



**SCIENTIFIC COMMITTEE  
TWELFTH REGULAR SESSION**

Bali, Indonesia  
3-11 August 2016

---

**DEVELOPMENTS IN THE MULTIFAN-CL SOFTWARE 2015-16**

---

**WCPFC-SC12-2016/SA- IP-10**

**Nick Davies<sup>3</sup>, Dave Fournier<sup>2</sup>, Fabrice Bouyé<sup>1</sup>, and John Hampton<sup>1</sup>**

**Oceanic Fisheries Programme  
Secretariat of the Pacific Community, Noumea, New Caledonia**

<sup>1</sup> Oceanic Fisheries Programme, Pacific Community, Noumea, New Caledonia

<sup>2</sup> Otter Research Ltd, North Saanich, British Columbia, Canada

<sup>3</sup> Te Takina Ltd, Whangarei, New Zealand

# Table of Contents

Executive summary.....	4
1 Introduction.....	5
2 Development overview.....	6
2.1 Team.....	6
2.2 Calendar.....	6
2.3 Collaboration and versioning.....	6
2.4 Merging the development branch to the trunk.....	6
2.5 Compilation framework and Source code repository.....	7
2.5.1 Compilation framework.....	7
2.5.2 32-bit architecture.....	8
2.5.3 Source code repository.....	8
2.6 Developer’s workshops.....	8
2.7 Postings to website.....	8
2.7.1 29 February 2016 – version 2.0.0.1.....	8
2.7.2 19 April 2016 – version 2.0.0.2.....	9
2.7.3 28 April 2016 – version 2.0.0.3.....	9
2.7.4 21 June 2016 – version 2.0.1.1.....	9
2.8 Independent Peer Review of the 2011 bigeye tuna stock assessment.....	9
2.9 Tool development.....	10
2.9.1 R4MFCL.....	10
2.9.2 Testing framework.....	10
2.9.3 Viewer.....	12
2.9.4 Condor parallel processing facility.....	12
2.10 User’s guide.....	12
3 Training workshop.....	12
4 New features.....	13
4.1 Composition data likelihood.....	13
4.1.1 Rationale.....	13
4.1.2 Methods and Testing.....	13
5 Other enhancements and bug fixes.....	16
5.1 Extension of feature for assumed terminal recruitments.....	16
5.2 Revised approach for simulation projections.....	17
5.3 Streamline the generation of stochastic simulation inputs.....	18
5.4 Restructuring of the temporal recruitment deviates.....	18
5.5 Output of tag releases at age.....	19
5.6 Consolidated the tagging negative-binomial likelihood.....	19
5.7 Report of variance of the BH-SRR.....	19

5.8	Estimation of age-specific selectivity deviation coefficients (sel_dev_coffs).....	19
5.9	Crash save facility and capability for fit-autonomy (self-restart).....	19
5.10	Extension to initial population conditions.....	20
5.11	Fix within robust-normal likelihood .....	20
5.12	Fix within population projections .....	20
5.13	Increased independent variables.....	20
6	Application of new features .....	20
6.1	Terminal recruitments fixed at the arithmetic mean.....	21
6.2	Output of tag releases by age-class.....	21
6.3	Report of BH-SRR variance .....	21
6.4	Extension to initial population conditions.....	21
7	Future work .....	21
8	Discussion.....	21
9	References .....	22
10	Annex.....	30
10.1	Benchmark testing.....	30
10.2	Self-scaling Multinomial with random effects – flag allocations.....	32
10.3	Agenda for Training Workshop, March 2016 .....	34
	Agenda .....	34
10.3.1	Day1.....	34
10.3.2	Day 2.....	34
10.3.3	Day 3.....	35

## 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, Pacific Community) from August 2015 to July 2016 and updates the report of Davies et al. (2015). Highlights of this period are:

- Implementation of an innovative approach for fitting to observations of size composition
- Completion of the merging of the software's development version to the release version with posting of version 2.0.0.1
- A 4-day training workshop for stock assessment analysts at Nouméa, March 2016

The project's primary focus in 2015-16 related to developing the self-scaling multinomial with random effects for size composition data (SSM-RE). This innovative approach offers two key improvements over the current method. Firstly, effective sample sizes are estimated and therefore it has self-scaling properties which addresses a major source of uncertainty in stock assessment models, i.e. the arbitrary assignment of "relative importance" to the various data types in the integrated modelling approach. Secondly, it naturally deals with zero proportions, whereas an arbitrary constant is added in the robust-normal likelihood. The SSM-RE development has taken account of the several weaknesses in using the multinomial distribution, and the simulation testing (independent of MULTIFAN-CL) has indicated good performance. It has been implemented in the MULTIFAN-CL software and trials using a recent complex tuna model as an example, show satisfactory performance. A draft paper is currently in preparation.

As most of the new features developed in the past 3-4 years were available only in the development version, it was essential and urgent to merge this development branch into the trunk version. This specific sub-project entailed benchmark testing of the development version, establishing a new Windows cross-compilation, updating the User's Guide, and merging the development branch to the repository trunk. Upon completion, the new release version 2.0.0.1 was posted in February, 2016. This was a high priority in 2015-16, a landmark achievement, and made available the substantial new features to MULTIFAN-CL users.

The two primary developers presented a four-day training workshop to stock assessment analysts, and other observers, at OFP (Nouméa), being the first such workshop undertaken. This had utility in improving their skills in using MULTIFAN-CL, gave hands-on experience, and introduced the various new features in version 2.0.0.1, many of which have utility for the 2016 stock assessments.

In addition to these highlights, the following important advances were made during 2015-16:

- Four benchmark tests were completed for the release and development versions, which ensures the integrity of existing features and parity among the versions.
- Four release versions 2.0.0.1 to 2.0.1.1 were posted on the website.
- A number of new features were added as required for specific stock assessment applications, and several bug fixes were made, mostly associated with new features.
- The compilation framework was upgraded on a new computer, which included the MinGW cross-compilation of the Windows executable, and the SVN source code repository was successfully migrated to a GitHub repository.
- The algorithm for stochasticity in projections of the simulated recruitments as deviates about predictions of the Beverton-Holt stock-recruitment relationship was improved.

Ianelli et al. (2012) reported thirteen recommendations from an independent peer review panel specifically relating to MULTIFAN-CL. The rationale for these recommendations is 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. These have been central to the MULTIFAN-CL developments in the past 3-4 years, and over this period, of the thirteen recommendations, six have been implemented and tested, 3 have been implemented but testing yet remains in progress, and 4 remain yet to be developed (Davies et al. 2015). The status of

implementing the recommendations yet to be completed is listed below. Developments for several other new features are also listed.

<b>Peer review recommendation</b>	<b>Status of completion</b>
i. Tail compression	Development 100%; Testing 50%
j. Multinomial distribution for size composition data	Development 100%; Testing 60%
l. Tag likelihood relative weighting	Development 100%; Testing 50%

Upon completion of the SSM-RE development in 2016-17, the testing of the above Peer Review recommendations will be resumed.

It is suggested that the priority development tasks for the forthcoming 2016-17 year include the remaining four Peer Review recommendations yet to be developed. These are outlined in the following list among others, including the testing of the above three recommendations.

<b>Peer review recommendation</b>	<b>Implementation</b>
b. Non-uniform size bins	Development
c. Long-term tag loss	Development
d. Tags inform movement	Development
i. Tail compression	Testing
j. Multinomial distribution for size composition data	Testing
k. Maturity-at-length	Development
l. Tag likelihood relative weighting	Testing
<b>Other new features</b>	<b>Implementation</b>
Self-scaling multinomial with random effects	Draft report
Initial biomass approximates the equilibrium unexploited biomass	Testing
Exclude reporting rates from tag predictions during mixing period	Testing
Region-specific yield estimation	Testing
Tags inform growth	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 Dave Fournier of Otter Research Ltd for application to south Pacific albacore tuna (Fournier et al. 1998).

MULTIFAN-CL is typically fitted to total catch, catch rate, size-frequency and tagging data stratified by fishery, region and time period. For example, recent tropical tuna assessments (e.g. Davies et al. 2014, McKechnie et al. 2016) encompass a time period of 1952 or 1972 to 2012 or 2015 in quarterly time steps, and model multiple separate fisheries occurring in 5 to 9 spatial regions. 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), 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 likelihood comprised of the fits to the various data types, and penalized likelihood distributions for various parameters.

Each year the MULTIFAN-CL development team works to improve the model to accommodate changes in understanding of the fishery, to fix software errors, and to improve model features and usability. This document records changes made since August 2015 to the software and other components of the MULTIFAN-CL project both for the current release version (2.0.1.1), and the current unreleased development version, and updates the report for the previous period, 2014-15, (Davies et al. 2015).

## **2 DEVELOPMENT OVERVIEW**

### **2.1 Team**

The senior developer of MULTIFAN-CL is Dr Dave Fournier, of Otter Software in Canada. Assisting with programming is Nick Davies, with occasional programming carried out by John Hampton (SPC, New Caledonia). Other tasks include testing and debugging (ND, JH, and Fabrice Bouye (SPC)); documentation (ND); and planning and coordination (DF, ND, and JH). Related project software are developed or managed by FB (MULTIFAN-CL Viewer, Condor, GitHub, Jenkins), ND, and Robert Scott (R4MFCL).

### **2.2 Calendar**

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

December – February: Testing and ongoing code development

March: Developer’s workshop

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

### **2.3 Collaboration and versioning**

The project management website was originally based on the open source GForge software and was established in 2008-09, and provided the nucleus for source code management and versioning. The repository for MULTIFAN-CL source code development was held on the website and uses the open source software SVN (<http://tortoisesvn.net/>). During 2015-16, the repository was migrated to a GitHub site as part of a software merge sub-project (see section 2.4) and all code developments are consecutively committed to the repository while tracing the different versions chronologically. The repository and overall development are coordinated via the GitHub website on GitHub.com at <https://github.com/spc-ofp-sam/mfcl> which is administered by Fabrice Bouye [fabriceb@spc.int](mailto:fabriceb@spc.int).

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 (2014-15) have been addressed in the current period in the way of model developments. A main trunk exists for the MULTIFAN-CL source code, and a development branch has been created to hold recent developments to the source currently being developed and tested. A formal testing procedure has been designed before source code is committed from the development branch to the trunk, and a manual for the testing of new compilations, standardizing the source code compilation procedure, and posting of executables has been drafted.

A version of the source code for ADMB (<http://admb-project.org/>) has been added to the project management website in a separate repository. Minor modifications were required to the ADMB source (currently held in a development branch) to facilitate the recent MULTIFAN-CL developments.

