THE ATLANTIC-WIDE RESEARCH PROGRAMME FOR BLUEFIN TUNA
(GBYP Phase 11)

MODELLING AND MSE - M3 & ABTMSE R PACKAGE CODE REVIEW
(ICCAT GBYP 03/2021)

Final report

November 30th, 2021

Dr. Emilius Aalto, Stanford University
Technical Contractor for
The Ocean Foundation

This project is co-funded
by the European Union

Summary

The code review was commenced on July 20th following receipt of the code from Dr. Tom Carruthers, the primary contractor for the M3 & ABTMSE package. This report provides the final review of the complete codebase. The review includes the Technical Specifications Document (TSD), the M3 ADMB model (Component 1), and the R code organizing data and model inputs for model conditioning (Component 2). All reviewed code was checked for mathematical correctness (i.e., all formulae matched the equations specified in the TSD) and programming correctness (i.e., no coding errors), and suggestions were made to improve clarity and reduce the possibility of future errors. No major errors were found in the files; numerous minor comments and suggestions are attached in Appendices 1-6. Because no major changes were necessary, no updates were requested from Dr. Carruthers. The ABTMSE package was analyzed for improvements in computational efficiency, with particular focus on speeding up the MSE process which will be used by third parties to develop and test candidate management proposals (CMPs). The results and suggestions for improvement are given below, and further detailed in Appendix 7.

Objectives

Component 1: The TSD and M3.tpl

Status: Review complete (little changed from initial report)

The TSD is comprehensive, well-organized, and clearly written. Almost all of the model components are described, with a few minor typos and other suggestions given in Appendix 1. The TSD was then used as the basis for evaluating mathematical correctness in M3.tpl, the core ADMB file describing the M3 model. In general, all the formulae in M3.tpl matched with the TSD with a few which were not fully described (Appendix 2). There were numerous areas where additional comments could improve clarity, both in the code and the TSD.

Component 2: *R files to organize data and model inputs for use in OM conditioning*

Status: Review complete (slight change from initial report)

Review of the core M3 conditioning R files began with the three scripts identified by the M3 MSE contractor ('Step 1 Build Base OMI.R', 'Step 2 Build OMs.R' and 'Step 3 Build robustness OMs.R'). Each of these core scripts was reviewed (notes in Appendix 3) as well as all additional scripts called from the 'Data Processing' folder (Appendix 4) and elsewhere (Appendix 6). All math was checked for correctness, and no major errors were found.

Component 3: *The ABTMSE R package for CMP testing*
Status: Review complete (new)

Review of the ABTMSE focused primarily on the core objects ('MSE', 'OM', etc.) and associated functions found in the 'R' directory. Each of these files was reviewed (notes in Appendix 5) as well as additional scripts called from other directories (Appendix 6). All math was checked for correctness, and no major errors were found.

Summary of Computational Efficiency Analysis

Although conditioning of the M3 model is a lengthy process, it is performed only occasionally as new data become available and thus is not the principal target for improving runtimes. As described in the CMP Developer's Guide, the primary way in which CMP developers use the ABTMSE package is by designing novel CMPs, then running the CMPs against the standard set of reference OMs (48) and robustness OMs (44). Each OM is tested via the creation of a new MSE object, which automatically conducts CMP testing against the OM-specific future projections. This process could easily be run 1000s of times, as multiple different stakeholders develop and test multiple different CMPs. Consequently, the most important area of the ABTMSE package to improve is the runtime of the 'new('MSE')' call.

The *lineprof* package (Wickham 2021) was used to examine the MSE_Object.R source code (and supporting code) to determine which lines of code were responsible for the greatest use of processor time. It was determined that, on average, 67% of processor time was spent on just 21 lines of code, with a single line responsible for 15% of total time (Table 1, Appendix 7). After examining each of the 21 lines, it became clear that there were several common elements:

1. Multiplication and division of multiple matrices
    a. The fastest gain in speed here would likely be to reimplement large parts of the core code here (roughly 1060-1411) in C, since this region includes multiple layers of expensive loops. This may be impractical, however. Similarly, translating array math into vector transformations would be difficult, error-prone, and result in unreadable code. There may be some gains possible by collapsing certain loops into higher-dimension array multiplication. Transformations of any matrix which is static in nature and doesn't depend on the previous year could be pre-calculated outside the loop. However, most of the relevant matrices (N, Z, etc.) are dynamic in nature and thus difficult to remove from the loop.
2. Use of *apply(…, sum)* and *apply(…,mean)*
    a. Most of the matrix math lines are not in the '>1% of processor time' category. Rather, the slowest lines are those which call the 'apply' function on a matrix. This provides an easy opportunity for speed improvement, because *apply(…, sum)* and *apply(…, mean)* are much slower than their vectorized equivalents, *colSums* and

colMeans. However, the complexity of dimensions used in the *apply* calls require a complicated mapping to *colSums* using *aperm* to take advantage of the speed gains while still producing the same output. Preliminary analysis (Figure 1, Appendix 7) suggested that, for arrays of size similar to those in MSE_Object, switching to these functions would speed the execution of these lines between 20 and 600%. Overall improvement to MSE speed would reflect the speed gains from multiple *apply* replacements.

3. Use of the *domov* function
    a. *domov* is a convenience function which performs three operations: it creates an array, multiplies two arrays, and calls *apply(…, sum).* It can be sped up by replacing the *apply* call with *colSums* and potentially by eliminating the repeated array creation, instead re-using a pre-allocated array. However, preliminary analysis of the latter change was not promising, and it was abandoned.

The proposed changes to *apply* in #2 and #3 were implemented using edited versions of the MSE_Object code, checked against the output of the original lines to guarantee accuracy. The following functions were added:

```
# A substitute for apply() for sum or mean (only!)
# In order to use colSums/Means correctly, it needs to permute the starting
# array based off the margin vector to match the same output as apply().
# Hard-coding the number of dimensions as an input value would be slightly faster.
applySub <- function(theArray, marginV, useMean=FALSE) {
 dims          <- length(dim(theArray))
 dimV          <- seq(1:dims)
 nonMarginDimV <- dimV[!(dimV %in% marginV)]
 permV         <- c(rev(nonMarginDimV), marginV)
 if (useMean) return(colMeans(aperm(theArray, permV),
                        dims=(dims-length(marginV)), na.rm=T))
 else return(colSums(aperm(theArray, permV),
              dims=(dims-length(marginV)), na.rm=T))
}

# A substitute for domov() which uses applySub() instead
domovSub <- function(Ntemp,movtemp) {
 nareas <- dim(movtemp)[5]
 applySub(array(Ntemp, c(dim(Ntemp),nareas))*movtemp, c(1,2,3,5))
}
```

Replacing the instances of *apply*(…, *sum*) and *apply*(…, *mean*) in the key 21 lines of MSE_Object with *applySub* (13 lines) and the instances of *domov* with *domovSub* (3 lines) resulted in a consistent speed increase of approximately 15% (Table 2, Appendix 7). Replacing all additional

instances of *apply* (for *sum* and *mean* only; 81 more lines) and *domov* (2 more lines) within MSE_Object gained an additional 4%, highlighting the importance of the key 21 lines. Making this substitution more widely throughout the codebase could also benefit other aspects such as conditioning time.

