



**SCIENTIFIC COMMITTEE
SIXTEENTH REGULAR SESSION**

**ELECTRONIC MEETING
11-20 August 2020**

DEVELOPMENTS IN THE MULTIFAN-CL SOFTWARE 2019-20

WCPFC-SC16-2020/SA-IP-01

Nick Davies³, Dave Fournier², Fabrice Bouyé¹, and John Hampton¹

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

Table of Contents

Executive summary	5
1 Introduction.....	6
2 Development overview	6
2.1 Team	6
2.2 Calendar.....	6
2.3 Collaboration and versioning	7
2.4 Compilation framework and Source code repository	7
2.4.1 Compilation framework	7
2.4.2 Compilation of dependent libraries	7
2.4.3 Compilation of Linux executable	8
2.4.4 Compilation of Mac OS executable	8
2.4.5 Visual Studio 2019 Windows compilation.....	8
2.4.6 Development version	9
2.4.7 Source code repository	9
2.5 Developer’s workshops	10
2.6 Benchmark testing during 2019-20	10
2.6.1 Version 2.0.7.0.....	10
2.7 Postings to website.....	11
2.7.1 25 July 2019 – version 2.0.6.0	11
2.7.2 11 March 2020 – version 2.0.7.0.....	11
2.8 Independent Peer Review of the 2011 bigeye tuna stock assessment.....	12
2.9 Tool development	12
2.9.1 R4MFCL.....	12
2.9.2 Testing framework	12
2.9.3 Viewer	14
2.9.4 Condor parallel processing facility	14
2.10 User’s guide	14
3 Training workshop.....	14
4 New features	14
4.1 Self-scaling Multinomial plus random effects (SSMULT_RE)	15
4.1.1 Rationale	15
4.1.2 Methods and Testing.....	15
4.1.2.1 Revised SSMULT formulation as a likelihood approximation	15
4.2 Censored gamma tagging likelihood	16
4.2.1 Rationale	16
4.2.2 Methods and testing	17

4.3	Catch-conditioned fishing mortality calculation	17
4.3.1	Rationale	17
4.3.2	Methods and Testing.....	18
4.3.2.1	Initial population at equilibrium.....	18
4.3.2.2	Fishing mortality levels and effort relationship	18
4.3.2.3	Preliminary testing	19
4.4	Length-structured model.....	20
4.4.1	Rationale	20
4.4.2	Methods and Testing.....	20
4.4.2.1	Validating the analyser.....	21
4.4.2.2	Age-structured simulations.....	21
5	Enhancements and bug fixes.....	21
5.1	Stochastic Variability in N_{terminal}	21
5.2	ADMB - print_matrix_derivatives()	22
5.3	Optimised the generation of stochastic inputs	22
5.4	De-activate effort deviation penalties.....	22
5.5	Report of operating model "true" simulation effort without observation error	22
5.6	Natural mortality at age splines – parameter bounds	22
5.7	Enhancements for the multi-species/sexes/stocks feature	23
5.7.1	Capability for no tagging data, single region, no movement.....	23
5.7.2	Changes to scalars for age_pars(3, 4) in parameter placement	23
5.8	Tagging negative binomial - upper bound on overdispersion parameter.....	23
5.9	Bug fixes.....	23
5.9.1	Fix to total catch likelihood component for multi-stock model.....	23
5.9.2	Fix to zero fishing mortality calculations with the Cobb-Douglas feature	23
5.9.3	Fix to shared effort_dev_coffs in multi-stock model	23
5.9.4	Fix to random number seeds for simulation tagging data	23
5.9.5	Fix to grouped and de-activated effort_dev_coffs.....	23
5.9.6	Fix to duplicate records in "sizemult" report	24
5.9.7	Fix to multi-species Richards growth parameter estimation	24
5.9.8	Fix to plot.rep for multi-species unique nage	24
5.9.9	Fix to tag_CDF_gamma diagnostic report	24
5.9.10	Fix to mean weight used in Newton-Raphson for projections.....	24
5.9.11	Fix to fm_levels assignment within the Newton-Raphson for projections.....	24
5.9.12	Fix to SSMULT without random effects estimation ESS diagnostic reports.....	24
6	Application of new features	25
7	Future work	25
8	Discussion	25

9	References.....	26
10	Tables.....	28
11	Figures	32

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 2019 to July 2020, and updates the report of Davies et al. (2019).

The progress made on implementing new features related largely to improving its capability for stock assessments. Highlights during 2019-20 are several innovative and promising developments:

- Refining the self-scaling multinomial for composition data (SSMULT) for the improved likelihood formulation is now essentially complete, with positive results for the simulation testing, its implementation in the MULTIFAN-CL code, and its successful application to a large, complex tuna model.
- A censored Gamma likelihood for tagging data that explicitly deals with the large numbers of small or zero observations that occur due to patterns in the tag recapture data and the model's structure.
- A proof-of-concept length-structured model that produced positive results in simulation tests, and which offers a promising and innovative approach for estimating length-based growth within a population model.
- The catch-conditioned method for estimating fishing mortalities for which preliminary testing with a real tuna example has been positive, with a 70% reduction in the number of parameters and with similar, or improved, minimisation performance.

In addition, another highlight was the inclusion of the macOS compilation executable within the expanded benchmark testing framework, resulting in its release, the first ever, within the posting of version 2.0.7.0 along with the Linux and Windows platform executables.

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 a large part of the MULTIFAN-CL workplan for the past 8 years. However, two of the recommendations remain yet to be implemented. During 2019-20 resources were rather directed to the new features listed above, with the SSMULT and catch-conditioned features being a priority. Flexibility is necessarily maintained in setting and reviewing the priorities of project tasks during the year. Many of the incomplete items from 2019-20 are therefore suggested to be rolled over into the work plan for 2020-21, and the two remaining recommendations of Ianelli et al. (2012) are included.

The fifth 3-day training workshop for stock assessment analysts to be held at Nouméa, during April 2020 was unfortunately cancelled due to travel restrictions associated with the COVID-19 pandemic.

In respect of future work, the key components of the MULTIFAN-CL project proposed for 2020-21 are itemised below:

Peer review recommendation (Ianelli et al. 2012)	Implementation
c. Long-term tag loss	Development
d. Tags inform movement	Testing, draft report
Other new features	Implementation
Self-scaling multinomial with random effects	Draft report
Recruitment random effects estimation	Draft report
Constrain recruitment and effort deviates	Development
Tagging data informs growth estimation	Development
Length-structured model	Assess feasibility for implementation

Recruitment environmental correlates	Development
Movement coefficient correlates	Development
Outstanding testing of existing features – length-based selectivities	Review code, testing, draft report
Stochasticity in effort_dev_coffs	Testing
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 (Otter Research Ltd) and Dr John Hampton (The Pacific Community) for its initial application to south Pacific albacore tuna (Fournier et al. 1998). It has since provided the basis for undertaking stock assessments in the Western and Central Pacific Ocean.

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. Ducharm-Bath et al. 2020, Vincent et al. 2020) encompass long time periods, e.g. 1952 to 2019 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 2019 to the software and other components of the MULTIFAN-CL project both for the current release version (2.0.7.0), and the current unreleased development version, and updates the report for the previous period, 2018-19 (Davies et al. 2019).

2 DEVELOPMENT OVERVIEW

2.1 Team

The senior developer of MULTIFAN-CL is Dr David Fournier, of Otter Research Ltd, (Canada). Assisting with programming is Nick Davies. Other tasks include testing and debugging (ND, John Hampton, and Fabrice Bouyé (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 (2018-19) have been addressed in the current period through the model developments completed in 2019-20. 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) and the macOS release executables. The Windows10 VM used for undertaking the benchmark testing framework (see section 2.6) and the Mac Mini provided the platforms for undertaking the routine compilation administered by Jenkins (see section 2.4.4). These automated builds were maintained throughout 2019-20.

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

Davies et al. (2019) described the upgrade of the dependent library compilations. An overview follows.

For compilation of the dependent **OpenBLAS** library, the "dynamic architecture feature" is included to the routine compilations that builds several kernels for various processor types, and allows selection of 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 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 currently used for compilations is Ubuntu 18.04, with the gcc compiler version 7.3.0. The version of glibc installed with ubuntu18.04 is glibc 2.27, and provides better (faster) Math functions than the earlier versions.

2.4.4 Compilation of Mac OS executable

During 2018-19, routine macOS compilations of the "master" and "development" branches were added to the compilation framework of the MULTIFAN-CL project. This has been valuable and beneficial to the project, and in 2019-20 a macOS executable (version 2.0.7.0) was added to the release portfolio. This will benefit a number of users of MULTIFAN-CL who work on Mac OS platforms.

The version 2.0.7.0 compilation was done on the MULTIFAN-CL Testing PC (Mac Mini) that has the macOS Mojave installed ("macOS 10.14.6 Mojave"). The PC has two compilation directories:

- Local compilation - stand-alone directory for testing development versions
- Jenkins compilation - for routine automated Jenkins compilations of the master branch (checked out from the repository, see section 2.4.1)

A Software ID certificate was assigned to the macOS compilation using an Apple Developer ID certificate for SPC. The macOS version is signed "Developer ID Application: The Pacific Community" issued by Apple. No differences were detected in the computations or performance among the signed and unsigned compilations.

Apple has now officially announced that they are ceasing production with Intel's CPUs in favour of their own RISC ARM-based CPUs. The macOS compilation of MULTIFAN-CL will ultimately need to accommodate this change. A draft strategy for changing to the ARM-mac compilation for MULTIFAN-CL may entail:

- Following the 2021 Apple conference decide on a machine purchase to be made by May 2022
- Explore the potential for compiling on the existing Intel-mac with output target set to the new ARM-CPU using flags in the most up-to-date or the next version of XCode (the Apple dev tools/compiler)
- In 2021, maintain a careful watch of the capability of the Rosetta 2 emulator for running the MULTIFAN-CL executable (compiled for the Intel CPU) on an ARM-mac; this could offer a "breathing space" for our switch to the ARM-mac compilation
- Consider the merits of upgrading the mac Mini from Mojave to Catalina or Big Sur
- Potentially consider the lead developers purchasing an ARM-mac (external of SPC) for developing the compilation
- Aim for making the switch to the ARM-mac compilation for MULTIFAN-CL in mid-late 2022

2.4.5 Visual Studio 2019 Windows compilation

Few difficulties were encountered with the VS2019 compilations during 2019-20, and generally the MULTIFAN-CL code was compatible among the three compilation platforms. The VS2019 compiler treats

differently the casting of constant variables passed as formal arguments, and this was readily addressed using the tool: ADUNCONST(dvar_len_fish_stock_history,fsh).

A Software ID certificate was assigned to the VS2019 compilation using a Windows Developer ID certificate for SPC. The Windows version is signed: “The Secretariat of the Pacific Community” issued by DigiCert. No differences were detected in the computations or performance among the signed and un-signed compilations.