### **2.4 Merging the development branch to the trunk**

A specific sub-project was developed during 2014-15 that manages the merging of the development version of MULTIFAN-CL source code to the repository trunk, and subsequently transforms this into the release version for posting on the website for public access. As most of the new features developed in the past 3-4 years were available only in the development version, it was essential and urgent to restore this development branch into the trunk. This task was achieved in February 2016 with the posting of version 2.0.0.1 (see section 2.7), with the following steps complete:

- An Ubuntu Virtual Machine (VM) has been established and dedicated only to the MULTIFAN-CL project for compilations and testing. This integrates the Linux and Windows compilations, and local working copies of the SVN repository, onto a single platform.
- A series of 6 benchmark tests have been done, ensuring the integrity of the development version after adding numerous new features in the past 3-4 years, and with tests among the Linux and Windows compilations.
- A Windows compilation has been developed. Advancements made on the development version are compiled in Linux (gcc). For posting a release version, Windows executables are compiled using the MinGW-64 cross-compiler (g++-mingw-w64-x86-64), in 64-bit. This offers a streamlined approach for compiling directly from the Linux-based development version. A static build was achieved to improve portability (by reducing the number of necessary \*.dll files) by using -static-libstdc++ and -static-libgcc that removed the dependencies to libstdc++-6.dll and libgcc\_s\_sjlj-1.dll in the produced executable (at the expense of the executable getting slightly larger). But the dependency to pthreadgc2.dll was persistent, and could not be built statically. It was resolved to accept that this will need to be ported with the Windows executable. The optimized build options were reviewed and an executable of size similar to that of the Linux executable was possible. In addition, the MinGW compilation directories were structured to be identical that of the Linux compilation structure, and so the MinGW compilations can be integrated seamlessly within the Jenkins compilation facility. This link will be established later in 2016.
- The User's Guide was updated to include descriptions of the new features and guidelines for applying them when using version 2.0.0.1.
- Merging of the development branch to the repository trunk was done within the existing SVN repository. Thereafter, a Github repository was created to which the trunk was successfully migrated (see section 2.5)
- Upon completing the merge, the compilation using the trunk repository was posted on the website as version 2.0.0.1 being a substantial revision of MULTIFAN-CL (see section 2.7).

Completing the merge was considered to be the highest priority for first part of 2016 and will now simplify the administrative processes for future developments.

## 2.5 Compilation framework and Source code repository

### 2.5.1 Compilation framework

Starting in the second quarter of 2015, a continuous integration facility was established that allows for automatic nightly compilations of MULTIFAN-CL. This automation is done using the software called Jenkins (<https://jenkins-ci.org/>): an Open Source continuous integration tool that comes bundled with a web server used for administration. This software is now installed on the Virtual Machine (VM) that is dedicated to MULTIFAN-CL development.

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

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

It is intended, later in 2016, to extend this facility to support automated builds of the Windows release executable, as well as running automated tests 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.

### 2.5.2 32-bit architecture

Both Linux and Windows executables in 32-bit architecture have been previously been shipped with postings of the MULTIFAN-CL release version up to 1.1.5.9. Upon posting version 2.0.0.1 the demand for 32-bit architecture was reconsidered and deemed to be negligible as most users now operate on 64-bit architecture. Therefore, at the time of posting release version 2.0.0.1, a note was added in the README and RELEASE\_NOTES, with a request for users to contact the project team if they require the 32-bit version. None have yet been received, but any requests will be assessed for their integrity and utility to the project before investing the resources to supply 32-bit versions.

### 2.5.3 Source code repository

Since 2008, the MULTIFAN-CL source code has been held in an SVN repository with management via Gforge (see section 2.3). In April 2016, the SVN repository was replaced with a GitHub repository that now hosts the entire MULTIFAN-CL project. To reduce potential complications, the merging of the development branch to the trunk was first done using the existing SVN repository, rather than as part of the migration to the new GitHub repository. Once successfully merged to the trunk, the development branch was deleted from the SVN repository. The SVN repository trunk was then successfully migrated across to GitHub, with full retention of the trunk history.

This migration had implications for the automated Jenkins compilations. Migrating our automated Jenkins compilation framework from SVN to Git was possible without much change of the Jenkins facility as GitHub allows using an SVN compatible URL for cloning/checkout.

The MFCL project is now hosted on GitHub.com at <https://github.com/spc-ofp-sam/mfcl>. This site is only accessible to registered members of the OFP-SAM team.

## 2.6 **Developer's workshops**

Developer's workshops were held at Nanaimo, British Columbia, Canada, 12 – 22 October 2015, and at Nouméa, New Caledonia, 25 March to 4 April 2016. The participants were the primary developer Dr Dave Fournier and Mr Nick Davies, with Dr John Hampton attending the second workshop.

The main areas of development during the first workshop were:

- Extension of the feature for assumed terminal recruitments
- Revised approach for simulation projections (stochastic recruitments)
- Self-scaling multinomial likelihood - effective sample size estimation

The main areas of development during the second workshop were:

- Report of tagging diagnostics, a listing of the age-specific tag releases
- Revised the report of objective function components – tag likelihood
- Corrections to tagging likelihood – new formulation for pooled tag group
- Self-scaling multinomial likelihood – implementation in MULTIFAN-CL

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

## 2.7 **Postings to website**

There have been four postings of the MULTIFAN-CL release version to the website since August 2015.

### 2.7.1 29 February 2016 – version 2.0.0.1

Completion of merging the development branch to the repository trunk entailed adding substantial developments of new features, and major revisions to some existing features, made to the branch since 2011. In this period, regular testing relative to benchmarks of the development version, and versus the release versions have been completed. The main changes relative to the previous version 1.1.5.9 included:

- Dimensions added for multi-species/sexes
- Multi-sex equilibrium yield calculations
- Time-variant selectivities
- Including age-length data in the model fit
- Tagging negative binomial likelihood - revised formulation and allows for increased over-dispersion
- Likelihood components report
- Tagging diagnostics report - temporary\_tag\_report
- Fitting the BH-SRR to calendar year (annual) recruitments
- Fixed terminal recruitment deviates
- Non-integer tag release frequencies allowed
- Size composition normal likelihood with a student-t probability density function
- Zero selectivity-at-age specified for particular young age classes
- Tail compression of size composition data
- Minimum sample size threshold applying to size composition data
- Fitting during the control phases with fixed growth
- Cobb-Douglas biomass-related effect on catchability - substantial revisions

When posting version 2.0.0.1, it was noted that the bigeye tuna model example previously shipped with earlier postings of the release version, appeared to be out of date. For the 2.0.0.1 posting, instead the skipjack model from the 2014 assessments (Rice et al. 2014), was selected as the new example to be shipped with the new posting.

#### 2.7.2 19 April 2016 – version 2.0.0.2

Added the new tagging data negative-binomial likelihood formulation (using  $\tau$ ) for the fit of the pooled tag group when tag recaptures are not grouped. Added to the output report “test\_plot\_output” the likelihood component for ungrouped tag recaptures. Added a new output report “temporary\_tag\_report” containing the predicted age-specific tag releases by tag group, with indices for the species number associated with each tag group for the cases of multi-species or multi-sex dimensions.

#### 2.7.3 28 April 2016 – version 2.0.0.3

A minor revision to add the variance of the estimated Beverton-Holt stock-recruitment relationship to the equilibrium yield section of the “plot.rep” report.

#### 2.7.4 21 June 2016 – version 2.0.1.1

The independent variable “recr” was restructured to be recr(2,nyears). The feature implemented by parest\_flags(400) was extended into the multi-species and multi-sex dimensions. The functionality of age\_flags(95) as supplying a multiplier to M or Z for the initial equilibrium population conditions was transferred to age\_flags(128). The existing functionality of age\_flags(95) to specify the period over which the initial population is averaged is retained.

## 2.8 Independent Peer Review of the 2011 bigeye tuna stock assessment

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

Of the thirteen recommendations, six have been implemented and tested, 3 have been implemented but testing yet remains in progress, and 4 remain yet to be developed (Davies et al. 2015). Those recommendations identified for the testing to be undertaken for 2015-16 are (Table 1):

- Tail compression (recommendation “i”)
- Multinomial distribution for size composition data (recommendation “j”)
- Tag likelihood relative weighting (recommendation “l”)

As the majority of the developments implemented in 2015-16 related to the self-scaling multinomial with random effects for size composition data (SSM-RE, see section 4.1), this has taken priority over testing the implementation of the above three features. Indeed, the recommendation for a multinomial distribution for size composition data has been extended beyond the scope of that initially recommended through the development of the SSM-RE which has proved to be a better approach. Upon completion of the SSM-RE development in 2016, the testing of these recommendations will be resumed.

Those recommendations identified for the implementation and testing to be undertaken for 2016-17 are (Table 2):

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

A description of the proposed work towards these recommendations in 2016-17 is outlined in Section 7.

## 2.9 Tool development

### 2.9.1 R4MFCL

The R scripts for working with MULTIFAN-CL, developed and released on the internet (<http://code.google.com/p/r4mfcl/>), 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 2016 stock assessments that consolidated new features to the utilities package. The repository was updated in 2016 for the current and development versions of the package.

### 2.9.2 Testing framework

The testing framework for MULTIFAN-CL compilations first developed in 2011-12, was applied extensively during 2015-16. This framework ensures the repeatability and traceability of testing by streamlining the process for new source code developments through a system of model testing procedures and directories. The testing criterion is based upon pair-wise comparisons of model run results obtained using an existing MULTIFAN-CL compilation (usually the current release version) versus those from a development version compilation. Tests are undertaken over multiple processor platforms and architectures, with application to multiple input testing data sets, and with various options for the MULTIFAN-CL operation, viz. single or multiple model evaluations, or full doital model fits to convergence. This ensures a thorough integrity-check of model quantities and components of the objective function prior to the distribution of new versions.

Since March 2013 MULTIFAN-CL source code has undergone substantial developments for multi-threading of the tagged population calculations, combined tagging observations for multiple species/sexes, multi-sex equilibrium yield calculation, time-variant selectivities for individual fisheries, and adding a likelihood term for age-length composition data. These developments have been described (Davies et al. 2015) and more recent ones are described in Section 4.

In adding these new features to the development version, regular testing of this versus the release version was undertaken to ensure the integrity of existing operations. Known as "benchmark tests", these are summarized in Annex 10.1. The development version was last tested in June 2016 versus the release version, which defines the development version as the **benchmark** source code. Subsequent development versions were then tested relative to the benchmark to establish their integrity, after which they are defined as the new benchmark development version. The recent developments were validated using 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 single-species data for ALB2012, BET2011, YFT2011, SKJ2011, STM2012, SWO2013, BET2014, YFT2014 and SKJ2014 to conclude that single model evaluations and the fitted solutions were 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. The benchmark tests undertaken in 2014-15 are summarized in Annex 10.1, with the most recent being done in June 2016. These tests included: two for the development version versus the release version 1.1.5.8, and three tests of the development version at successive stages of development. Within architectures and operating systems, the development and release versions produced identical solutions. Differences among architectures and operating systems were found, but are attributable to the test model not being fully converged, and are small enough to be unimportant.

Tests using multi-species data disaggregated among species were 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 (Figure 1). 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.

