested, it has not as yet been posted as the next release version.

2.7 Postings to website

There have been two postings of the MULTIFAN-CL release version to the website since July 2018.

2.7.1 20 July 2018 – version 2.0.4.0

The main changes relative to the previous version 2.0.3.1 included:

- 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

The above changes and the benchmark testing of version 2.0.4.0 have been described previously by Davies et al. (2018).

2.7.2 1 September 2018 – version 2.0.5.1

The main changes relative to the previous version 2.0.4.0 included:

- Self-scaling Multinomial M-estimator for composition data – development of an auto-regressive – auto-regressive method for estimating an autocorrelation in the size composition residuals
- A diagnostic check of the input/output of the .par file – controlled by a parest_flags
- A model fit diagnostic of the Hessian using the singular value decomposition – option for producing a report
- Optimising simulation and operating model calculations – options to exclude certain calculations and report generation to improve performance
- Sharing effort_dev_coffs parameters for the multi-sex model – to improve robustness when only aggregated data are available
- Extended simulation mode – generating pseudo-observations for the estimation model time periods, including tagging data that includes reporting rate probabilities
- Implementation of maturity-at-length feature – enables input of maturity at length ogive with conversion to maturity at age done internally using growth estimates

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 been the basis of MULTIFAN-CL developments since the review, and an outline of the status in fulfilling these recommendations is provided.

At the beginning of 2018-19, of the thirteen recommendations, 10 had been implemented and tested, 1 was largely complete, and 2 remained yet to be developed. The recommendations that were identified to be undertaken for 2018-19 included (Table 1):

- Maturity at length (recommendation "k")
- Long-term and initial tag loss (recommendation "c")

The 2 remaining yet to be developed included:

- Non-uniform size bins (recommendation "b")
- Tags inform movement only (recommendation "d")

Development and testing for the maturity at length feature was largely completed during 2017-18 (Davies et al. 2018). This feature was included in the work plan for 2018-19 to make further improvements for its implementation for multi-species and multi-sex models, and to accommodate the Richards growth function (see 5.1 and 5.2).

No progress was made on the feature for: adding long-term and initial tag loss into MULTIFAN-CL, "c", which was initially included in the work-plan for 2018-19. Higher priority was rather given to other unforeseen tasks that arose: fishing in non-movement time periods; orthogonal-polynomial parameterization for movement; tagging data informs movement; diagnostics of model parameterisation and convergence; and, development of a length-structured model (in progress). These are described in section 4.

While not included in the plan, the recommendation: Tags inform movement only (recommendation "d"), has been completed during 2018-19 and is described in section 4.4. Work on recommendation "c" is now proposed for the remainder of 2019-20.

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 and 2019 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 2018-19. 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 doitall 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, and have been described in earlier reports (e.g. Davies et al. 2017) and more recent developments 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 2018-19 are described in section 2.6. The development version was last tested in February 2019 versus the release version 2.0.5.1, which defines the development version as the **benchmark** source code, version 2.0.6.0. 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.

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 2018-19 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 2018-19 and is also unlikely in the remaining part of 2019. It is suggested that they be considered in the 2019-20 work plan for the MULTIFAN-CL project.

A Windows10 VM replaced the existing Windows7 machine used for undertaking the benchmark testing framework. A mirror copy of the framework was created on the new VM and Condor was installed as required for submitting multiple and parallel testing jobs over the fleet of machines.

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 2018-19, additional Linux Virtual Machines were added to the grid to increase the number of model runs possible using the Linux development version executable. Also, the Mac OS mini-computer was added to enable Mac compilations to be included in the benchmark testing framework.

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 2.0.5.1. Proposed future revisions include: incorporating the suggestions arising from the Training workshop (see section 3); the recent features added to version 2.0.6.0 (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 fourth training workshop for users of MULTIFAN-CL at the Oceanic Fisheries Programme (Pacific Community) was held on 8 – 10 April 2019, at Nouméa, New Caledonia. The aim was to achieve proficiency with the new features in MULTIFAN-CL versions 2.0.5.1 and 2.0.6.0, particularly with the features: maturity-at-length, sharing effort_dev_coffs parameters for the multi-sex model, a model fit diagnostic of the Hessian solution, an orthogonal-polynomial parameterisation of movement, optimising simulation and operating model calculations, and generating pseudo-observations for the estimation model time periods (see agenda in Annex 12.2). The workshop was delivered by Nick Davies and was well attended by seven stock assessment scientists.

The morning of the first day was spent on the first agenda item: General Introduction to MULTIFAN-CL, that focused on the needs of novice users and was primarily for the benefit of the attendees from the Japan FSFRI. However, all the other participants were present for this item, and expressed appreciation from reviewing the fundamentals of using the package. Good feedback was received on how comprehensive and useful it was to have a condensed overview.

The afternoon of the first day was partly spent setting up all participants with the MULTIFAN-CL compilation package on Linux Virtual Machines and Mac OS, and installing the ddd debugger. This was essential for them to implement the Module's practical examples, and for participating in the debugging module. Thereafter, the agenda items were covered. The timing for the remainder of the workshop was reasonably close to that set out in the agenda, and sufficient time was available for undertaking the practical examples within each module.

Compared to previous workshops, the participants were reasonably advanced in proficiency and 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 some were able to locate the causes of bugs that interrupt or crash the program. There was great interest in the module where the performance of the three compilation platforms were compared. The Japanese participants confirmed the Mac OS executable was the quickest in undertaking multiple evaluations of a skipjack model! Following the debug session on the afternoon of the final day, participants preferred to remain at the workshop and a very useful and informal discussion was enjoyed by all.

While no major problems relating to using MULTIFAN-CL or its operation were raised during the workshop, several smaller issues were identified that warrant attention or feedback:

- When age-specific natural mortality is estimated using the spline parameterisation, is the average constant natural mortality parameter estimated?
- Confirm that the R4MFCL package routine read_nmd.par (as adapted by myself) can take the input of the latest version of the .par file: 1059.
- Once the benchmark testing framework includes the Mac OS executable, circulate to OFP the condor_sub and *.bat files used for distributing the Mac OS jobs over condor.
- Implement a routine check of the projection catches in catch.rep and the *.frq files reveals they are plausibly similar. This would be a useful "sanity" check of the MSE projection fishing mortality calculations.

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 posted is 2.0.5.1. Most of the developments described below are currently implemented in version 2.0.5.1 and 2.0.6.0 (yet to be posted), while others have been made to the development version since January 2019 (listed in section 2.4.6), 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. Improvements were made during 2017-18 (Davies et al. 2018), and further improvements have been made in 2018-19, which are described here.

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 under-estimate 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 Autoregressive-Autoregressive correlation estimation

Problems have been experienced with using the AR(1) process in estimating correlation in the random effects, and therefore an alternative using an autoregressive-autoregressive (AU-AU) method was developed during 2017-18 (Davies et al. 2018). This method for estimating the autocorrelation, $\boldsymbol{\rho}$, of the residuals is akin to "long memory" processes, i.e. processes for which the autocorrelation dies out more slowly than for an AR(1) process. It is a tractable calculation that can be readily done within a real stock assessment application, i.e. within each minimization step. 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 is that the single $\boldsymbol{\rho}$ is incapable of adequately describing this decay in the correlation with time, which is high initially, but then weak as the fish age and grow.

Further progress was made on the AU-AU development during 2018-19 using the simulation model developed in ADMB. The simple simulation analyzer indicated the AU-AU method was largely unbiased in estimating the autocorrelation in multinomial data. For this test, an AU-AU simple simulator was constructed that for the case of k-levels, generates random autocorrelated Multinomial samples. The samples generated using a 4th order AU-AU function were then analysed using the R **arima()** function, which solved for the AU-AU parameters that were used to simulate the samples. The **arima()** parameter estimates were almost exact (Table 3), providing a compelling demonstration that the AU-AU formulation developed by Dave Fournier is consistent with the standard autoregressive-autoregressive correlation structure. The simulator and R script for this demonstration will be made available to reviewers of the SSMULT_RE paper currently being drafted.

The simple fish model simulator and estimator model developed in 2017-18 that included an AU-AU process in the simulation data, and in the random effects component of the SSMULT_RE estimator, was used to test the SSMULT_RE using AU-AU correlations for estimating the effective sample sizes. For the small simulated sample sizes there was very little difference because the process error is small relative to the sampling error, but for the large sample sizes the range of the percentage difference was -34.1% to -46.9%. This is an expected result because as the sample sizes get larger, the process error begins to dominate, as it should, and hence the effective sample size is estimated to be smaller.

During 2017-18, the AU-AU was integrated into the MULTIFAN-CL code with implementation for length and weight composition data, and during 2018-19 it was applied using the BET2017 model, over a range of 5 to 7 AU-AU levels, and the $\boldsymbol{\rho}$ estimates appear plausible (length frequency: 0.35 – 0.88, weight frequency: 0.77 – 0.96), and do not hit the parameter bounds.

In summary, the SSMULT_RE AU-AU method for estimating the autocorrelation, $\boldsymbol{\rho}$, of the residuals appears valid, and its implementation in MULTIFAN-CL successful. Preliminary fits to length and weight composition data in a tuna model appear plausible. The draft paper is to be completed during 2019-20, and this is recommended as a high priority for the project.

4.1.2.2 Revised M-estimator formulation as a likelihood approximation

During 2018-19, the formulation of the SSMULT_RE M-estimator was reviewed, resulting in a substantial simplification and possibly improved performance.

The SSMULT_RE M-estimator was previously formulated having 5 parameters estimated by simulation of multinomial samples with known proportions, for which the parameters were solved in each cell within a large grid of: numbers of slots, and observed sample sizes. This was termed the "learner" component for developing the M-estimator. Implementation entailed applying the parameter set corresponding to the observation sample having a given number of slots and sample size, and then solving for the effective sample size. This approach while performing well, was cumbersome and not readily intuitive.