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 2019-20 where development versions were created included:

- Following the posting of version 2.0.6.0
- Preceding the October 2019 developer’s workshop
- Following the benchmark testing of version 2.0.7.0
- Preceding the April 2020 developer’s workshop

The main developments currently implemented in the development version since version 2.0.7.0 that have not yet been benchmark tested include:

- Catch-conditioned fishing mortality calculation
- Correction to diagnostic reports for SSMULT_noRE
- Correction Newton-Raphson for projection with mixed catch-effort incidents
- Generate simulation file of true effort without observation error
- Optimised generation of stochastic files for cases excluding stochastic N_{terminal}
- Fix to mean_weight_yr used in Newton-Raphson for projections
- Fix to censored gamma tag likelihood added constant
- Fix to robustified censored gamma tag likelihood diagnostic variables
- Fix to reports for multi-species models with unequal nage
- Fix to multi-species models with Richards growth parameter estimation

2.4.7 Source code repository

The MFCL project is 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>

The branches of the repository are managed such that following benchmark testing, the development version that has tested positive and 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 during 2019-20, with the current development version being maintained in the “ongoing-dev” branch (version 2.0.7.3).

A total of 34 source code commits were made to the master and development branches (71 modified source code files, and 16 new source code files), including merges to the master branch preceding the release of version 2.0.7.0., since the posting of version 2.0.6.0. Substantial commits occurred following the developer’s workshop in October 2019.

2.5 Developer's workshops

Two developer's workshops were held during 2019-20. The first being in Nanaimo, British Columbia, Canada, from 8 – 24 October 2019, and second being held "on-line" from 6 April – 12 June 2020. The participants were the primary developer Dr Dave Fournier and Mr Nick Davies.

The main items during the first workshop were:

- Length-structured model
- SSMULT likelihood formulation
- Generating stochastic variability in N_{terminal}

The item developed during the second workshop was the catch-conditioned calculation of fishing mortality, being an alternative to the catch-errors method.

Due to the COVID-19 pandemic (since early 2020) travelling to the second workshop was not possible, and therefore an "on-line" capability was developed for undertaking this workshop. This entailed setting up a Linux **Nomachine** remote desktop protocol server on a mainframe, that enabled both developers to share, with equal participation, a Linux session simultaneously, thus allowing collaborative work on the same desktop. With a concurrent Skype session for verbal and visual communication, this facilitated a "virtual" workshop with considerable success. Due to the substantial developments necessary, the workshop was run for more than twice as long as usual, totalling 28 days.

The details of the developments undertaken in the workshops are described in Sections 4 and 5.

2.6 Benchmark testing during 2019-20

The benchmark testing framework is described in section 2.9.2, and one set of benchmark tests was undertaken in 2019-20. A brief description of the test, and the features tested, is provided in this section.

2.6.1 Version 2.0.7.0

In February 2020, and preceding the second developer's workshop, a benchmark test was done of the development version versus the release version 2.0.6.0. Substantial developments and bug fixes had been made since the previous testing in February 2019. The major changes in version 2.0.7.0 are listed in section 2.7.2. Some minor coding issues were detected during testing relating to adjustments needed for compatibility with the VS2019 Windows compilation in respect of the casting of constant variables included in routine formal arguments.

Single evaluation tests, with or without gradient calculations and a minimisation step, indicated no differences among the versions or compilation platforms. Differences in the multi-species/sexes doital solutions among the versions were attributed to the changes in version 2.0.7.0 relating to the placement of parameters for the minimiser that may result in extremely small differences in the gradient calculation. Consequently, slightly different minimisation paths may be taken that affect the solution obtained by version 2.0.7.0, with this being most apparent for these more complex, or unstable, models (i.e. multi-species/sexes test data). Among the platforms but within versions: the differences among the platforms were either minor (4% or less) or moderate (up to 11%) for more complex models. These were not due to the code changes made to the development version, but were rather due to the floating point calculations of each platform, that result in different minimisation paths being taken.

Additional testing was undertaken using the executables for both versions for the macOS platform using three single species test data (ALB2015, BET2017, SKJ2016), and for the macOS version 2.0.7.0 executable using the multi-species test data. For all single species test data the solutions using the macOS platform were identical among versions. The differences among the platforms were either minor (4% or less) or moderate (up to 10%) for more complex models. These differences were of an order consistent with that obtained for comparisons among the Linux and Windows platforms.

The relative performance of the macOS executable was markedly higher (nearly twice as fast) compared to the Windows and Linux executables. This could be due to the CPU of the MacMini, but is most likely due to the specifications applied for optimising the macOS compilation.

Tests of the development version were consistent with respect to the benchmark version 2.0.6.0, as all existing features remained intact, and therefore the development version was advanced to the new MULTIFAN-CL release version **2.0.7.0**.

2.7 Postings to website

There have been two postings of the MULTIFAN-CL release version to the website since July 2019.

2.7.1 25 July 2019 – version 2.0.6.0

The main changes relative to the previous version 2.0.5.1 included:

- Allow fishing in non-movement time periods
- Parameterisation diagnostic
- Movement – an alternative orthogonal-polynomial parameterisation
- Maturity-at-length extended to multi-species
- Richards growth curve variance calculation
- Multi-species growth curve variance calculation – correction
- Multi-species maximum age – unique and different for each species
- Estimating orthogonal-polynomial recruitments
- Various other enhancements and corrections

The above changes and the benchmark testing of version 2.0.6.0 have been described previously by Davies et al. (2019).

2.7.2 11 March 2020 – version 2.0.7.0

The main changes relative to the previous version 2.0.6.0 included:

- Self-scaling multinomial likelihood – the formulation of the self-scaling multinomial M-estimator for fitting composition data was improved to take the form of a likelihood function; and, this formulation was updated for both the options that include, or exclude, the estimation of random effects for the positive correlation in the residuals.
- Fitting tagging data to estimate movement only – an alternative approach for fitting tagging data that uses the SS3-like approach using the dirichlet-multinomial likelihood; applied for the grouped and ungrouped data scenarios, and also for pooled tags. Included the new likelihood in the test_plot_output report of the component objective function terms.
- Censored gamma likelihood for tagging data – an innovative likelihood for fitting tagging data using a Gamma distribution with a censor value outside of which data are accumulated. This offers a potential improvement for dealing with large over-dispersion due to zeroes.
- Generating stochastic simulation inputs – optimised the inverse Hessian calculation using lapack_symmetric_eigen routines that greatly improves computation time.
- Negative-binomial tagging likelihood – enhancement to specify the upper bound on the overdispersion parameter.
- Simulation of tagging data – enhancement to input the iseed values via a command line argument.
- Correction to test_plot_output report – added the component terms for the total catch likelihood, both the grouped and ungrouped cases of the pooled tag likelihood.

- Correction to tail compression class – set to zero if unallocated; shifted constructors to `optalloc.cpp` routine and to exclude the zero-fishing case.
- Correction to tail compression execution – imposed this conditional upon the `parest_flags(301, 311)`, and also the related SSMULT reports in `plot.rep`.
- Corrections to parameter placement – routines `set_value()` and `set_value_inv()` for the cases of:
 - grouping, grouping and mixed activation flags; for the sharing of `effort_dev_coeffs()` for the multi-spp/stock/sex case
 - incorrectly juxtaposed formal arguments
- Corrections to intermediate `fish_mort_calcs` – calls to the correct sub-routine in the case of Cobb-Douglas catchability with zero-fishing mortality.
- Correction to robust-normal effective sample size calculation – performed over the correct size intervals.
- Debug tools – added new print routines: `mpnz()`, `mpgt()`.

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 2019-20, of the thirteen recommendations, 11 had been implemented and tested, and 2 remained yet to be developed. The recommendation "d", Tags inform movement only, had reached the point of application to pooled and unpooled tags for single species; but testing of its implementation for the multi-species case was yet to be done. The other recommendations that were identified to be undertaken for 2019-20 included (Table 1):

- Non-uniform size bins (recommendation "b")
- Long-term and initial tag loss (recommendation "c")

No progress was made on these recommendations during 2019-20, because higher priority was given to other tasks (some of which were unforeseen) that arose: the revised formulation of the self-scaling multinomial likelihood; the censored Gamma likelihood for tagging data; the catch-conditioned method for estimating fishing mortality; and, the development of a length-structured model (in progress). These are described in section 4.

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 2019 and 2020 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 2019-20. 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 (64-bit architecture only), with application to multiple input

testing data sets, and with various options for the MULTIFAN-CL operation, viz. single or multiple model evaluations, or full doital model fits to convergence. This ensures a thorough integrity-check of model quantities and components of the objective function prior to the distribution of new versions.

Since March 2013, the MULTIFAN-CL source code has undergone substantial developments, and those have been described in earlier reports (e.g. Davies et al. 2019), and the recent developments in 2019-20 are described in Sections 4 and 5.

Following the addition of these new features to the development version, regular testing of this versus the release version aims to ensure the integrity of existing operations. Known as “benchmark tests”, those undertaken in 2019-20 are described in section 2.6. The development version was last tested in February 2020 versus the release version 2.0.6.0. The positive result then defines the development version as being the **benchmark** source code, then posted as version 2.0.7.0. Subsequent development versions will then be tested relative to the benchmark to establish their integrity, after which they may be 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 including: 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. Note 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.

A positive test result is when the benchmark tests conclude that the development version conserves the existing features, and so 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 2019-20 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 and is also unlikely in the remaining part of 2020. It is suggested that they be considered in future work plans for the MULTIFAN-CL project.

With the routine compilation and development of a macOS executable now firmly implemented in the project, the testing framework was extended to include the MacMini host and the macOS executable within tests among platforms and versions. The framework's R functions were successfully upgraded to include the macOS executable, and following preliminary tests on the MacMini hosts discretely, a subset of the benchmark tests for version 2.0.7.0 were conducted upon all 3 platforms simultaneously over the Condor network. The framework now performs pair-wise comparisons among versions and over three platforms: Linux, Windows, and macOS.

2.9.3 Viewer

A new version of the MULTIFAN-CL viewer has been released that can display the results of a multi-species or multi-sex application updated to display the new output added to the MULTIFAN-CL report files.

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 2019-20, additional Linux Virtual Machines were added to the grid to increase the number of model runs possible using the Linux development version executable.

2.10 User's guide

A revision to the MULTIFAN-CL User's Guide (Kleiber et al. 2018) has not yet been completed to include the developments since version 2.0.5.1. Proposed future revisions include: incorporating the suggestions arising from the earlier Training workshops; and the recent features added to versions 2.0.6.0 and 2.0.7.0. The revised version will be posted on the website <http://www.multifan-cl.org/>.

3 TRAINING WORKSHOP

The fifth training workshop for users of MULTIFAN-CL at the Oceanic Fisheries Programme (Pacific Community) was scheduled for April 2020, to be held at OFP in Nouméa, New Caledonia. However, due to the COVID-19 pandemic and the related travel restrictions, attending the workshop was not possible. Consequently, the workshop was postponed to a future date.

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 2.0.7.0. Most of the developments described below are currently implemented in this version, while others have been made to the development version, (since February 2020 and as listed in section 2.4.6), and will be merged to the next release version upon the completion of the benchmark testing.

4.1 Self-scaling Multinomial plus random effects (SSMULT_RE)

4.1.1 Rationale

Ianelli et al. (2012) made a 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, and it has since formed a significant component of the project. It was mostly completed during 2016-17, but further improvements were made during 2017-18 and 2018-19 (Davies et al. 2018, 2019), and a final improvement was completed in 2019-20, which is described here. As background, the rationale provided in earlier reports is repeated here.