Conclusion

The M3 model and ABTMSE code base were found to be correctly implemented at every level, with generally accurate (if occasionally insufficient) description in the TSD. A few minor errors were found and described, including typos in the TSD. Many minor improvements to the code were suggested, mainly for readability and maintainability. Although major gains in speed would require reimplementation of core code in a faster language such as C, widespread replacement of the *apply* function with a faster alternative promises to substantially improve runtime. Nothing was found in the review to suggest any reservations for the use of this package in ICCAT management.

Acknowledgements

This work has been carried out under the ICCAT Atlantic-Wide Research Programme for Bluefin Tuna (GBYP), which is funded by the European Union, several ICCAT CPCs, the ICCAT Secretariat, and other entities (see https://www.iccat.int/gbyp/en/overview.asp). The content of this paper does not necessarily reflect ICCAT's point of view or that of any of the other sponsors, who carry no responsibility. In addition, it does not indicate the Commission's future policy in this area.

Hadley Wickham (2021). *lineprof*: An alternative display for line profiling information. R package version 0.1.9001.

Comment added since the initial progress report are in *italics* (only two, one p22 and one p38)

Appendix 1: Notes on the Technical Specifications Document

p4: Figure 1.2 caption doesn't include description of sub-graph C
-no description of (B) and (C) as alternatives?
p5: could clarify that *any* days in the GOM or Med allow assignment, even if the location for the quarter was elsewhere
p9: Table 2.2 doesn't clearly explain why the recent Canadian acoustic survey has zero weight
-typo: loose paranthesis in "...asymptotic quarterly distribution w_eight)"
-actually, that entire term is confusing
-D_psa is not defined in equation 2.2. Presumably the quarterly distribution described above?
-there are VPA/SS estimates from later than 2015 now, right? Update this?
p11-12: presumably highlighted yellow sections of Table 2.5 are waiting to be filled in
p13: Table 2.9: Are the orange cells correct? Two 0 values aren't orange, and one 1 value is.
-3.1: update M3 model # from 6.5 to 7.0?
-Western stock can spawn in the W_ATL but not Eastern stock?
-3.2: in Eq. 2.1-2.3, 'p' is stock and 's' is quarter. Starting with 3.1, 's' is stock and 'm' is quarter. This is confusing, and also doesn't match the code in M3.tpl (which uses 'np'/'ns' for stock/quarter)
Suggestion: A lot to change, though. Maybe best to change 2.1-2.3 and not meddle with the code. Still, it would be ideal to change the TSD formulations to be consistent with the code and parameter files.
-the N_arrow matrix is a little confusing, as its definition comes later
Suggestion: add '(defined in Eq. 3.18)' after N_arrow
-Eq. 3.3: The term F(l,y,m,l,r,f) is confusing, with 'l' replacing both 's' and 'a'
Suggestion: change it to F_L(y,m,l,r,f), similar to F_D and F_A in Eq. 3.4
-the LAK is described as using the Richards and/or von Bert. growth curves, with references and parameters given. However, it would be nice to provide those two formulas here as well.
p14: Comma needed after 'F_D (constrained to mean 1)'
-Eq. 3.4: comma needed in I(y,m,rf)
-unclear if Eqs. 3.6-3.8 are estimated on a fleet-specific basis, to match the values in 3.5
p15: Unclear how V_rs is calculated for the West. 50/50? Estimated by the model?
p16: It states that 'a large negative number' is assigned to the matrix for implausible movements, but Eq. 3.22 shows '1e-10', a very small but non-negative number. Presumably this stems from the log nature of the matrix.
-the text mentions an "additional movement exclusion matrix". It is unclear what this is and how it differs from line 1 of 3.22
Some sub-headers would be useful, to distinguish the different model sections.
Things like: recruitment, mortality, movement, initializaiton...
p17: Eq. 3.23 is not fully explained. It's unclear what R_s is, for example. Mean recruitment?
p18: Typo: 'lengths have *been* observed...'

p21: the '-' after 'annual value of 0.5' should be removed

p22: there are two equations numbered 5.6

p23: Eq. 5.7 - comma missing in E(R,U,s,yj)

-Eq. 5.8 has its label on the next line

p28: this page is mostly blank for no particular reason

p29: Eq. 7.5 - comma missing in E(U,i,yj)

p32: Eqs. 8.9 and 8.10 both seem to be doubled up. Perhaps a mis-rejected deletion?

-Eq. 8.13 might need an 'r' in the ET term? Not clear where it's summing. Also, movement prior to this has generally been from 'k' to 'r', so it's counter-intuitive

-Eq. 8.13 includes year, but that's not included in the M3.dat data

-Eq. 8.14 has a confusing use of 'i' for strata. It also does not match Eq. 3 in Carruthers&Butterworth 2019

p33: Table 8.5 - This doesn't seem to include some of the functions

-Eq. 8.17 is 'lnL_SSBmu', but Table 8.4 has 'lnL_muSSB'

p34: Table 8.6 - It might be nice to explain how those weights were selected.

-2 entries for 'F deviation from master index (prior)'. One is w_FD and the other w_FA but should add 'annual' to latter.

-9.1: It states 'four major uncertainty axes' but five are listed. Western stock mixing was dropped to the robustness tests, yes?

p35: Table 9.1 and text

-For Eastern stock recruitment, level 1&3 both say the recruitment "switches" from 50-87, 88-present, and after

-however, the same steepness value (h=0.98) is given for both scenarios. This is confusing.

-presumably the text should read "B-H with h=0.98 fixed, low R0"?

-note: this is reflected in the code

p38: Should be 'Box 3.1 of that book' for Walters and Martell reference

No comments have been added since the initial progress report.

Appendix 2: Notes on the core M3 code, M3.tpl

Data Section

Lines 111-119:

-use of 'np' and 'ns' for stock/quarter doesn't match equations 3.1+ (which use 's' and 'm')

Suggestion: make a note, but both TSD and code too extensive to change

-comments use different terminology:

Suggestion: add '(here, quarters)' to 'Number of sub-year time steps' for clarity?