The following benchmark tests were undertaken during 2015-16:

- Nov.2015 – to test the **devvsn** (multi\_spp\_2015\_09\_15, Oct.2015) versus the existing **benchmark** (multi\_spp\_2014\_11\_07, Feb.2015) to ensure the integrity of existing features. The major changes to the **devvsn** code that may have affected existing features include substantial developments to: scaling and penalties of model parameters (achieves a more stable minimisation), corrections to logistic selectivity calculation using new `bs_selcoff` parameter structure, fitting an annualised BH-SRR, revised tagging likelihood formulation, and a Cobb-Douglas effect on catchability in respect of biomass. All tests proved positive.
- Feb.2016 – to confirm the integrity of the Windows executable of the Benchmark compilation. These Nov. 2015 tests were made in respect of the Linux executable only. A Windows executable was subsequently compiled from the benchmark version, and prior to being released, the integrity of the calculations and minimisation solutions were confirmed relative to the benchmark Linux executable. The new version provided identical calculations among the Windows and Linux compilations, with slight differences in the doital solutions among the compilation platforms being explained by differences in the minimisation “paths” of each compilation. These differences are small in respect of model quantities, (up to 1%)
- Jun.2016 – to test the `devvsn` code (with restructuring of the vector of temporal recruitment deviates, see Section 4.4) in respect of existing features. Different doital fit solutions were obtained among the versions; but were almost negligible (<0.5%), and related largely to the equilibrium quantities.

All the benchmark tests concluded that the development version has integrity for single and multi-species operations where disaggregated data are used, and can therefore be used as the benchmark development version of the MULTIFAN-CL source code.

## 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 2015-16 this level of testing was done for the stochastic recruitments feature and the self-scaling multinomial M-estimator 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.

While action on these tasks is unlikely in 2016, it is proposed that they be considered in the 2017 work plan for the MULTIFAN-CL project.

### 2.9.3 Viewer

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

The following updates were made:

- Beta version enables opening of TUMAS template and result files
- Projection period is shown on charts that support this feature
- Displays time-variant selectivities in respect of time-blocks and seasons (new plot.rep format)

Immediate priority should be given to displaying the two sections relating to the exploitable population to accommodate fisheries with time-variant selectivities as the format for this has changed since this feature was added to MULTIFAN-CL. There are also issues yet to be addressed for the BETA version, e.g. the memory consumption for some charts is much larger than in the “release” version.

### 2.9.4 Condor parallel processing facility

The Condor ([www.condor.wisc.edu](http://www.condor.wisc.edu)) facility has been used routinely for managing multiple MULTIFAN-CL model runs on a grid currently numbering over 100 processors; being windows or linux platforms, and either 32- or 64-bit architecture. This grid enables intensive model runs for: testing MULTIFAN-CL development versions prior to release; undertaking stock assessments that entail multiple model runs (e.g. sensitivity analyses), structural uncertainty analyses, and management strategy evaluation. Additional Linux Virtual Machines were added to the grid to increase the number of model runs possible using the Linux development version executable. The Condor version used by SPC-OFP was recently updated in 2014.

## 2.10 User's guide

A revision the MULTIFAN-CL User's Guide (Kleiber et al. 2016) has been completed that documents the developments in version 2.0.0.1 and this has been posted on the website <http://www.multifan-cl.org/>. Proposed future revisions include incorporating the suggestions arising from the Training workshop (see section 3) and the recent features added to versions 2.0.0.2 to 2.0.1.1.

## 3 TRAINING WORKSHOP

The first training workshop for users of MULTIFAN-CL at the Oceanic Fisheries Programme (Pacific Community) was held 21-24 March 2016, at Nouméa, New Caledonia. The aim was for the stock assessment analyst to be confident in developing a Reference Case model using MULTIFAN-CL (see agenda in Annex 9.3). The workshop was delivered by Nick Davies and Dr Dave Fournier and was well attended with four core analysts and several observers. While scheduled over three days, the workshop required four days, with a detailed consideration of an actual example filling the final section. While good

feedback was received regarding achieving the aim, the workshop had additional utility in highlighting a number of issues relating to flags and parameterisations in MULTIFAN-CL that are to be clarified for the benefit of users. Also several developments were proposed that would be of utility in undertaking assessments:

- An option for generating a new report of the independent variables being estimated (in the X-vector) at each function evaluation
- Assess and improve the run-time performance of the MinGW compilation executable
- Revise the likelihood profile calculation so that this can be applied to dependent variables; such as current biomass

Some other new features that were suggested by participants during the workshop have been already been implemented in the posted versions 2.0.0.2 to 2.0.1.1.

## 4 NEW FEATURES

All new features that have been implemented in the MULTIFAN-CL source code have been added first to the development version. Once these features have been tested for their integrity, with no impacts on existing features, then the development version is merged into the release version of the code. The current release version is 2.0.1.1. The developments described below have been made to the development version during 2015-16 and will be merged to the release version upon the completion of benchmark testing.

### 4.1 Composition data likelihood

#### 4.1.1 Rationale

Ianelli et al. (2012) recommendation “j” was 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 formed a significant component of the developments for 2015-16. While providing the initial impetus for development, the three known weaknesses in the multinomial likelihood must be considered (Francis 2014):

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

Size composition data are pivotal to the tuna stock assessments using MULTIFAN-CL, in particular for estimating growth and recruitments. Improvements that might offset these weaknesses would enable the correct use of these data in the integrated modelling approach, and this became the focus of this development. The lead developer, Dr Dave Fournier, formulated an innovative size composition minimizing function, being the self-scaling multinomial plus random effects (SSM-RE). Indeed, in developing the SSM-RE, 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

Highly innovative work by the primary developer of MULTIFAN-CL, Dr Dave Fournier, aims to improve the existing multinomial likelihood used for fitting the model to size composition data. Development of the SSM-RE addresses the above-mentioned three weaknesses by modifying the multinomial to have three strong attributes: self-scaling properties (estimating effective-sample sizes), maintaining overdispersion, and to estimate autocorrelated random effects. We refer to this as the self-scaling multinomial, a form of M-estimator (Huber 2009), with estimation of random effects (SSM-RE). In addition to addressing the noted deficiencies in the multinomial, the SSM-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.

In statistics, M-estimators are a broad class of estimators, which are obtained as the minima of sums of functions of the data. The definition of M-estimators was motivated by robust statistics, which contributed new types of M-estimators (Huber 2009). The SSM-RE was formulated using an orthogonal polynomial M-estimator for which one can estimate the sample size in multinomial samples, dependent upon the number of the composition's intervals. A "learner" minimisation is done that estimates the polynomial parameters over a representative range of effective sample sizes given an example multinomial sample, and this is done only once. These parameters are then supplied to the SSM-RE (the "analyser"), for solving for the effective sample size for a given set of observed multinomial samples. A caveat for the M-estimator must be noted, being that the SSM-RE is not a likelihood term, and therefore one can not make likelihood comparisons among models. This may limit the capability for assessing model uncertainty upon the goodness of fit to the various data types supplied to the integrated modelling approach, or assessing the relative improvement in fit in respect of adding new parameters for model process estimation. However, this is considered to be a small disadvantage compared to the three attributes described above, and the capability of dealing with zero proportions.

At both of the Developer's Workshops (Oct. 2015, Mar-Apr. 2016), the SSM-RE was the primary task, with development of simulation testing software during the former, and to update the SSM-RE to include the new polynomial formulation of the M-estimator with full implementation into MULTIFAN-CL during the latter.

#### *First developer's workshop*

At the first developer's workshop, three components of the SSM-RE structure were constructed.

- "Learner" component of M-estimator for calculating polynomial parameters. ADMB code was developed for the "learner" minimisation that estimates the polynomial parameters over a representative range of effective sample sizes given simulated multinomial samples. A report is generated of the sample size estimates over the range considered with the % error (and parameter estimates). Over the sample size range  $n = 5$  to 2000, the average absolute % error was around 4%.
- "Analyzer" component that simply estimates the Multinomial sample size by non-linear minimisation given the known proportions. The polynomial parameter estimates from the "learner" are supplied to the self-scaling multinomial M-estimator (the "analyser"), for solving for the effective sample size for a simulated "observed" multinomial samples. The simple "analyser" algorithm simulates multinomial composition data given a fixed sample size, and runs the "analyzer" to estimate the sample sizes given the "known" proportions, over a large number of simulations. Summaries of the sample size estimates with the absolute % bias from the true values are reported. This was coded in both R and ADMB. It was found to be working exceptionally well with effective sample size being estimated within <6% absolute bias.
- Fish model simulator and analyzer. While the above simulation testing of the "analyser" demonstrated its ability to estimate the effective sample size when the true multinomial proportions were "known", it was necessary to test its ability to also estimate these proportions. So a simple fish population model was used to both simulate "pseudo-observations" of composition data (the "simulator"), and a copy of the fish population model was used as an "estimator" using the SSM-RE M-estimator to fit the size compositions, and estimate the predicted proportions and multinomial sample sizes. The basis for the fish model was the simple example age-structured model that is shipped with ADMB, and the SSM M-estimator was added for fitting the estimator to the age-composition "pseudo-observations" while having the polynomial parameters supplied from the "learner". A log-normal relative abundance likelihood was also included in the estimator. Initially this was coded in R but it was too simplistic and onerous, so the simulator and estimator were coded in ADMB. The simulator included options for observation error (sample size, total catch, and relative abundance) and process error (fishing mortality, natural mortality, and recruitment). Preliminary simulations were run during the workshop for a reasonably realistic and typical configuration for the simulator fish population model, and the M-estimator was found to perform as expected in estimating the effective sample size in the absence of

process error, with around 25% positive bias. For simulations including process error, and for known sample sizes up to  $n = 100$ , effective sample size was estimated with around 6% positive bias. For larger known sample sizes, a key result was that smaller effective sample sizes were estimated. This supports the notion that if you had real multinomial sampling of, say,  $n=1000$  and use the multinomial model to fit the data, if there is additional process error so that the model can not produce predicted compositions equal to the true compositions, assuming the observed sample size means you are overfitting the model. So as extra process error is included one should see the estimates of effective sample sizes getting smaller. In other words, the estimator is incapable of describing the additional process error, so this is “assigned” by the estimator as higher observation error, hence the lower effective sample size estimates. The preliminary tests assumed a constant effective sample size, but it is possible to estimate varying sample sizes by adding the “actual” sample size as a covariate to the M-estimator, and this was done for the implementation of the SSM-RE in MULTIFAN-CL.

### *Optimisation of SSM-RE*

The SSM-RE is computationally intensive and therefore an optimised method for the calculations was developed so as to facilitate its implementation in MULTIFAN-CL for application in typical complex fish stock assessment models.