While the recommendation provided the initial impetus for a development to use the Multinomial likelihood, it was expanded to address the three known weaknesses with this method (Francis 2014):

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

Size composition data are pivotal to the tuna stock assessments using MULTIFAN-CL, in particular for estimating growth and recruitments. Improvements that might offset these weaknesses would enable the correct use of these data in the integrated modelling approach, and this became the focus of this development. The lead developer, Dr David 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). 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 *Revised SSMULT formulation as a likelihood approximation*

During 2018-19, the formulation of the SSMULT M-estimator was reviewed, resulting in a substantial simplification and improved performance. The SSMULT was previously formulated as 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 deriving a set of coefficients used in the formulation as an M-estimator (Huber, 2009). This formulation entailed applying the coefficients corresponding to the observation sample having a given number of slots and sample size, and then solving for the effective sample size. This approach performed relatively well, but was somewhat cumbersome and not readily intuitive.

The revised approach developed in 2018-19 (Davies et al. 2019) is a Multinomial likelihood based on an expression being 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 SSMULT_RE is then implemented directly as a likelihood term in fitting to size compositions (SSMULT_like), and estimates the effective sample sizes directly, thus dispensing with the coefficients specific to the number of slots and sample sizes as required for the M-estimator.

During the first developer's workshop in 2019-20 and in subsequent months, the following was completed: testing using simulation of multinomial samples and a simple fish model simulator; implementing the likelihood formulation into MULTIFAN-CL; and, testing using a real tuna model. The implementation of the SSMULT_like formulation into MULTIFAN-CL was done to ensure the existing size composition likelihood options remained intact. This implementation was required for both the cases of SSMULT with or without random-effects estimation.

A simple simulation fish model (with 10 age classes) was used to produce simulation data over a range of sample sizes ($n=10$ to 1000) both with, and without, process error (fishing and natural mortality, mortality, recruitment, total catch observations, and relative abundance observations). A comparison was made of the estimates obtained from a comparable analyser using the SSMULT_like for the effective sample sizes and total population sizes in the final year, (Table 3). For small sample sizes there is little difference as the process error is small relative to the sampling error, but the ratios of the last sample size to the first in the range tested were 100 and 43.9 for the true and estimated values, respectively; as the sample sizes get larger the process error begins to dominate as it should. While for the mid and low sample sizes the estimated total abundance was moderately over-estimated (around 25%), for the high-range values the estimates were less over-estimated (around 10%).

The simple fish model simulations were repeated including positive autocorrelation in the multinomial composition data, and with the analyser using SSMULT_like and including estimation of random effects with AU-AU autocorrelation (as described by Davies et al. 2019). Again, the estimates of effective sample size and total population abundance in the final model year were compared with the true simulation values, for scenarios with and without process and observation error. Without error, the analyzer estimated well the small sample sizes, and moderately over-estimated the large sample sizes, while for the simulations with error, the difference was larger, particularly for the large sample sizes, (up to -67.1%), because the process error begins to dominate, as it should (Figure 1). Without error, the analyzer total population abundance estimates were moderately lower, (up to -13.5%), but with error, the estimates were moderately higher at very low sample sizes (14.5%), but similar, albeit lower, in the mid to large sample sizes, (Figure 1).

The SSMULT_like with random effects estimation (AU-AU) as implemented in MULTIFAN-CL was used in fitting a complex tuna model (bigeye, McKechnie et al. 2017) having both length and weight composition data. This stock assessment model is complex being spatially disaggregated, having estimated movement, and with 32 defined fisheries, which offers a useful test case for the method. A key feature of the SSMULT_like model is the low effective sample size estimates for the size-frequency data. This was pronounced for the exceptionally large observed weight-frequency sample sizes as mentioned above, with the estimates being noticeably lower than the actual values for all the fisheries. A comparison was made relative to the same model fitted using a robust-normal composition likelihood (RobNorm), and the model quantities were reasonably similar, but with notable differences that can be attributed to the difference in how the size composition data were fitted. Differences in estimated selectivity-at-age were clear for some fisheries, that most likely contributed to the differences in the fitted proportions to the composition data, (Figure 2 and Figure 3), being visually somewhat worse for the SSMULT_like model. This comparison with the RobNorm suggested that using the SSMULT_like method has reduced the relative influence of the composition data in the integrated model fit. The implementation of this method for the large and complex tuna example amply demonstrates it is tractable in production-level stock assessment modelling. To achieve this a substantial amount of optimised coding was implemented including the tridiagonal solve and the Sherman-Morrison formula.

4.2 Censored gamma tagging likelihood

4.2.1 Rationale

The negative binomial likelihood has been used for fitting tag-recapture data but this approach does not effectively deal with the general problem in tagging data where a large number of small or zero observations occur due to patterns in the tag recapture data. An alternative and parsimonious approach using a modification of the Gamma distribution (with binning) developed by Dr David Fournier during 2019-20 that explicitly deals with these observations. Also, it is common for tagging data to be "pre-adjusted" for effects such as shedding, tagger performance, and some aspects of non-reporting, that renders the data as non-integers. The Gamma

distribution may deal more naturally with these data. This approach was termed the “censored gamma tagging likelihood” (tag_CDF_gamma).

4.2.2 Methods and testing

A proof-of-concept was demonstrated with a simple simulator of negative-binomial data and the binned Gamma distribution analyzer under several scenarios for observation error and the binning threshold. The positive results obtained prompted implementing the method in the MULTIFAN-CL code, and testing by using a real tuna fisheries model (skipjack, Vincent et al. 2019), with a comparison with the existing model that uses the “conventional” negative-binomial likelihood.

The testing design entailed fitting three models:

- **refcase** - solution obtained from a full doital fit (Vincent et al. 2019), with tagging data fitted using negative-binomial likelihood
- **CDF_gamma** - starting from the refcase solution, the tag_CDF_gamma approach was used and run to convergence
- **CDF_gamma_robust** - starting from the refcase solution, the tag_CDF_gamma approach was used with robustness to outliers, and run to convergence

For the tag_CDF_gamma models the censor value (binning threshold) was set to 0.9 effectively binning all values <1 and zero values. Comparisons were made among the converged solutions of the three models.

The proportion of bins in the observations that were censored (i.e. placed into the cumulative bin) was 0.996 for the cutoff value for censoring = 0.9, thus placing almost 100% of observations into the binned term of the likelihood. Noting that a recapture observation is indexed by: region, time period, fishing incident and length, such that for nearly all of the observed recaptures at length, the observation is zero. Also, the observations are "pre-adjusted" for the effects of: tag shedding, tagger performance, etc.; which scales down the observed recapture frequencies to real numbers, and in some cases less than 1. So, the observations are no longer integer frequency data.

The two tag_CDF_gamma models were similar absolutely and in their relative difference to the refcase model. Absolute biomass estimates were moderately higher (20 – 30%), with only minor (0.5 – 5%) differences in the reference point values of relative depletion. The fits of the tag_CDF_gamma models to the observed tag recaptures in respect of time at liberty appear noticeably “worse” than the refcase model for the short periods at liberty (Figure 4). While this effect will be case-specific depending upon the data quality and types in the integrated fit, this result illustrates the difference in the way the high numbers of zero recapture observations are accounted for in the tag likelihood and how it may alter the relative influence these data may have on the overall model fit. A direct comparison among the tau parameters for the negative-binomial and the tag_CDF_gamma, where tau is the relationship between the mean and the variance in each case, is not entirely correct. This is because the negative-binomial is discrete and the tag_CDF_gamma is continuous. The key difference among the two likelihoods relates to how each deals with values close to zero, and it is difficult to know how this might affect tau in every case.

4.3 **Catch-conditioned fishing mortality calculation**

4.3.1 Rationale

MULTIFAN-CL has a method called the “catch-errors” approach for calculating fishing mortalities that entails the estimation of a range of effort-related parameters: fishing incident-specific deviates on effort and catchability, and fishery-specific average catchability. Depending upon the model complexity in terms of numbers of fisheries and time periods, this fishing mortality parameterisation can result in a large number of parameters being estimated. For example, a recent complex tuna model, bigeye (McKechnie et al. 2017) entailed the estimation of around 9000 independent variables. While the catch-errors approach is effective, and especially in cases where effort is better observed than total catch, an alternative formulation with fewer variables could be an improvement, hopefully in terms of convergence characteristics and ultimately run time. During the second workshop of 2020, an approach that calculates fishing mortality conditioned upon the observed catches was formulated and implemented in MULTIFAN-CL. While the major part of this development

has been completed and preliminary tests with real examples have been completed, it is still in progress at the time of this report.

4.3.2 Methods and Testing

Essentially the catch-conditioned method is that employed in most contemporary fish population models, such that the fishing mortalities are solved conditional upon the observed catches. The formulation used in this development assumed that the Baranov catch equation held and the Newton-Raphson (N-R) procedure was used to solve for the fishing mortality levels among the various fisheries for which there were incidents within a time period.

There is a fundamental difference in the dynamic model calculations when using the catch-conditioned method compared to the catch-errors method. With the latter, all the estimated parameters for calculating fishing mortality (catchability, effort deviates etc.) are available at the beginning of the model calculation period, and therefore the numbers-at-age calculation is a linear sequence starting from the first model year. In contrast for the catch-conditioned method, none of the fishing mortalities are available, and these are calculated sequentially for each fishing incident with each time period starting in the first year. The numbers-at-age calculation will therefore include both the dynamic age-matrix and the fishing mortality calculations in each time period. This entailed a fundamental structural development to the model and hence substantial changes to the code.

For the fishery impact analysis, there is a fundamental difference with respect to this calculation undertaken when using the catch-errors method, where the **catchabilities** for certain fisheries are set to 0. Therefore, the fishing mortality for the fisheries with non-zero catches will be the same as for the normal fishing run. The predicted catches calculated within the impact analysis will therefore be larger (due to the larger population size). However, when using the catch-conditioned approach, in the impact analysis the **catches** for certain fisheries are set to 0. While the catches for the fisheries with non-zero catches will be the same as for the normal fishing run, the predicted fishing mortalities for these non-zero fisheries will therefore be smaller.

4.3.2.1 *Initial population at equilibrium*

Since the average initial total mortality, i.e. the survival rate, is not known at the beginning of year 1, a method was developed to allow for the estimation of the survival rate to be used for calculating the equilibrium population within the overall integrated model fit. The standard approach is to specify a period at the beginning of the model from which the average total mortality will be used for calculating the initial exploited population at equilibrium. A spline formulation was used for the age-specific estimated survival rates with the nodes being the independent variables. The initial population at equilibrium is then calculated with these rates, and the subsequent population calculation is done for all time periods following. A penalty function ensures the survival rate estimates are closely consistent with that of the actual total mortalities for the specified period used for the equilibrium calculation. Provision is made for the case where the spline degree can be increased from that of a previous solution.

For the fishery impact analysis, there is the added complication that the equilibrium survival rates estimated for the normal fishing run, will now be invalid because catches for particular fisheries are set to zero. For the impact analysis scenario, the initial survival rate must be recalculated as it now excludes the fishing mortalities for the fisheries with zero catches. An approach was developed that calculates the initial population using a nominal survival rate for the first iteration, and it is then solved for iteratively using the actual fishing mortalities of the dynamic model within the impact analysis, i.e. replacing the survival rates with the actual values at each iteration. Preliminary tests indicated this takes several iterations to converge.

Attention was also given to the formulation of the equilibrium population under conditions of multiple recruitments, regions, and movement within a calculation time period. A prototype formulation was developed that may improve on that currently being implemented in MULTIFAN-CL. This is still in progress.