Suggestion: add '(referred to as strata)' to '//Number of areas')

-Typo in 119: 'repapture' instead of 'recapture'

Line 126: helpful to clarify that it gives probabilities of different lengths for each age

Lines 141&142: same comment as 140. Meanings of RDts and RDno unclear

Lines 156-192: meaning of the hard-coded 7 is unclear. Specify columns in comments (i.e., "cobs, year, subyear, area, fleet, XX, YY")

-looking in M3.dat, the first 5 columns are obvious but the 0.01 and 1 at the end are not described

-similarly for 160 and the '10'. M3.tpl and M3.dat comments don't quite match

-165&168 match M3.dat better

-170 comment says 'fleet' when it should be 'region'?

-177 mismatched (type not clearly explained, unclear what the 0 second to the end is)

-184 description in M3.dat not accurate (excludes 'y', doesn't indicated two probabilities)

-188 unclear what the final two columns are (one is N, but the other?)

-why are there so few PSAT2 and nTag entries?

-192 comments don't match data in either M3.tpl or M3.dat

Parameter Section

Lines 221-244:

-layouts of SSBprior and Depprior aren't commented

-237 comment in M3.dat doesn't match

-244 there are 18 weights but only 10 commented

Lines 247-258:

-not really clear what Phase1234 are

-unclear what's going on with datacheck (also line 500)

Line 480: clarify difference with CWpred in comments (catch length composition)

Procedure Section

Line 536: typo in comment

Line 556: perhaps have the function definitions follow the order in 502-548?

assignPars(): math checked

Line 565: why are we adding 1 to lnR0(sr) if pp==1?

Line 599: typo in comment

-math checked: Eq. 3.15

calcSurvival(): math checked; Eq.
Line 662: typo in comment
calcMovement(): math checked; Eqs. 3.19-3.22
Line 683-4: mildly confusing, as this makes the assumption that GOM is always region 1 and Med always 'nr'
Line 717: excluded movements appear to receive e^-20, not e^-10 as in the TSD (Eq. 3.22)
calcSensitivities(): math checked
Lines 786-793: Eqs. 3.9-3.11
Lines 804-821: Eqs. 3.5-3.8
-slightly confusing that Lmax is parameter 1 here and parameter 2 for the logistic selectivity
-812&816 both rely on operator left-to-right ordering to square the exponential term; adding parantheses could clarify
calcF(): math checked
Line 865: hard to match up with Eq. 3.4
-sel==s, qE==q is obvious
-Fmod(ll) appears to == FD(m,r). The 'll' term is confusing since it looks like length
-FDYt(rr,ss,yy) appears to be FA(y,m,r), rather than FD
-Eobs(i,6) appears to map to I(y,m,r,f), but I see nothing which converted it to a scaled index (edit: it's converted in Partial Fs.R)
-the comment for Eobs in M3.dat lists it as CPUE as well, not effort
Line 903&907: math checked; Eq.s 3.2&3.3
-not totally clear that sigma(l) is already factored into ALK (so ALK = sigma(l)*P(l|a))
initModel(): math checked
Line 966: so 20 is just an arbitrary 'big enough' number?
Line 994-999: math checked; Eqs. 3.12 & 3.23
Line 999: based off of Eq. 3.12, but indefinite integral not shown in TSD
-seems like it is derived from N_t/N_t_1 = e^-M / (1-e^-M)?
-ok here because this is an equilibrium estimate at the beginning?
Line 1025: clarify in comment why M/2 instead of M (for F event, correct?)
Line 1031: this seems to be the top of Eq. 3.24 (i==1)
Line 1033: why is pen / by 10^6 here and not in 1103 or 1159?
Line 1082: so spawning fraction in GOM vs. W_ATL is purely based on fraction of SSB? This assumes that western stock fish are equally happy to spawn in each region, so spawning location is driven by the movement data. This seems questionable. It's not even clear that Slope Sea spawners, for example, are western stock. Not really a code question, though.
Line 1111: should this end in 'sdur(ss-1)'? It doesn't match 1041
Lines 1136-1139: math checked; Eq. 3.13 w/ no random deviation term
Line 1167: same question as 1111
Lines 1015-1058 and 1143-1185 could be combined into one section, with a 'priorS' variable which is either 'ss-1' or 'ns' depending on the value of 'ss'. In fact, this could be combined with the spawning season as well, since the only difference is the addition of new recruits and the

aging. It's confusing and potentially error-prone to duplicate most of the math three times. It's also very confusing that yy increases from 2->nHY, but we keep using N(1) values. I figured out that you're just over-writing N(1) each time, but make that clear in the comments or it looks wrong.

Eq. 3.24 gives three different scenarios, but the way the code is written makes it hard to match them up.

Lines 1200-1241: math checked; Eqs. 8.1-8.3

Lines 1200-1202: took a bit to untangle this. A comment would help

calcTransitions(): math checked

Same observation about seasonal change split into three similar sections

Line 1328: Eq. 3.13 with deviations added in now

calcRecaptureProb(): this function is commented out and should be removed?

calcObjective(): math checked

Catch

Lines 1526-1529: better to have the math in one line, and vary the SD only

Not clear what 'last phase()' means: not in M3.tpl and not explained in the TSD

CPUE

Lines 1563&1567: could be clearer in the TSD. Mentioned briefly in 7.2

Line 1573: the calculation of qCPUE isn't explained in the TSD

Lines 1586-1587: math checked; Eqs. 8.4-8.6, 8.9

FI Indices

A question: why skip values when loading the ints? Lines 1609 clearly indicate what each entry is for, except skip over 7,8,10&12. Why not include them here for clarity? Same for all the other regions like this. It makes it more confusing when they show up later.

Line 1641: comment is the same as case 3, when should be "vulnerable biomass"

CLobs

Line 1694: not discussed in TSD

Line 1698: math checked; Eqs. 8.11-8.12

Line 1701: case 3 isn't mentioned in the comments or TSD

SOO

Line 1732-1734: these aren't explained at all, and don't appear to be used (instead, macat)

Line 1743-1744: math checked; Eq. 8.14

PSAT

Lines 1773*1777: should use PSATpred(i) instead of movm(etc)

Line 1773: math checked; Eq. 8.13

Eq. 8.13 includes year for ET observations, but not included in PSAT data

Lines 1777*1781: not discussed in TSD

SSB & Depppriors

Lines 1795-1833: out of order (LHw) and not commented or with debug messages

Line 1801: clarify in comment that SSBprior is in tons, so need to converted

Lines 1808&1826: same 'last_phase' comment

Lines 1809&1811: math checked; Eq. 8.17

Lines 1827&1829: no equation given in the TSD; why divide by SSB_EW(pp,2) instead of 1?