A revised approach is a Multinomial likelihood based on an expression which is an approximation to the log-likelihood function using ideas from the Kulbach Liebler form of the log-likelihood. It is a good approximation if the predicted and observed proportions are close. The M-estimator is then implemented directly as a likelihood term in fitting to size compositions, and estimates the effective sample sizes. Thus, dispensing with the parameter sets for number of slots and sample sizes. Simulation testing of the new approach using simulation multinomial samples is in progress, with excellent results to date. Subsequent stages for this development are: testing within the simple fish model simulator; implementing the revised M-estimator into MULTIFAN-CL; and, testing using a real tuna model.

4.2 Parameterisation and convergence diagnostics

4.2.1 Rationale

Typically, fish population models may be large, complex and entail estimating large numbers of parameters, e.g. more than 4000 in the case of some Pacific tuna models. The impetus for this development was to diagnose possible correlations among the parameters and model over-parameterisation.

A ready diagnostic generated by the ADMB package (Fournier et al. 2012) is the "positive-definite Hessian" result for the model's Hessian solution, that indicates adequate convergence and probably a well-defined solution, e.g. parameters not being highly correlated. It was identified during the 2018 assessments that such a ready model diagnostic was needed for identifying a non-positive-definite Hessian solutions in tuna models developed using MULTIFAN-CL. The existing method entailed time-consuming calculations of the inverse Hessian, which can be substantial for large models.

4.2.2 Parameter correlation heuristic

A heuristic approach for examining correlations among the estimated parameters was developed that can be done without having a converged model solution. A real-time method for examining correlations among estimated parameters was developed that entails writing a report of the parameter values from every nth evaluation during the course of a minimization procedure. The matrix of n evaluations by the dimension of the parameter vector is then analysed for the correlations among the parameters. This is activated by parest_flags(196) that defines the nth evaluation interval, and with the report generated: "*.xbsamples ". Upon completing the minimization (or interrupting it) the correlation analysis is done using parest_flags(145) = 10 that calls routine correlation_report() that reports the correlations to "*.par_corr". This report can be subset using parest_flags(194) for the minimum level of correlation to be reported.

This "real-time" method can be run in any one of the phases during a doitall fitting procedure, however, it would be best done during the final phase when all the parameters are being estimated so as to calculate all the correlations. The feature was tested using the BET2018 orthogonal-polynomial model, and strong correlations (>0.995) were found among the movement diffusion coefficients.

4.2.3 Hessian diagnostic

The Hessian diagnostic method in version 2.0.5.1 that used the singular value decomposition (SVD) was revised to use the Choleski Decomposition with more efficient calculations using the fast routines in the OpenBLAS libraries. The method entails:

- Calculating the Hessian of the converged solution
- Using the OpenBLAS library routine for the Lapacke calculation of the eigenvalues and eigenvectors of the symmetric matrix, and using those to calculate the inverse Hessian

The test is to identify any negative eigenvalues, and to examine the corresponding eigenvector and find the parameters associated. A report is produced of the eigenvalues and eigenvectors with details for readily identifying the parameters of influence that affect the non-positive solution. The eigenvalues are listed in ascending order (smallest at the top), so as to report any negative values at the beginning of the file. For each eigenvalue, the associated parameters with their index and coordinates are presented. An example of a negative result found using this diagnostic might be that movement is poorly determined - say if there are identical abundances in adjacent regions, then the reciprocal movements possible may not alter the objective function value.

The feature has been tested for examples: striped marlin (npar = 2339) and skipjack tuna model (npar = 5266) and required only around 1 and 13 minutes (respectively) to produce the report.

4.3 Movement parameterisation

4.3.1 Rationale

The impetus for this feature was the parameter correlation diagnostic (see 4.2.2) that indicated strong correlations (>0.995) among movement coefficients for a complex tuna model. This prompted exploration of an alternative movement parameterization to the existing case where the coefficients are independent variables. The alternative formulation was an orthogonal-polynomial parameterisation of the movement coefficients that aims to maintain the coefficients as orthogonal variables.

4.3.2 Methods and Testing

The dimensions of the movement coefficients matrix vary depending upon a given model's spatial structure, and therefore a parameterisation must take account of this. The key factor in the new parameterisation is the capability of a Gramm-Schmidt orthogonalisation basis matrix that is flexible to any possible regional structure, and therefore flexible to the parameterisation of the movement coefficients among the regions, and also to the number of time periods in which movement occurs. A modification was made of the Gramm-Schmidt matrix to exclude "zero" coefficients where movement does not occur among the regions, i.e. rows that are zeroes.

The matrix includes rows for the region- and time-specific effects, where nmp = number of movement periods, nrows = 1 (all effects) + nmp + (2 * nmp * nregs) + (nvar). This is populated with the movement incidence matrix for the non-zero indices for the between-region movement such that:

- First row will all be 1 (overall effect for all the parameters)
- effects for movement periods, rows 2 to nmp + 1
- the row effects; (region effects by row for each movement period)
- the column effects; (region effects by column for each movement period)
- the identity matrix (diagonal for nvar)

The modified Gramm-Schmidt orthogonalisation with the zero effects deleted is then derived, that excludes those instances where no movement occurs among particular combinations of regions. Thus, the parameterisation is generic to whatever temporal and spatial stratification is specified in the model.

The feature includes a facility that allows seamless switching between the two possible parameterisations:

- Movement coefficients are independent variables
- Orthogonal-polynomial parameterisation of movement coefficients

In case 1, the corresponding orthogonal-polynomial parameters are derived from the independent movement coefficients, and while they may not be used in the model, they are nevertheless reported to the ***.par*** file.

In case 2, the coefficients are derived in a routine that calculates the independent variables in respect of the estimated orthogonal-polynomial parameters.

Therefore, a transformation is done of the parameters from one form to the other form in the report to the ***.par*** file, that ensures both forms of the movement coefficients are available to the next model run using the ***.par*** irrespective of which form is specified. This facilitates exploration among the two options during model development.

A comparison among model estimates using both parameterisations (case 1: **std_implicit**, case2: **orth_poly**) was done for the BET2018 example model. In respect of the movement estimates, overall the rates are similar among the models except for regions 6 and 7 (Figure 1), where higher movement out of region 6 and lower movement out of region 7 is estimated to occur for the orth_poly model. Estimated recruitments are about 20% lower overall for the orth_poly model, resulting in lower (~26%) absolute biomass (Figure 2), and the estimated impacts of fishing are greater by ~18% for the orth_poly model.

4.4 Tagging likelihood conditioned in respect of recaptures only

4.4.1 Rationale

Ianelli et al. (2012) made recommendation "**d**": An option which allows the tagging data to inform movement only rather than movement and mortality. Recent focus on spatial stock assessments, has identified a strong requirement for assessments using tagging data to explore solutions with the tagging likelihood in "recapture-conditioned" mode, as an alternative to solutions with total mortality being implicit. This alternative approach is sometimes loosely termed as using tagging data to "inform" the model in respect of movement only.

4.4.2 Methods and Testing

The differences among the two methods are evident between:

- using the probabilities of caught tagged fish being caught in a particular year (can be independent of the total mortality in the population)
- using the probability of live tagged fish being caught in particular year (is therefore dependent upon total mortality)

The second approach is sounder, and consistent with the assumptions of the tagging experiment, while the first approach is implemented by conditioning the tag likelihood in respect of the relative recapture rate for all releases made at a single time event. The intent of the first approach is to fit tagging data conditioned in respect of recaptures only within each time period, i.e. so that the observations "inform" movement processes only, and is denoted here as the **_ss3** option since it may be implemented in Stock Synthesis 3 (Methot et al. 2013). The methodology uses a modified Dirichlet-Multinomial likelihood conditioned in respect of recaptures only within each time period. This differs from the second approach's likelihood, that is in respect of each tagging group over all time periods, and therefore includes the temporal effects of fishing mortality.

This development was first implemented for the single-species model, but then extended for the multi-species case, and for grouped recaptures and for tags being pooled into an aggregate tag group after a specified number of time periods at liberty.

Using the SKJ2016 model with tags pooled and grouped, the **_ss3** option was applied, and compared with respect to the standard negative-binomial tagging likelihood. Generally, the **_ss3** estimates were plausible and comparable, with visible differences in the estimated movement (Figure 3). Absolute biomass was estimated to be lower (Figure 4), and age-specific natural mortality was estimated to be lower for young age classes, but substantially higher for older age classes (Figure 5). The difference in the statistical inference among the two approaches was evident in the worsening of the fit to the observed recaptures in respect of period at liberty for the **_ss3** approach (Figure 6). Overall, this was a surprising result given how important the tagging data are for estimating total mortality in this particular example, and although a somewhat worse fit to the tagging data occurred, the **_ss3** approach nevertheless produced plausible abundance estimates.

4.5 Fishing in non-movement time periods

4.5.1 Rationale

During the 2018 assessments, it was required to use fisheries data (e.g. size compositions) on a finer temporal scale than movement processes. For example, monthly size compositions may illustrate modal progressions associated with growth, which may not be obvious, or able to be inferred, on a coarser temporal scale (Figure 7). This entails removing the constraint for concurrent fishing mortality within the same time period that movement occurs, i.e. to allow for fishing to occur in non-movement time periods.

4.5.2 Methods and Testing

The feature affects the order in which the movement coefficients are applied in calculating the diffusion matrix. This entailed a correction to a validity check within the code for allowing fishing incidents in time periods that lack movement processes.

To ensure this feature achieves backward compatibility for a previous solution obtained using a version that precedes the feature, a parest_flags(357) is used internally in MULTIFAN-CL. Backward compatibility is activated for an input ***.par*** file having version <1057, that allows its input and produces the identical solution from a single model evaluation (assuming solution is converged). This compatibility option uses the parest_flags(357) = 1 to allow for transfer of the backward compatibility status within the ***.par*** file to be passed among sequential model runs. Consequently, the solution will be maintained with subsequent runs using a sequence of ***.par*** files starting from the original ***.par*** version <1057. This was tested with a projection model having an input ***.par*** < 1057 and the identical model quantities for both the estimation and projection time periods were obtained. The feature was also extended into the simulation mode calculations, and for generating pseudo observations of tagging data.