4.3.2.2 *Fishing mortality levels and effort relationship*

The N-R solution for the fishing mortality levels applies in the cases where an observed catch is available. The exceptions are the cases where observed catch is unavailable. Dr David Fournier formulated an orthogonal-polynomial regression model using a Gram-Schmidt (G-S) design matrix and a robustified fit with a mixture

distribution to fit a fishery-specific relationship between the fishing mortality levels (estimated from the N-R) for those incidents where catch is available, to the observed effort (the **fm_level_rltnsnp**). The degrees for the estimated polynomial for each fishery are specified and the parameters are estimated within the integrated model fit. Then for those incidents having missing catch, the fishing mortality level prediction of the relationship using the observed effort is applied to predict the catch. Tests of this algorithm indicated that reasonable derivatives were obtained for the polynomial parameters. The calculation was also extended to cope with the case of grouped fisheries. The method entailed the following algorithm:

- Obtain the fishery and grouping specifications and construct the fm_level_rltnsnp parameters
- Build the G-S design matrix for each fishery/grouping – store
- Enter the fitting procedure
- Obtain the first model iteration values for the fm_level_rltnsnp parameters
- For incidents with observed catches use the N-R to calculate the fishing mortality levels
- Using the fm_level_rltnsnp parameters, the G-S design matrix, and observed effort, predict the fishing mortality levels for the incidents with missing catch
- Calculate the robust mixture distribution function for the fm_level_rltnsnp and add to the integrated model fit
- Enter next model iteration
- Repeat iterations until convergence

A simple simulation model was used to demonstrate the proof-of-concept and the fm_level_rltnsnp prediction for a single incident with missing observed catch illustrates the method (Figure 5). In the implementation of the method within MULTIFAN-CL a diagnostic report for all incidents having missing catches is produced containing the predictions obtained from the fm_level_rltnsnp. A penalty on the maximum age-specific fishing mortality is applied that prevents the N-R estimates exceeding levels above which the population cannot sustain, and therefore causing minimisation instability. The weight of this penalty is assigned by a flag setting.

The fm_level_rltnsnp plays a fundamental role in estimating fishing mortality for fisheries with standardised indices of relative abundance. This aspect of the feature is still in development and testing. Presently the sigma (maximum likelihood estimate of the standard deviation) for the regression is calculated and reported for each fishery. In theory, the fm_level_rltnsnp should be self-scaling, but how to achieve the correct relativity between the normal and the standardised fisheries remains to be developed. The set of fishery-specific fm_level_rltnsnps estimated for a testing example using a real tuna model (skipjack, Vincent et al. 2019) illustrate the implementation of the catch-conditioned method for a complex model with missing catches and effort in many incidents (Figure 6). Some of the fishery groups assumed seasonality in catchability, while others lacked effort in all but the final year.

4.3.2.3 Preliminary testing

Preliminary testing was staged, firstly using an example where all observed catches were available, and the feature dealt appropriately with incidents having missing effort. Secondly, the example introduced incidents having missing catches for which the estimated fm_level_rltnsnp appropriately predicted the fishing mortality. Thus, the catch-conditioned method demonstrated the estimation of fishing mortality for either instance.

The real tuna model example was then fit to convergence using a doital sequence of phases and compared with the corresponding solution using the catch-errors method. No informative comparisons among the model quantities are possible because the catch-conditioned method is still in the prototype phase, but the characteristics of the minimisation were compared. The catch-conditioned model entailed estimating around 70% fewer parameters ($n=2351$ versus 7602), and converged more quickly (Table 4). A comparison of the total likelihood is not valid because the models include different components (catch-errors model includes terms for effort deviates and total catch), however, the catch-conditioned model obtained an improved fit to the tagging data, with a somewhat worse fit to the size composition data. Other examples using the orthogonal-polynomial parameterisation have been explored, that has further reduced the number of parameters to around 600, being an order of magnitude lower than the current tuna models.

4.4 Length-structured model

4.4.1 Rationale

The initial development stage of this feature was described previously (Davies et al. 2019), but the rationale is repeated here to provide a background for the continuation of this feature in 2019-20.

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 were made for the underlying length-based growth functions needed for such a model, and in 2019-20 this was progressed to fish model simulation evaluations of the growth estimation capability, with a view to its implementation in MULTIFAN-CL.

4.4.2 Methods and Testing

This development required a simple fish model, and necessitated a simulation testing framework for evaluating the model. During 2019-20 this was the primary focus of development – to construct an analyser, being a fish model having a length-based growth transition matrix that is fully differentiable and that generates plausible derivatives for the length-based growth parameters, and that can estimate length-based growth parameters in an integrated model fit. 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. Simulation testing demonstrated that plausible growth curves were estimated relative to the simulator model, with a reasonable fit of the length-based analyser to the simulated size compositions, relative abundance indices, and total catches.

The simulation project was consolidated by conditioning the simulator for a plausible population state, and to test the analyser for coding errors using deterministic evaluations having the conditioned, fixed parameters. Stochastic simulations were then undertaken to evaluate the length-based model formulations. On the basis of these results, the feasibility for implementing the length-based structure into MULTIFAN-CL will be assessed.

4.4.2.1 Validating the analyser

A length-structured simulator model was developed so as to validate the code of the length-structured fisheries model analyzer. Using fixed input parameters for the population dynamics and length-based growth, a deterministic simulation run was done to simulated data without error. The length-structured analyser was fitted to these data, and produced a solution essentially identical to the simulator model. This validated the integrity of the length-structured analyser code.

4.4.2.2 Age-structured simulations

A true test of the length-structured analyser is to estimate the growth function of an age-structured model by fitting to simulated length-compositions generated by a von Bertalanffy growth function with assumed $sd(\text{mean_length_at_age})$. Using the age-structured simple fish model simulator, simulation length composition data were produced. Simulations were undertaken and comparisons between the simulation and analyser quantities were made. The modal length-frequencies for each age class estimated by the length-structured analyser were visibly very similar to those of the age-structured growth function (Figure 7), and the estimated mean lengths-at-age of the length-based growth function were consistent with those of the age-structured simulator (Figure 8). Given this consistency in the growth estimates, reasonably good fits to the simulation composition data were therefore obtained (Figure 9). This simulation experiment clearly illustrated the utility of the length-based growth function for approximating age-based growth.

In addition to estimating growth and population parameters, the SSMULT_like approach was used in the length-structured analyser for fitting the simulation composition data. The lower effective sample size estimates (ESS) obtained relative to the age-structured simulator may be explained because:

- The simulated composition data are not a "true" multinomial as they are generated from a parameterised model and not a natural multinomial distribution;
- The selectivity functions of the simulator and analyser are not directly comparable; and,
- The analyser includes other terms in the integrated total likelihood.

Also, the direction of change in the analyser ESS estimates were correct in that:

- With zero simulation error the analyser values were -59% with respect to the simulator; and,
- Including simulator process error, the analyzer values are -80.9% with respect to the simulator.

This achieved the primary aim during 2019-20, that was to validate the length-structured analyser as a reasonable estimator model, and to determine that there is considerable merit for its integration into the MULTIFAN-CL software. The next stage of this development is to evaluate the feasibility of including this structure within the code. The coding for the simulation framework was done in C++ routines within the ADMB project, so as to easily import the length-structured components into the MULTIFAN-CL project.

5 ENHANCEMENTS AND BUG FIXES

5.1 Stochastic Variability in N_{terminal}

The feature that generates stochastic variability in the terminal numbers at age (N_{terminal}) was reviewed and improved. The code that calculated the variance of N_{terminal} was computationally very intensive and somewhat out of date. Since then, the OpenBLAS library has been added to the MULTIFAN-CL project that makes available more efficient matrix calculation routines. The `inverse()` function used for the Hessian matrix was replaced with the `lapack_choleski_inverse()` routine. To confirm the operation of these routines, a trivial example was tested externally of MULTIFAN-CL that ensured accuracy to an acceptable level of precision.

The feature was improved by including the Hessian diagnostics within the operation since the stochasticity in N_{terminal} is determined from the Hessian solution. This now allows checking of the eigenvalues prior to generating the simulation files. If an error is reported (negative values), the user is advised not to proceed with using this solution for undertaking projections, and should investigate the parameterisation that produced this ill-determined solution. The user is asked to enter a "1" from the keyboard to continue anyway of that is desired. The existing algorithm is then completed that calculates the stochastic N_{terminal} .

The feature also undertakes the equivalent calculations for the case of zero-fishing to generate stochastic N_{terminal} under zero fishing mortality, and is operating correctly for all the possible population structures available in MULTIFAN-CL.

5.2 ADMB - print_matrix_derivatives()

Diagnostics of the derivatives are often required when introducing a new feature with its associated parameter estimation. A display of the derivative values during run-time is useful for diagnosing bad or zero derivatives. New routines: `df_print_array_derivatives()`; `df_print_matrix_derivatives()`, were added to the ADMB dependent library that readily allow examination of the derivatives of parameters that have array or matrix structures.

5.3 Optimised the generation of stochastic inputs

When simulations exclude stochasticity in N_{terminal} , there is some benefit in removing the requirement for the *.hes file to be available when generating the stochastic input files. In undertaking simulations over a large grid of models, there are real logistical problems due to:

- Generating the *.hes and subsequently the input files for stochastic projections
- Porting the *.hes to host machines, and the generation of the input files for stochastic projections on remote machines

Consequently, in cases where simulations exclude stochasticity in N_{terminal} , the operation described above in 5.1 was optimised out, thus removing the requirement for the *.hes file to be available when generating the stochastic simulation input files.

5.4 De-activate effort deviation penalties

As part of the catch-errors method for fishing mortality estimation, penalties are calculated for the effort deviate parameters being estimated. The exception to this is when observed effort is missing for particular fishing incidents. It may be that for certain fisheries, no observed effort will be available to the model, either due to the data collection regime, or as specified by the analyst. In these cases, a useful option is to specify particular fisheries for which the penalties on the "effort_dev_coffs" parameters are not calculated. This calculation was therefore made conditional upon the settings of `fish_flags(i,83)`, thus making the "effort_devs_coffs" penalty a fishery-specific function.

5.5 Report of operating model "true" simulation effort without observation error

In undertaking management strategy evaluations, it may be required to evaluate the performance indicators of a candidate strategy in respect of the Operating Model CPUE index. While simulation fishing effort data are routinely generated, this includes the observation error specified for the stochastic simulations. A new report was therefore created that provides the simulation fishing effort without observation error, i.e. the "true" Operating Model fishing effort. A setting for a flag option produces the report "effort_sim_true" having the same format as that for the stochastic simulation ("effort_sim"), but without the observation error. This enhancement avoids the complication of running the projection Operating Model twice, once with error and then again without error.

5.6 Natural mortality at age splines – parameter bounds

While testing the development of the catch-conditioned fishing mortality calculation, it was identified that the minimization became unstable when estimating age-specific natural mortality. This was due to exceptional iteration values for the mortality spline functions that cause the Newton-Raphson calculation to fail (not enough fish). The bounds for the spline parameters were therefore regulated to prevent such exceptional values during the minimization, thus achieving the stability needed.

5.7 Enhancements for the multi-species/sexes/stocks feature

5.7.1 Capability for no tagging data, single region, no movement

It became evident that no provision had been made for the case of: multiple species, sexes or stocks; lacking spatial structure or fish movement; and, with no available tagging data. The code development had excluded examples with these conditions, and therefore caused an error. Allowance was made for this special case and the associated routine calls were modified.