Recruitment deviations

Line 1839: what is this doing?

Line 1847: math checked; Table 8.4

Movement parameters

Lines 1861-1868: a1 and a3 store deviations from a2, but a2 itself is different, right? This weighting isn't clearly explained in the TSD. Unclear why a2 values should be penalized for deviating from mean=0.

Spatial priors & Spatial fraction

Lines 1892-1909: is this lnL_natal in the TSD? Eq. 8.17a? Hard to follow

Lines 1917-1955: I think this isn't in the TSD?

Selectivity etc.

Lines 1963-1990: math checked; Table 8.4, Eq. 8.15

Line 2006: math checked; Eq. 8.16

Line 2034: objPSAT2 always =0

Lines 2036&2037: incorrect description in comments

All the following are fine:

simsam()

popNodes()

Report section

Runtime, main, globals sections

Comments added since the initial report are in *italics.*

Appendix 3: Notes on R-Scripts 1-3

Step 1: Building the OM

It would be nicer if all this read in from a file like M3.dat (but for the R script), rather than being hard-coded. Or at least have all the constants declared up at the top so it's easy to edit the ones you want.

-Line 51: M3.dat gives number of length classes nl=15. Does this set it to 16?
 -I see that line 72 returns the nlen value of 15. A little unclear what's happening.
-Line 114: defaulting to younger maturity for both stocks?
-Lines 197-204: Code author stated that this section was deprecated
-Lines 237&238: the comment says 'ages 0-3' but macat gives 1-4, 5-8, etc. Fix comment?
-Line 252: unclear why this value exists and where it comes from. Is it important to have at least one nTag value?
-Lines 273&274: not clear where these values are coming from (not in TSD)
-Line 310: is this deprecated? Remove it?
-Line 317: Isn't this just 0.5/(yblock^0.5)?
-Line 323: MICV isn't stated in table 8.4 or elsewhere in the TSD.
-Line 337-340: consider removing deprecated entries prior to finalization
-Line 345 (and others): PSAT2 appears to be deprecated as well; consider removing (doesn't match Table 8.6).
 -others which don't quite match up or have a clear entry: SSB, SSBinc, Fmod, BSFrac, MICV, SpatPr
 -some of these, the naming doesn't match: SSB==SSBmu, BSFrac==mix, SpatPr==natal?

Step 2: Building the reference grid

-Line 39-41: same confusion about the eastern recruitment "switch" as in the TSD
-Line 87: make clear where Rec_Ref, MatM_Ref, etc. come from in comments (Ref_OM_funcs.R)
-Line 91: I see what's happening here (mapping option 2->option 4 in MatM_Ref) but it should be more clearly commented. This is a bit of a hack which will confuse people referring to the Ref_OM_funcs vs. the TSD
-Line 128: Is this the big conditioning run? Need to comment what's happening
-Report section: checked briefly

Step 3: Building the robustness set

-Line 28: robustness OMs aren't discussed in the TSD. It appears that you are testing four combinations for most:
  Recruitment scenarios 1&2 vs. spawning/mortality A&B, with no comparison for scaling and length comparison
 This should be documented in TSD section 9.2

-Line 29: there are 4 options but only 2 different values.
-Line 30: for this one only we bring in the length comp L&H. Obviously no A&B, but why L&H here but not for TVregime or others?
-Line 32: the TSD lists 12 robustness tests, while only the first 11 are here
-Line 104: implementation checked
-Line 105: implementation checked
-why is the source code for the Obs classes ('GoodObs', 'BadObs', 'Unreported_20', etc.) in an R-script ('Build Observation Models.R') instead of in the package itself (the way the IE models are in 'Implementation Models.R')?
-Line 136: Brazilian code checked
-Lines 137&146: assess the importance of this comments. Unresolved issue?
-Lines 150-161: checked against 'Partial Fs.R'
-Lines 292&338: Tests 7&8 are switched, presumably because 8 follows from 6?
-Line 350: Implementation checked (see line 105, though)
-Line 357: does this need reconditioning or not? Unclear. Comment to clarify.
-Lines 417-419: math checked
-Lines 471-472: Length (44) is hard-coded instead of nRoms
-Lines 494-502: comment and explain these
-Lines 510&514: comment and explain the difference between OM and OMd
-Lines 531-535, 546-547, 566: Length (44) is hard-coded instead of nRoms
-Line 551: explain why the loop begins at 17 (skipping first 4 robustness tests?)
-Lines 597, 608, 619: do these numbers match the introtext? For example, aren't 1-8 WestGW and Qinc?

No comments added since the initial report.

Appendix 4: Notes on R files for Data Processing

-*Area definitions.R*: coordinates for various areas; not checking.
-*Define fleets.R*: reads the fleets in from FleetDefs.csv
 -values match Table 3.1, except LLJPN is referred to as LLJPNold
-*Fishery indices.R*: makes CPUExxx + Ilencat; checked
 -Line 13: sets a minimum CV of 0.25. Is this mentioned in the TSD? I don't see it near Eq. 8.9 or Table 8.5
-*FI indices.R*: makes Iobs; checked
 -Line 12: also uses a minimum CV of 0.25 for conditioning.
-*Historical catches.R*: makes HCobs; checked
 -Lines 5&6: the multiple folders for ICCAT_2021_etc are confusing. Could this be cleaned up to remove deprecated folders?
 -a comment pointing out that this is creating a 4d array instead of the usual table would be helpful
-*Historical catches Brazilian to East.R: checked*
-*iALK.R*: creates age-length conversion table; math checked
 -needs better commenting
 -Line 7: it would be nice to know where TEG comes from; appears to be *Internal.R*
 -Lines 8-14: math checked against Richards eq. in Ailloud (with assumption that A1==0)
 -Line 17: math checked against vB equation
 -Lines 19-20: this appears to assume that if it's not growth 1, it's growth 2 (rather than checking). Works fine here but potentially a source of bugs if another growth form is added
 -Line 25: math checked against Eq. 3.15
 -Line 31: these would be easier to read if you declared the SD on a separate line; math checked
-*Length observations.R*: makes CLobs
 -Line 5: path appears to be incorrect. No CLobs file in ICCAT_2021_3, though there's one in ICCAT_2021_2
 -Lines 24-27: that's a clever way to do that. I may have to steal it!
-*Master index.R*: makes the master catch index, RAI
 -Line 9: Unclear that source for *'ts2017.Rdata*' is *'Copy data to package.R*'.
 -Line 17: math checked against Eq. 2.1
 -Line 21: The arbitrary 20 timesteps of movement to stabilize?
 -Lines 25-30: math checked
 -Lines 51-53: math checked against Eq. 2.2 & 2.3
 -Lines 59-69: adjust for seasonal priors; is this covered in the TSD?
-*Movement definitions.R*: makes nMP values; checked against Eq. 3.20 and description
 -Line 16: 'Tracks' is created by *PSAT.R*. It would be nice if that were commented.
 -Lines 28-50: some commenting here would help. The logic is hard to follow.

