CSIRO
MARINE RESEARCH

RESULTS FROM FURTHER TESTING OF CANDIDATE MANAGEMENT PROCEDURES FOR SOUTHERN BLUEFIN TUNA

Tom Polacheck
Dan Ricard
Paige Eveson
Marinelle Basson
Dale Kolody
Jason Hartog

Prepared for the CCSBT SAG
25-29 August 2003, Christchurch, New Zealand

CCSBT-ESC/0309/29

Table of Contents

List of Appendices

Abstract

Results are presented from the second phase of testing a number of candidate management procedures for SBT based on a range of different underlying decision rules. Results from 5 of the decision rules presented from the first phase of testing are presented, plus results from 6 new or substantially modified ones. However, exploration of the full performance of the various rules, particularly with respect to robustness trials, was limited by computational constraints. As was found in the first phase of testing, substantial improvement in average performance could be achieved by adopting a feedback approach. Within any decision rule, a wide range of performance was achievable for the 18 reference case operating model scenarios in terms of trade-off in catch versus stock status (e.g. rebuilding) by varying the tuning parameters. Similar average performance could be achieved from two different decision rules, but with quite different performance within particular scenarios. In addition to results for the 18 reference case scenarios, results are presented for the set of 26 robustness trials defined at the April 2003 Management Procedure Workshop (including the MCMC run) . The results from each robustness trial were compared with the results from the most analogous scenario within the reference cases, and substantial differences were often found in the performance of a candidate management procedure. This indicates that it will be important to consider how to characterize the uncertainty represented by these robustness trials within the final set of operating models.

As was found previously, the wide range of uncertainty about the SBT stock dynamics embedded in the set of operating model scenarios being considered substantially limits what can be achieved from management procedures to both ensure that adequate rebuilding of the stock occurs and that catches are not unnecessarily foregone. Recommendation on a choice of management procedure requires clarification of management objectives in terms of robustness and risk, combined with an agreed process for synthesising the results across the full range of operating model scenarios.

Introduction

The agreed approach for the development and evaluation of a management procedure for the stock of southern bluefun tuna (SBT) by the Commission for the Conservation of Southern Bluefin Tuna (CCSBT) involves the simulation testing of a range of candidate management procedures. The testing process underlying the set of results presented here is very similar to that used during the initial testing, and presented in Polacheck et al (2003a).

Based on the results of the initial testing and further exploration of the operating model (Anon 2003, Polacheck and Kolody 2003a), a revised set of operating model scenarios was defined at the 2nd CCSBT Management Procedure Workshop (April 2003) to use in the next stage of management procedure testing (Anon 2003). This consisted of 18 reference, or base, case scenarios and an additional 26 scenarios for robustness testing. In addition, some technical modifications were made to the operating model and additional performance indices were defined. The software was again distributed to members of the CCSBT Scientific Committee to allow a range of candidate management procedures to be developed and their performance tested with the new set of operating models.

In Polacheck et al. (2003a), we explored a large number of candidate management procedures based on 11 different underlying decision rules. This provided some indication of how different information and feedback mechanisms affect trade-offs in performance. We have continued to explore these decision rules and develop additional rules based on the experience gained in the initial testing. In the current paper, we present results for 11 different decision rules (5 from the original 11 presented in March 2003 and an additional 6 new or substantially revised ones). We have tested the performance of these decision rules across a range of their internal tuning parameters both for the 18 base case scenarios and the set of 26 robustness trials. The details and results of each rule are given in the Appendices to this document.

Methods

Results for eleven different decisions rules are presented. A decision rule is defined here as a basic algorithm that can be used to determine the TAC in the next year given available information (e.g. past catches, CPUE trends, etc.). All of the decision rules considered here have “control” or “tuning” parameters that need to be specified before they can actually be used. These determine the actual TAC given the algorithm and the available information. We define a management procedure to be a fully specified decision rule (i.e. specific values assigned to the tuning parameters). Thus, for any general decision rule there are potentially an infinite number of possible versions or candidate management procedures depending upon the specific values of the tuning parameters. For example, setting the TAC to a constant value constitutes the simplest decision rule that one might consider. In this case, there would be one tuning parameter (i.e. the actual constant value for the TAC) and any specific value for the constant TAC level would constitute an individual candidate management procedure.

A generic feature built into all of the decisions that we developed was a tuning parameter to control the interval over which the TAC is kept constant and a tuning parameter to control the maximum change in the TAC allowed between any two successive years. However, in all the testing that we performed, we have set the interval to be one year and the maximum level of change to be 3000t. The latter figure was agreed upon at the Second Management Procedure Workshop to enable comparability (Anon. 2003). Additionally, the year-to-year changes in TAC were constrained to the agreed minimum of 100t (Anon. 2003). Fixing these constraints provided a common basis for comparisons. However, at a latter stage in the testing process it may be worthwhile to vary these constraints. Also, in all of our initial testing, we have maintained the distribution of the TAC among fisheries constant at their current level. This was done to facilitate comparison of the basic performance of different decision rules. In addition, there was little basis for evaluating performance of a management procedure if the distribution of catches was varied without any guidance from the Commission on what might constitute an appropriate metric for judging performance in this regard.

The 11 decision rules were given the following names:

1. Const (constant catch)
2. CPUE (nominal LL1 CPUE)
3. CPUE_2020 (CPUE forecasted to 2020)
4. Fox (Fox production model)
5. Fox_pe (modified Fox-based rule)

6. Fox_cpue (composite Fox and CPUE-based rule)
7. ACRLRT (Aggregate CPUE with Rebuilding Lag and Historical Rebuilding Target - formerly known as “Stinky”).
8. ASCURE (Age-Structured CPUE with Unabashed Reverse Engineering)
9. Kaltac (Kalman Filter ‘assessment’)
10. VirtualSPN (VPA and spawner per recruit 1)
11. SPNadapt (VPA and spawner per recruit 2)

Rule 1, which involves no feedback, is based on a constant catch scenario. It has been included because we found that it provided useful references for helping to evaluate the performance of rules with an actual feedback component and to understand the limits in performance that could be achieved within the set of initial operating model scenarios. In particular, the constant catch decision rule provided a standard for comparison for evaluating the degree to which the feedback control mechanism incorporated into a particular decision rule improved (or degraded) performance. Rules 2, 3, 7 and 8 are empirically CPUE based rules in that they use changes in CPUE as the primary information for adjusting CPUE and do this outside of any underlying population dynamic model. Rule 8 differs from the other three empirically based rules in that it uses estimates of age specific CPUE indices while the other three use the age aggregated index. Rules 4, 5, 9, 10 and 11 are “model” based in that an underlying population dynamic or assessment model is fitted to the available information and the estimated parameters from the model fitting are used to determine the TAC. Rules 10 and 11 differ in that the underlying model is explicitly age structured (i.e. a VPA) and uses estimates of catch-at-age based on cohort slicing. Rule 6 is a composite rule in that it combines trends in the CPUE with parameter estimates from fitting a population dynamics model to set the TAC. Also note that some rules have explicit targets (often as tuning parameters) which they aim for, whereas others do not. Detail descriptions of each of these rules are contained in the appendices.