The main observation is that for an autoregressive first-order,  $ar(1)$ , process while the correlation matrix is dense, the inverse of the correlation matrix is banded and the choleski decomposition of the correlation has only one non-zero row off the diagonal. So in principle, one can use the choleski decomposition of the inverse of the correlation matrix, and the banded lower triangular matrix routines. The non-optimised calculations involves solving a matrix equation

$$y = H^{-1}x$$

and calculating the log-determinant of H, where H is an  $n \times n$  matrix, with  $n$  being the number of slots or categories for the multinomial. In general these are both order of  $n^3$  problems, so for  $n=100$  they are on the order of  $10^6$ . In contrast, the optimized approach using an  $ar(1)$  process to get the covariance matrix for the random effects, one need only to carry out these calculations for  $ar$ , and 1 update of a tridiagonal matrix. This reduces the number of calculations from  $n^6$  to about  $10n$ , so for  $n=100$  it reduces the number of calculations by a factor of 1000-10,000. At later stages, when implementing the SSM-RE in MULTIFAN-CL, the optimized approach, when tested using a complex model example (YFT2014), indicated a notable (up to 3-fold) improvement in computational performance making it feasible for complex model applications.

### *Second developer's workshop*

At the second developer's workshop, implementing SSM-RE in the MULTIFAN-CL source code was completed, and preliminary testing was undertaken using the SKJ2014 stock assessment model (Rice et al. 2014).

The first step was to introduce the M-estimator coefficients (orthogonal polynomial parameters). These were estimated separately using the “learner” component coded in ADMB-RE for a range of slot sizes (5 to 250), and with alternatives for the assumed constant (ftmp). To negate porting the coefficients as a separated data file or library with the MULTIFAN-CL executable, they are included in the source code within a routine (learner\_code.cpp) as a data structure, and are therefore compiled as part of the executable. The coefficients are placed in vectors, with each being specific to a given number of slots that is then related to each observed size composition sample. This flexibility allows for tail-compression to be applied, which results in the number of slots varying among samples.

M-estimator coefficients were estimated for alternative values for the constant ftmp of 0.1 and 0.001. The selection of which set of coefficients used in the SSM-RE is specified by `parest_flags(338)`. It was found that the option assuming `ftmp=0.001` resulted in the minimiser being more stable. Therefore the coefficients for this option were used for applying the SSM-RE to the SKJ model example.

The second step was to implement the SSM-RE “analyzer” for both length- and weight-frequency composition data. The observed sample size ( $i^{\text{th}}$  sample) was input as a covariate expressed in the ratio to the average observed sample size, and the exponent of the fraction was an estimated parameter specific to each fishery, `fish_pars(20)`:

$$\left( \frac{S_{i,fi}}{\bar{S}_{fi}} \right)^{\text{fish\_pars}(fi,20)}$$

Improvements were required for the “inner-minimisation”, relating to the preliminary optimization over the random effects parameters. The “inner minimization” does a limited-memory Newton minimization to get to the point where one is close enough to the minimum for the Newton method to converge. Typically one goes from a maximum gradient magnitude of about 1.e-4 at the end of the limited-memory Newton, to 1.e-12 at the end of the Newton method. The reason one wants to get so close to the minimum is that it has a great effect on how smooth the solution appears to the outer function minimizer. If the function is not smooth enough one can get arithmetic errors in the function minimizer calculations.

The selection of the SSM-RE for size composition data and the input of the M-estimator coefficients is conditional on `parest_flags(139, 141)==9`, for weight- and length-frequency data respectively. Numerous new flags have been assigned for controlling SSM-RE parameter estimation (see Annex 10.2).

A routine was added for explicitly implementing the self-scaling multinomial without the estimation of random effects (for the case of length-frequency data only). This option is conditional on `parest_flags(141)==10`, and enabled a consideration of the relative influence of the random effects estimation upon the model fit to size composition data. This was useful in assessing the SSM-RE relative to the current robust-normal method.

During the workshop, the need for applying tail compression was considered as a means to avoid positive bias in the auto-correlation parameter ( $\rho$ ). However, implementing this in the complex fish model in the early stages was problematic. It was noted that this potential would best be investigated firstly using the simple fish model in simulation tests.

Preliminary results are presented for the complex fish model example used, showing the SSM-RE estimated effective sample sizes being substantially lower compared to the observed values (Figure 2). This reflects the results found in the simple fish model simulations described above, in accounting for additional process error in the model. Visual comparisons were made of the model fits to the summarized length-frequencies obtained using the robust-normal likelihood with that of the SSM-RE (Figure 3), showing the quality of the fit to be comparable. Estimated spawning biomass is similar over the model estimation period, with estimates for the recent period obtained for the model fit using the SSM-RE being around 6% higher (Figure 4). Further refinements in applying the SSM-RE to this example are currently being completed.

This development represents a substantial advance in the use of size composition data in population models, and the reporting of this in the scientific literature is currently in progress.

## 5 OTHER ENHANCEMENTS AND BUG FIXES

### 5.1 Extension of feature for assumed terminal recruitments

A set of consecutive terminal recruitments may be fixed at the default value for the temporal deviates being 0 on the log-scale, i.e. total recruitment over all regions is fixed at the average of the `log(recruitments)`. This feature has been extended so that they can be fixed at the arithmetic mean recruitment, conditional upon the setting of `parest_flags(398)` to apply the mean in normal space to the set of terminal recruitments specified by `parest_flags(400)`.

This feature was also extended to include the multi-species and multi-sex dimensions.

## 5.2 Revised approach for simulation projections

The existing approach developed in 2012 entailed using the stochastic absolute recruitments to calculate the deviates from the average  $\log(\text{recruitment})$  over all regions. For the estimation period the recruitments are:

$$N_{1,t,r} = \bar{R} + \alpha_r + \tau_t + \gamma_{t,r} \quad \text{Eq.1}$$

where:

$\bar{R}$  is average annual recruitment,

$\alpha_r$  is the proportion of annual recruitment assigned to region  $r$ , constrained such that  $\sum_r e^{\alpha_r} = 1$

$\tau_t$  is the annual deviate of spatially aggregated recruitments from  $\bar{R}$ , constrained such that  $\sum_t \tau_t = 0$

$\gamma_{t,r}$  is the deviate on regional recruitment proportions with the constraint that  $\prod_t \gamma_{t,r} = 1$  and  $\sum_r e^{\alpha_r \gamma_{t,r}} = 1$

The total deviate around the mean recruitment can be denoted as  $\delta_{t,r}$  being the sum of the deviates and regional proportions in Eq.1.

$$\delta_{t,r} = \alpha_r + \tau_t + \gamma_{t,r} \quad \text{Eq.2}$$

For the stochastic simulations, random absolute recruitments  $\hat{N}_{1,t,r}$  are resampled from the time series of historical estimates of  $N_{1,t,r}$ . One can derive the stochastic deviates from this set of random recruitments:

$$\hat{\delta}_{t,r} = \hat{N}_{1,t,r} - \bar{R} \quad \text{Eq.3}$$

Using the average recruitment as predicted from the BH-SRR in any future time period  $t$ , the deviates may be applied to this prediction so as to generate a distribution about the predictions:

$$\hat{N}_{1,t,r}^{SRR} = \hat{R}^{SRR} + \delta_{t,r} \quad \text{Eq.4}$$

Previous testing had shown the BH-SRR predictions of recruitment from a deterministic projection were very similar to the average of stochastic recruitments from simulations. However during the workshop, two problems were identified with this approach:

- In the case of the annualised BH-SRR, if the assumed lag period between spawning and recruitment is not a multiple of the number of recruitments in a calendar year, the absolute annual recruitments in the stochastic samples are not consistent with those used in fitting the BH-SRR. Therefore the deviates will not be the same.
- While  $\bar{R}$  and  $\hat{R}^{SRR}$  have been found to be generally consistent over the estimation period used for fitting the BH-SRR (for most models developed at OFP), this will not always be the case, particularly for populations having low values of steepness or estimated biomasses  $< 0.2B_0$ .

These problems necessitated a complete revision of how stochastic recruitments should be formulated in simulation projections by using Eq.4 while  $\delta_{t,r}$  is not derived from Eq.3, but is rather output directly from the fit of the BH-SRR, and subsequently randomly sampled with replacement.

Before developing source new code, the existing approach was rigorously tested during the workshop to ensure it must be improved upon. The stochastic recruitments and the calculated recruitment predictions were compared between the existing and revised approaches. The ALB2015 model data were used as this entails the annualised BH-SRR. Following lengthy testing, while similar on average, it became clear the two methods for obtaining the deviates were not comparable (mostly because of the first bullet point above), and the existing approach would be error-prone in some exceptional, yet possible, situations.

An additional development was made for the special case of using the annualised BH-SRR in projections. The existing approach used an approximation that assigned the total predicted recruitment

uniformly among seasons within a calendar year. This simplification ignores seasonality in recruitments, which was highly evident in the ALB2015 model, for example. An improvement was made that calculates the average seasonal distribution based upon the estimated recruitments, which is used to allocate stochastic predictions of the annualised BH-SRR in projection years; thus maintaining the seasonal recruitment pattern in the future.

In routine `get_numbers_at_age()` (`lesmatrix.cpp`) a new routine was added: `get_average_recruitment_by_season()` that returns a matrix, `avg_recruitment_by_season`, being the average distribution of estimated recruitment among seasons and regions. For both deterministic and stochastic projections, this assigned the total annualised BH-SRR predictions among regions and seasons in each calendar year of the projections. This option was made conditional upon `age_flags(183)`, which is convenient because the option for an annualised BH-SRR fit is conditional upon `age_flags(182)`.

Substantial code development was needed for revising the stochastic recruitment calculations used in simulation projections so as to use the BH-SRR deviates directly. Initially it was necessary to make available the random sequencing to the `get_numbers_at_age()` routine. From the Hessian option `parest_flags(145)=8` command, the routine `make_covariance_report_for_projections()` (`gradcforproj.cpp`) generates the stochastic sequence, so a file “simyears” was reported containing the random fish periods, (or calendar years in the case of `age_flags(182)`), as is used for resampling the total absolute recruitments. This is subsequently read in by the routine `minimizing_routine()` (`newl3.cpp`) as class member of `dvar_len_fish_stock_history` so as to be available in routine `get_numbers_at_age()`.

For the case of the annualised BH-SRR, the calendar year and season associated with each fish period was indexed, so that when resampling the BH-SRR deviates (`bh_recr_devs`) using the “simyears” pointer, these can be assigned to the corresponding fish period. For deterministic projections using recruitments predicted by the annualised BH-SRR, the predictions are divided by the number of recruitments per calendar year, i.e.  $r/\text{age\_flags}(57)$ ; and is applied in the first and subsequent projection periods. The ALB2015 example, having the annualised BH-SRR, was used to test the new approach, and it was found to be working in proof-of-concept. This algorithm has since been tested for the standard case, i.e. a BH-SRR fitted to recruitments from each fish period (not annualised), and exhibits the proper operation.

This development is currently only held in the development version, and testing will be completed in the latter part of 2016.