-Is this the part where one value is arbitrarily set =1 and the rest are scaled? (end of p16)
-Might be cleaner if you initialized movind/mov1=NULL? It seems like the c(1,1,1,1,1) is just for rbinding, which accepts NULL
-Lines 53-70: this appears to undo everything done by 28-50. Why isn't there an if/else? More commenting please!

-*Movement exclusions.R*: makes nMovExc
-Line 9: -20 doesn't match the 10^-10 in the TSD
-Lines 19&22: aren't these already covered by lines 12&13 in '*Movement definitions.R*'?
-Lines 31-33: more hard-coded regional mapping (see *'Movement definitions.R*' and '*PSAT.R*')
-Line 40: isn't there code to generate observed transitions in '*Movement definitions.R*'? Why are we using a file instead? If there are other sources of valid tag transitions, this needs to be better explained
-Line 61: I don't see any such transitions in the CSV file. Why is this within the 'if' statement? Shouldn't it be with the others?

-*Partial Fs.R*: makes Eobs; math checked
-Eobs appears to be equivalent to I(y,m,r,f) in Eq. 3.4
-Needs more commenting and/or clearer equation in the TSD
-Line 12: This appears to be inverting 'C = B*(1-e^-Z) to calculate Z. Comment would help, and the '-' for '-log' is cryptic.
-Line 19: parantheses would clarify the / and * operations. Where does 0.2 come from?

-*PSAT.R*: creates PSAT values and the PSAT data files
-Lines 41&42: this is hard-coded, and something similar in '*Movement exclusions.R*'
-Would be better to have regional mapping in one place in case ICCAT adds back some of the regions
-Line 53: This took me a minute to figure out. Using 'sum(x>vec)+1' seems simpler and avoids the c(0,vec) line, though perhaps slightly slower.
-Line 103: Spawning region IDs hard-coded here rather than flexible.
-Lines 127-131: This is confusing to read. Cleaner if you set temptrack=NULL before the loop starts, then eliminate the j==2 branch
-Lines 153-155: Unclear where 'Impute' comes from, but *'impSOO.r*' appears to be deprecated (only in the pre-reconditioning folder)

-*SOO.R*: makes nSOO
-Line 39: SOO_old seems to not be being used. Delete?
-Lines 50&51: yet another hard-coded regional mapping. Really should be centralized.
-Line 149: math checked against Carruthers&Butterworth 2019 Eq.2; btw, there appears to be a typo in Eq. 1
-Line 150: where does this '5' value for sd come from? Or this entire adjustment?

-*Spatial priors.R*: makes SpatPr; math checked
-This uses 'data/Other_2021_1/...csv', while Master Index.R uses 'data/Processed/Priors/...csv'. Should these be the same?

New since initial report

Appendix 5: Notes on R files for in 'R' folder of ABTMSE

Folder: 'R'

-*Data_documentation.R*: checked. No actual code here.

-*Data_formatting.R*: checked

-Lines 52&73: ICCATtoGEO functions; these are very similar, and not clear from commenting which to use or if one is deprecated

-Line 101: assign_area() - shouldn't this exit once it assigns? Unspecified behavior if more than one poly matches

-Line 116: assign_quarter() - some commenting would help clarify what's happening here

-*External.R*: checked

-Line 98: seems important to comment what this function is doing

-Lines 100-110: all these hard-coded defaults need comments to explain

-*HCRs.R*: checked

-*Implementation_models.R*: checked

-why are these all implemented individually? Increases the likelihood of errors.

-Suggestion: one 'imp_error' function which all the rest call

-perhaps this is where the Obs source code belongs?

-*Internal.R*: checked

-Lines 32-79: math checked

-Lines 119&133: these two functions are very similar, but expect slightly different versions of iALK. I assume one is deprecated?

-They should be either commented or one deleted.

-Line 182: here's another version

-Lines 221, 228, 235: a little confusingly named, but at least the comments make it clear

-*IRW.R*: checked

-why is this a separate file, and not part of 'Iterative_reweighting.R'?

-Line 32: 1/3 is hard-coded. Perhaps a parameter? Or at least a comment.

-Line 47: reweights for both high and low CVs equally?

-*Iterative_reweighting.R*: checked

-I looked for obvious errors, but the reweighting process isn't described in the TSD

-it would be helpful to have more commenting and/or the relevant equations

-*M3_tools.R*: includes runM3p and read/write for M3 data

-checked briefly

-Line 603: hard-coding of weighting vector names

-Line 901: same here

-*Methods.R*: checked

-*MPs.R*: briefly looked over

-Line 36: Comment should say 'Index 4', not 7

-This is true for a few other 'Index X' comments as well

-*MSE_auto_report.R*: checked
-*MSE_comp_plots.R*: checked
-Lines 125-126: <ERROR> These should be MSEObj1 & MSEObj2 at the end
-*MSE_functions.R*: checked
-Line 27: you could really use clearer function names than 'getperf', given that this is documented and everything.
-I don't know why R programmers never use names like 'getPerformanceMetrics'...
-Ah, I figured out that the median is hidden inside these 'quantile' function calls...
-Line 95: <ERROR?> these don't seem to exactly match the metrics. Where are AvC10, AvgBr, and OFT?
-Lines 141 & 164: MPnamsj is never used. Also, should it be '...*npop' instead of '...*2' to keep flexibility?
-Lines 199&200, 229&230: confusing that these have ',pp=2' and the rest just use the number
-Lines 509, 575: cleaner to use a parameter with default for values like this
-Wouldn't the start year change down the line as the MSE process goes along?
-Line 606: '92' should just be 'nOMs'
-Concerned about this being hard-coded, though. Doesn't this change sometimes?
-*MSE_Object.R*: checked
-Lines 175&176: might be good to turn these into a year/TAC list for future lexibility
-Line 191: <ERROR> Should be 'OM', not 'OMd'
-Lines 260&261: here survival is again
-Line 323: nice if this were a function call as well
-Lines 376-385: the naming structure is nice. Why not put them in alphabetical or column order as well for clarity?
-Line 415: stocks hard-coded to 2 here, instead of npop
-Line 421: good comment, but maybe put it on several lines
-Line 422: checked against Eq. 5.7 / 7.5
-Lines 468-550: code duplication as discussed under Objects.R
-Line 519: checked against Eq. 3.13
-Line 526: math checked
-Lines 651&660: math checked (duplicated)
-Line 691: this (3 years) probably shouldn't be hard-coded
-Line 715: Wow. That is an impressively dense and hard-to-read line. Feels like Perl!
-Lines 883-898: checked
-Lines 902-927: checked all correct CVs used
-Lines 953-968: B0 math checked
-Lines 983-1053: checked F and TAC distribution code
-Lines 1070-1076: year hard-coding here
-Lines 1060-1148: checked assessment code
-Lines 1150-1246: checked
-Lines 1254-1429: checked