In theory, this feature should not affect the ultimate solution relative to that obtained using the old code. However, the temporal sequence of the movement coefficients is altered between the regions due to the code change in its implementation among time periods. While the code ensured backwardly compatibility with existing ***.par*** solutions (version < 1057) to produce identical model results, the feature alters the temporal sequence of the movement coefficients in the dynamic model calculations. The effect of this was tested with a full doitall fit using an existing example for which fishing occurs in every movement and recruitment time period.

Estimated recruitments are a little higher (around 5%) which alters the fitted BH-SRR and therefore the estimated equilibrium yield quantities, with a 1.5% increase in MSY, and a 6.4% increase in Fmult (Table

4). This also produces a minor increase in the estimated absolute biomass (around 3.3%). Negligible differences are evident in estimates of natural mortality and fishery-specific selectivity. Overall, the effect of the feature was slight, and will be case-specific, and probably largely in respect of its effect on the fit to tagging data.

4.6 Length-structured model

4.6.1 Rationale

Spatial heterogeneity in growth rates within a fish stock can be described using two scenarios (inter alia) for variable growth among regions:

- Growth is inherent (genetically determined) within parts of the stock and persist throughout the lifetime of each fish in respect of its region of origin
- An underlying growth rate exists, but it is also a function of the environment (or region) within which the fish exists at any point of its lifetime

In the first scenario, this can be modelled adequately within an age-structure model with multiple stocks, such that growth rates are stock-specific. Movement of the stocks among regions may occur, but all fish within a stock retain the stock-specific growth rate (associated with the region of origin) irrespective of the ambient region in which they occur.

In the second scenario, an age-structured model is unsuitable because movement of fish among regions will create temporal discontinuity in the size of fish dependent upon the ambient region in which they occur. In other words, the size-at-age will be decoupled in respect of that which existed in the ambient region before movement occurred, i.e. compared to its size before movement, the fish having the same age may shrink or enlarge when the region-specific growth rate for the current region is applied. To correctly describe this growth scenario, a length-structured model is most suitable, such that the population state matrix is in respect of fish length, and growth is a length-based process. Following movement of fish among regions, the effect of region-specific growth can be correctly modelled by applying the growth of the ambient region into which fish have moved, in respect of the fish's length. Thus, temporal continuity in the growth history of the fish is maintained, and it is consistent with the movements among regions throughout its lifetime.

MULTIFAN-CL has the capability of describing the first scenario because it has the capacity for spatial stratification of multiple stocks within the population, and applies stock-specific biological characteristics, including growth. However, MULTIFAN-CL is age-structured and no capability currently exists for a population state matrix in respect of fish lengths, or for a length-based growth function.

Spatial heterogeneity in tuna population growth rates has been suggested as a possible explanation for spatial differences in size compositions among regions, e.g. bigeye tuna (Hoyle 2011). The mechanisms for this heterogeneity are most likely best described by the second scenario (above), and therefore a length-structured modelling approach is needed. During 2018-19 preliminary developments have been made for the underlying length-based growth functions needed for such a model, with a view to implementing this capability in MULTIFAN-CL.

4.6.2 Methods and Testing

This development required a prototype model, and necessitated a simulation testing framework for evaluating the model. During 2018-19 this was the primary focus.

The core element of the length-structured model is a length-based growth transition matrix. To validate the concept and algebra, an ADMB framework was constructed consisting of a simple age-structured fisheries model simulator to generate size composition data, and a length-structured fisheries model analyzer. Length-composition data were simulated given an assumed von Bertalanffy growth function and fishery selectivity. The length-structured analyzer was developed having: the initial equilibrium population conditions; annual population and fishery dynamics; and fitting to the simulated data. The key achievement was the length-based growth transition matrix that is fully differentiable and that generates plausible derivatives for the length-based growth parameters. Length-based growth is achieved by a family of curves, each having its own transition matrix, and the aggregate matrix being used for growing the population state matrix. A relative weight for each curve in the family is either assumed or may be estimated. The framework was developed to provide the proof-of-concept for the length-structured growth function. While preliminary, this has already been demonstrated, with plausible growth curves being produced relative to the simulator model, a reasonable fit of the length-based analyzer to the simulated size compositions, relative abundance indices, and total catches (Figure 8, Figure 9, and Figure 10, respectively).

However, a comparison of the simulator and analyzer total abundance indicated a large discrepancy (Figure 11). Despite attempts to refine model parameterisations to achieve full parity in the respective model structures and parameters, it seemed that using a simulator that is age-structured (while convenient initially), is not ideal for testing the proof-of-concept length-structured analyser. Developments to change the simulator to being length-structured have begun and will continue in 2019-20.

The coding for the simulation framework was done in C++ routines within the ADMB project, so as to facilitate importing the length-structured model into the MULTIFAN-CL project once it is validated.

5 ENHANCEMENTS AND BUG FIXES

5.1 Richards growth curve implementation

The existing feature for estimating growth using a Richards curve formulation was enhanced by extending its implementation to the multi-species, multi-sex, or multi-stock cases. This function was not operating properly, and the enhancement made a substantial difference to a multi-species model fit that incorporate the feature. It resulted in marked differences in mean weights-at-age, population biomass, and equilibrium yield quantities. The feature is now operating correctly for all the possible population structures available in MULTIFAN-CL.

5.2 Multi-species maturity-at-length

The feature that allows percent maturity to be modelled in respect of an input schedule of maturity-at-length and the growth function estimated within the model, was implemented in MULTIFAN-CL in 2017-18 (Davies et al. 2018). This offers a better approach than assuming a fixed maturity-at-age when growth is an independently estimated process.

This feature required further development to extend its implementation to allow for estimating growth using a Richards curve formulation, and for applying the feature for the multi-species and multi-stock cases. This enables separate schedules of maturity-at-length and growth estimates specific to each species or stock to be used in deriving the percent maturity for each.

5.3 Multi-species unique maximum ages

Steps were taken to improve the discreteness of the multi-species or multi-stock model structures, beginning with maximum age class. This is to better allow for different biology for each species, such that the population state for each can be unique in having a maximum number of age classes. This fundamental change to the population structures impacted on many aspects of model operations including: percent maturity, growth, selectivity, and also tagging calculations, such as the number of time periods for mixing. To create the appropriate “ragged” data structures, several new imatrix and i3_array constructors were required for the ADMB library. This enhancement was tested using the multi-species model employed within the benchmark testing framework.

5.4 Optimisation of MSE estimator model fit

The aim of this task was to improve the efficiency of the “assessment” estimation model (EM) embedded in a MSE management procedure. This entails only fitting parameters relating to the “new” data provided for the projection time periods, e.g. for effort deviates, catchability deviates, and recruitment deviates, while holding all other parameters fixed at the initial values. An example was used for the development (SKJ2016 projection model) having 30 years of projection data and a fitted solution to the entire time series. The feature was developed that estimates a subset of parameters for the projection time periods only.

Two pointers were created: first_unfixed_year and first_unfixed_fish_time that were assigned according to the input parest_flags(243) value in a new routine. The assignment to the estimated x-vector of the parameters: recruitment temporal deviates (recr) and regional recruitment proportion deviates (region_rec_diff_coffs) were made dependent upon first_unfixed_year. The assignment to the estimated x-vector of the effort_dev_coffs parameters were made dependent upon first_unfixed_fish_time. A nominal value for parest_flags(243)=2018 was tested by running the estimation for a small number of evaluations. The correct parameter placements were being done, and the number of estimated parameters was substantially reduced.

While the feature appears to work as intended, further work on it has been suspended because its implications in respect of the time series of estimated parameters must be fully considered, particularly in respect of the effort_dev_coffs parameters and catchability deviates. For these, the deviates are estimated in respect of the mean effort and the mean catchability (respectively) over the full model time period. Fixing a subset of the deviates, and supplying new effort data for the projection time periods having a mean different to that used for the fixed deviates, is likely to cause unexpected outcomes. This de-couples the effort data from the effort deviates that are now fixed. It is not clear how the estimated deviates, and their prior penalty functions, will accommodate this condition. It was concluded that this feature will require further understanding and input from the rest of the team. It might be suitable for the recruitment deviates since the total population scaling parameter will remain an independent variable.

5.5 Check for zeroes in composition pseudo-observations

In an earlier version, an input check was implemented in respect of fisheries data that will report an input error when the size composition data for a fishing incident are all zero. The screen report displays the fishery and time period relating to the error, and then the program is terminated. While helpful when developing a population model, this error can occur when running MULTIFAN-CL in simulation mode, such that pseudo-observations of size compositions as generated for some fisheries for the estimation time periods can sometimes be simulated as all zeroes, simply due to the observation error applied. This may cause program terminations when running Management Strategy Evaluation (MSE) simulations on remote locations, that is inconvenient.

Therefore, an optional catch was created that was conditioned upon the cases where simulation pseudo-observations of size compositions being produced with all zero proportions. In these instances, the option for not reporting these pseudo-observations is possible.

5.6 Increased maximum of fishery data dimensions

During the development of a complex and large multi-stock model, the limit was exceeded for the total size of fishing incidents possible, around 10,000 records. For this example model, having 40 fisheries, 12 regions and 2 stocks, the total volume of fishery data exceeded this maximum, therefore the upper limit was extended to 20,000 records.

5.7 Optimisation of multi-species tagging calculations

For the cases of multiple species, sexes or stocks, with tagging data specific to each structure, the tagging calculations can become computationally intensive. Code was therefore developed for enabling the optimisation of multi-species tag calculations over only those regions relating to each species. This avoids unnecessary looping over regions where species are always absent. The optimization was extended for both the pooled and non-pooled cases.

5.8 Optimised memory for tagging calculations

A computational limitation was encountered when developing the complex and large multi-stock model (see 5.6), such that it was requiring ~15GB of memory, making it implausible to run on standard desktop computers. Adjoint code was therefore drafted for dealing with the derivative calculations for the tagging data so as to optimize the memory management. This greatly reduced the amount of memory required for this model and improved its run-time performance. This enhancement required developing a new dvar_matrix data type to be added to the ADMB library.