5.7.2 Changes to scalars for age_pars(3, 4) in parameter placement

The age_pars(3) hold the parameter estimates for the independent deviations from the von Bertalanffy curve in certain young age classes, while age_pars(4) hold the estimated growth curve deviations on mean length-at-age. For the case of multiple species, sexes or stocks, it was apparent that the scalars for the parameters were different for species 1 relative to species 2 or more. A correction was made that ensured the scalars were consistent among the species. This was found to alter the minimized solution for a model having multiple species, sexes or stocks.

5.8 Tagging negative binomial - upper bound on overdispersion parameter

A lower bound may be specified for the estimated overdispersion parameter of the negative binomial likelihood for tagging data that allows the exploration of scenarios in which these data may have lower relative influence in the integrated model fit. For the cases where the overdispersion estimate is hitting the default upper bound, or for delimiting overdispersion to be within a desired range, it may be necessary to specify an upper bound. A flag was therefore assigned to this function such that the upper bound is: $\log(\text{parest_flags}(358)/100.)$.

5.9 Bug fixes

5.9.1 Fix to total catch likelihood component for multi-stock model

A correction was made to the report of the total likelihood components (“test_plot_output”) that included the total catch likelihood values for the cases of multiple species, sexes or stocks. While the report included the component for single species models, it lacked the components for species 2 or more.

5.9.2 Fix to zero fishing mortality calculations with the Cobb-Douglas feature

In the special case of projections undertaken with both: zero fishing mortality (fishing impact analysis); and, the biomass-dependent catchability feature (Cobb-Douglas); an error was evident in the routine that returns the fishing mortalities (do_fish_mort_intermediate_calcs(int ir,int ip)). This was a syntax error causing an incorrect assignment and was corrected.

5.9.3 Fix to shared effort_dev_coeffs in multi-stock model

For the case of multiple sexes or stocks with the parameters “effort_dev_coeffs” being shared among the sexes or stocks, and with fishery grouping, the parameters were being incorrectly placed in the x-vector used in the minimization procedure. This was identified to be caused by a reversal in the order of the formal arguments to the set_value() routine such that the parameter was not being placed correctly, i.e. the model parameters did not depend upon the x-vector and so the derivatives with respect to the model function were zero. This was corrected in the routine’s signature.

5.9.4 Fix to random number seeds for simulation tagging data

It was identified when running simulation projections that generate stochastic tagging data that, despite changing “-tag_seed” argument value in the command line, the simulated tagging data does not alter. It was identified that the code prevented a change from the default seed value. It was amended to take the “-tag_seed” and to use it appropriately for the random number generation.

5.9.5 Fix to grouped and de-activated effort_dev_coeffs

It is possible that the “effort_dev_coeffs” parameters may be estimated according to particular fishery groupings and with a mixture of certain groups being estimated while others are fixed. In this case, a mis-

specification of the parameter indices in the `set_value_inv()` routines was occurring due to the incorrect grouping order being assigned. A logic check was inserted that indexes the activation flags relating to the fishery in respect of its grouping order not in respect of the fisheries number. Also, a new class was implemented that holds the shared grouping/active flags: `pgroup_manager_1`; which is a more efficient method for managing a mixture of flags.

5.9.6 Fix to duplicate records in "sizemult" report

The SSMULT method produces the "sizemult" report containing the observed and estimated effective sample sizes for the composition data. With the development of the SSMULT_like formulation during 2019-20, the code that writes to the report was duplicated such that duplicate records were produced. This was corrected for both the length and weight frequency composition data.

5.9.7 Fix to multi-species Richards growth parameter estimation

While the estimation of the Richards growth curve was improved for version 2.0.6.0, an oversight occurred with the placement of the parameter in the x-vector for the case of the multi-species/stocks/sexes model. This was corrected in the `reset()`, `xinit()` and `nvcal()` routines.

5.9.8 Fix to plot.rep for multi-species unique nage

An improvement was made to the multi-species feature in version 2.0.6.0 that allowed unique maximum ages for each species. However, the reporting of model quantities to the "plot.rep" was not fully tested in this case, and it fails in writing particular quantities. This was corrected for the quantities: natural mortality at age and the population state matrix, but fails for the fishing mortality at age and several other quantities. This will be corrected in subsequent versions.

5.9.9 Fix to tag_CDF_gamma diagnostic report

A diagnostic report for the censored Gamma likelihood for tagging data (`tag_CDF_gamma`) is produced that summarises the censor value and the proportions of observations binned in respect of it. The assignments needed for generating this report were implemented for the censored Gamma option, but had not been done for the robustified option. This was corrected by replicating the `sensor_report` assignments into the `censored_gamma_mix()` routine.

5.9.10 Fix to mean weight used in Newton-Raphson for projections

In the case where projections include incidents with missing effort and the number of recruitments per year is out of phase with the fishing periods, an error occurred with the mean weight variable used in the Newton-Raphson calculation of fishing mortality in these projection period fishing incidents. The correct subscript for the `mean_weight_yr` variable was made to deal with this case.

5.9.11 Fix to fm_levels assignment within the Newton-Raphson for projections

For projection fishing periods where there is a mixture of fishing incidents for which observed catches are both available and missing, the Newton-Raphson routine solves for the fishing mortality levels (`fm_levels`) for the available catches, while the fishing mortality is calculated using the observed effort for the unavailable catches. Using the `fm_levels`, the correct catches were being removed from the population, but the assignment to the class member was not being made, such that an error was occurring in the simulation data report that required the `fm_levels`. This was corrected, made no difference to the projection results, but produced the simulation reports as needed.

5.9.12 Fix to SSMULT without random effects estimation ESS diagnostic reports

Only for the SSMULT method without random effects estimation, was an error produced in the "sizemult" report that contains the observed and estimated effective sample sizes for the composition data. This report had been tested positive for the SSMULT method with random effects estimation, but for the routine without random effects, `new_len_self_scaling_multinomial_nore()`, a reporting variable was being assigned within a local scope, and so was not available for the report. The correction was made for the reports for both the length and weight composition data.

6 APPLICATION OF NEW FEATURES

A number of the new features and enhancements described in sections 4 and 5 were employed for developing the yellowfin and bigeye stock assessments, and for the MSE work, undertaken at SPC-OFP in 2020. These included:

- The SSMULT_like was used to explore one-off sensitivities of the assessment models. The minimization performance using this feature was low relative to the robust-normal, and therefore it was not possible for it to be used uniformly over the entire structural uncertainty grid. This feature was also used across the MSE grid of models to determine the ESS that specified the generation of stochastic length composition data.
- The tag_CDF_gamma was used in early exploratory model runs, but was not employed for the final assessment models for which the negative-binomial likelihood was employed.
- The enhancement for the optimized generation of stochastic simulation input files that precludes the *.hes file (section **Error! Reference source not found.**) was employed within the MSE grid of models.

The application of these features for each assessment are described by Ducharme-Barth et al. (2020), and Vincent et al. (2020), and for the MSE by Scott et al. (2020).

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 2020-21, 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: completing the catch-conditioned fishing mortality calculation; assessing the feasibility of implementing length-structure in MULTIFAN-CL; various improvements in stochastic functionality as needed for management strategy evaluations (MSE); and, completing the draft paper of the SSMULT method. 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 2019-20 period that have been added to the work plan for 2020-21 include:

- Estimation of region-specific stock-recruitment relationships
- Refining the format of the plot.rep report and adding more detail to the xinit.rpt

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

- Non-uniform size bins (recommendation "b")
- Long-term tag loss (recommendation "c") – proposed for 2020-21

Important developments that have not been addressed in previous years and that should have priority in 2020-21 relate to the simulation mode for its application to MSE. These are:

- 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, random selectivity deviates, and, overdispersion in the probability of tag recaptures.

8 DISCUSSION

The highlights of 2019-20 are several innovative and promising developments from the lead developer, Dr David Fournier. Some of these developments were unforeseen.

The development of the self-scaling multinomial for composition data (SSMULT) for the improved likelihood formulation is now essentially complete, with: positive results for the simulation testing; its implementation in the MULTIFAN-CL code; and, its successful application to a large, complex tuna model. This innovative approach demonstrates a truly self-scaling likelihood function. It offers a substantial advance in

fisheries modelling methodology that integrates multiple data types in the non-linear regression by addressing the weaknesses of the multinomial distribution, and removing the need for assumptions regarding data precision. These results have been drafted in a report with the introduction and discussion sections in progress. This brings a substantial body of development work near to completion.

The censored Gamma likelihood for tagging data is another innovative development. Examples used for testing the feature illustrated the prevalence of zero observations in tagging data (>99%), which this method formally assigns into a single aggregate term. Preliminary testing indicates the method is tractable with real tuna data. There is certainly merit in further exploring its utility in a range of examples which can be used to elucidate its effects on the inferences made from these data.

A proof-of-concept length-structured population model with a fully differentiable growth transition matrix produced positive results in simulation tests, thus offering a promising and innovative approach for estimating length-based growth within a population model. The next stage is to assess the feasibility of implementing it within the data structures in MULTIFAN-CL which might determine if this work is to be continued in the current, or a new, software project.

In late 2019 the priority was raised for the implementation of the catch-conditioned method for estimating fishing mortalities. While initially thought to entail only testing and refining the existing code, it in fact resulted in an unforeseen and high demand on resources in the latter half of 2019-20. The method is fundamentally different in respect of the dynamic model calculations compared to the existing catch-errors method, and entailed a complete revision of the prototype code previously drafted for the method in much earlier versions (circa. 2008). Preliminary testing with a real tuna example has been positive with a 70% reduction in the number of parameters, and with similar or improved minimisation performance. This development is currently still in progress.

The unforeseen tasks during 2019-20 included the tag_CDF_gamma likelihood and the raised priority of the catch-conditioned feature. This resulted in many of the items in the 2019-20 workplan not being addressed, and which are recommended to be rolled over into the work plan for 2020-21, including the two outstanding BET Peer Review recommendations. In respect of its utility within the MSE framework, no progress was made on the planned items, with only several enhancements or support matters accomplished. As in previous years, flexibility in the work plan was required during 2019-20, given the unforeseen items arising that attracted priority. Also, a large number of corrections were required (12 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.

As in previous years, improving the support structures of the MULTIFAN-CL project have received limited attention during 2019-20. However, a good advance was made in the testing aspect of the project with the benchmark testing framework being expanded to include the macOS executable of the development version. Thus, the tests now undertake pair-wise comparisons among versions and on all three platforms: Linux, Windows and macOS. As always, this framework uses the parallel processing capacity of the Condor facility at OFP. Minimal differences were apparent among the platforms and the posting of version 2.0.7.0 in February 2020 included all three executables. The Windows and macOS releases included digital signatures for the Pacific Community. The highlight of this aspect of the project was the posting of version 2.0.7.0 with the first ever release of the macOS executable of MULTIFAN-CL. This greatly improves the utility of MULTIFAN-CL among users operating on any of the three platforms. The forthcoming challenge is to prepare for the change by Apple machines to use the RISC ARM-based CPU, and a strategy for the MULTIFAN-CL project to accommodate this has been suggested.

Short-term priorities for work in 2020-21 are to complete the primary items currently in progress: the catch-conditioned fishing mortality calculation; and, completing the draft paper that introduces SSMULT_like feature. Other immediate priorities in the support aspect of the project include: benchmark testing the development version (that now includes the catch-conditioned feature); and, updating the User Guide that lacks adequate descriptions for many of the new features in versions 2.0.6.0 and 2.0.7.0.