### 5.3 Streamline the generation of stochastic simulation inputs

Where stochastic simulations are undertaken without stochasticity in the population numbers-at-age in the first projection year, certain calculations of the Hessian matrix were excluded. This increases the efficiency of generating inputs for simulations that only include stochasticity in recruitments, for example.

However, the algorithm requires that a \*.hes file is present in the working directory and derived for the model solution being used for the stochastic simulations. This requirement may be avoided which further simplifies preparations for stochastic simulations. Coding to achieve this is proposed for the latter part of 2016.

### 5.4 Restructuring of the temporal recruitment deviates

The vector of temporal recruitment deviates was restructured. The temporal deviate in the first model year (`year1`) was found to be de-coupled from the numbers-at-age of 1 year old fish in that year ( $N_{\text{year1},1}$ ). This may be expected given that the initial population numbers-at-age is derived independently of the temporal deviate (being a separate independent variable). This de-coupling was revealed in a very large temporal deviate estimated in `year1` (e.g. 6.03) relative to the rest of the deviates (e.g. -1.30 to 0.48), which is different to the relative index absolute recruitment taken from  $N_{\text{year1},1}$  of say -0.46.

The vector of temporal recruitment deviates was therefore restructured to be 2:nyears with the  $N_{\text{year1},1}$  being determined (as was always the case) by the initial population conditions assumptions. Excluding this element slightly alters the initial population numbers and “recmean”, and initially creates

a "Total catch main" likelihood value that is high relative to a solution obtained using the previous version. However, within 20 evaluations the original solution is restored almost exactly.

The .par version number was incremented to 1051 to reflect the change to the vector's structure, with backward compatibility to earlier .par versions ensured.

New code was tested on examples for quarterly and annual recruitments, single-, multi-species and multi-sex populations. Compiled, benchmark test (using YFT2014), and released as version 2.0.1.1.

## 5.5 Output of tag releases at age

While the tag release samples are input in respect of length class, they are predicted by MULTIFAN-CL in respect of age class given the distribution of length-at-age derived from the estimated growth function. In a new routine `test_tag_report()`, a listing of the variable "initial\_tag\_release\_by\_age" with respect to each tag release group was reported to a new output file "temporary\_tag\_report". This has utility for making comparisons with other population models that are not length-based in respect of tagging data. In producing the report, the species identifier was applied to the release groups (including those duplicated for release groups having combined species or sexes) in the instances of multi-species or multi-sex dimensions.

## 5.6 Consolidated the tagging negative-binomial likelihood

The developments made in 2015 of the tagging negative-binomial likelihood that entails the new formulation in respect of the parameter  $\tau$  (Davies et al. 2015), were consolidated in the case of the ungrouped tag recaptures being fitted for the pooled tag group. A correction was made to the pooled tags calculation, where an unnecessary loop in respect of ages was being done. Also, the report of the ungrouped tag likelihood was added to the "test\_plot\_output" report, making it available for model diagnostics.

## 5.7 Report of variance of the BH-SRR

A consideration of the variance of the fitted Beverton-Holt stock-recruitment relationship (BH-SRR) is sometimes warranted, especially in cases where it is expected to be low, e.g. mammals and sharks. This was added to the existing report of the BH-SRR parameters in the "plot.rep" file as follows:  
"# alpha = " << alpha << " beta = " << beta << " steepness = " << steepness << " variance = " << fsh.bh\_variance

## 5.8 Estimation of age-specific selectivity deviation coefficients (sel\_dev\_coefs)

The existing feature for estimating deviates of the fishery-specific selectivity-at-age function has not been widely used, despite its potential utility for estimating process error in size composition data. The feature was consolidated as part of the development of the SSM-RE approach for fitting composition data (see section 4.1). The SSM-RE estimates the positively correlated sampling errors for the composition data. It is possible this may be confounded with positively correlated deviations from the predicted lengths due to variations from the average fishing mortality specific to particular cohorts. Estimating the `sel_dev_coefs` may affect this confounding, and therefore the source code for the feature was refined and implemented for the fish model example used for the SSM-RE development.

Routines at the interface of the independent variables were updated for estimating `sel_dev_coefs` – controlled by `fish_flags(19)` which if  $> 0$  specifies the number of initial age classes include. The penalty is set with `fish_flags(20)`, and the grouping by `fish_flags(51)`. The code was improved to ensure the `sel_dev_coefs` relate only to fishing incidents for which size observations were available.

## 5.9 Crash save facility and capability for fit-autonomy (self-restart)

In the course of implementing the SSM-RE in a complex fish model example (SKJ2014), minimization instability was encountered, particularly when the autocorrelation parameter approached the value of 1. The facility for dealing with minimization failures (crashes) was improved so as to produce a file called "crash\_save" containing the most recent and best ".par" solution which may then be used for re-starting the minimisation in the instance of a crash. Bash script was also used to monitor the run

directory and to re-submit the model fit for further evaluations using the “crash\_save” file, thus achieving a form of autonomy. This feature also produces an output of the current parameters (.par) at regular intervals in a file called “routine\_save”, which may also be used for manually re-starting the fit. In the minimizing routine a new routine was added, save\_best\_estimate(), that writes the current best estimate to “crash\_save” when the crash-related routine catch() is called. Also, a test routine: test\_adexception\_flag(), was added for capturing a solution when an exception error occurs, dependent upon parest\_flags(333). Other refinements made to improve minimization stability include the revised scaling of independent variables at the interface for: q0, bs\_selcoff, totpop\_coff, nat\_mort\_coff, fish\_pars(1, 2, 14, 15), region\_pars(1), sv(21), and age\_pars(2). The revised scaling is likely to change the minimization path relative to earlier versions, however, negligible differences in the global solution may be expected.

### 5.10 Extension to initial population conditions

For the option for initial equilibrium population conditions calculated as a multiple of the natural fishing mortality (age\_flags(94)=1), the multiplier is specified by age\_flags(95). However, the setting of age\_flags(95) also specifies a period for calculating an average population numbers-at-age. This created conflict for particular models having short calculation periods, e.g. maximum number of year ~20. The functionality of age\_flags(95) as supplying a multiplier to M for the initial equilibrium population conditions was transferred to age\_flags(128). The existing functionality of age\_flags(95) to specify the period over which the initial population is averaged is retained.

### 5.11 Fix within robust-normal likelihood

A minor inefficiency was corrected in the robust-normal likelihood calculation where the denominator term (varQ) was added to the aggregate likelihood. This term should only be added when the variable is, in fact, allocated, and the code was adjusted as such.

### 5.12 Fix within population projections

A correction was made to the population numbers-at-age matrix (\*pN ) for the aggregate age class during projections, where this was being assigned before the aggregate class was calculated. This fixed a failure in the simulation projection numbers-at-age calculation but did not affect the simulation projected numbers-at-age output to the report files, i.e. the problem was not influencing the stochastic projection results. The results were correct because the variable “\*pnun\_fish” is a reference to what is reported, and includes the calculated numbers-at-age for the aggregate age class. The failure occurred because \*pN was in the signature of the “routine get\_bh\_recruitment\_for\_projections()” and held *inf* values for the aggregate age class. The code was adjusted to assign \*pnun\_fish to \*pN in the correct order.

### 5.13 Increased independent variables

For a typical tropical tuna model having spatial disaggregation and complex fishery definitions, and including the estimation of sel\_dev\_coffs, the number of independent variables increased to in excess of 30,000 parameters. Consequently, the maximum dimension of the parameter vector was increased to 60,000 for the development version. An improvement noted here would be a facility that obtains the maximum number of parameters from the “mfcl.cfg” file to make this an object-oriented specification of the executable.

## 6 APPLICATION OF NEW FEATURES

The following new features were employed for the 2016 assessments for WCPO skipjack tuna, SKJ, (McKechnie et al. 2016) and south Pacific blue shark, BSH, (Takeuchi et al. 2016a), and for a comparative study of SKJ models developed using MULTIFAN-CL and SS3 (Methot, 2013) by Takeuchi et al. (2016b).

## **6.1 Terminal recruitments fixed at the arithmetic mean**

Both the SKJ and BSH models assumed a specified number of terminal recruitments were fixed at the arithmetic mean value rather than the mean of the log(recruitments).

## **6.2 Output of tag releases by age-class**

In comparing models for WCPO SKJ developed using MULTIFAN-CL with those developed using SS3 (Methot 2013), Takeuchi et al. (2016b) made comparisons of the predicted tag releases specific to each age class.

## **6.3 Report of BH-SRR variance**

During the development of the BSH assessment Takeuchi et al. (2016a) examined estimates of the BH-SRR variance for plausibility, and various levels of assumed penalties applied to the deviates.

## **6.4 Extension to initial population conditions**

For the BSH assessment, Takeuchi et al. (2016a) used the assumption for the initial equilibrium population conditions to be calculated as a multiple of the natural fishing mortality ( $\text{age\_flags}(94)=1$ ), with the multiplier being specified by  $\text{age\_flags}(128)$ . They considered a range of multipliers of up to 4-fold during the development phases of the assessment.

# **7 FUTURE WORK**

A future work plan for the development of new features in MULTIFAN-CL is suggested in Table 2 with those having high priority being undertaken in 2016-17, while others may be addressed in subsequent years. Tasks of immediate priority and relating to the new features recently added (Section 4) are discussed in the following section.

# **8 DISCUSSION**

A major highlight of 2015-16 was the merging of the development version source code to the repository trunk, with the subsequent release of version 2.0.0.1, as posted on the website that made available substantial new features (see section 2.7.1) to MULTIFAN-CL users. This addressed the large dichotomy between the previous release version 1.1.5.9 and the development version of the source code. The development branch had separated from the trunk in 2011 to isolate the substantial and inevitable restructuring needed for implementing multi-sex/multi-species capability. During this period, parity between the development and release versions had been maintained through regular benchmark testing that compares among the versions, highlighting any differences, which were subsequently and immediately addressed. This ensured that when implementing new features, the existing calculations are not compromised. However, maintaining parallel versions is unwieldy and undesirable. Small development branches are intended in future that will facilitate the rapid confirmation of new features. As part of the merge sub-project, the software's administrative tools (compilation framework and the source code repository) have been restructured and improved, which has increased the efficiency of producing new release versions. The completion of this sub-project has been something of a landmark for the MULTIFAN-CL project.

The other major highlight of 2015-16 has been the innovative approach for fitting to composition data as formulated by the primary developer Dr Dave Fournier; called the self-scaling multinomial with random effects, (SSM-RE). Model uncertainty due to assumptions made for the relative importance amongst the various data types included in the integrated modelling approach has dominated most fisheries assessment models, and selecting appropriate "weights" for these data is a "challenging aspect" of fisheries modelling (Punt 2015). Alternatives for the assumed effective sample sizes of length- or weight-frequency data have been presented in key model runs making up the range of models used for formulating advice for most of the recent WCPO tuna assessments. However, these assumptions are based upon prior information regarding the reliability or precision of these data, and may differ among analysts undertaking the assessments. The SSM-RE offers a means for estimating the effective sample sizes based upon a robust statistical approach, and the simulation testing using a simple fish model indicates the method to be relatively unbiased. With optimization of the calculations, the computationally

intensive approach has been made efficient and can be implemented in a complex fish model, making it feasible for use in routine stock assessments. This promises to be a substantial advance in reducing this area of model uncertainty for not only WCPO tuna assessments using MULTIFAN-CL, but also has wider utility for other applications that fit to composition data. Dr Dave Fournier has a draft paper in preparation to submit for peer review and subsequent publication.