Decision rules were tested using the projection software developed and provided by Vivian Haist for this purpose as defined by the Second Management Procedure Workshop (Anon. 2003). This workshop defined 18 reference, or base, case scenarios which represented a full cross of 3 steepness values for the stock and recruitment curve, three natural mortality vectors and 2 assumptions about trends in catchability. In addition, 26 robustness (or “tick-test”) scenarios were defined based on a permutation of one the base case scenarios. Decision rules were tested over the first and third uncertainty hierarchies as defined at the Second Management Procedure Workshop (Anon. 2003). Note that hierarchies 1 and 3 have been modified from those defined at the initial management procedure workshop (Anon 2002b), and that hierarchy 2 was agreed to be dropped. Hierarchy 4 applies only to the MCMC run, which is included as one of the robustness scenarios. Exploration of the full performance of the various rules, particularly with respect to the robustness trials, was limited by computational constraints – i.e. substantial amounts of computing time were required to run a full set of trials for the third uncertainty hierarchy level for many of the decision rules.

The graphical summaries and associated software initially developed by Eveson and Ricard (2003) and modified by Eveson (2003) were used to evaluate the performance of different decision rules as tuning parameters were varied, and to compare results between different decision rules.

Results and Discussion

Each decision rule was explored over a range of values for the tuning parameters embedded within each rule. Summaries of the performance of each decision rule over a range of values for their tuning parameters are presented in the appendices in which these rules are described. In the current paper, we have not attempted to provide an overall ranking or performance evaluation of the different rules that we have considered. This is because the basis for such an overall comparison across the range of operating models has not been decided upon and a ranking is likely to be dependent upon both the general approach and specific criteria adopted for synthesising results (see Polacheck and Kolody 2003b, Polacheck et al. 2003b). Instead, we limit our discussion here to some general observations and conclusions.

As was seen previously (e.g. Polacheck et al. 2003a) and was expected, variation in the tuning parameters yield negative trade-off in performance between catch indicators and biological status indicators (i.e. higher catches are associated with lower stock status). This was true both within a given operating model scenario and when averages were calculated across scenarios (see the various summary plots within each appendix). The trade-off tends to be smooth, such that small changes in the catch performance indicators yield approximately linear changes in stock status indicators within the range that is likely to be of interest. Differences were found in how this trade-off occurred between different rules among the various scenarios. In addition, we found that the tuning parameters for some rules easily allowed for defining management procedures over a wide range of the stock status versus catch trade-off axis, while for others, their behaviour tended to be more restricted to a particular region of the stock status axis[1].

For an individual model scenario, decisions rules that involved a feedback component in the setting of TACs tended to result in improved performance relative to a constant catch rule (i.e. they tended to take more catch from highly productive scenarios at the expense of less rebuilding and less catch from less productive scenarios with greater rebuilding or a reduced decline). However, when averaged over operating model scenarios, the performance of feedback decision rules in terms of the stock status indicators was often quite similar to the performance obtained for a constant catch rule with the same average catch. This is illustrated in Figures 1 and 2. These two figures compare the performance of seven different feedback rules with that of a constant catch rule. In all cases, including constant catch, the rules have been tuned so that the average catch over all model scenarios is approximately equal to the current catch level (i.e. the 2001 level). Figure 1 shows the median performance of each rule using the B_{2020}:B_{1980} performance indicator and Figure 2 shows the results using the B_{2022}:B_{2002} performance indicator. In terms of average performance, there are only small differences across all of the seven management procedures, and their average performances do not differ substantially from that under constant current catch (labelled as “const v1” on the figures).

[1] Note however, that by adding an additional tuning parameter (e.g. a constant to be added or subtracted from the TAC specified by the rule or a constant multiplier) presumably these rules could also be tuned to provide results across the full range of status values.

It is important to realize that the relatively similar average performances of the candidate management procedures in Figures 1 and 2 are achieved in different ways. Some management procedures result in a much wider range of median catch levels and final spawning stock levels than others. In many cases, procedures appear to be “too aggressive” with the higher productivity scenarios (i.e. those with a large potential for rebuilding). In particular, procedures tended to result in essentially no or negative rebuilding for scenario H55M10, although it has a large potential to rebuild. Similarly, most procedures were unable to reduce the TAC quickly enough to prevent further declines of the stock for the least productive scenarios. Examples of more detailed comparisons of how these different procedures performed within a particular scenario are presented in Figures 4-6.

It is also important to note, however, that more substantial differences exist among the different procedures when their performances were compared in terms of risk of not achieving an objective rather than in terms of average or median performance. This is illustrated in Figure 3 in which the same results as Figure 2 are shown but for the lower 10^{th} percentiles of the performance indicators rather than the medians (i.e. the lower bounds on the 80% confidence intervals are plotted for both the biomass indicator and the average catch indicator). In this case, for example, the management procedure shown for the Fox_pe and the Fox_cpue decision rules provide more robust performance (i.e. relatively higher probability that the stock will not decline to a low level) than the ACRLRT and ASURE rules. This difference in “risk” among procedures is also evident in comparing the lower bound of the 80% confidence intervals in Figures 4-6.

Figure 7 provides a summary of the performance of these seven management procedures in terms of the robustness criteria (i.e. “stoplights”) defined at the Second Management Procedure Workshop. In this case (and more generally), the robustness criteria did not provide a useful distinction among different procedures. When decision rules were tuned to similar average performance, they tended to either fail or pass all of the criteria.

With respect to the set of robustness trials (i.e. “tick tests”), all procedures tended to exhibit similar sensitivities (see Figure 8 and the figures in the Appendices). Thus, the management procedures appear to have been reasonably robust to only the alternative specifications for selectivity change (i.e. H30M10Q0_SC and HM10Q0_SC) and the fecundity-based spawning biomass assumptions (H30M10Q0_Fec and H30M10Q0_Fec). However, we note that the fecundity assumption was handled in a rather ad hoc fashion and probably did not represent the extent of plausible uncertainty in this dimension. With respect to the other scenarios in the robustness trials, most procedures appeared to be reasonably sensitive to the alternative hypotheses embedded within them (particularly with respect to the stock status related performance indicators). As such, it would appear necessary to carry the remaining uncertainty dimensions contained in these robustness trials into the final trials (or at least until there is further clarification on the management objectives and the process for synthesizing across scenarios of varying plausibility) in order to ensure a comprehensive and robust set of tests have been completed.

In addition to looking at average performance, time trajectories of catches and biomass trends can be quite variable among different procedures (e.g. Figures 9 and

10). Procedures differ both in the amount of variability around their average trajectory and in the timeframes during which major changes in TACs take place. Some of the model-based procedures may, in fact, suffer somewhat more from high interannual variability in TACs due to large changes in estimated parameters from year to year. It may be worth considering applying some 'smoothing' to results to improve performance.