-*MSMPs.R*: briefly looked over
-*MSY_calcs.R*: checked
 -Lines 5&6: surv math checked. This same calculation appears all over the place. Wouldn't it be better to use one function to reduce risk of error?
 -Lines 12-23: math checked
 -Lines 87-100: math checked
 -Lines 108-114: checked vs Walters and Martell
 -Lines 111-113: this is duplicated from F1calc() above. why not just use F1calc() as a function which takes Flow and Flower (with defaults)?
 -Lines 115-168: checked vs. Walters and Martell
 -Line 188: inconsistent use of '=' instead of '<-'
 -Line 217: more year hard-coding
 -Lines 221-228: good to comment the assumptions here regarding h=0.98 and h=0.3
 -Lines 259-266: checked
 -Line 359: not sure what this is doing
 -Lines 373-379: it would be clearer if these were the same as 119-122 (or vice versa)
 -Lines 381-384: <ERROR?> This plusgroup adjustment is *not* in F01Calcs
 -Lines 409-411: why isn't this the same as 144-145?
 -Lines 443-446: no plusgroup adjustment here either
 -Lines 463-468: ok, these are the same as 119-122
-*Objects.R*: checked
 -Line 305: would be better to have a constant for total iterations, rather than just =100
 -Lines 573-574: another hard-coded survival calculation
 -Lines 588-589: what are y and m? Confusing.
  -Looks like they're used only to initialize SFAYMR and SPAYMR (624 & 627)
  -After that, overwritten by the 'for' loops at 647 & 649.
 -Lines 653-657: there are a lot of places where movement matrices get initialized over some stretch of time
  -it would be nice to make this a function call and code it once
 -Lines 683-740: seen this code before as well
  -as mentioned, much cleaner if the bulk of the code (e.g., 685-696) wasn't duplicated
  -just have 'prevM' be set to either 'm-1' or 'nsubyears' based on m (and y too, of course)
  -then add a single 'if (m==SPAWNING_SUBYEAR)' for the last bit
  -should also not hard-code 'm==2' as the spawning subyear, since you're keeping the value of nsubyears as a flexible parameter
-*Obs_Err.R*: checked
 -Line 2: should this equation be in the TSD?
 -Lines 16-18: math checked against Eq. 5.5 & 5.6
 -Lines 21-32: is this deprecated?
 -Lines 54-65: math checked against Eq. 5.7
-*OM_functions.R*: checked

-*OM_object.R*: checked
 -Lines 180&181: why make both 'npop' *and* 'np', and same for 'nages' and 'na'?
  -'np' is only used twice after this point, on lines 201 and 641. 'na' only in line 201.
   -And once you've made local copies of all these, why keep getting them from 'out'?
 -Line 268: should this be 'nsim<=nmcmc'? Seems like the message in the 'else' statement is only for nsim > nmcmc
 -'=' instead of "<-" on lines 313-316, 330, 345, 355, 361, 375, 380
 -Lines 407-416: I don't see any difference between these two branches. Should there be?
 -Line 426: I'm not really sure what's happening here
 -Line 444: here the MED is assumed to be the final area, whereas it's hard-coded as 7 elsewhere (as mentioned)
 -Lines 561-566: math checked against Eq. 3.9-3.11
 -Lines 570-579: math checked against Eq. 3.5-3.8
 -Lines 672-673: math checked against Eq. 3.12
 -Line 687: math checked against Eq. 5.1
-*Other_object_classes.R*: checked
 -Lines 93&95: not clear to me why these two have to be set this way
 -Lines 122&123: surv calculated again
 -Line 150: the comment should be "s p y m r f"
 -Lines 160-170: you're just trying to make my head hurt here, aren't you? :-)
 -Line 184: another place for a comment making it clear that you're cycling stemp in place.
 -Lines 190&194: this would be cleaner if you called this once, after setting the 'subyear' index to nsubyears or m-1 depending on m
 -It looks to me like the rest of this file is part of the R implementation of the M3 code, which has been confirmed to produce the same output
 -Consequently, I will end the review here. Should have done that for some of the other files!
-*Performance_metrics.R*: checked
 -checked all against Table 10.1
 -unclear how this file interacts with /inst/Diagnostics.R
-*Plotting.R*: checked
-*Ref_OM_funcs.R*: design document for reference OM permutations
 -MatM_Ref: the fact that reference option 2 is really option 4 (mapped in 'Step 2') is confusing
  -at the very least it should be heavily commented here to prevent someone "fixing" it later
  -Lines 115,122,129,136: the mortality arrays should be put at the top like the maturity arrays
  -Line 140&141: math checked
 -Bmu_Ref: Line 171: there seems to be a deprecated 700/90 level. Delete?
 -Section B: not checking here since it's mostly deprecated
-*Reporting_funcs.R*: checked
-*Rob_OM_funcs.R*: robustness grid permutations, in theory, but appears to be empty
 -Robustness values seem to be only in Step 3 file
-*RunShiny.R*: checked

Extensive changes since initial report

Appendix 6: notes on additional R files

Folder: 'RScripts/Observation_models.R'
-*Build observation models.R*: checked
-*Build observation models 2021.R*: checked
 -A lot hard-coded here which could be parameterized more clearly
 -Better commenting of what is important in the observation models would be helpful
-*Build Annual Catches.R*: checked
 -Lines 7&8: hard-coded W/E regions, which could cause errors if we return to 10 regions

Folder: 'RScripts/Package checks'
-*MSE check v7x.R*: checked
 -needs better commenting

Folder: 'RScripts/Packaging'
-*Copy data to package.R*: checked
 -seems a bit hap-hazard; is this meant to be part of the package? If so, comment more clearly
 -Lines 9-10: hard-coded paths
 -Line 58: hard-coded OM count
 -Lines 119-122: these values should be stored somewhere else that's more clearly marked
 -Line 290: where does Targ come from? Also, explain function more clearly in comments
 -Line 314: ah, I see, this is East and West.