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
- Davies, N., Fournier, D., Takeuchi, Y., Bouyé, F., and Hampton, J. 2019. Developments in the MULTIFAN-CL software 2018-19. WCPFC-SC15-2019/SA-IP-02
- Ducharme-Barth, N., Vincent, M., Hampton, J., Hamer, P., Williams, P. and Pilling, G. 2020. Stock assessment of bigeye tuna in the western and central Pacific Ocean. WCPFC-SC16-2020/SA-WP-03, Online, 11–20 August 2020.
- 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
- Hrafinkelsson, 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., 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, R., Scott, F., Pilling, G. and Hampton, J. 2020. Updating the WCPO skipjack operating models for the 2019 stock assessment. WCPFC-SC16-2020/MI-IP-08, Online, 11–20 August 2020.
- 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.
- Vincent, M., Ducharme-Bath, N., Hampton, J., Hamer, P., Williams, P. and Pilling, G. 2020. Stock assessment of yellowfin tuna in the western and central Pacific Ocean. WCPFC-SC16-2020/SA-WP-04, Online, 11--20 August 2020.

10 TABLES

Table 1. New features added to MULTIFAN-CL with respect to their state of completion as of July 2020.

Peer review recommendations		
Task	Description	Status of completion
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
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%
Binned likelihood for tagging data	Implement a censored Gamma likelihood for tagging data that better deals with large numbers of zero observations.	Development 100%; Testing 100%
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 revised formulation for SSMULT_like.	Development 100%; Testing 100%
Length-structured model	Simulation project - design experiment; undertake simulations; draft report; assess feasibility for implementation in MULTIFAN-CL code	Development 100%; Testing 100%
Outstanding testing of existing features	Catch-conditioned estimation of fishing mortality	Development 70%; Testing 50%
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: <ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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_{terminal} (more stable method is to use terminal year less 5 as the period for obtaining variance) and eff_devs 	Development 0%; Testing 0% Development 100%; Testing 90%

Table 2. New features to be added to MULTIFAN-CL in 2020-21 and subsequent years, and those for which implementation and testing is to be completed.

Peer review recommendations		
Task	Description	Implementation
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.	2021-22
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.	2021-22
d. Tags inform movement	Testing for the case of multi-species/stocks/sexes.	Development 100%; Testing 90%
Other developments		
Task	Description	Implementation
Length-structured model	A length-structured model with a differentiable growth transition matrix – assess feasibility for implementation in MULTIFAN-CL	2020-21
Recruitment random effects	Report on the feasibility of its implementation in MULTIFAN-CL.	2020-21
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.	2020-21
Tags inform growth	Development of a feature that incorporates size data from tag recaptures to inform growth estimation.	2021-22
Self-scaling multinomial with random effects	Complete draft paper for peer review.	2020-21
Recruitment correlates	Region-specific environmental recruitment correlates estimated within the orthogonal polynomial parameterisation for recruitments	2020-21
Movement correlates	Add a time-series structure (e.g. random walk, time blocks or using environmental correlates) to movement coefficients	2020-21
Recruitment deviate penalties	Allow for time-variant penalties on recruitment deviate estimates	2021-22
Tagging multi-sex	Account for instances of differences between size composition the tag releases and the sex-specific populations	2021-22
Region specific SRR	Allow that region-specific spawning biomass is responsible for recruitment within that region. This is consistent with the	2021-22

	assumption that stocks may not be truly panmictic. This would estimate region-specific SRRs.	
Outstanding testing of existing features	<ul style="list-style-type: none"> - Catch-conditioned fishing mortality for multi-species - 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 	<p>Complete development 2020-21</p> <p>2021-22</p>
Report comments	Add comment descriptions of the selectivity parameter configurations in the output .par and .rep reports	2021-22
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.	2020-21
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.	2020-21
Stochastic projection functionality	<ul style="list-style-type: none"> - implement process error in future recruitments with application of the derived autocorrelation coefficient in historical recruitment estimates - consolidate the generation of stochasticity in effort_dev_coffs 	<p>2020-21</p> <p>2020-21</p>
Turing test	<p>Ensure the quality of pseudo-observations to be made more realistic by:</p> <ul style="list-style-type: none"> - Including the sel_dev_coffs and effort_dev_coffs estimates in applying process error in projection size compositions and effort - Including over-dispersion error in tagging data 	2020-21

Table 3. True and estimated values for sample sizes (ESS) and total population sizes in final year (Nterm) using a simple simulation fish model having 10 bins and a fitted analyzer using the SSMULT_like, with and without process error.

		ESS				Nterm
True	no_err	err	True_no_err	est_no_err	True_err	est_err
10	6.4	6.2	30810	36536	31015	38295
20	17.8	16.3	30810	37144	31015	39332
50	58.8	48.8	30810	31774	31015	36977
100	125.5	90	30810	35991	31015	41153
500	649	222	30810	30685	31015	33795
1000	1296.3	271.9	30810	31634	31015	34692

Table 4. The minimisation solutions for the SKJ2019 example fitted using alternative methods for estimating fishing mortality: the catch-errors (catch_err) and catch-conditioned (catch_cond) methods; with the relative percentage difference with respect to the catch_err model.

	catch_err	catch_cond	%diff
Objective function	-272830.1	-271741.3	-0.40
Maximum gradient	9.795e-04	7.817e-04	-20.19
No. parameters	7602	2351	-69.07
Evaluations in final phase	5942	4459	-24.96

11 FIGURES

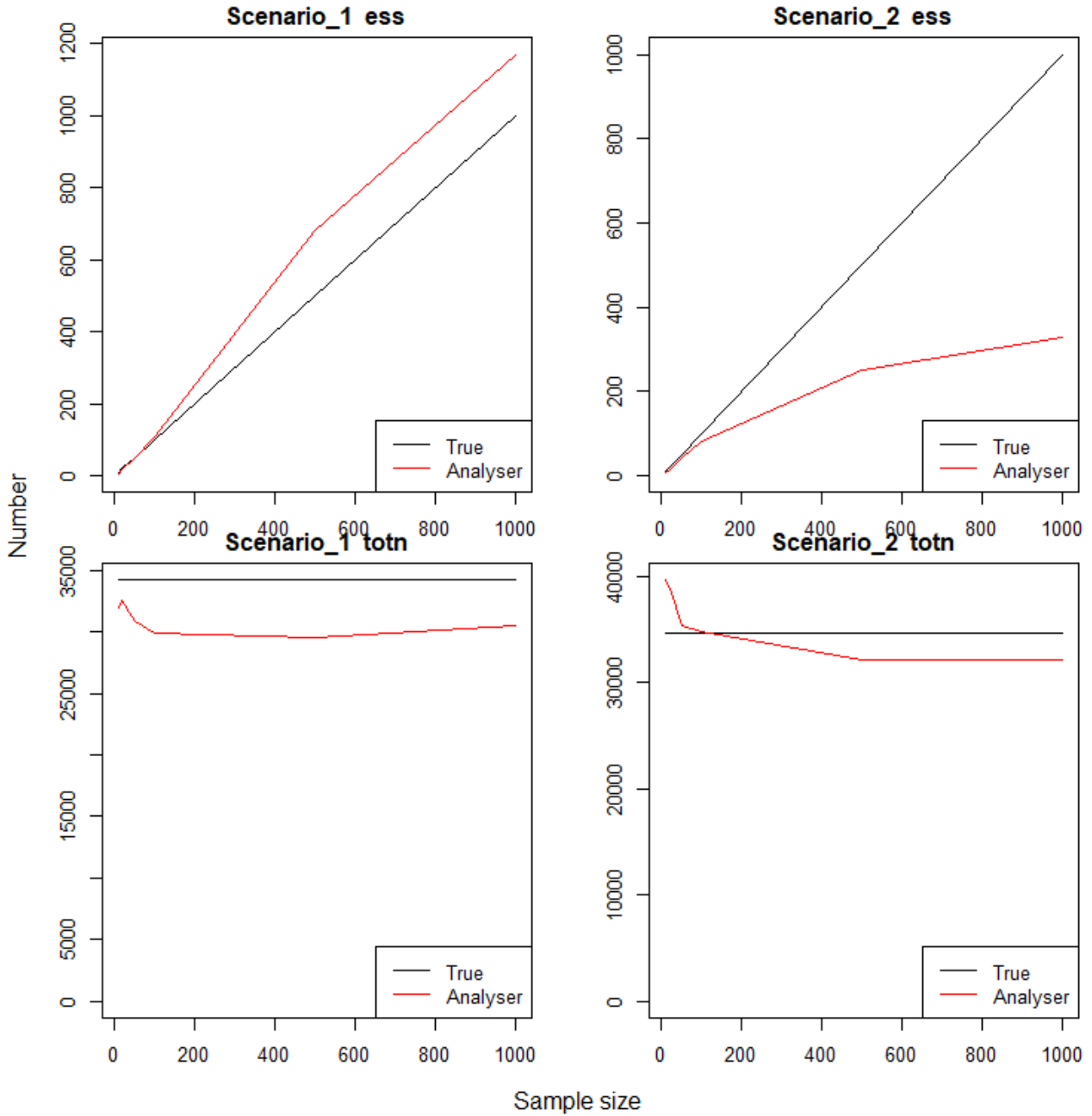


Figure 1. Simple fish model simulator (True) and analyser (Analyser) values for the effective sample size (ess, top panels) and total population abundance in the final model year (totn, bottom panels) under scenarios without (Scenario_1, left panels) and with (Scenario_2, right panels) process and observation error.