5.9 Re-formatted pseudo-observations

Tools external of MULTIFAN-CL have been developed for operating the complex procedures in running MSEs, in particular the input and output files needed for simulation operating and estimation models. An enhancement was made to the report file containing size composition pseudo-observations that better facilitates this aspect of the MSE operation using MULTIFAN-CL. The output file of the length and weight composition pseudo-observations was reformatted slightly, and in cases where if the sum of the samples is too low, to output a vector of zeroes (currently nothing is output in this case). Instead of overwriting the current output file, an additional output file, "**test_lw_sim_alt**", is produced.

5.10 Bug fixes

5.10.1 Fix to SSMULT_RE reports

A correction was made that addressed an aspect of model fitted using the SSMULT_RE feature, revealed when developing the SKJ2019 assessment model. During the reporting stage of the converged model solution, the program terminated unexpectedly. The problem related to an unallocated object caused by missing size composition data. A revision was made to the development version code that adjusted the report conditional upon the object being allocated, i.e. is zero if size composition data do not exist, and therefore the report is not generated.

5.10.2 Fix to simulation inputs

A problem was identified that causes program terminations when running MSE simulations on remote locations. This is inconvenient and often difficult to diagnose. It related to jobs being shipped to host platforms, but with missing input files required for running simulations. A specific if-condition was added in respect of age_flags(20)>0 (i.e. simulations are to be undertaken), that directed the operation to an ad_exit(1) for cases when simulation input files don't exist.

5.10.3 Fix to projection recruitments

When running deterministic projections, an option exists for specifying the recruitments as are predicted by the Beverton-Holt stock-recruitment relationship (BH-SRR) in the projection time period, but with the total recruitment being distributed according to a specified regional distribution. This option is activated by age_flags(195) = 1. An error was detected such that when !age_flags(190) is satisfied, the routine call that supplies the BH-SRR parameters was not activated. The if-condition was amended to include the special case of age_flags(195)>1, so that the correct calculation of recruitment is done using the routine average_recruitment_for_projections().

5.10.4 Fix to allocation of tagging variables

For models with no tagging data, a matrix bound exception occurred when attempting an allocation of tagging-related variables over zero dimensions. This was corrected by making the call to the allocation routine conditional upon a switch variable: tag_group_flag=dataswitch(4), that is zero in the case of no tagging data. This switch variable was also applied for initialising a pointer for indexing tagging events, where there are zero events.

5.10.5 Fix to multi-species single region model

A special case may occur for a multi-species model with a single region, and therefore no movement. This was not accommodated for in deriving the equilibrium population age compositions, and resulted in an unexpected program termination. In the routine get_equilibrium_structure_for_yield(), an if-condition was therefore added for the case of pmsd->num_real_regions=1 (multi-species with single region) such that the single region equilibrium age-structure calculation is performed for each species sequentially, without movement.

5.10.6 Fix to un-initialised variable

During the benchmark testing of the Visual Studio 2019 compilation, an error was found in the matrix used for deriving the modified Gram-Schmidt orthogonalization matrix for the movement coefficients. While not evident in the Linux compilation, for the Windows executable this matrix held implausibly large numbers and negative zeroes. It was identified that it hadn't been initialized after the constructor, and this was corrected.

5.10.7 Fix to Cobb-Douglas with zero-fishing

An unexpected program termination occurred in a model that employed the feature for biomass-dependent catchabilities (Cobb-Douglas parameterization) during the fishing impact analysis, i.e. zero fishing mortality. It was identified that for this case, the fishing impact analysis was not yet accommodated. The if-condition relating to setting the catchabilities and fishing mortalities to zero (catchability_q0 and fish_mort_q0) was therefore modified to include the switch af170q0==0, such that they were set to zero for all years.

5.10.8 Fix to Cobb-Douglas parameter placement

In a model that employed the feature for biomass-dependent catchabilities (Cobb-Douglas parameterization) and where the exponent was estimated as an independent variable, it was noticed that the assumed fishery grouping did not correspond correctly to the parameters. This was traced to the set_value() routines where the flags(i) vector references the sequential position in "key" but it should be referencing the fishery within the flags vector that holds the grouping, i.e. should be flags(key(i)). This was corrected in various set_value() routines, and tested positive with the example model.

5.10.9 Fix to size composition tail-compression

An error occurred during the effective sample size calculations for the robust-normal likelihood that must account for the flexible shapes of tail-compressed size composition samples. This was traced to the tail-compression variables being unallocated because the necessary switch settings (input from -switch or the doitall file) had not been populated into the internal flags before the tail-compression routines had been called. The necessary code changes were made that shifted: the allocation of the tail-compression variables until after the switch changes had populated the internal parest_flags; and, the tail-compression calculations of the size compositions until after the related variables had been allocated.

A second related problem occurred during the impact analysis (under zero fishing mortality), where the allocation of the tail-compression variables occurs a second time preceding the model evaluation under zero fishing mortalities. This caused a de-allocation of the partially-allocated tail-compression variables. This was remedied by making the de-allocation conditional upon the impact analysis flag being null.

5.10.10 Fix to multi-species growth variances

During the enhancements to the Richards growth curve calculations (see 5.1) an error was identified where the global variables for multi-species growth parameters were incorrectly dimensioned. This was corrected in the routine for the multi-species growth curve variance calculation, in respect of the calculation of stdev(mean length-at-age) specific to each species, and for the global variables to which they are assigned.

5.10.11 Fix to multi-sex effort deviates

A correction was required to the routine zero_effdev_sanity_check() in respect of the parameter grouping for the effort deviates that are shared among sexes in the multi-sex model.

5.10.12 Fix to recruitment auto-correlation parameter reports

A minor correction was made to the format of the ***plot.rep*** report relating to the BH-SRR autocorrelation estimates. A space was added that prevents errors when accessing the values using tools external of MULTIFAN-CL (e.g. R4MFCL).

5.10.13 Fix to set_value() routines

The set_value() routines undertake the transfer of independent variables to and from the parameter objects and the x-vector used during the minimization procedure. The routines are accessed in respect of the formal argument data types specified in the routine's signature. A potential error was identified where some signatures were not completely unique or potentially ambiguous, and safeguards were put in place to ensure the correct routines are called in all cases.

6 APPLICATION OF NEW FEATURES

A number of the new features and enhancements described in sections 4 and 5 were employed for the striped marlin and skipjack stock assessments, and the harvest strategy work, undertaken at SPC-OFP in 2019. These included: the Hessian diagnostics for evaluating the converged solutions; the input of a schedule for maturity-at-length; allowing fishing in non-movement time periods; sharing effort_dev_coffs parameters for the multi-sex model; the maturity-at-length feature extended to multi-species; corrections to the multi-species growth curve variance calculation; for the cases of no tagging data supplied for the single region model with no movement; and the optimized simulation and operating model calculations. The application of these features for each assessment are described by Ducharme-Barth and Pilling (2019), Ducharme-Barth et al. (2019) and Vincent et al. (2019), and for the harvest strategy work by Scott et al. (2019a,b,c,d).

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 2019-20, while others may be addressed in subsequent years.

The primary focus will be upon developments that improve the stock assessment model. Of high priority are: a proof-of-concept length-structured model; and, completing the report of the SSMULT_RE. Other notable developments are: informing growth estimation with tagging 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 2018-19 period include:

- Allowing for time-variant penalties on recruitment deviate estimates
- Region-specific environmental recruitment correlates
- Add a time-series structure (e.g. random walk, time blocks or using environmental correlates) to movement coefficients
- 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 two 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") – proposed for 2019-20

Important developments to the simulation mode for its application to MSE 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.

8 DISCUSSION

Whereas in previous years, improving the support structures of the MULTIFAN-CL project have received limited attention, during 2018-19 good progress was made with the compilation framework (section 2.4). The value of this is evident in achieving wider utility for MULTIFAN-CL among users on any of the Linux, Windows or Mac OS platforms, and streamlines the routine compilation procedures. A good result was obtained from the preliminary testing of the Mac OS executable, that confirms this may now be included in the benchmark testing and posting of release versions. The automated routine compilations now include all the dependent libraries, and full portability of the entire project is now possible. Attention was also directed to an upgrade to Ubuntu 18.04 with the gcc-7.3.0 version compiler, which has improved the runtime performance of the Linux executable. Collectively, these achievements have consolidated and centralized the compilation framework which provides added security and performance to the project.

The MULTIFAN-CL workplan is divided into the categories: improving its capability for stock assessment models; and, improving its utility within the MSE framework. In respect of the first category, there were several highlights during 2018-19. The work done in 2017-18 on detailed model fit diagnostics were reviewed and substantially improved for enabling users to diagnose the Hessian solution, to identify parameter correlations, and therefore to better identify and assess poorly defined models. A feature was added that may improve model parameterisation by reducing the potential for correlations, i.e. the orthogonal-polynomial parameterisation of movement coefficients. The SSMULT_RE feature is now close to completion, with an improved formulation of the M-estimator, akin to a likelihood, having been developed during 2018-19. Simulation testing was completed of the AU-AU method and has proven positive, and the approach has been applied successfully to a large tuna model. Promising developments were made on a proof-of-concept length-structured model, with a differentiable length-based growth transition matrix. This was constructed in an ADMB project and simulation testing in a simple fish model is producing positive results. Collectively, these achievements offer real improvements for the development of stock assessment models using MULTIFAN-CL.

In respect of its utility within the MSE framework, limited progress was made on the planned items, with only several enhancements completed. Similarly, the item relating to the Independent Peer Review recommendations (section 2.8) was not addressed. As in previous years, flexibility in the work plan was required during 2018-19, given that unforeseen items arose that attracted priority. These included: fishing in non-movement time periods; orthogonal-polynomial parameterization for movement; tagging data informs movement, diagnostics of model parameterisation and convergence; and, development of a length-structured model. Some were needed for constructing models that required either new features, or adaptations of existing features (e.g. fishing in non-movement time periods). Also, a large number of corrections were required (13 items in section 5) due in part to the application of MULTIFAN-CL to new model configurations (e.g. multi-sex and single region). These were necessary, albeit time consuming, tasks.

The rapid pace of the developments impacted on the project's support role during 2018-19, as indicated by the delay in posting the next release version 2.0.6.0 (benchmark tested in February 2019), the long period since the last benchmark test of the development version, and the delay in 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 has prevented it. It is now an immediate and important priority task.