While the major focus in 2014-15 was the implementation of new features in respect of the thirteen Peer Review recommendations, this aspect of the project received less attention in 2015-16. Aside from the merge sub-project and developing the SSM-RE, substantive resources were directed to refining the algorithm simulating stochastic recruitments (see section 5.2), being a pivotal source of uncertainty in projections and management strategy evaluations. Therefore, the status of work towards the Peer Review recommendations is largely unchanged since August 2015, with nine of the recommendations having been implemented, three of which yet require testing, and four recommendations remaining to be developed. During 2016-17 it is proposed to address this outstanding work by completing the testing and implementing the remaining recommendations.

Table 2 lists a back-log for testing of seven important new features, some of which will be used in forthcoming assessments or for management strategy evaluations. Other significant features yet to be tested that share this potential include: orthogonal recruitments, and length-based selectivities. High priority should be assigned to addressing this back-log to make these new features available to MULTIFAN-CL users.

A lag continues to persist in the development of the MULTIFAN-CL support tools. For example, although some advances have been made with the Viewer's capability for multi-sex model reports and time-variant selectivities, substantial further development is needed for this to be fully operational in applying this model structure in an assessment context. Other examples are: the region-specific yield report which must be added to the Viewer; and, functions to be added to the R4MFCL package that summarise the various new reports, e.g. the fit to conditional age-length composition data. It is suggested that priority be assigned to these advances in the support tools for MULTIFAN-CL to ensure the new features in release version 2.0.1.1 can be fully and effectively used by analysts.

## 9 REFERENCES

- Davies, N., Fournier, D., Bouyé, F., Hampton, J., and Harley, S. 2015. Developments in the MULTIFAN-CL software 2014-15. WCPFC-SC11-2015/SA-IP-01
- Davies, N., Harley, S., Hampton, J. and McKechnie, S. 2014. Stock assessment of yellowfin tuna in the western and central Pacific Ocean. WCPFC-SC10-2014/SA-WP-04
- Fournier, D.A., Hampton, J., and Sibert, J.R. 1998. MULTIFAN-CL: a length-based, age-structured model for fisheries stock assessment, with application to South Pacific albacore, *Thunnus alalunga*. *Can. J. Fish. Aquat. Sci.* **55**:2105-2116
- Francis, R.I.C.C. 2014. Replacing the multinomial in stock assessment models: a first step. *Fisheries Research* **151**: 70-84
- Huber, P.J. 2009. Robust statistics. 2<sup>nd</sup> edition/Huber, P.J., Ronchetti, E.M., John Wiley & Sons, Inc., Hoboken, New Jersey
- Ianelli, J., Maunder, M., and Punt, A. 2012. Independent review of 2011 WCPO bigeye tuna assessment. WCPFC-SC8-SA-WP-01
- Kleiber, P., Hampton, J., Davies, N., Hoyle, S., and Fournier, D. 2016. MULTIFAN-CL User's Guide. <http://www.multifan-cl.org/>
- McKechnie, S., Hampton, J., Pilling, G. and Davies, N. 2016. Assessment of skipjack tuna in the western and central Pacific Ocean. WCPFC-SC12-2016/SA-WP-04
- Methot, R. User Manual for Stock Synthesis - model version 3.24s. [http://nft.nefsc.noaa.gov/Stock\\_Synthesis\\_3.htm](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>

Rice, J., Harley, S., Davies, N., and Hampton, J 2014. Stock assessment of skipjack tuna in the western and central Pacific Ocean. WCPFC-SC10-2014/SA- WP-05

Takeuchi, Y., Tremblay-Boyer, L., Pilling, G. and Hampton, J. 2016a. Assessment of blue shark in the southwestern Pacific. WCPFC-SC12-2016/SA-WP-08

Takeuchi, Y. et al. 2016b. Comparison of MULTIFAN-CL and Stock Synthesis platforms for the 2014 skipjack assessment. WCPFC-SC12-2016/SA-IP-07

**Table 1.** New features added to MULTIFAN-CL with respect to their state of completion as of August 2016.

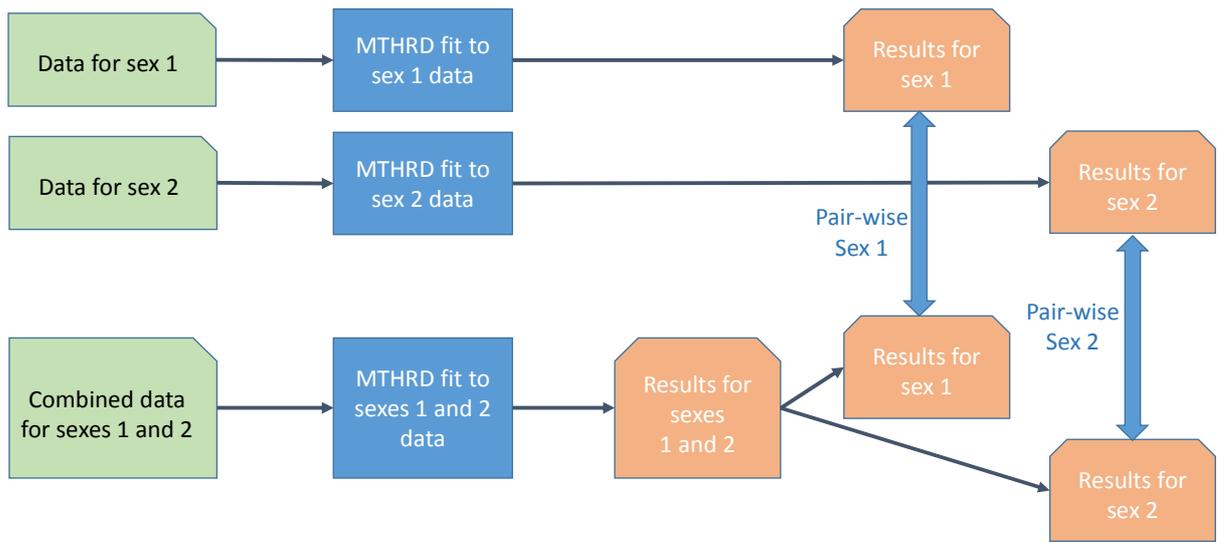
<b>Peer review recommendations</b>		
Task	Description	Status of completion
i. Tail compression	Include a “tail compression” option, which would pool all length- and weight-data for large and small sizes based on a specified percentage.	Development 100%; Testing 50%
j. Multinomial distribution for size composition data	Add an option which allows the analyst to assume a multinomial likelihood for the compositional data in the first phases	Development 100%; Testing 60%
l. Tag likelihood relative weighting	An option to add a likelihood weight to the tagging data component should be added	Development 100%; Testing 50%
<b>Other developments</b>		
Task	Description	Status of completion
Initial biomass approximates the equilibrium unexploited biomass	Abundance in the first model period is equivalent to that estimated using the estimated BH-SRR in an unexploited equilibrium state	Development 90%; Testing 20%
Region-specific yield estimation	Calculate the equilibrium productivity and biomass from each region in a spatially disaggregated model	Development 100%; Testing 20%
Self-scaling multinomial with random effects	Add a multinomial likelihood for composition data that is modified to be self-scaling and estimates auto-correlated random effects.	Development 100%; Testing 90%
Exclude reporting rates from tag predictions during mixing period	Do not to account for the estimated tag reporting rate when solving the Newton-Raphson catch algorithm for tags recaptured during the mixing period	Development 100%; Testing 20%

**Table 2.** New features to be added to MULTIFAN-CL in 2016-17 and those for which testing is to be completed.

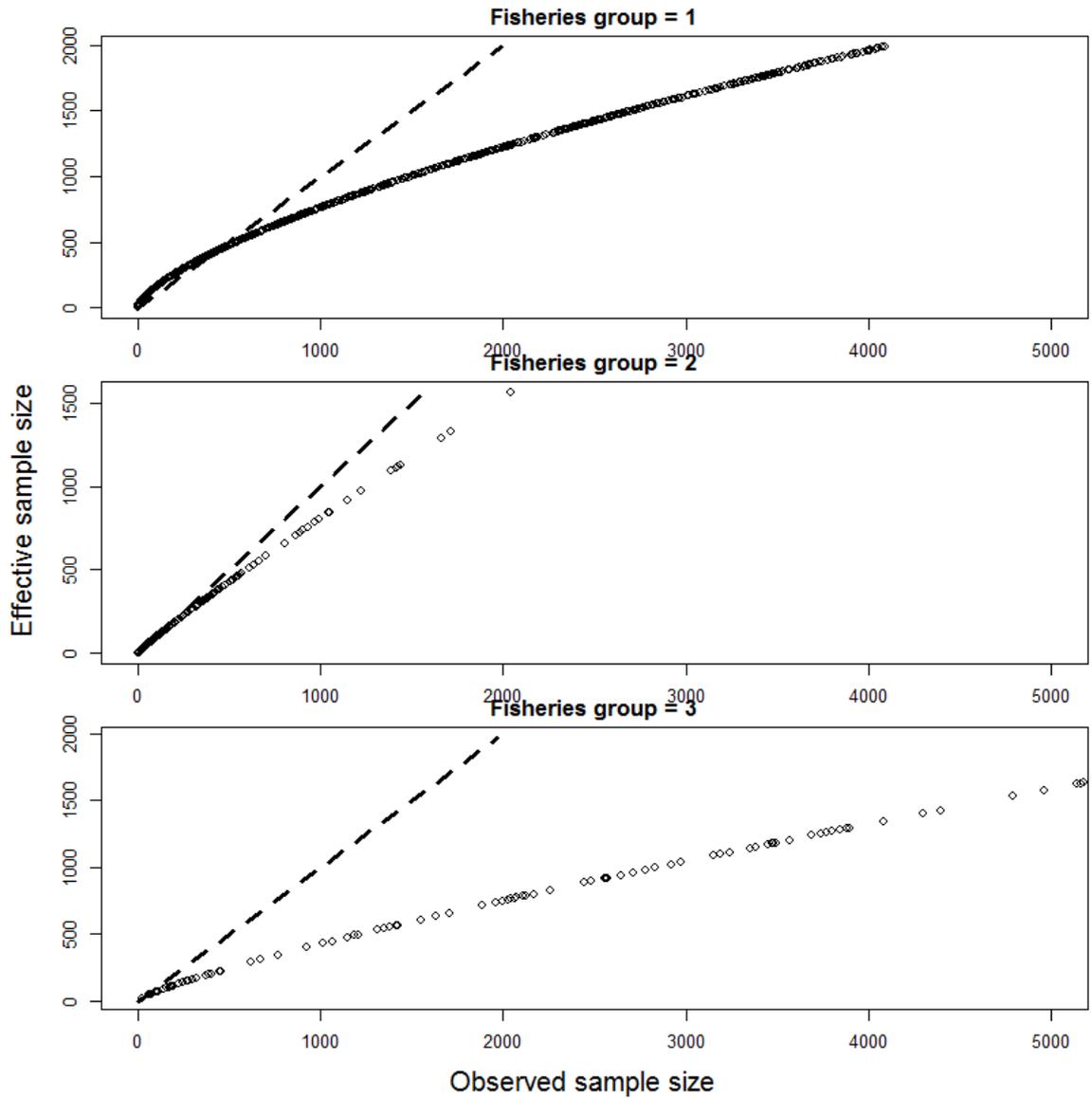
<b>Peer review recommendations</b>		
Task	Description	Implementation
b. Non-uniform size bins	Allow the length bins to be of different widths. One might, for example, want many narrow length bins for the smaller lengths, but fewer but wider length bins for the larger lengths.	2016-17
c. Long-term tag 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.	2016-17
d. Tags inform movement	Include an option which allows the tagging data to inform movement only rather than movement and mortality	2016-17
i. Tail compression	Include a “tail compression” option, which would pool all	Complete testing 2016-17