As in our initial attempts at developing a management procedure for SBT, the results from this second stage testing further demonstrate the difficulty in developing a procedure that can provide good performance with respect to catches and still ensure “acceptable” behaviour with respect to the stock status performance indicators over all scenarios. With respect to meeting biomass conservation targets, a management procedure with perfect performance would result in appropriately low catches for low productivity scenarios and high catches for high productivity scenarios. In terms of the summary graphics (e.g. Figure 1), this idealized case would be represented by a flat star (i.e. a constant biomass index, with variability in productivity among scenarios represented by variability in catch). . However, achieving this level of control continues to prove elusive, as procedures tend to be either too “conservative” with respect to the high productivity scenarios or too “aggressive” in the low productivity scenarios. “Conservative” is used in the sense of overshooting (in some cases substantially) the CCSBT rebuilding target and thus having foregone catches. “Aggressive” is used in the sense that the stock did not recover to the rebuilding target and/or the final spawning stock biomass (SSB) was below its current level. We recognize that what we describe above as perfect performance with respect to biomass targets is a rather hypothetical concept at this point, requiring further clarification from managers.

We have noted that some rules have explicit targets (often as tuning parameters) which they aim for, whereas others do not. In most cases where the target is defined as some level of SSB (e.g. SSB in 1980 or SSB in some other year), the current SSB is below that target, and depending on where the target is set, the current set of model scenarios may or may not all enable the rule to reach its target within the time-frame of the tests (20 years, as default). Once the management objectives have been defined, it may be necessary to conduct trials with slightly longer time-frames to ensure that the rules behave acceptably once they have reached or over-shot the target.

In conclusion, initial testing of candidate management procedures indicates that there is scope for improved performance by adopting a feedback approach. However, given the wide range of uncertainty about the SBT stock dynamics embedded in the current model scenarios, there appears to be substantial limits to what can be achieved to ensure that the CCSBT rebuilding target is met where possible (or at least that some rebuilding of the stock occurs) and to simultaneously ensure that the resource in terms of catch is not under utilized. Further progress in the development process and providing recommendations on a choice of management procedure will require clarification of management objectives in terms of robustness and risk combined with a clearly defined process of synthesis and evaluation that takes into account these objectives and the plausibility of different scenarios.

Literature Cited

Anon. 2002a. Report of the third Stock Assessment Group Meeting. CCSBT. 3-7 September, 2003. Canberra, Australia.

Anon. 2002b. Report of the First Meeting of the Management Procedure Workshop. CCSBT. 3-4 & 6-8 March 2002. Tokyo, Japan.

Anon. 2003. Report of the Second management procedure workshop. CCSBT. 7-9, 12, & 14-15 March 2003. Queenstown, New Zealand.

Eveson. P. 2003. An update of the graphics used for evaluating the performance of candidate management procedures for southern bluefin tuna. CCSBT-ESC/0309/23.

Eveson, P. and Ricard, D. 2003. An overview of potential graphics for evaluating the performance of candidate management procedures for southern bluefin tuna. CCSBT-MP/0304/05.

Polacheck, T. and D. Kolody. 2003a. The behaviour and fit of alternate operating model specifications for testing the performance of southern bluefin tuna candidate management procedures. CCSBT-MP/0304/8.

Polacheck, T. and D. Kolody. 2003b. Synthesising Performance of Candidate Management Procedures Across Different Operating Model Scenarios. CCSBT-MP/0304/8.

Polacheck, T., D. Ricard, P. Eveson, M. Basson and D. Kolody. 2003a. Results from Initial Testing of Some Candidate Management Procedures for Southern Bluefin. CCSBT-MP/0304/6.

Polacheck, T., D. Kolody and M. Basson. 2003b. Issues in the selection of final trials for testing SBT management procedures and for the process of synthesizing results from the simulation testing. CCSBT-ESC/0309/27.

CSIRO
MARINE RESEARCH

APPENDICES FOR:

RESULTS FROM FURTHER TESTING OF CANDIDATE MANAGEMENT PROCEDURES FOR SOUTHERN BLUEFIN TUNA

Tom Polacheck
Dan Ricard
Paige Eveson
Marinelle Basson
Dale Kolody
Jason Hartog

Prepared for the CCSBT SAG
25-29 August 2003, Christchurch, New Zealand

CCSBT-ESC/0309/29

A1. Const (constant catch)

A1.1. Description of rule

A1.1.1. Overview

Const simply sets the TAC to a constant value *C* in all years, where *C* is a tuning parameter of the rule. Const is meant to serve as a reference point for evaluating other decision rules; the zero catch case ($C = 0$) is particularly useful in this regard.

A1.1.2. Mathematical description

For year *t*, TAC[*t*] = *C*

A1.1.3. Versions (tuning parameter values)

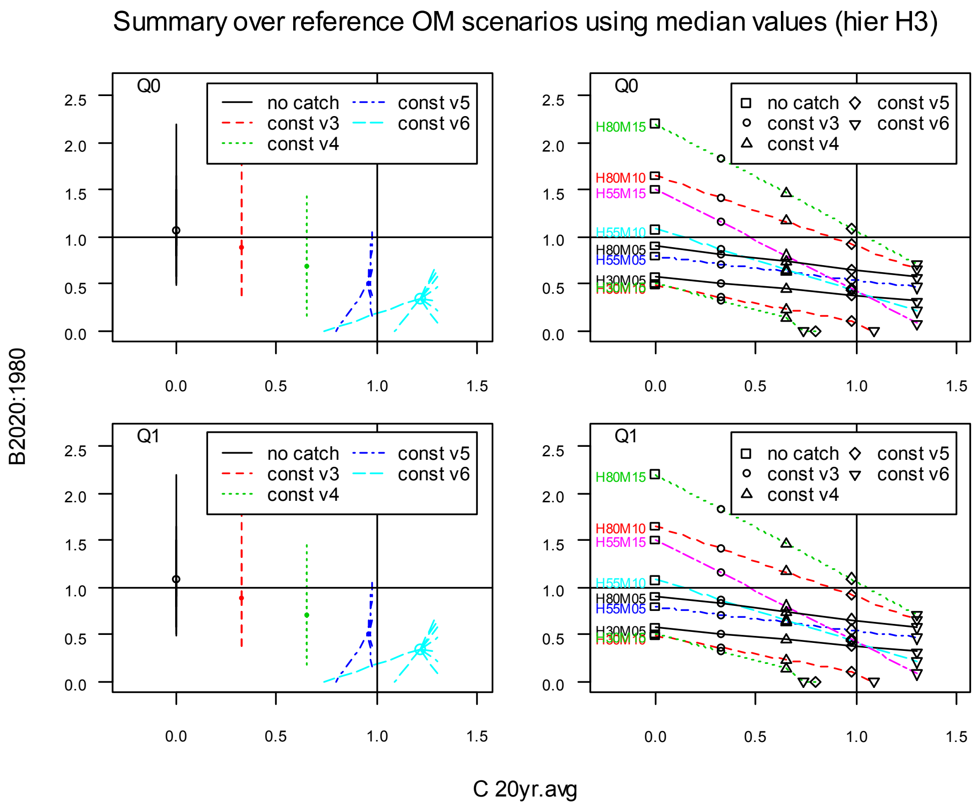
Version	*C* (in MT)
1	Catch for 2001 = 15385.7*
2	0
3	5000
4	10000
5	15000
6	20000

*The problem with the catch for 2001 in MT differing between operating model scenarios has been resolved.