Short-term priorities for work in 2019-20 are to complete the items currently in progress, in particular the SSMULT_RE. The improved M-estimator formulation requires testing within the simple fish model simulator, implementing into MULTIFAN-CL; and, testing using a real tuna model. Completing the draft paper that introduces this size-composition M-estimator consolidates this large volume of work done over the past 4-5 years. Another item in progress is to fit the EM to projection data only, which may enhance the use of MULTIFAN-CL as an EM in an MSE, by reducing the computational overhead of the model fit to simulated projection data.

Medium-term priorities include developing the length-structured model to a stage where it may be implemented into MULTIFAN-CL. This is an exciting prospect that will greatly expand its utility for stock assessments. Another priority 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 listed in Table 2. 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.

Several items included in the 2019-20 plan, entail developing features that estimate environmental correlates in recruitment and movement processes. These are arguably timely developments given the large scale and long-term oceanographic changes occurring that may impact upon these processes. The orthogonal-polynomial parameterisations available in MULTIFAN-CL for recruitment and movement are suitably formulated for including such correlates.

9 REFERENCES

Davies, N., Fournier, D., Takeuchi, Y., Bouyé, F., and Hampton, J. 2018. Developments in the MULTIFAN-CL software 2017-18. WCPFC-SC14-2018/SA-IP-02

Ducharme-Barth, N. and Pilling, G. 2019. Background analyses for the 2019 stock assessment of SW Pacific striped marlin. WCPFC-SC15-2019/SA-IP-07, Pohnpei, Federated States of Micronesia.

Ducharme-Barth, N., Pilling, G. and Hampton, J. 2019. Stock assessment of SW Pacific striped marlin in the WCPO. WCPFC-SC15-2019/SA-WP-07, Pohnpei, Federated States of Micronesia.

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

Fournier, D.A., Skaug, H.J., Ancheta, J., Ianelli, J., Magnusson, A., Maunder, M., Nielsen, A. and Sibert, J. 2012. AD Model Builder: using automatic differentiation for statistical inference of highly parameterized complex nonlinear models. Optimization Methods and Software, 27(2): 233-249

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

Hoyle, S. 2011. Research outline for size data in WCPO length-based stock assessments. WCPFC-SC7-2011/SA IP-06.

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

Hrafnkelsson, 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.

Scott, R. D., Scott, F., Yao, N., Pilling, G., Hampton, J. and Davies, N. 2019a. The WCPO skipjack MSE modelling framework. WCPFC-SC15-2019/MI-IP-02, Pohnpei, Federated States of Micronesia, 12–20 August 2019.

Scott, R. D., Scott, F., Yao, N., Pilling, G., and Hampton, J. 2019b. Results of Initial evaluations of management procedures for skipjack. WCPFC-SC15-2019/MI-WP-05, Pohnpei, Federated States of Micronesia, 12–20 August 2019.

Scott, R. D., Scott, F., Yao, N., and Pilling, G. 2019c. South Pacific albacore management strategy evaluation framework. WCPFC-SC15-2019/MI-WP-08, Pohnpei, Federated States of Micronesia, 12–20 August 2019.

Scott, F., Scott, R., Yao, N., Pilling, G., and Hampton J. 2019d. Key uncertainties in the skipjack MSE framework. WCPFC-SC15-2019/MI-WP-06, Pohnpei, Federated States of Micronesia, 12–20 August 2019.

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.

Vincent, M. T., Pilling, G. M. and Hampton, J. 2019. Stock assessment of skipjack tuna in the western and central Pacific Ocean. WCPFC-SC15-2019/SA-WP-05, Pohnpei, Federated States of Micronesia, 12--20 August 2019.

10 TABLES

Table 1. New features added to MULTIFAN-CL with respect to their state of completion as of July 2019.

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% Extended to multiple species, sexes, stocks
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 100%; Testing 90%
Other developments		
Task	Description	Status of completion
Hessian solution diagnostic	Diagnostic for identifying a non-positive definite Hessian solution	Development 100%; Testing 90%

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.	Development 60%; Feasibility has been assessed
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.	Development 10%; Testing 0%
Tags inform growth	Development of a feature that incorporates size data from tag recaptures to inform growth estimation.	Development 0%; Testing 0%
Self-scaling multinomial with random effects	Consolidate the self-scaling methods for fitting composition data - in particular the AR-AR method for rho estimation in the SSMULT M-estimator.	Development 100%; Testing 100%
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, catchability devs, recruitment devs, while holding all other parameters fixed at the initial values.	Development 80%; Testing 0%
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.	Development 0%; Testing 0%
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	Development 0%; Testing 0%
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 (more stable method is to use terminal year less 5 as the period for obtaining variance) and eff_devs	Development 0%; Testing 0%

Table 2. New features to be added to MULTIFAN-CL in 2019-20, and those for which implementation and testing is to be completed.

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.	2019-20
Other developments		
Task	Description	Implementation
Length-structured	A length-structured model with a differentiable growth transition	2019-20 – complete proof-

model	matrix	of-concept
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.	2019-20 - complete feasibility assessment
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.	2019-20
Tags inform growth	Development of a feature that incorporates size data from tag recaptures to inform growth estimation.	2019-20
Self-scaling multinomial with random effects	Consolidate the self-scaling methods for fitting composition data – revised M-estimator. Draft paper peer review.	2019-20
Recruitment correlates	Region-specific environmental recruitment correlates estimated within the orthogonal polynomial parameterisation for recruitments	2019-20
Movement correlates	Add a time-series structure (e.g. random walk, time blocks or using environmental correlates) to movement coefficients	2019-20
Recruitment deviate penalties	Allow for time-variant penalties on recruitment deviate estimates	2019-20
Tagging multi-sex	Account for instances of differences between size composition the tag releases and the sex-specific populations	2019-20
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) - Multi-sex model projections	2019-20
Report comments	Add comment descriptions of the selectivity parameter configurations in the output .par and .rep reports	2019-20
EM fits only projection parameters	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, catchability devs, recruitment devs, while holding all other parameters fixed at the initial values.	2019-20
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.	2019-20
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	2019-20
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	2019-20

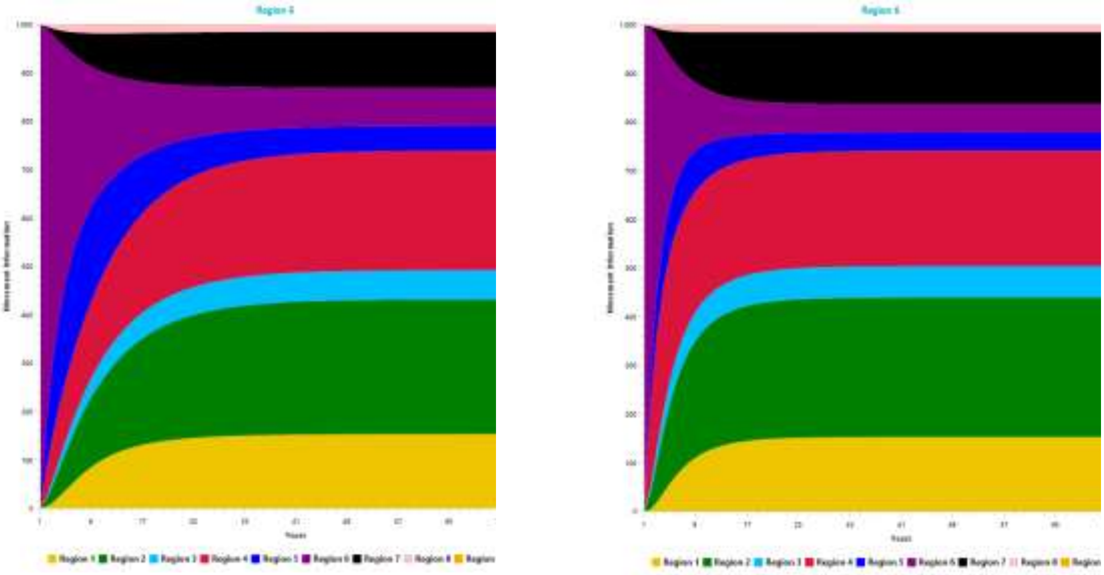
Table 3. Comparison of 4th order autoregressive-autoregressive parameters as used to simulate multinomial data (Simulated), and as estimated by the R arima() function (Estimated).

	First	Second	Third	Fourth	Intercept
Simulated	2.96923	-3.28944	1.6125	-0.295267	0.0162462
Estimated	2.97010	-3.29230	1.6156	-0.296400	0.0054000

Table 4. Estimated model quantities and likelihood terms for the a model preceding the feature for fishing in non-movement periods (refcase) and a model using the version that includes the new feature (fish_nonmv).