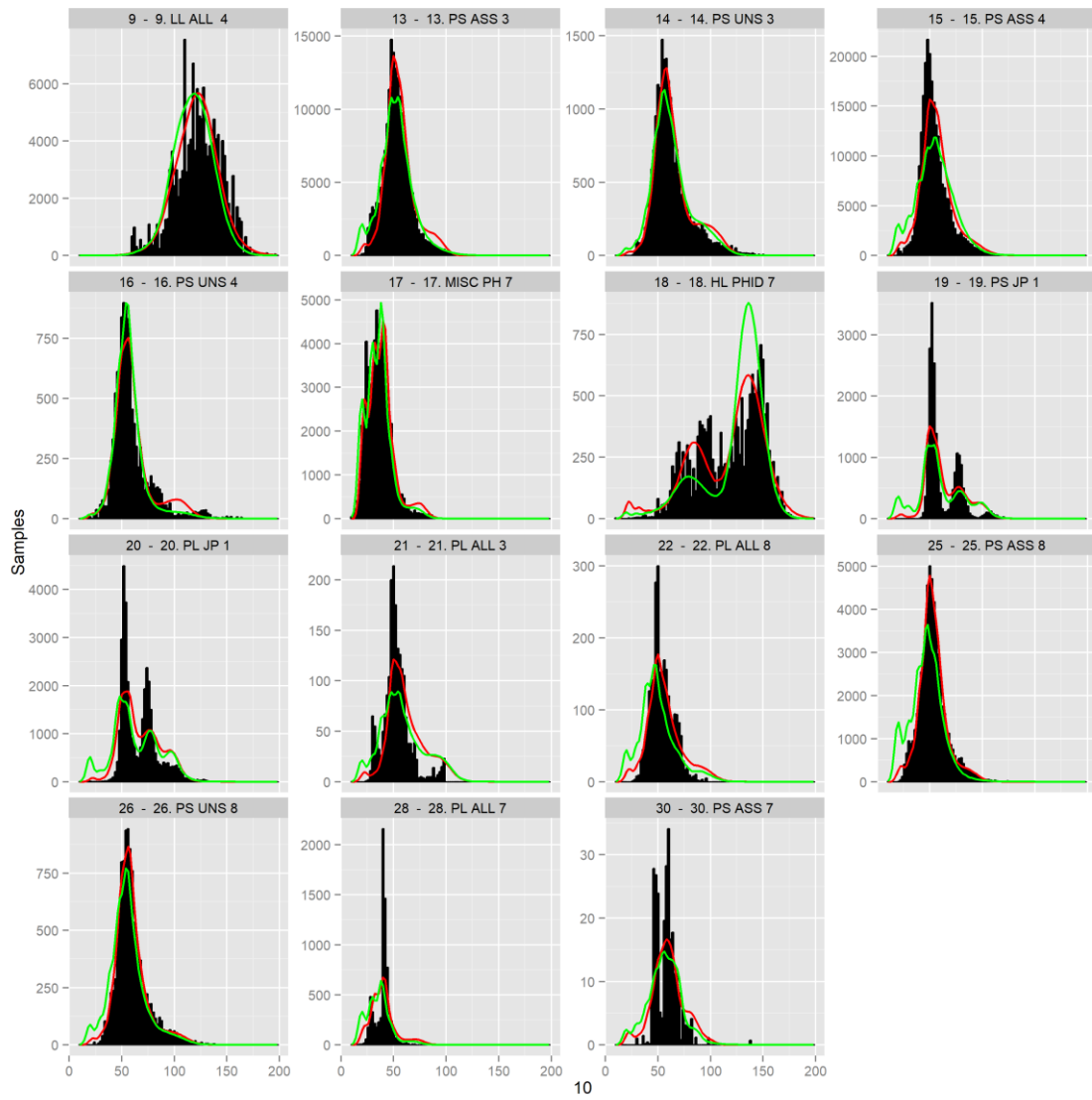
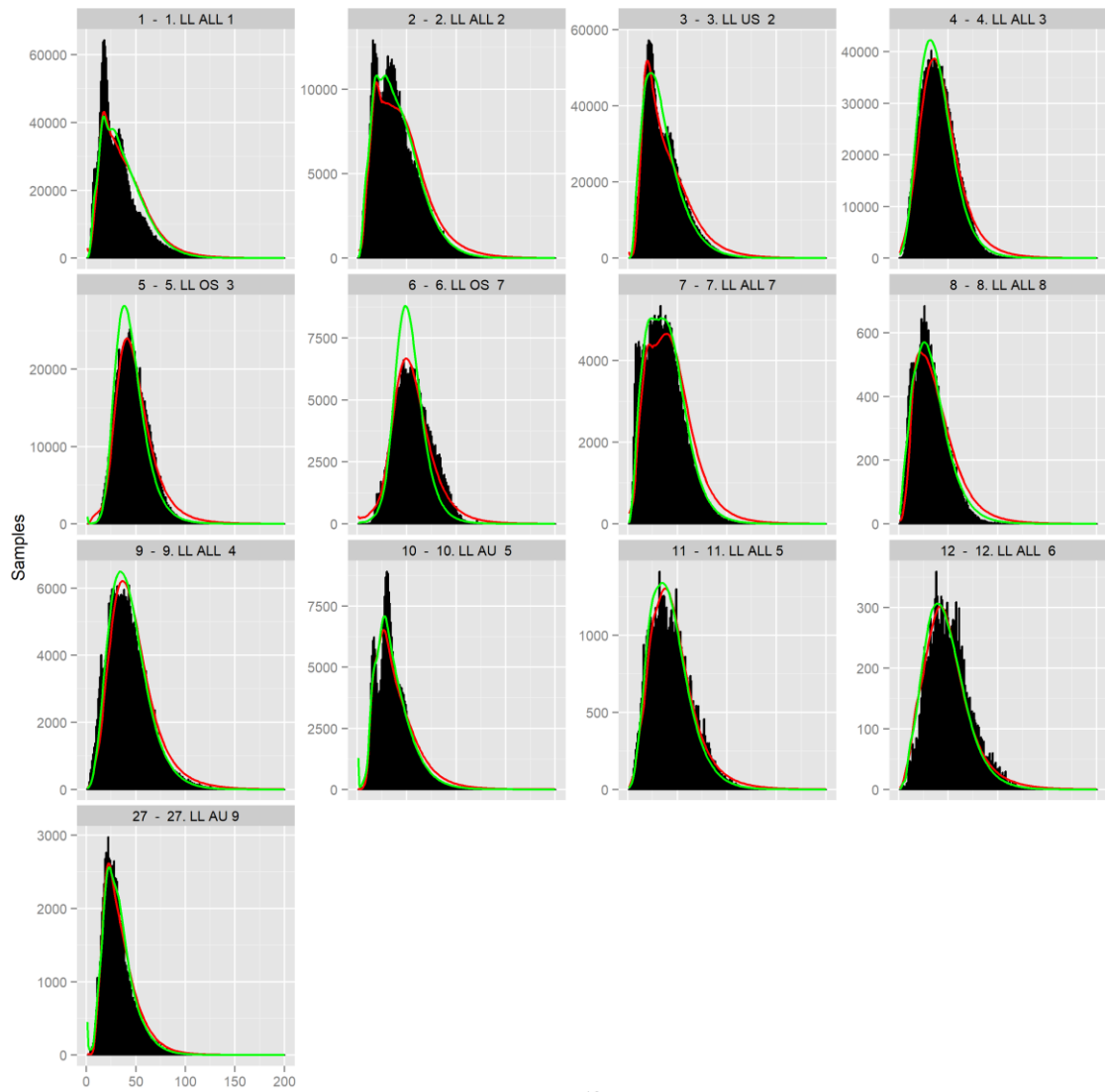


Figure 2. The fit to the aggregated length frequency data from each fishery (histograms in individual panels) for the RobNorm (red lines) and SSMULT_like (green lines) models.



10

Figure 3. The fit to the aggregated weight frequency data from each fishery (histograms in individual panels) for the RobNorm (red lines) and SSMULT_like (green lines) models.

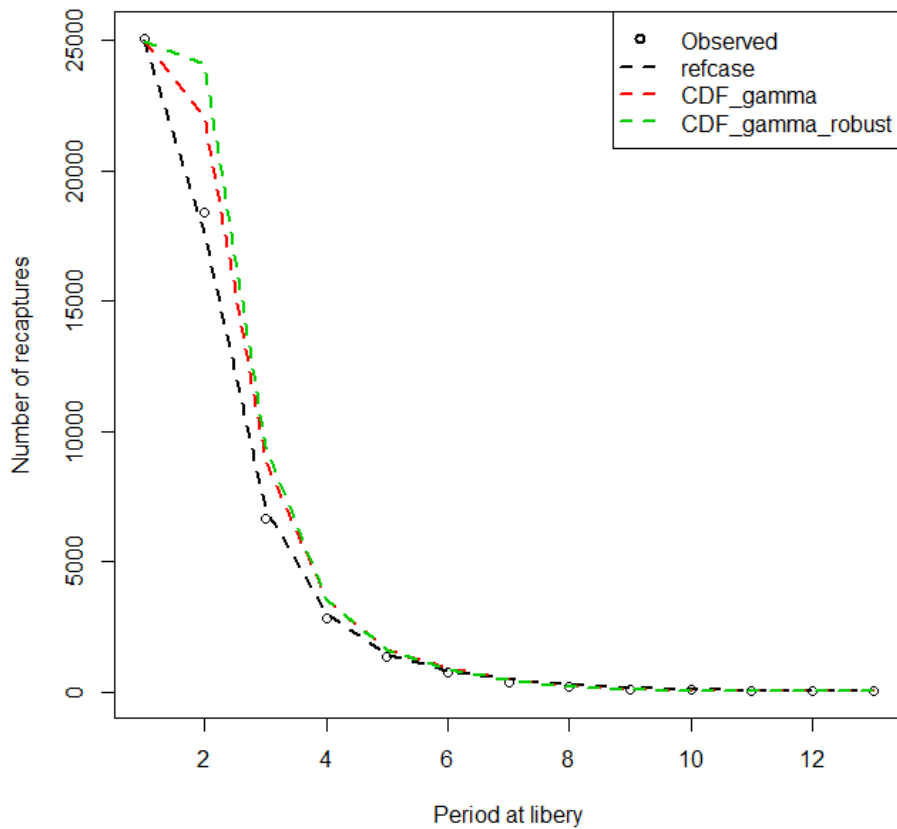


Figure 4. Comparison of the observed and fitted tag returns in respect of periods at liberty for the refcase model, the models employing: the tag_CDF_gamma likelihood (CDF_gamma); and, the tag_CDF_gamma likelihood with robustness to outliers (CDF_gamma_robust).

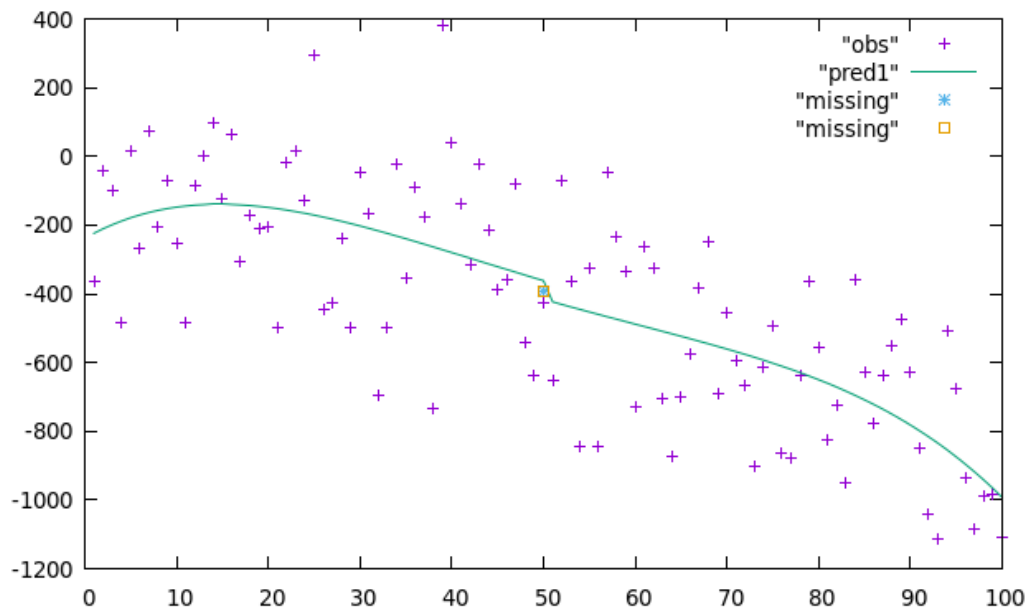


Figure 5. A simple simulation model of the fishing mortality level and effort relationship, “pred1”, fitted to observed catches “obs”, with a prediction, “missing”, for a single incident with missing observed catch.