	length- and weight-data for large and small sizes based on a specified percentage.	
j. Multinomial distribution for size composition data	Add an option which allows the analyst to assume a multinomial likelihood for the compositional data in the first phases	Complete testing 2016-17
k. Maturity at length	When maturity data are based on length, converting to ages should be done within the model.	2016-17
l. Tag likelihood relative weighting	An option to add a likelihood weight to the tagging data component should be added	Complete testing 2016-17
<b>Other developments</b>		
Task	Description	Implementation
Cobb-Douglas scalar	A biomass-related effect on catchability implying hyper-stability in catch rates.	Complete testing 2016-17
Self-scaling multinomial with random effects	Add a multinomial likelihood for composition data that is modified to be self-scaling and estimates auto-correlated random effects.	2016-17
Initial biomass approximates the equilibrium unexploited biomass	Calculate the equilibrium productivity and biomass from each region in a spatially disaggregated model	Complete testing 2016-17
Exclude reporting rates from tag predictions during mixing period	Do not account for the estimated tag reporting rate when solving the Newton-Raphson catch algorithm for tags recaptured during the mixing period	Complete testing 2016-17
Region-specific yield estimation		Complete testing 2016-17
Tags inform growth	Add length increments of tag recaptures to inform growth estimation.	2016-17

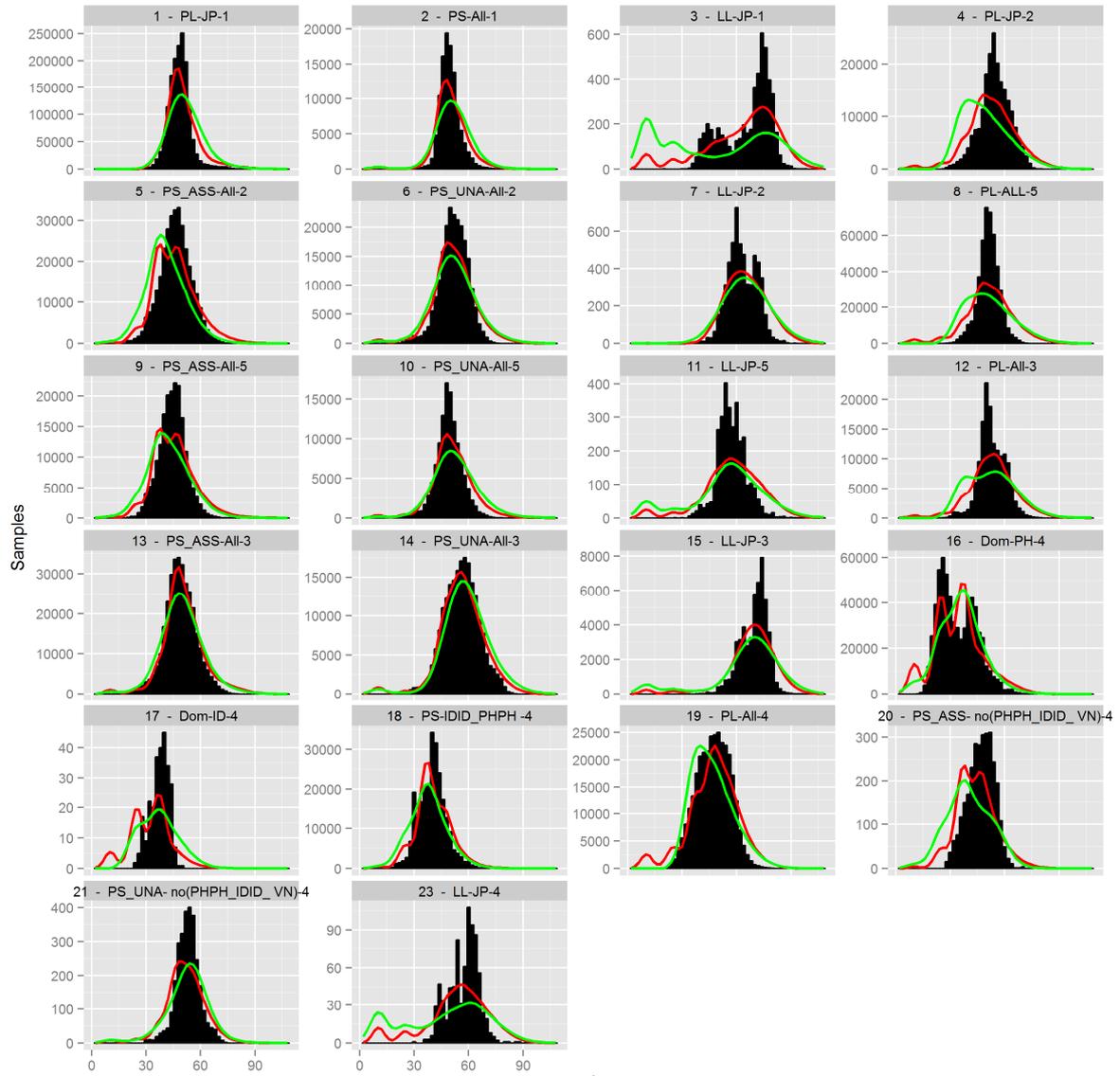
### Flow diagram – multi-sex model testing



**Figure 1.** Flow diagram for testing the integrity of sex-disaggregated and sex-aggregated models.



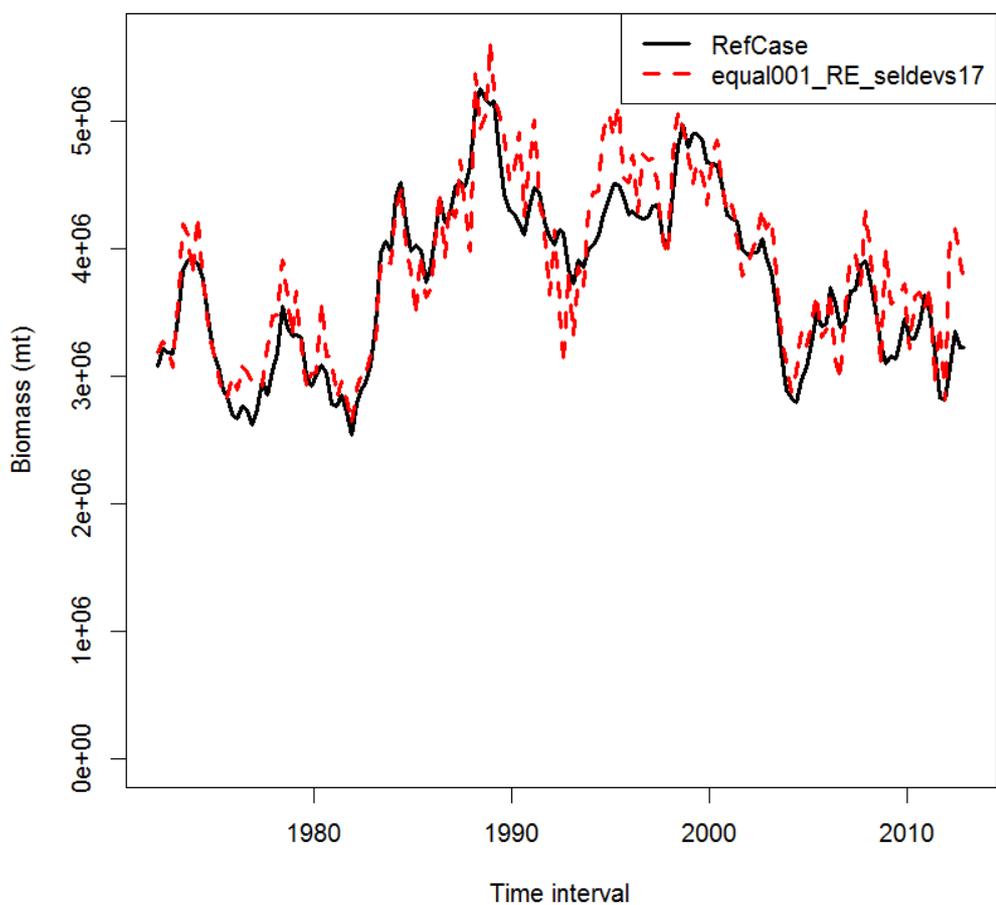
**Figure 2.** The effective sample size of length-frequency samples estimated by the SSM-RE fitted to skipjack tuna observations, relative to the observed sample size for the three fisheries groups assumed for the SSM-RE parameter estimation. The dashed black line indicates a 1:1 relationship.



2

**Figure 3.** Comparison of the fit to fishery-specific observations for length-frequency for skipjack tuna (histograms, over all years) among models: that using the robust-normal likelihood (green line), and that using the SSM-RE M-estimator (red line).

Comparison Adult biomass



**Figure 4.** Comparison of the skipjack tuna spawning biomass estimated from a model using the robust-normal likelihood in fitting to observed size composition data (solid black line, RefCase) with that of a model using the SSM-RE M-estimator for fitting these data (dashed red line, equal001\_RE\_seldevs17).

# 10 ANNEX

## 10.1 Benchmark testing

Table 9.1.1. Summary of benchmark tests of the development version using single species data undertaken August 2013 to July 2014.