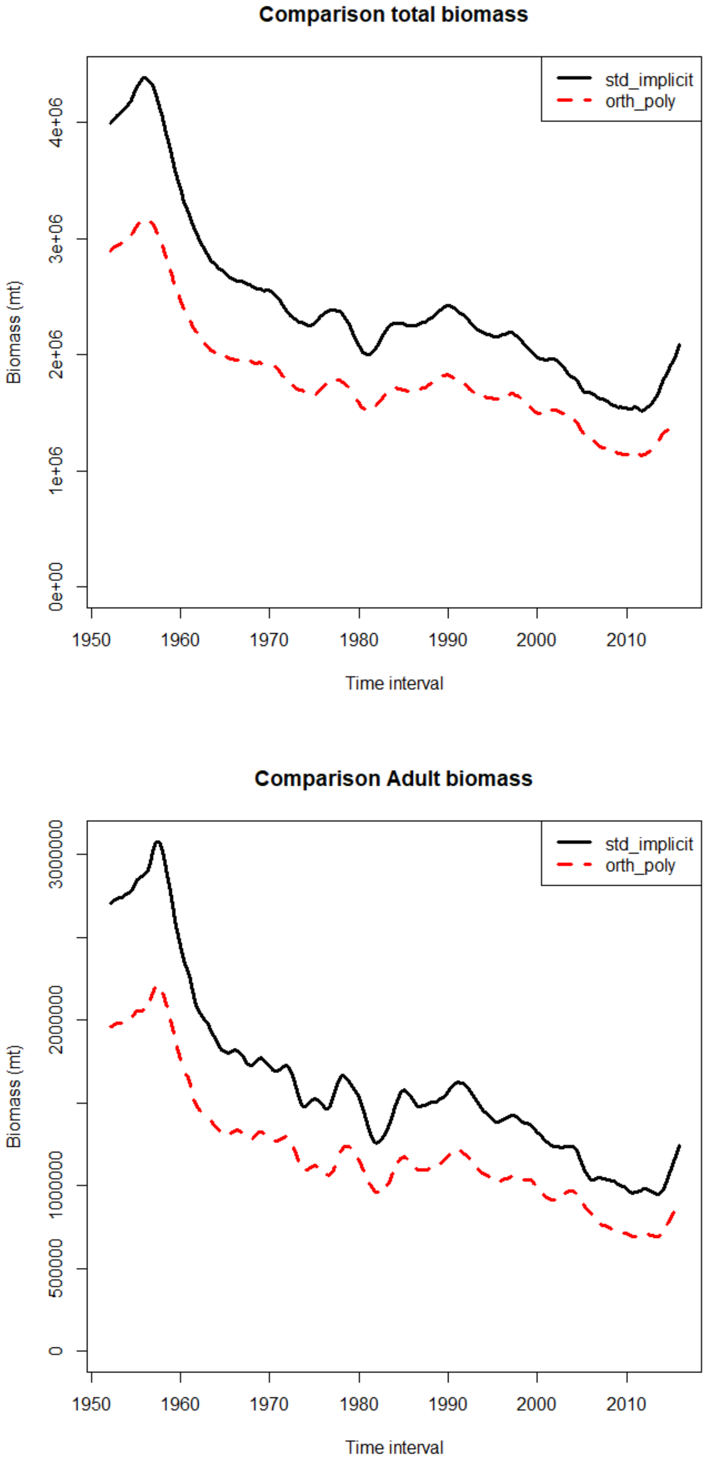
Model quantity	refcase	fish_nonmv	%diff
MSY	474900	481800	1.45
Ccurr.MSY	3.633	3.581	-1.43
Fmsy	0.245	0.244	-0.16
Fmult	2.255	2.399	6.39
Fcurr.Fmsy	0.443	0.417	-6.00
B0	7058000	7131000	1.03
Bmsy	1940000	1972000	1.65
Bcurr	4146396	4282168	3.27
SB0	6724000	6785000	0.91
SBmsy	1625000	1645000	1.23
SBcurr	3769110	3893035	3.29
Bcurr.Bmsy	2.137	2.171	1.60
SBcurr.SBmsy	2.319	2.367	2.03
SBcurr.SBcurrF0	0.520	0.532	2.39
SBlatest.SBlatestF0	0.498	0.500	0.48
obj_bhsteep	0.214	0.208	-2.59
obj_effdev	1628.950	1582.212	-2.87
obj_catdev	57.972	57.392	-1.00
obj_lencomp	-251015.543	-250895.751	-0.05
obj_tagdata	24481.507	24420.526	-0.25
obj_agelngdata	0.000	0.000	-
Obj	-224344.853	-224326.370	-0.01
gradient	0.0094386	0.0092254	-2.26
Lmin	10.000	10.000	0.00
Lmax	88.317	88.317	0.00
K	0.197	0.197	0.00

11 FIGURES



Region 6

Figure 1. The estimated movement among regions derived from coefficient parameters as independent variables (left panel) and orthogonal-polynomial parameters (right panel) for region 6 of the BET2018 example model.



Comparison total biomass

Comparison Adult biomass

Figure 2. Comparison of the estimated total (top) and adult (bottom) absolute biomass for the BET2018 model having movement coefficients parameterised as independent variables (std_implicit) and orthogonal-polynomial parameters (orth-poly).

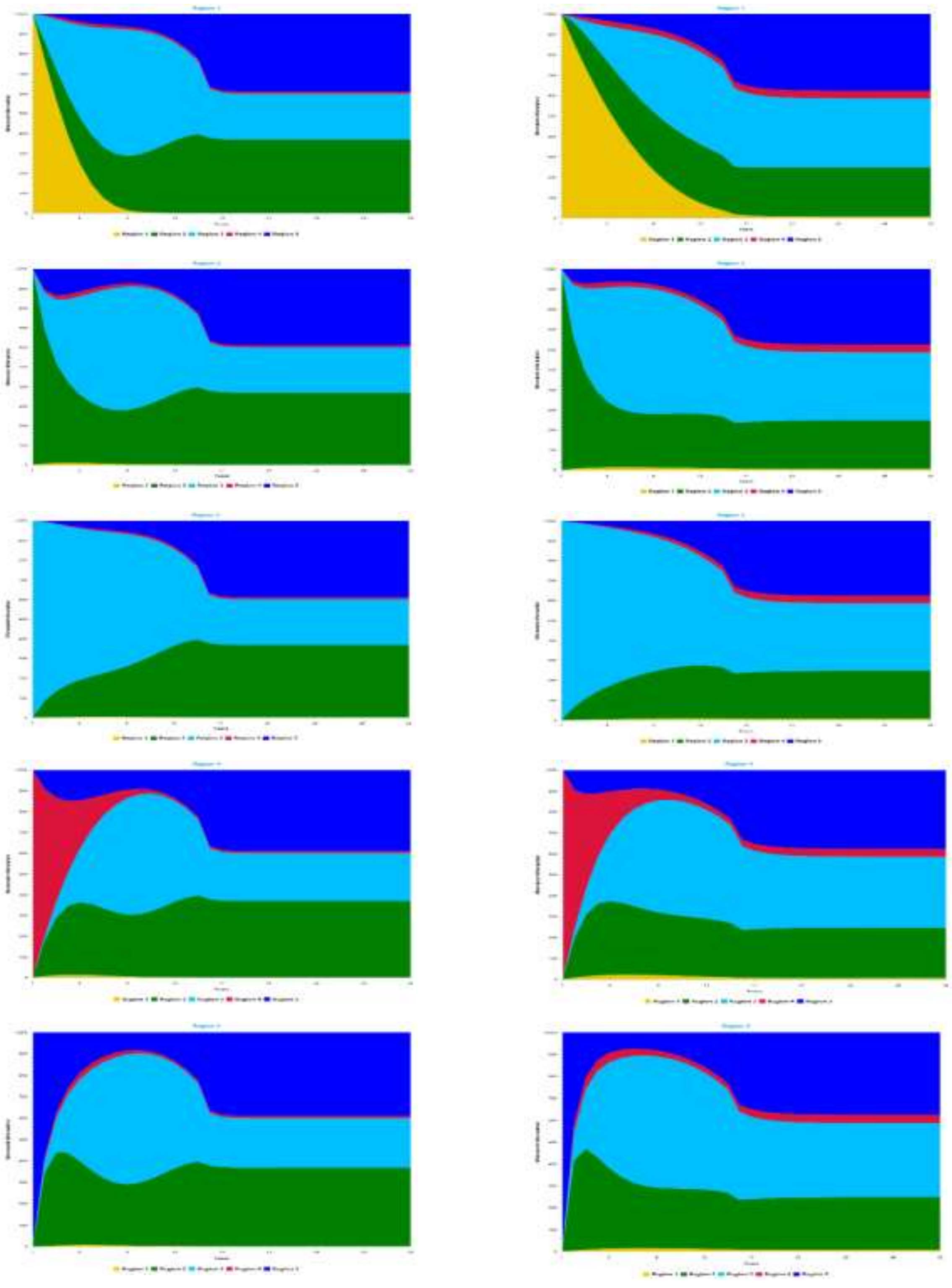


Figure 3. Comparison of estimated movement shown in terms of regional numbers of fish within a cohort's lifetime (rows for regions 1 to 5) for a SKJ2016 model using the negative-binomial tagging likelihood (left-hand panels), and a model using the modified Dirichlet likelihood conditioned in respect of recaptures only (right-hand panels).

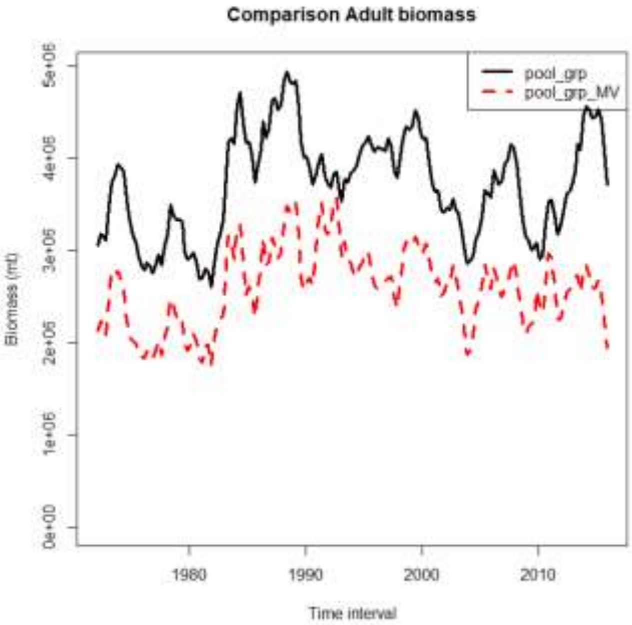


Figure 4. Comparison of the estimated spawning biomass for a SKJ2016 model using the negative-binomial tagging likelihood (pool_grp), and a model using the modified Dirichlet likelihood conditioned in respect of recaptures only (pool_grp_MV).