A1.2. Performance of rule

Const is intended to serve as a reference with which to compare other decision rules. The zero catch case is particularly useful in this regard as it shows the maximum biomass performance that can be achieved under a given operating model scenario. It is apparent from Figure A1-1 that a constant catch of 15000 MT or more is not sustainable under several of the model scenarios (i.e. the population crashes and the TAC cannot be achieved).

Figure Const 1. Performance of 'Const' with respect to catch and biomass statistics.



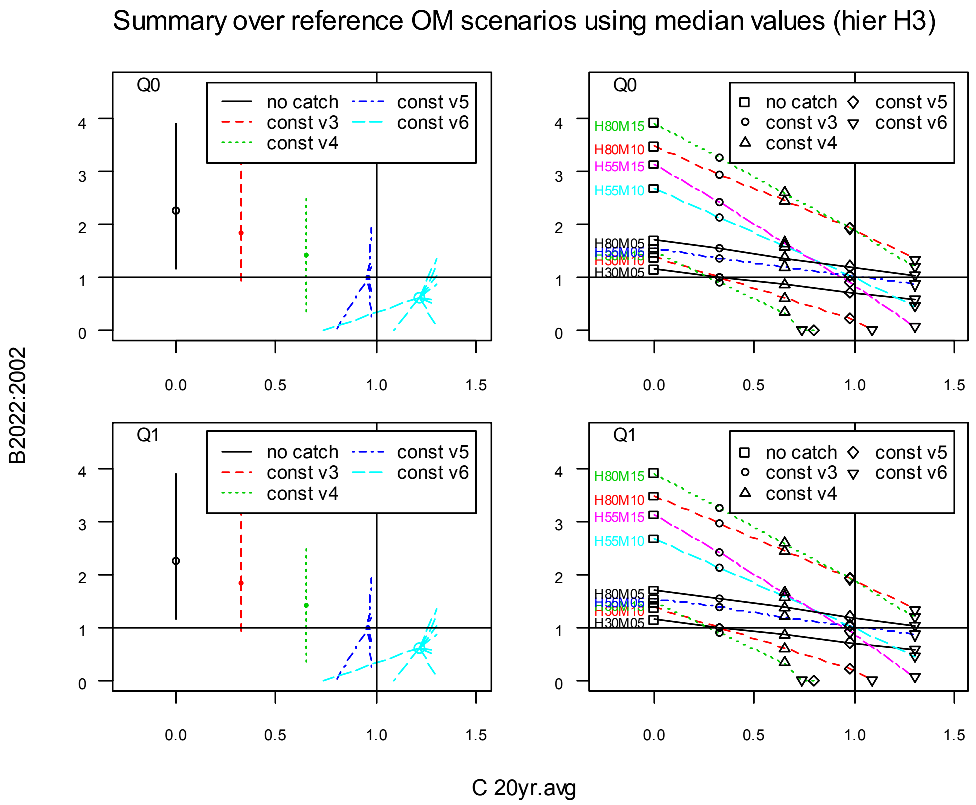
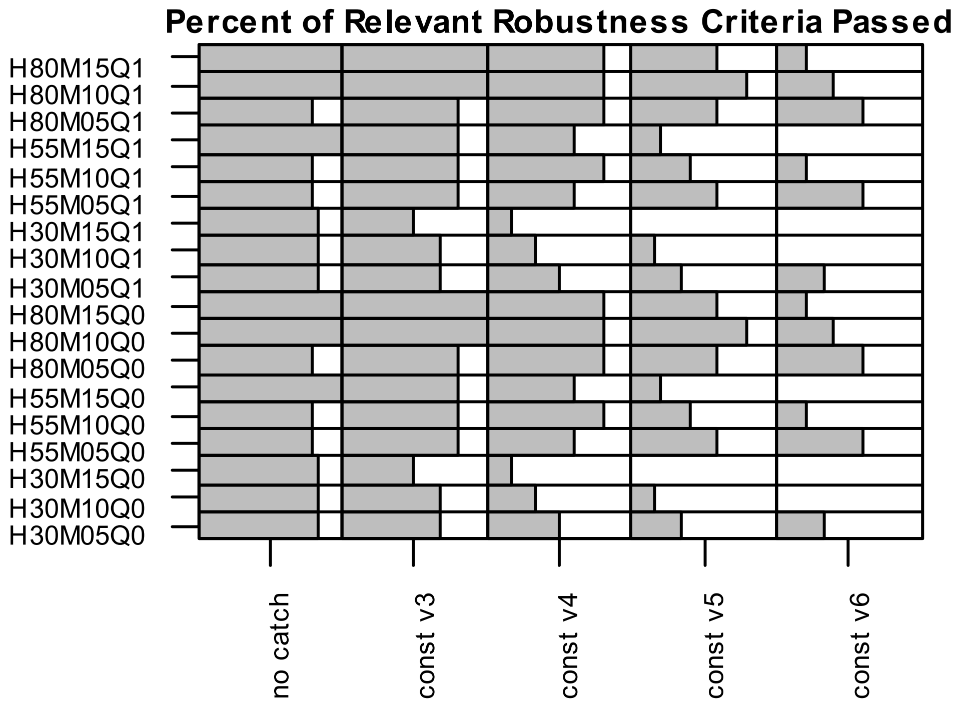


Figure Const 2. Performance of 'Const' with respect to robustness criteria.