Folders within the package:
Folder: 'inst'
R files
-*Diagnostics.R*: checked
 -much of this doesn't match Table 10.1
 -unclear how it interacts with /R/Performance_metrics.R
 -Line 88-100: <ERROR?> shouldn't these be medians, not means, given what it says in Table 10.1?
 -Line 306: this looks like a one-off function not intended to be included in the package. Remove?
-*IE_models.R*: checked
 -some commenting here would help
-*Methods.R*: checked briefly
-*MPs.R*: not checking
 -unclear how these interact with /R/MPs.R
-*MSE_source.R*: checked
 -needs more commenting to clarify each function

-Lines 145, 152: none of this Thompson fitting is described in the TSD or elsewhere
-Line 396: clarify what these values correspond to
-Lines 406-408: survival again
-Lines 484-519: another implementation of the annual dynamics
-Lines 466-612: math checked
-some of these functions (e.g., tomt, TEG, domov) are also in Internal.R
-Line 1476: 8 region names by default
-*Plotting.R*: checked

Rmd files: reviewed these more briefly, focusing on code for the data (not display code)
-*Free_Comp.Rmd*: checked
-*Index_fit_summary_manyOMs.Rmd*: checked
-*MSEreport.Rmd*: checked
-*OM_Comp.Rmd*: checked
-*OM_Comp_old.Rmd* # ignored
-*OMreport.Rmd*: checked
-Line 420: ? Comment this or remove
-Lines 558-563: this is pretty hard to follow without comments. Clarify that this is the Richards growth curve
-Line 1028-1029: should these values be the same? SOOobs and allobs aren't the same.
-Line 1092-1093: same here
-*OMreport_old* # ignored
-*RecDevs_manyOMs2.Rmd*: checked
-Line 170: this should be commented and explained

Folder: 'shiny_apps/ABTMSE' : checked briefly
-*global.R*
-*server.R*
-*ui.R*
-*Source/Figures.R*
-*Source/Misc.R*
-*Source/Tables.R*
-*Source/Weighted_estimators.R*

Folder: 'vignettes' : checked briefly
-*ABT-MSE.R*
-*ABT-MSE.Rmd*

Appendix 7: Computational efficiency analysis of specific lines of code

Table 1: Proportion of total run time used by specific lines of code as determined by *lineprof*.

Line #	*Run 1*	*Run 2*	*Run 3*	*Mean*
1124	0.152	0.152	0.153	**0.152**
1371	0.046	0.047	0.044	**0.046**
1391	0.043	0.049	0.042	**0.045**
309	0.037	0.038	0.036	**0.037**
629	0.023	0.035	0.039	**0.032**
1220	0.029	0.03	0.037	**0.032**
1284	0.031	0.033	0.03	**0.031**
1378	0.025	0.034	0.032	**0.030**
1262	0.031	0.03	0.03	**0.030**
1254	0.03	0.031	0.029	**0.030**
1392	0.025	0.031	0.029	**0.028**
1270	0.027	0.027	0.025	**0.026**
1261	0.023	0.024	0.023	**0.023**
1258	0.023	0.024	0.023	**0.023**
1380	0.017	0.027	0.026	**0.023**
1256	0.016	0.018	0.017	**0.017**
1286	0.011	0.015	0.015	**0.014**
624	0.012	0.013	0.014	**0.013**
1127	0.011	0.014	0.012	**0.012**
1336	0.013	0.011	0.013	**0.012**
549	0.011	0.012	0.011	**0.011**
Sum	0.636	0.695	0.68	**0.670**

Specific lines with notes:

Line 309:

```
FM[s,p,,y,,,]<-aperm(apply(Ftemp,1:4,sum),c(4,1,3,2)) #[m f r a]  to [a m r f]
```

-loops for each year, sim, and stock

-aperm seems efficient enough

-try using colSums instead of apply(, sum)

Line 549:

```
N[,,,1,m,]<-domov(N[,,,1,m,],mov[,,,mi,m,,])
```

-loops for each year and quarter

-'domov' is the slowdown here

-key line from 'domov': apply(array(Ntemp, c(dim(Ntemp),nareas))*movtemp,c(1,2,3,5),sum)

-allocating a new array each loop seems slow

-try re-using a constant array

-try using colSums instead of apply(, sum)

Lines 624-629:

```
624 N[,,,y,m,]<-domov(N[,,,y,m,],mov[,,,mi,m,,])
      Ftot<-apply(FM[,,,y,m,,],1:4,sum)
      Z[SPAYMR]<-Ftot[SPAR]+M[SPAY]/nsubyears
629 C[SPAYMRF2]<-N[SPAYMR2]*(1-exp(-
Z[SPAYMR2]))*(FM[SPAYMRF2]/Z[SPAYMR2])
```

-loops for each year and quarter
-see line 549 for domov issues
-no clear speed up on 629, as all values dependent on y and m

Line 1124 & 1127:

```
1124 Cobs[,upind]<-apply(
   array(C[,,,upind,,AA,],c(nsim,npop,nages,length(upind),nsubyears,nA,nfleets))*
   array(Wt_age[,,,nyears],c(nsim,npop,nages,length(upind),nsubyears,nA,nfleets)),
   c(1,4),sum,na.rm=T)*.Object@Cerr[,upind]
1127 dset[[AS]]<-list("Cobs"=cbind(Cobs,.Object@TAC[,MP,AS,y-nyears-1]),
            "Iobs"=Iobs[,,1:(y-2)],
            "K"=OM@Kmu[,AS]*.Object@Kb,       # for now these assume same growth by
stock
            "Linf"=OM@Linfmu[,AS]*.Object@Kb,  # for now these assume same growth by
stock
            "t0"=OM@t0[,AS],                # no error in t0
            "M"=OM@M[,AS,,(y-2)]*.Object@Mb,
            "a"=rep(OM@a,nsim),
            "b"=rep(OM@b,nsim),
            "nages"=OM@nages,
            "ageM"=OM@ageM[,AS]*.Object@ageMb,
            "Mat"=OM@mat[,AS,,nyears],
            "Bt_PI"=apply(N[,,,y-1,nsubyears,AA]*
                        array(Wt_age[,,,nyears],c(nsim,npop,nages,nA)),1,sum,na.rm=T),
            "Bty_PI"=apply(N[,,,1:(y-1),nsubyears,AA]*
                        array(Wt_age[,,,nyears],c(nsim,npop,nages,y-
1,nA)),c(1,4),sum,na.rm=T),
            "VBty_PI"=apply(N[,,4:nages,1:(y-1),nsubyears,AA]*
                         array(Wt_age[,,4:nages,nyears],c(nsim,npop,nages-3,y-1,nA)),
                         c(1,4),sum,na.rm=T),
            "MPrec"=TAC[,AS],
            "TAC"=matrix(.Object@TAC[,MP,AS,1:(y-nyears)],ncol=(y-nyears),nrow=nsim),
            "curTAC"=rep(TAC2020[AS],nsim)
          )
```

-loops for each MP, each projection year, each assessment
-only in 'upyrs'

-try using colSums instead of apply(, sum)
-check for things to move out of loop
-faster if seperated?