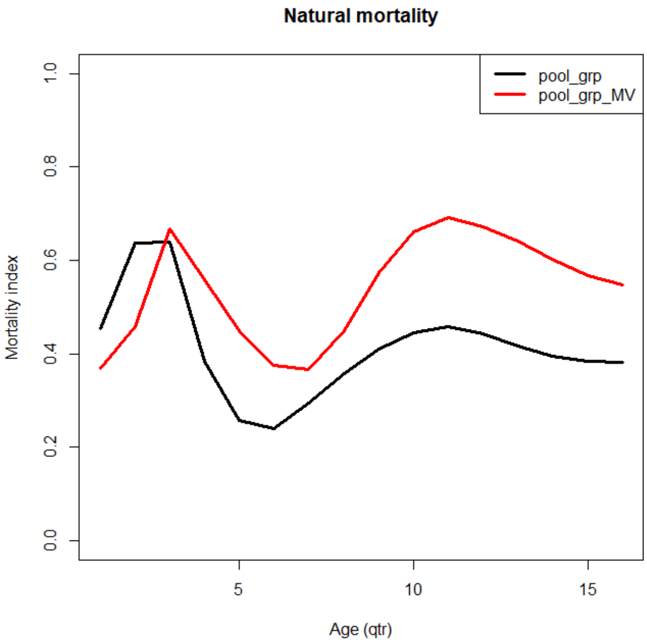


Figure 5. Comparison of the estimated age-specific natural mortality for a SKJ2016 model using the negative-binomial tagging likelihood (pool_grp), and a model using the modified Dirichlet likelihood conditioned in respect of recaptures only (pool_grp_MV).

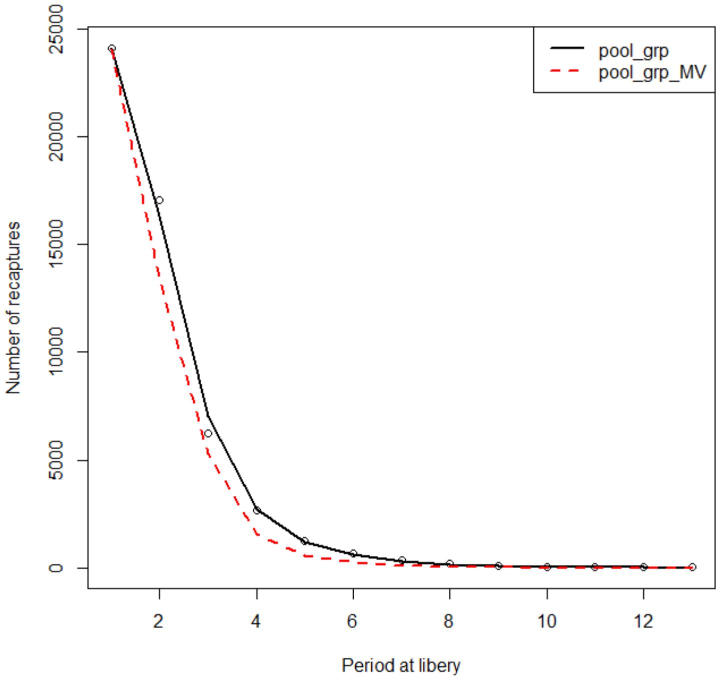


Figure 6. Comparison of the observed (circles) and predicted recaptures (lines) with respect to period at liberty for a SKJ2016 model using the negative-binomial tagging likelihood (pool_grp), and a model using the modified Dirichlet likelihood conditioned in respect of recaptures only (pool_grp_MV).

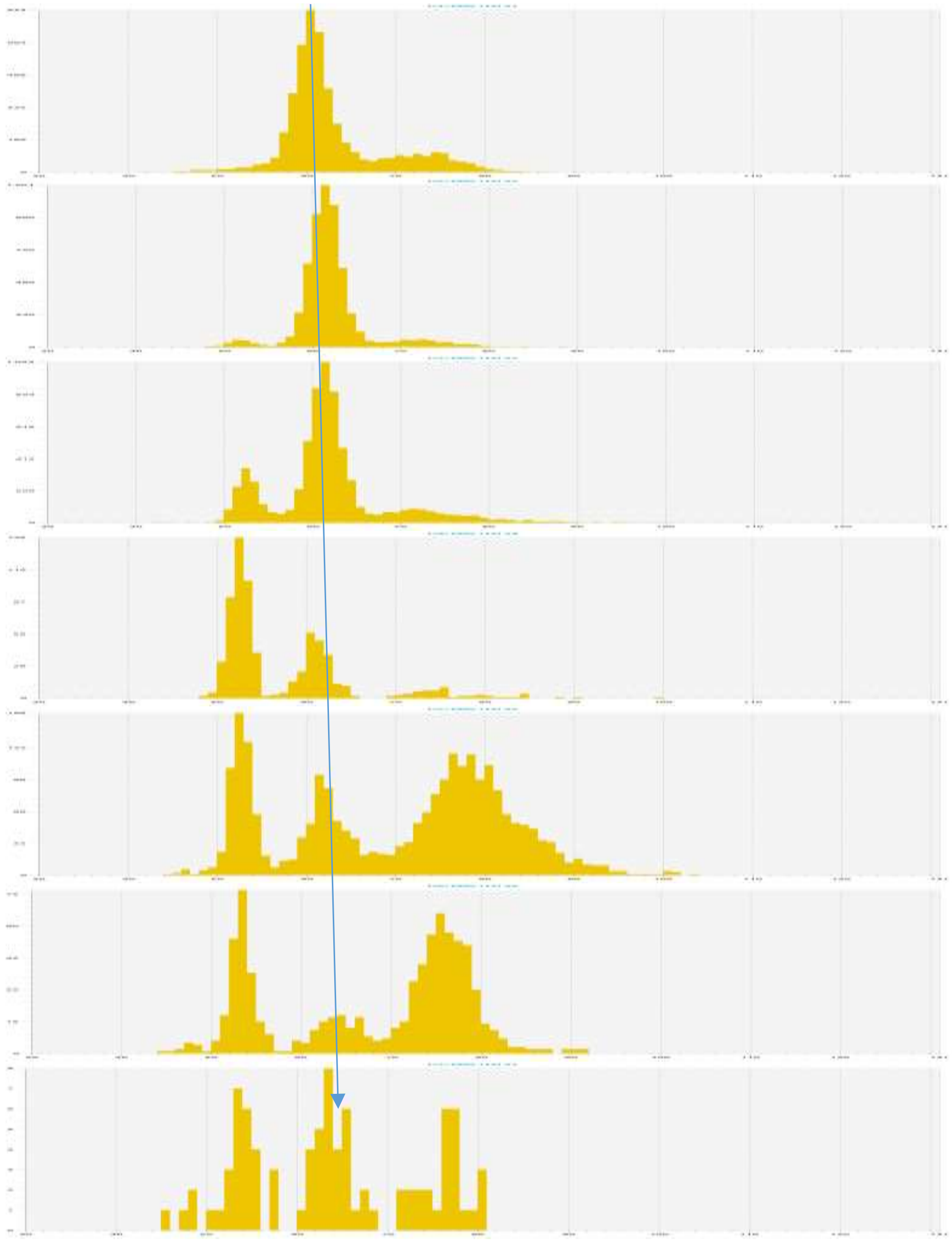


Figure 7. Monthly size compositions (histograms) from troll fishery samples showing modal progressions (arrow) for the ALB2015 example.

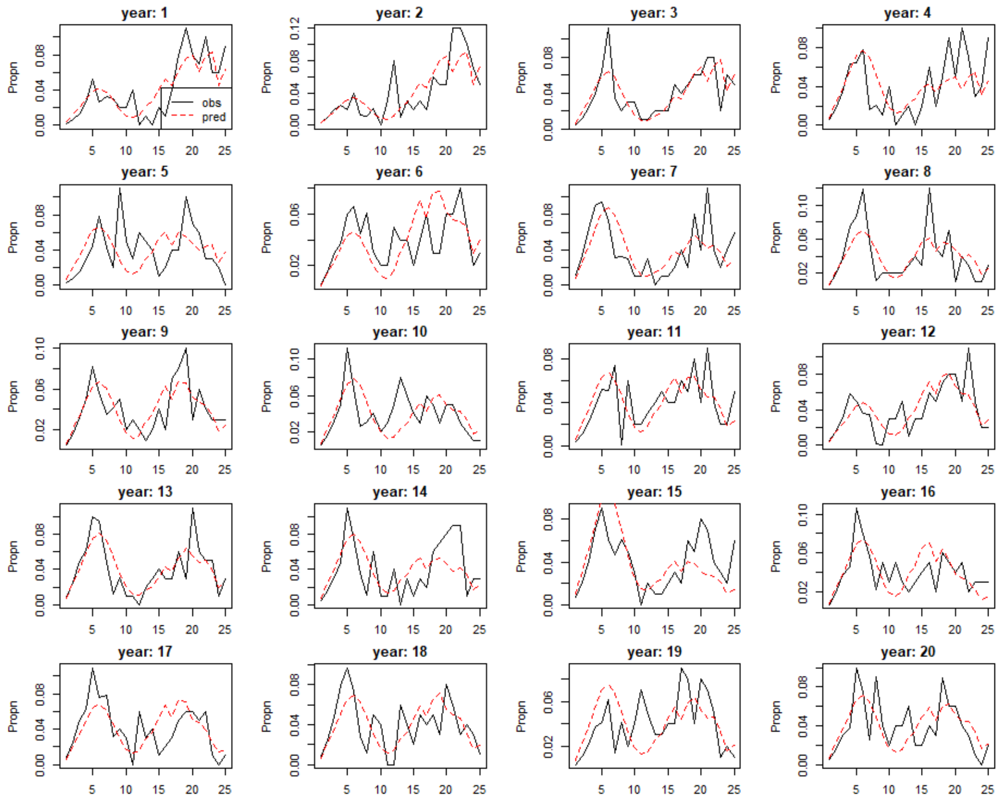


Figure 8. Fits of the length-based analyser model (pred, red dashed line) to the simulated size compositions (obs, solid black line) for each year of the simulation model run (nyears = 20).