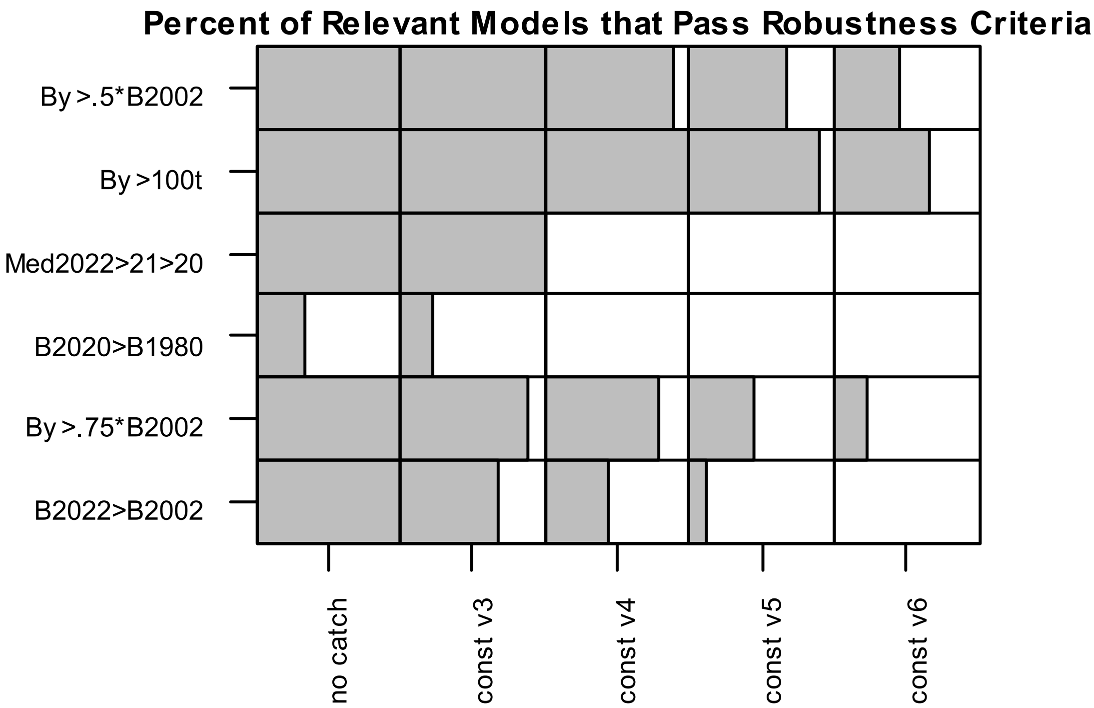
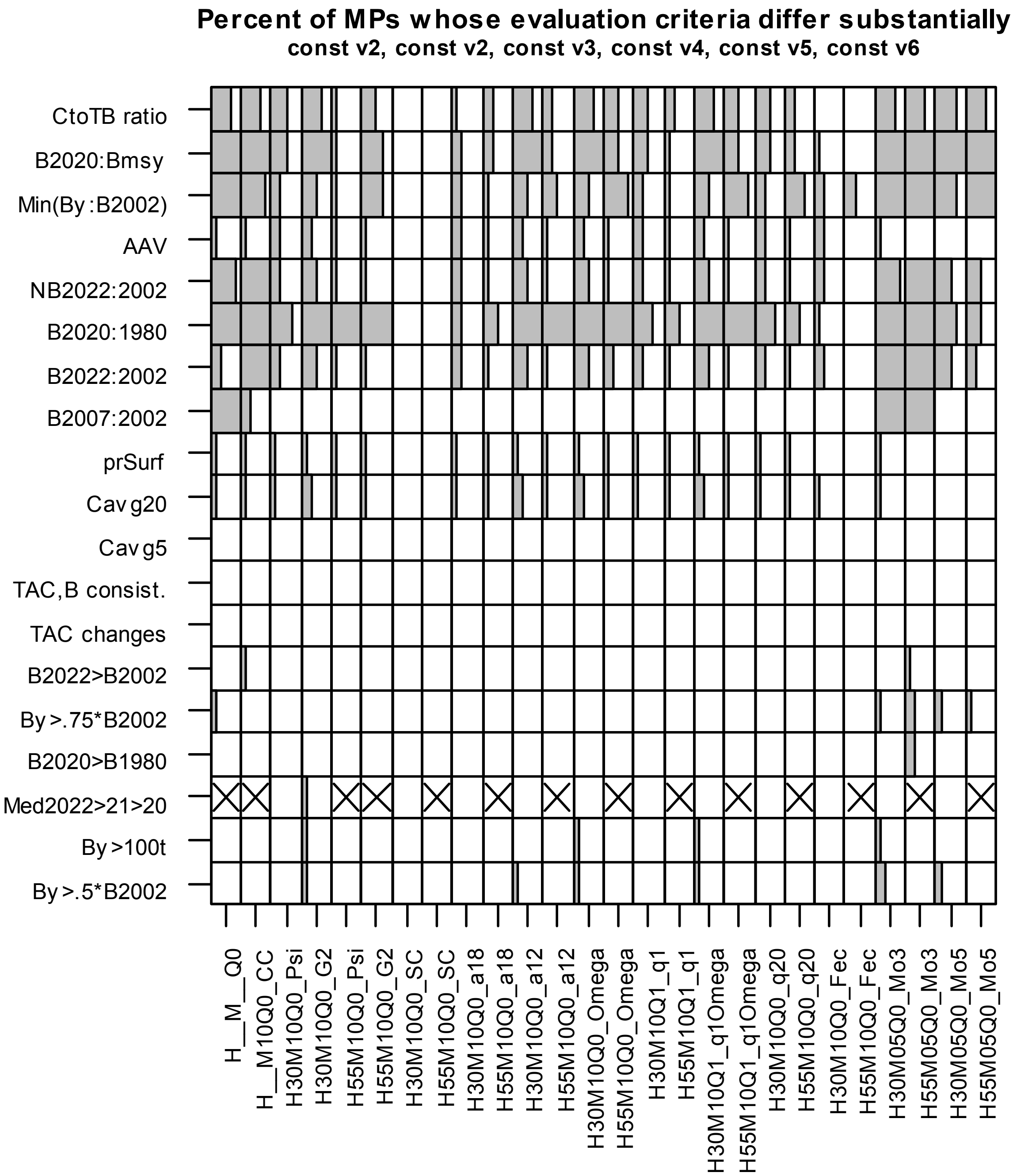


Figure Const 3. Robustness tests for versions 2 to 6 of 'Const' under hierarchy 3.



A2. CPUE (nominal LL1 CPUE)

A2.1. Description of the rule

A2.1.1. Overview

The trend in the log(nominal CPUE) over the last *n* years is used to adjust the TAC.

A2.1.2. Mathematical description

$$\begin{aligned} TAC_{y+1} &= \omega TAC_y + (1-\omega) TAC_y (1+\kappa\lambda_n) \\ &= TAC_y (1+(1-\omega)\kappa\lambda_n) \\ &= TAC_y (1+\alpha\lambda_n) \end{aligned}$$

where,

α is the weight given to the log(CPUE) slope, and
λ_n is the slope of the regression of log(CPUE) vs. time over the last *n* years.

If $TAC_{y+1} - TAC_y < 3000$ then $TAC_{y+1} = TAC_y - 3000$.
If $TAC_{y+1} - TAC_y > 3000$ then $TAC_{y+1} = TAC_y + 3000$.

(Note that this is equivalent to the CPUE rule presented at the April 2003 MP Workshop – we have simply reduced the number of tuning parameter from 2 to 1 via the calculations shown above.)