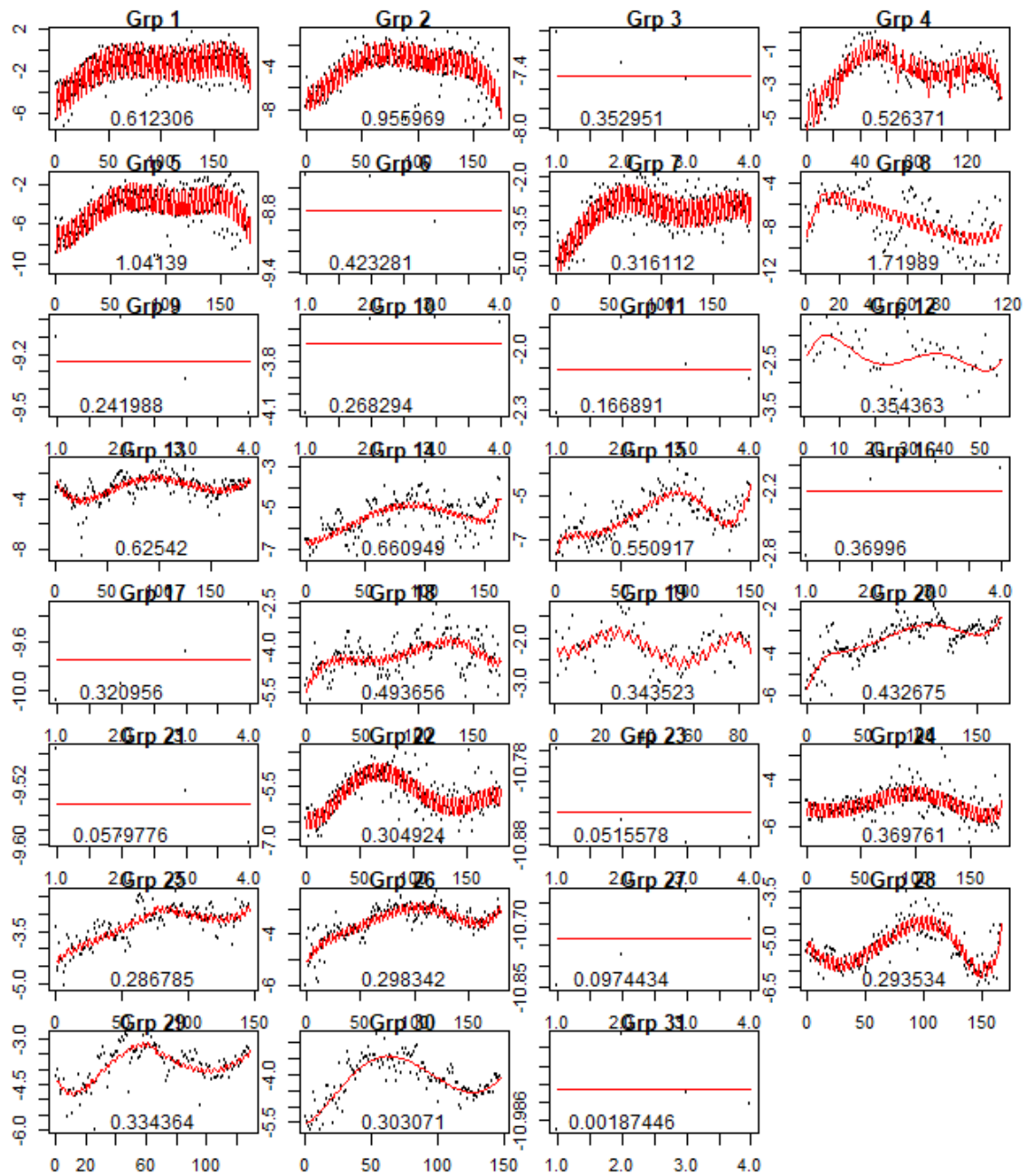


Figure 6. The estimated relationships between fishing mortality levels and observed effort (red lines) specific to fishery groupings as part of the catch-conditioned method of estimating fishing mortality. This was from a testing example using a real tuna model (skipjack, Vincent et al. 2019).

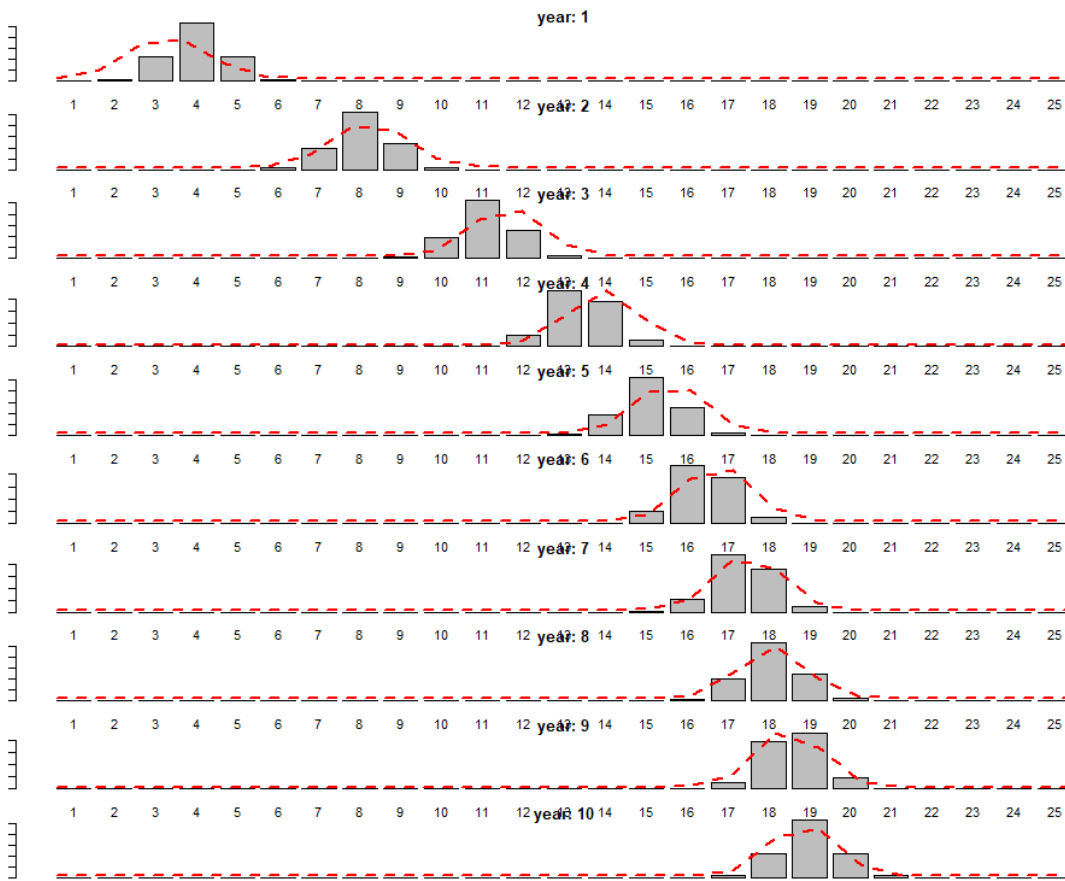


Figure 7. Modal length-frequencies for each age class as simulated from an age-structured growth model (histograms), and compared to those estimated by a length-structured analyser (dashed red lines) fitted to the simulation data..

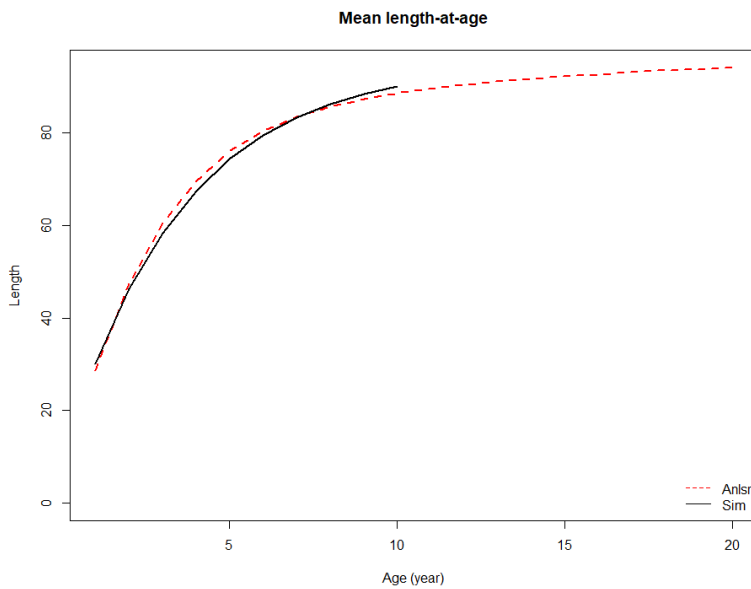


Figure 8. Comparison of the mean length-at-age estimated from the length-based analyser model (Anlsr, red dashed line) with that of the age-structured simulator (Sim, black line).

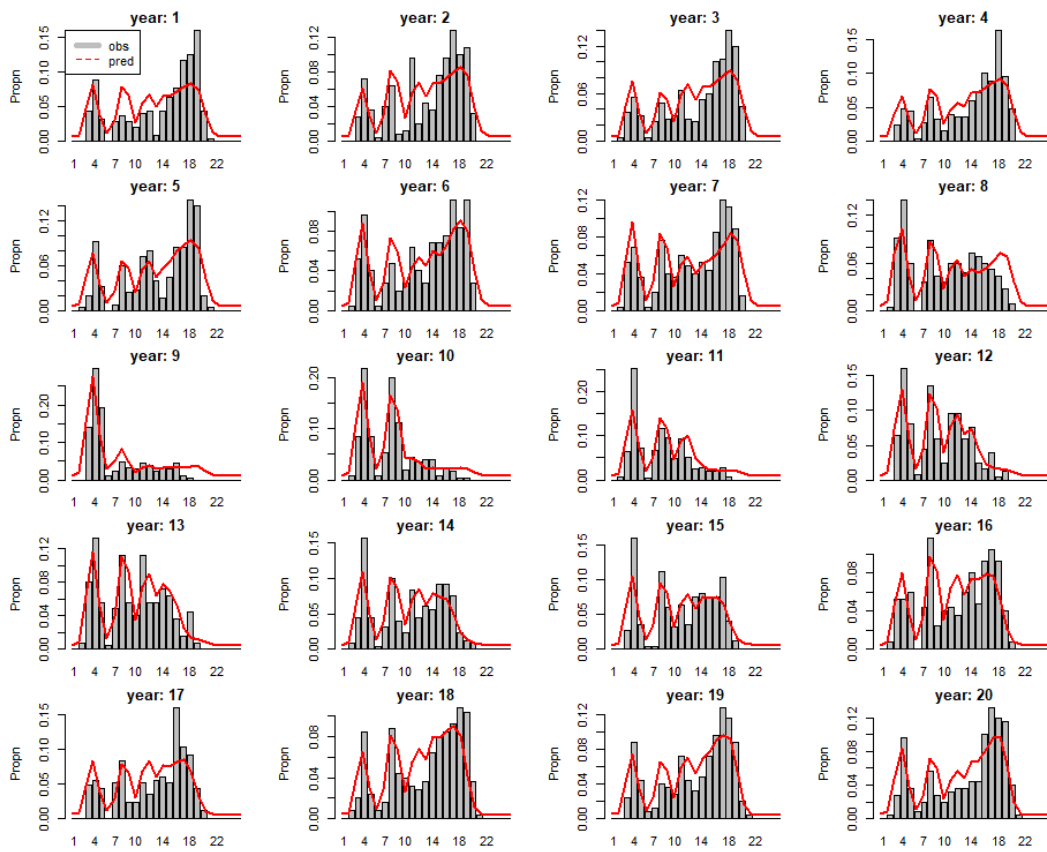


Figure 9. Fits of the length-based analyser model (pred, red dashed line) to the simulated size compositions (obs, histograms) for each year of the simulation model run (nyears = 20).