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

Benchmark and Development version testing 16 November - 2 December 2015							
MTHRD_2014_11_07	multi_spp_2014_11_07; exec_txf/mfclo64_2014_11_07	26 Feb.2015	2015-11-25_fit_bet/ bnchmrk_bet2011_fit	25/11/2015	af145 = 1; pen_wt=af145	2015-11-25_fit_bet	pen_wt=1; Obj.=1068553.274013987277; MSY=1.854e+04; B0=1.386e+06
MTHRD_2015_09_15	multi_spp_2015_09_15; exec_txf/mfclo64_2015_09_15	3 Nov.2015	2015-11-25_fit_bet/ devvsn_bet2011_fit	25/11/2015	af145 = 1; pen_wt=af145	2015-11-25_fit_bet	pen_wt=1; Obj.=1.06765392664784e+06; MSY=1.775e+04; B0=1.074e+06
Benchmark and Development version testing 17 February 2016							
MTHRD_2015_09_15	multi_spp_2014_11_07; exec_txf/mfclo64_2014_11_07	26 Feb.2015	2016-02-17_fit_skj2014/ release_skj2014_fit	18/02/2016	af145 = -1; pen_wt=10^af145	2016-02-17_fit_skj2014	pen_wt=0.1; Obj.=170692.859; MSY=4.047e+005; B0=6.585e+006
MTHRD_2015_09_15	multi_spp_2015_09_15; exec_txf/mfclo64_2015_09_15	2 Feb.2016	2016-02-17_fit_skj2014/ devvsn_skj2014_fit	18/02/2016	af145 = -1; pen_wt=10^af145	2016-02-17_fit_skj2014	pen_wt=0.1; Obj.=170247.097; MSY=4.056e+005; B0=6.603e+006
Benchmark and Development version testing 20 June 2016							
MTHRD_2015_09_15	multi_spp_2015_09_15; exec_txf/mfclo64_2015_09_15	2 Feb.2016	2016-06-10_fit_yft2014/ bnchmrk_yft2014_fit	17/06/2016	af145 = -2; pen_wt=10^af145	2016-06-10_fit_yft2014	pen_wt=0.01; Obj.=1195760.515; MSY= 1.469e+05; B0=4.322e+06
MTHRD_2015_09_15	multi_spp_2015_09_15; exec_txf/mfclo64_2015_09_15	21 Jun.2016	2016-06-10_fit_yft2014/ devvsn_yft2014_fit	17/06/2016	af145 = -2; pen_wt=10^af145	2016-06-10_fit_yft2014	pen_wt=0.01; Obj.=1195760.641; MSY= 1.467e+05; B0=4.319e+06

## 10.2 Self-scaling Multinomial with random effects – flag allocations

The following table lists the flags assigned within MULTIFAN-CL for controlling the implementation of the self-scaling multinomial M-estimator with random effects.

The parameters being estimated for the SSM-RE are specified by the `parest_flags` settings as detailed in the following table. Those used include: `log_length_variance`, `log_weight_variance`, `log_length_sample_sizevar`, `log_weight_sample_sizevar`, `length_rho`, `weight_rho`.

Table of flag settings for the SSM-RE size composition likelihoods.

Parameter	fish_pars	Length-frequency flags	Weight-frequency flags	Description
Likelihood selection	-	<code>parest_flags(141)</code>	<code>parest_flags(139)</code>	Set equal to 9 selects SSM-RE; 10 selects SSM but without random-effects
log-length variance log-weight variance	<code>fish_pars(18)</code> <code>fish_pars(19)</code>	<code>fish_flags(i,82)</code>	<code>fish_flags(i,84)</code>	if set to 1 it activates the estimation of this parameter
Bounds for log-length/weight variance	-	<code>parest_flags(316)</code>	<code>parest_flags(316)</code>	Sets the bounds for the estimates of log-length/weight variance
Length-rho Weight-rho correlation parameter	<code>fish_pars(16)</code> <code>fish_pars(17)</code>	<code>fish_flags(i,78)</code>	<code>fish_flags(i,80)</code>	if set to 1 activates the estimation of this parameter
Bounds for length-rho/weight-rho correlation parameter	-	<code>parest_flags(315)</code>	<code>parest_flags(315)</code>	Sets the bounds for the estimates of length-rho/weight-rho correlation parameter
log-length N variance log-weight N variance	<code>fish_pars(14)</code> <code>fish_pars(15)</code>	<code>fish_flags(i,67)</code>	<code>fish_flags(i,76)</code>	if set to 1 activates the estimation of this parameter
LF sample_sz covar WF sample_sz covar	<code>fish_pars(20)</code> <code>fish_pars(21)</code>			Sample size covariate exponent parameter
Bounds for: LF sample_sz covar; WF sample_sz covar	-	<code>parest_flags(317)</code>	<code>parest_flags(317)</code>	Sets the upper bounds for the estimates of the sample size covariate exponent parameter
Bounds for log-length N variance log-weight N variance parameters	-	<code>parest_flags(335)</code>	<code>parest_flags(335)</code>	Sets the upper bounds for the estimates of log-length/weight N variance parameters
Fishery grouping: log-length/weight variance; rho; and log-length/weight N variance	-	<code>fish_flags(i,68)</code>	<code>fish_flags(i,77)</code>	Fishery grouping for the log-length/weight variance; rho; and log-length/weight sample variance; grouping is shared for all parameters. parameters.
Minimum size in tail-compression penalty	-	<code>parest_flags(320)</code>	<code>parest_flags(330)</code>	Activates and defines the minimum size specified for the tail-compression penalty, e.g. 5
Tail-compression penalty term – second term	-	<code>parest_flags(321)</code> or <code>parest_flags(322)</code>	<code>parest_flags(331)</code> or <code>parest_flags(332)</code>	Defines the method for the tail-compression penalty, e.g. <code>parest_flags(321) = 10</code>
Selection of M-		<code>parest_flags(338)</code>	<code>parest_flags(338)</code>	Defines the form of the

estimator coefficients				SSMult M-estimator coefficients to be used; 1 uses the "lllower" form; 2 uses the "lllower_001" form; 3 uses the "equal_001" form.
------------------------	--	--	--	--

## 10.3 Agenda for Training Workshop, March 2016

### MULTIFAN-CL TRAINING WORKSHOP – 21 TO 24 MARCH 2016, OFP

**Aim: For the stock assessment analyst to be confident in developing a Reference Case model using MULTIFAN-CL.**

#### Agenda

##### 10.3.1 Day1

1. Workshop outline
  - Aim
  - Scope of material
  - Participant's requirements
  - Course structure (daily routine)
2. Introduction to MULTIFAN-CL
  - What is MULTIFAN-CL? An age-structured, length-based model, and a brief history.
  - Getting access to binaries and User Guide - website and OFP distribution folder
  - Overview of using MULTIFAN-CL – what it is and how to use it
  - Outputs of a fitted model and how they are presented for stock assessment advice
3. Data and Biological inputs
  - Catch, effort and size data – the *.frq* file
  - Release and recaptures of tagged fish – the *.tag* file
  - Observations of fish age at length – the *.age\_length* file
  - Biological parameters – the *.ini* file
4. Memory requirements
  - Specifying memory required – the *mfcl.cfg* file
5. Initial parameter file generation
  - Creating the initial parameter file – the `-makepar` operation

#### Afternoon practical session

- Build *.frq* and *.tag* from MUFDAGER outputs
- Build an *.ini* file
- Run a `-makepar` operation to obtain an initial *00.par*

##### 10.3.2 Day 2

6. Model description (Chapter 3)
  - Overview of the flexibility of model dimensioning with brief mention of the most relevant flags under each of the following items.
  - Dimensions for: time, space, sex/species, age, length, tagged and untagged populations
  - Parameterisations for: initial population, recruitments, growth, M, Beverton-Holt stock-recruitment relationship, fishing mortality (catchability, effort), selectivity, movement, tagging reporting rates and grouping
  - Technical Appendices A.1 and A.2
7. Flags and Settings
  - Flag construction and input via `-switch`, `-file`, and `PHASE`
  - User Guide sections of utility for using flags – section 4.5, Chapter 3, Appendix B
  - `parest_flags`

- `age_flags`
- `fish_flags`
- `region_flags`

## 8. Model Fitting

- Likelihood formulations available – catch, size composition, tagging, age-length data
- Forms of penalties applied – effort deviates, BH-SRR, recruitments, etc.
- The objective function value
- Gradient and convergence criterion
- Interpreting the screen output during minimisation, sending it to a file
- Report of likelihood components
- Control phases that are done by default during `PHASE 1` (following `00.par`)

## 9. Integrated model fitting

- Discussion session with Dr Dave Fournier on general issues of the statistical properties of these integrated models:
  - how do we develop models with appropriate levels of complexity,
  - issues around data weighting,
  - how do we know if we are over-parameterizing these models and what to do about it?

### Afternoon practical session

- Run a single model evaluation with input of a solution `.par` and apply alternative flags
- Build `PHASE 1` flag settings and run via `-file`, and `PHASE` arguments to obtain `01.par`

## 10.3.3 Day 3

### 10. Doitall – batch execution

- The anatomy of phasing and its syntax
- The purpose of phasing and the problem of local minima
- A typical phasing sequence used in OFP's tuna models
- Running a `doitall` – shell/bash script from cygwin (`win.exe`) and in Linux
- Manipulating `doitall` and input files from **R** – `stepwise-working.r`

### 11. Output reports

- Dynamic and equilibrium model quantities - `plot.rep`
- Observed and predicted size compositions - `length.fit` and `weight.fit`
- Observed and predicted total catches - `catch.rep`
- Other fitted and model quantities - `ests.rep`
- Likelihood components - `test_plot_output`
- Observed and predicted tag recaptures – `temporary_tag_report`
- Control of output report generation for “streamlined” operations
- Kobe analysis – generating time series of status relative to MSY quantities

### 12. Diagnostics

- Viewer – eyeballing catchability and effort deviates, and fits to catch, tagging, size composition data, checking the residual distributions, and the BH-SRR fit
- R4MFCL scripts – fits to tagging data in respect of temporal and spatial dimensions
- Likelihood profiles – example in respect of `totpop`
- Retrospective analysis

### 13. Interpreting stock status

- Estimation of MSY and related quantities
- Fishery impact and comparisons with SBF0-based reference points

- Fishery impact analysis (no- fishing model runs) and how to construct the comparison with the SBF0-based reference points
- The methodology behind attributing fishery impact to various defined fishery groups.
- The stock status history or “Majuro” plot (which has essentially replaced the Kobe plot)

#### 14. Hessian operations

- Variance calculations for a converged RefCase model
- Correlation report for estimated parameters
- Generating input files needed for stochastic projection runs – *simulated\_numbers\_at\_age*

#### 15. Projections and evaluating harvest strategies

- Building projection *.frq* and *.par* files
- Running a deterministic projection
- Running stochastic projections
  - Simple simulation analysis for single harvest strategy
  - Iterative simulations for Harvest Control Rule evaluation

#### Afternoon practical session

- Construct a *doitall* having all the phases required for minimisation
- Submit the *doitall* for a model fit
- Produce diagnostic summaries in the Viewer
- Generate projection model input files
- Run deterministic and stochastic projections