A2.1.3. Versions (tuning parameter values)

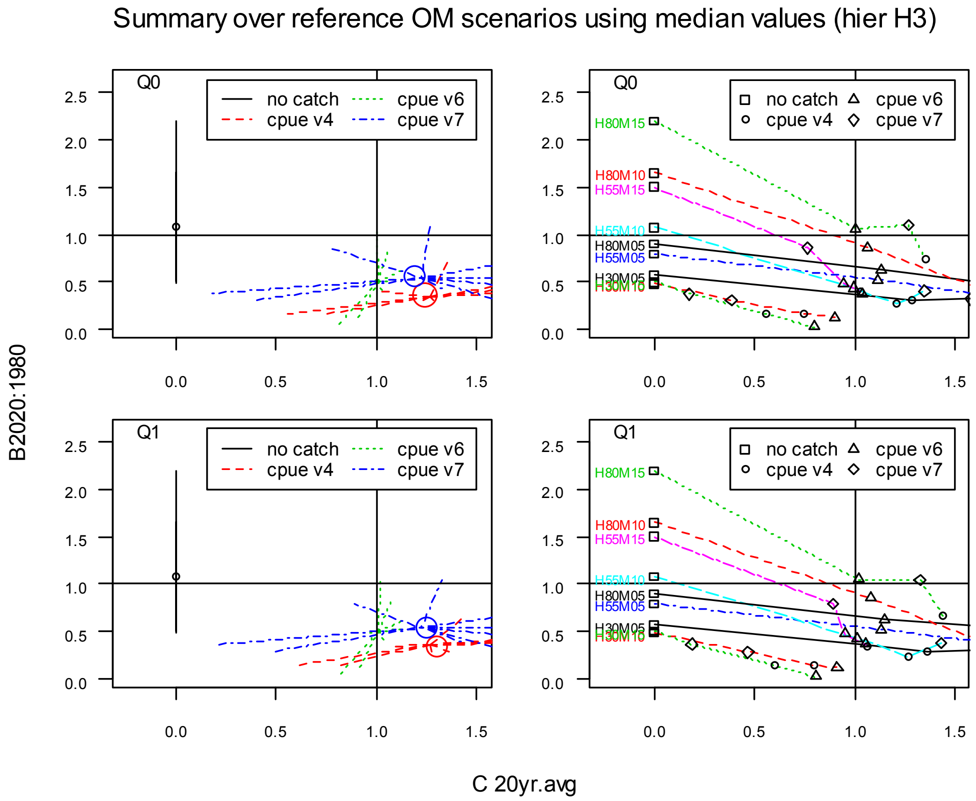
Version	α	n
1	0.5	5
2	1.0	5
3	2.0	5
4	5.0	5
5	10.0	5
6	0.5	20
7	10.0	20

A2.2. Performance of rule

This decision rule does not perform very well overall. On average, it takes too much catch so that there is no rebuilding of the stock (in fact, the stock size decreases below the current level under most of the operating model scenarios). There does, however, seem to be some information in the CPUE trend to indicate whether or not the stock is productive. In particular, under low steepness scenarios the CPUE trend tends to be negative, whereas under high steepness scenarios the CPUE trend tends to be positive. We attempted to make better use of the CPUE trend in the decision rule "CPUE_2020".

Versions 4, 6 and 7 represent the range of performance attained, as summarized in Figures 1 and 2. Version 7 appears to give the best overall performance in terms of the catch/biomass trade-off (Figure 1), and also in passing the most of the robustness criteria (Figure 2). Figure 3 shows how sensitive this decision rule is to the various robustness models that are being explored. As with most of the rules, the results show little sensitivity to robustness trials *_Fec and *_SC but a high degree of sensitivity to many of the others.

Figure CPUE 1. Performance of 'CPUE' with respect to catch and biomass statistics.



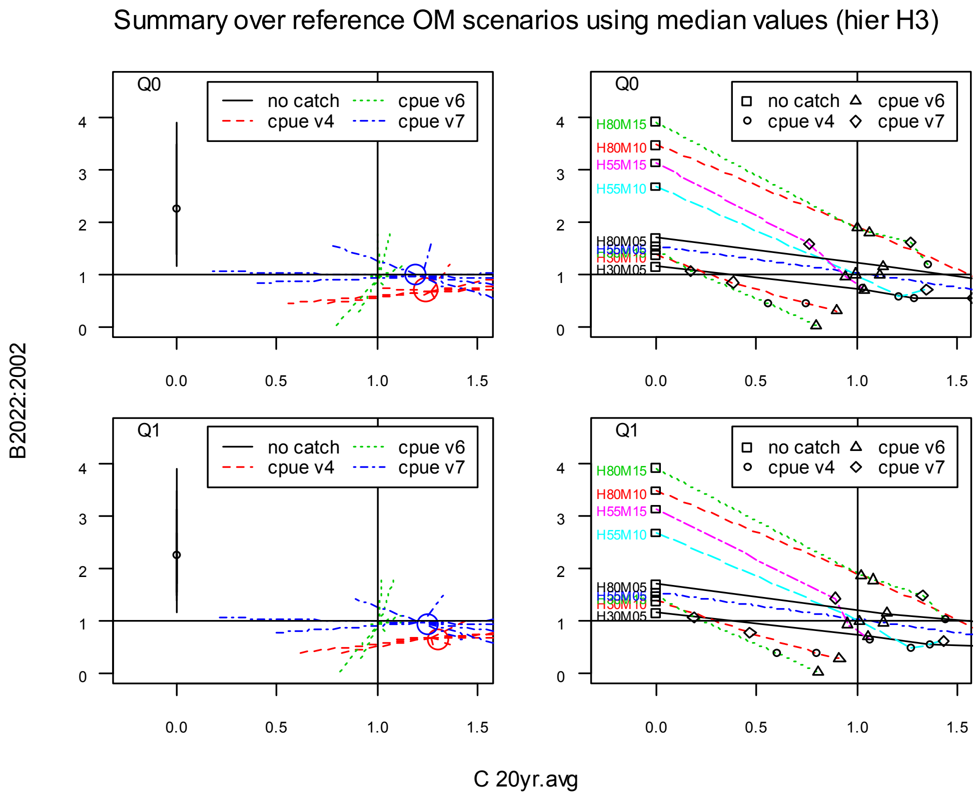
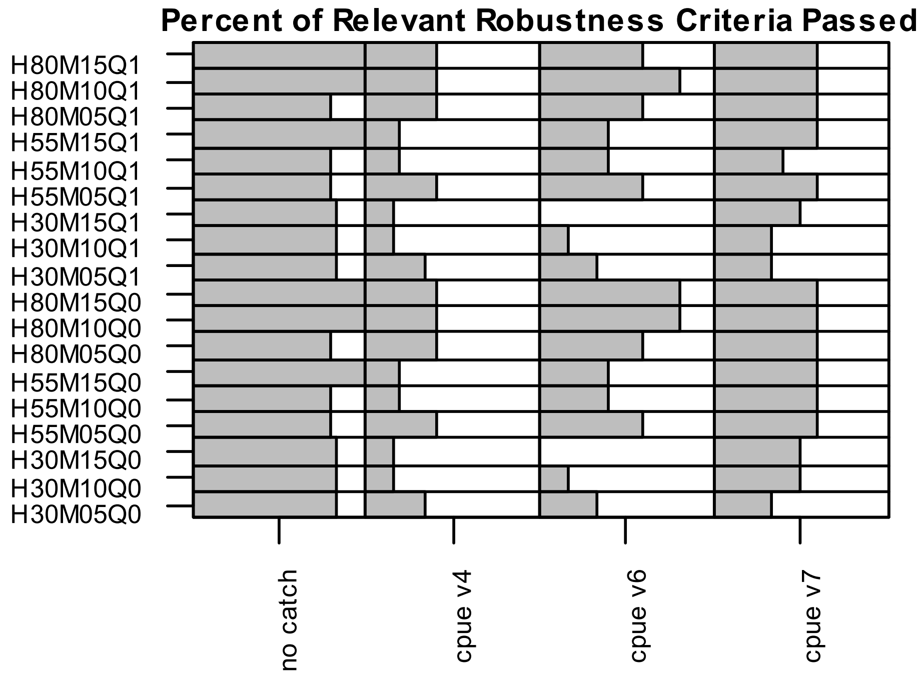