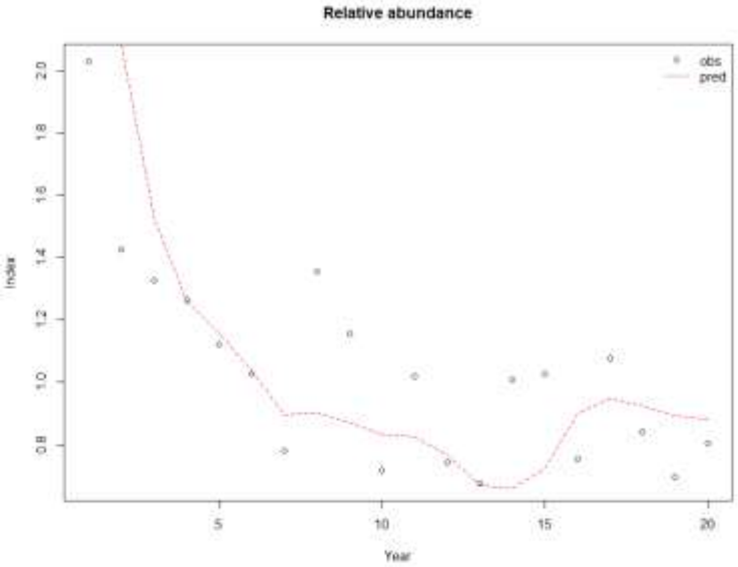


Figure 9. Fits of the length-based analyser model (pred, red dashed line) to the simulated time series relative abundance indices (obs, black circles).

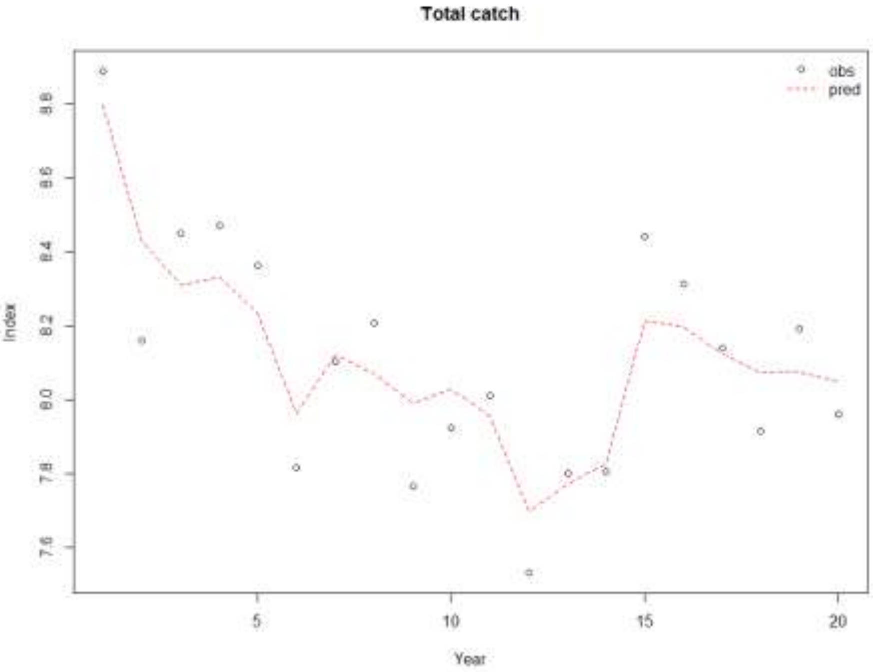


Figure 10. Fits of the length-based analyser model (pred, red dashed line) to the simulated time series total catches (obs, black circles).

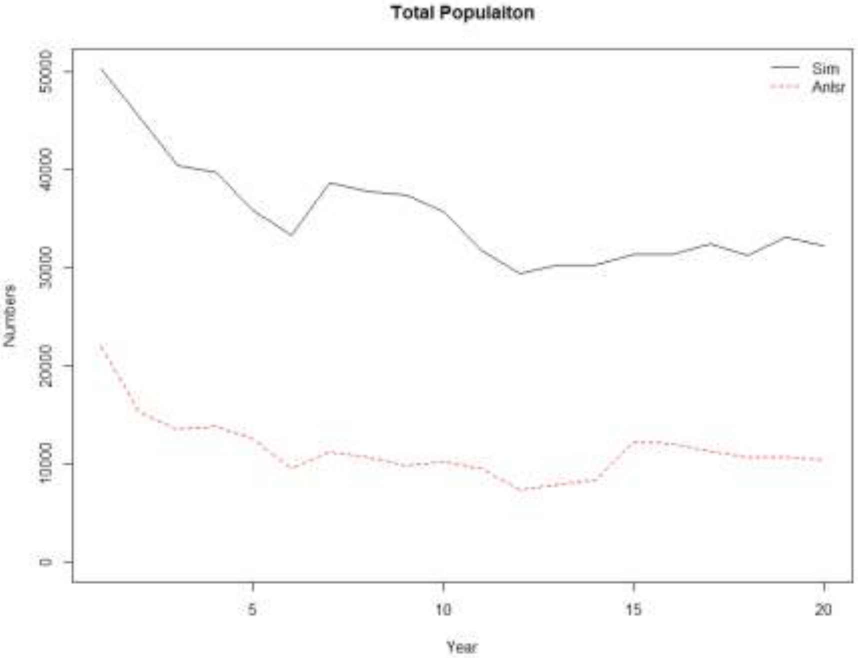


Figure 11. Comparison of the total population numbers of the length-based analyser model (Anlsr, red dashed line) to the age-structured simulator model (Sim, solid black line).

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\mfcl\; 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\mfcl\; 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.=170247.097; MSY=4.056e+005; B0=6.603e+006
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_yft2014_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_yft2014_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_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
MTHRD_2017_05_02	multi_spp_2017_05_02; ..\I_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/mfclo64_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; ..\I_Assessments\MFCL\2017_05_02_devvsn7	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 (devvvsn9) testing 23 Jan 2018							
MTHRD_2017_05_02	multi_spp_2017_05_02; ..\I_Assessments\MFCL\2017_05_02_devvsn7	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; ..\I_Assessments\MFCL\2017_10_20_devvsn9	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
Benchmark (vsn.2.0.4.0) and Development version (devvvsn10) testing 7 Aug 2018							
MTHRD_2017_10_20	multi_spp_2017_10_20; ..\I_Assessments\MFCL\2017-10-20_vsn2040/	15-Feb-18	2018-08-07_fit_yft2017/vsn2.0.4.0_yft2017_fit	**7/8/2018**	af145 = -2; pen_wt=10^af145	2018-08-07_fit_yft2017	pen_wt=0.01; Obj.=1202570.63; MSY= 189500; B0=5142000
mfcl-wrkshp-merged	mfcl-wrkshp-merged; ..\I_Assessments\MFCL\2018_08_07_devvsn10	8-Aug-18	2018-08-07_fit_yft2017/devvsn10_yft2017_fit	**7/8/2018**	af145 = -2; pen_wt=10^af145	2018-08-07_fit_yft2017	pen_wt=0.01; Obj.=1202570.63; MSY= 189500; B0=5142000
Benchmark (vsn.2.0.5.1) and Development version (devvvsn11) testing 31 Jan 2019							
mfcl-wrkshp-merged	mfcl-wrkshp-merged; ..\I_Assessments\MFCL\2018_08_07_devvsn10	8-Aug-18	2019-01-31_fit_yft2017/vsn2.0.5.1_yft2017_fit	**2/21/2019**	af145 = -2; pen_wt=10^af145	2019-01-31_fit_yft2017	pen_wt=0.01; Obj.=1202570.63; MSY= 189500; B0=5142000
multi_spp_2018_11_20	multi_spp_2018_11_20; ..\I_Assessments\MFCL\2019_01_31_devvsn11	21-Jan-19	2019-01-31_fit_yft2017/devvsn11_yft2017_fit	**2/21/2019**	af145 = -2; pen_wt=10^af145	2019-01-31_fit_yft2017	pen_wt=0.01; Obj.=1202659.71; MSY= 162900; B0=4377000

12.2 Agenda for Training Workshop, April 2019

MULTIFAN-CL training workshop – 8 to 10 April 2019, OFP

Venue: ICT room, SPC campus

Timetable: 8:30 am to 4:30 pm

Aims:

- **To introduce MULTIFAN-CL and familiarise users with the basic operations**
- **To achieve proficiency with the new features in MULTIFAN-CL versions 2.0.5.1 and 2.0.6.0**
- **To review issues arising from benchmark testing of vsn.2.0.6.0**
- **Performance of the recent versions and compilation platforms**
- **Familiarise with using the Linux debugger**

Presenter: Nick Davies

Agenda

1. Workshop outline **Mon. 8 April 8:30 to 9:00**
    - Aim and introductions
    - Scope of material and course structure
    - Participant's requirements

2. General Introduction to MULTIFAN-CL **Mon. 8 April 9:00 to 12:30**
    - Background to the model and software
    - Model development
    - Model fitting and reports
    - Using the Viewer

3. Features in version 2.0.5.1 **Mon. 8 April 13:30 to 16:30**

    Features developed during the 2018 assessments, and partly implemented. An introduction is provided to each feature, followed by an application using an example model.

    - maturity-at-length
    - sharing effort_dev_coffs parameters for the multi-sex model
    - optimising simulation and operating model calculations
    - generating pseudo-observations for the estimation model time periods

4. Features in version 2.0.6.0 **Tue. 9 April 8:30 to 14:30**

    New features implemented in the latest release version. An introduction is provided to each feature, followed by an application using an example model.

    - a model fit diagnostic of the Hessian solution
    - an orthogonal-polynomial parameterisation of movement
    - an heuristic parameterisation diagnostic

5. Issues, enhancements and corrections **Tue. 9 April 14:30 to 16:30**
    - review issues arising from benchmark testing of vsn.2.0.6.0:
        - effects of fishing in non-movement time period
        - new version 1059 of .par with format changes
        - new sections for orthogonal-polynomial movement coefficients
    - review of the enhancements to existing features and corrections made

6. Compilations and versions **Wed. 10 April 8:30 to 12:30**
    - demonstrate the VS2017 compilation of vsn.2.0.6.0
    - review of the relative performance of versions and compilation platforms

7. Debugging **Wed. 10 April 13:30 to 16:30**
    - using `ddd` with the MULTIFAN-CL debug compilation executable **mfclsdbg64**, with `backtrace` to locate problem; using the tools `call mp()`
    - practical sessions debugging examples
    - getting assistance with resolving the bug – what to prepare