Line 1220:

N[,,,y,m,]<-domov(N[,,,y,m,],mov[,,,mi,m,,])
-loops for each MP, each projection year, each quarter
-see line 549 for domov issues

Lines 1254-1286:

```
1254 CAdist[SPRFA2]<-N[SPAYMR2]*Wt_age[SPAL]*sel[SPAF2] # predicted vulnerable
biomass
       CAdist[CAdist==0]<-tiny # you have to do this otherwise zero fish lead to missing
catches
1256 CAdistsum<-apply(CAdist,c(1,3,4),sum)            # total in each sim, region and fleet

1258 CAdist[SPRFA2]<-CAdist[SPRFA2]/CAdistsum[SPRFA2[,c(1,3,4)]] # fraction in each
stock and age class per sim region and fleet
       CAdist[is.na(CAdist)]<-0
       C[SPAYMRF2]<-testC2[SRF2]*CAdist[SPRFA2]
       C[SPAYMRF2][is.na(C[SPAYMRF2])]<-0
      <comments removed>
1270 C[SPAYMRF2]<-C[SPAYMRF2]/Wt_age[SPAL] # divide by weight to get numbers
      <comments removed>
1281 Up<-C[,,,y,m,,]/array(N[,,,y,m,],c(nsim,npop,nages,nareas,nfleets)) # harvest rate
       Up[is.na(Up)|Up<tiny]<-tiny # otherwise you can't generate the automatic fishery
data
       Up[Up>0.9]<-0.9  # 90% max U by age
1284 FM[SPAYMRF2]<-(-log(1-Up[SPARF2])) # get F

1286 Ftot<-apply(FM[,,,y,m,,],1:4,sum,na.rm=T)
```

-loops for each MP, each projection year, each quarter

Line 1336:

Nttind<-as.matrix(expand.grid(1:nsim,pp,1:nages,y,MPQ[i],MParea[i],1:nlen))
-loops per MP, year, index, stock

Lines 1371-1392:

```
1371 SSBmu<-apply(SSB,c(1:4,6),mean) #nsim,npop,nages,allyears,nareas
       .Object@SSB[MP,,,]<-apply(SSBmu,c(1:2,4),sum)
      # SSB by assessment area:
      for(aa in 1:2).Object@SSBa[MP,,aa,]<-apply(SSBmu[,,,,MPareas==aa],c(1,4),sum)
```

```
      rm(SSBmu)

1378 Ctemp<-apply(C[,,,1:allyears,,,]*
        array(Wt_age[,,,nyears],dim(C[,,,1:allyears,,,])),c(1,2,4,6),sum) # s p y a
1380 .Object@CN[MP,,,,]<-apply(C[,,,1:allyears,,,],c(1,2,6,4),sum)  # PM s p r y
      .Object@CW[MP,,,]<-apply(Ctemp,1:3,sum) # s p y
      for(aa in 1:2){
        .Object@CWa[MP,,aa,]=apply(Ctemp[,,,MPareas==aa],c(1,3),sum)
        .Object@TACtaken[,MP,aa,1:(proyears+1)]<-.Object@CWa[MP,,aa,nyears+(-
0:proyears)]
      }

      Ctemp2<-apply(Ctemp,c(1,3,4),sum)# SYR
      for(aa in 1:2).Object@C[MP,,aa,]=apply(Ctemp2[,,MPareas==aa],1:2,sum)

1391 .Object@Fleet_comp[,MP,,,]<-apply(C,c(1,7,4,3),sum) # S,F,Y,A from SPAYMRF2
1392 .Object@Fleet_cat[,MP,,]<-apply(C[,,,1:allyears,,,]*
      array(Wt_age[,,,nyears],dim(C[,,,1:allyears,,,])),c(1,7,4),sum)  # SFY
 -loops per MP only
 -mainly big apply(,sum) and apply(,mean) calls
```

Speed increase from use of 'colSums' and 'colMeans' instead of 'apply'

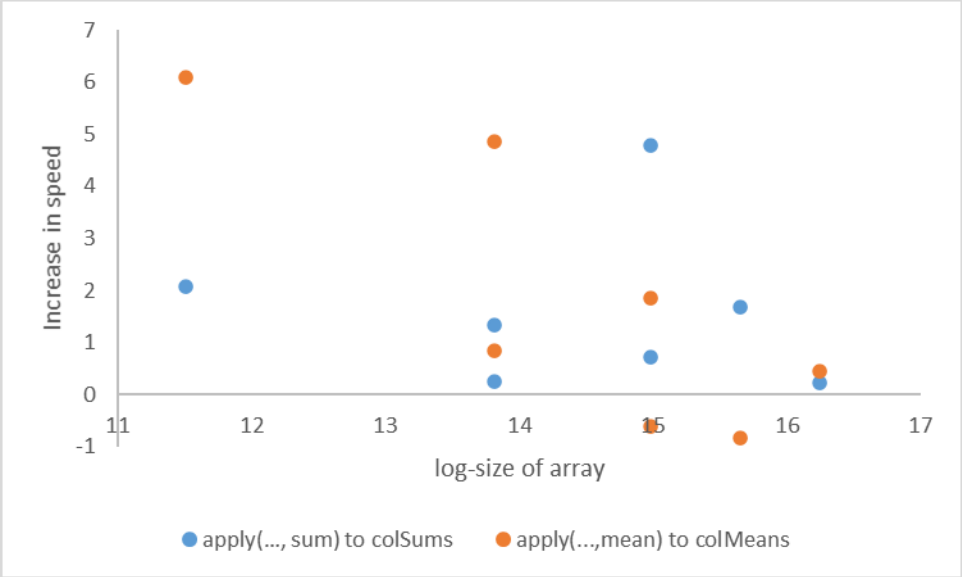

Figure 1. Increase in speed gained by switching from using *apply* to *col** in test code. Note that these times include the use of *aperm* to remap the input array so that *col** produces the same output.

Table 2: Increase in speed of new('MSE') from *apply* replacement in key lines of code.

	Mean time per run (+/- 1 s.d.)	Edited / original ratio
Original code (50 runs)	142.6 s +/- 1.038	-
16 lines edited (50 runs)	121.7 s +/- 0.630	0.854
All *apply* edited (50 runs)	115.8 s +/- 0.470	0.812