Figure CPUE 2. Performance of 'CPUE' with respect to robustness criteria.



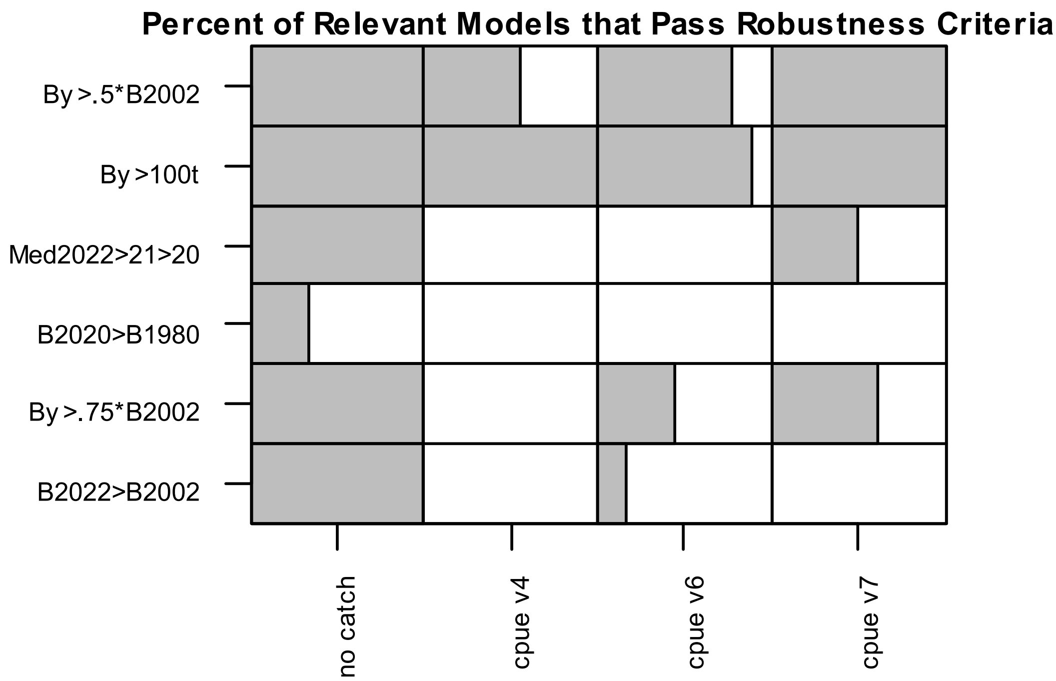
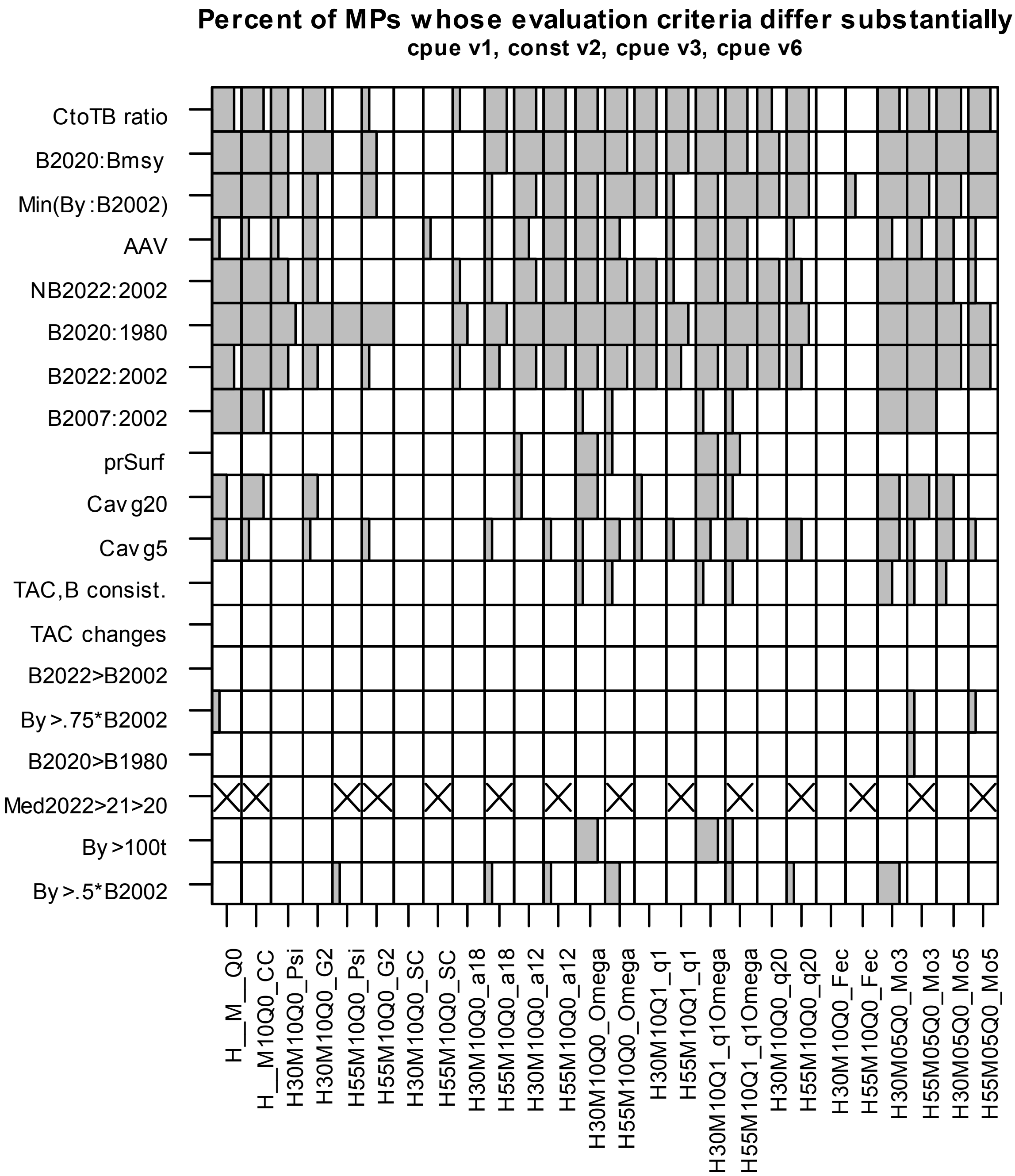


Figure CPUE 3. Robustness tests for 4 versions of 'CPUE' under hierarchy 3.



A3. CPUE_2020

A3.1. Description of the rule

A3.1.1. Overview

A linear trend is fitted through the log(nominal CPUE) over the last *n* years, and the fitted line is used to forecast the CPUE value in 2020. If the forecasted value is above a certain reference level, then the TAC is adjust upwards from the previous year; similarly, if the forecasted CPUE value is below the reference value, then the TAC is adjusted downwards. The amount that the TAC is adjusted is proportional to the amount that the forecasted CPUE value differs from the reference value, and the reference value depends on the tuning parameters chosen.

A3.1.2. Mathematical description

Let β_n and λ_n be the intercept and slope respectively of the linear regression of log(CPUE) vs. time over the last *n* years, where *n* is a tuning parameter. Then the predicted log(CPUE) value in 2020 is:

$$\log(\text{CPUE}_{2020}) = \beta_n + \lambda_n * 2020\,.$$

The reference log(CPUE) value to which we compare the 2020 forecasted value is given by the log of the mean CPUE over years y1 to y2, where y1 and y2 are tuning parameters. That is,

$$\log(\text{CPUE}_{ref}) = \log(\text{mean}(\text{CPUE}_{y1:y2}))\,.$$

The TAC is then set as follows:

$$TAC_{y+1} = TAC_y(1+\alpha)$$

where

$$\alpha = \frac{\log(\text{CPUE}_{2020}) - \log(\text{CPUE}_{ref})}{\left|\log(\text{CPUE}_{ref})\right|}\,.$$

If $TAC_{y+1} - TAC_y < 3000$ then $TAC_{y+1} = TAC_y - 3000$.

If $TAC_{y+1} - TAC_y > 3000$ then $TAC_{y+1} = TAC_y + 3000$.

A3.1.3. Versions (tuning parameter values)

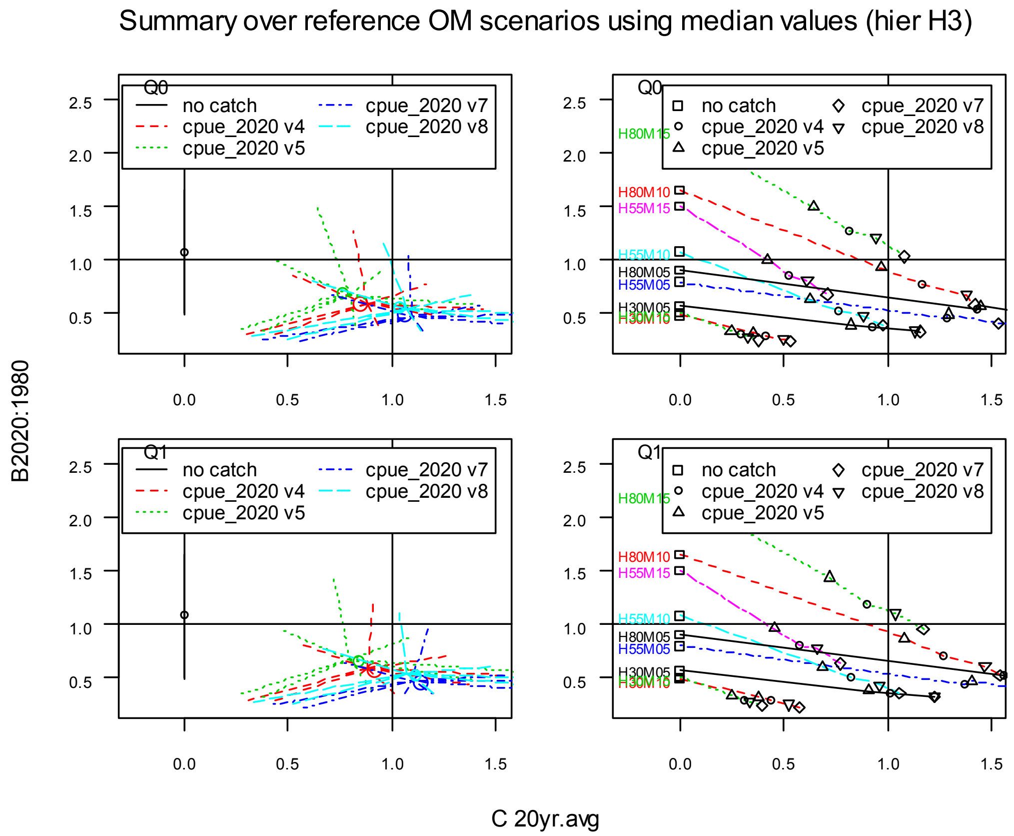
Version	n	$y1$	$y2$
1	5	1980	2002
2	10	1980	2002
3	20	1980	2002
4	5	1980	1980
5	10	1980	1980
6	20	1980	1980
7	5	1980	1990
8	10	1980	1990

A3.2. Performance of rule

The decision rule 'CPUE_2020' can easily be tuned to move along the biomass/catch trade-off line by adjusting the reference CPUE level up or down. For example, using the 1980 CPUE level (as in versions 4 and 5) gives less catch but greater biomass than using the average 1980 to 1990 level (as in versions 7 and 8). Once at a general position on the biomass/catch trade-off graph, finer tuning can be achieved by adjusting the number of years over which the CPUE trend is being calculated. This rule seems to respond reasonably well to the different operating model scenarios, allowing for greater catches when the productivity of the stock is assumed to be high, and reducing catches greatly when the productivity of the stock is low. This can be seen in the reasonably "flat" nature of the stars in Figure 1. Note that only versions 4, 5, 7 and 8 have been plotted as these were deemed to show the most reasonable performance while still showing the range of performance that can be attained.

Figure 3 shows how sensitive this decision rule is to the various robustness models that are being explored.

Figure CPUE_2020 1. Performance of 'CPUE_2020' with respect to catch and biomass statistics.



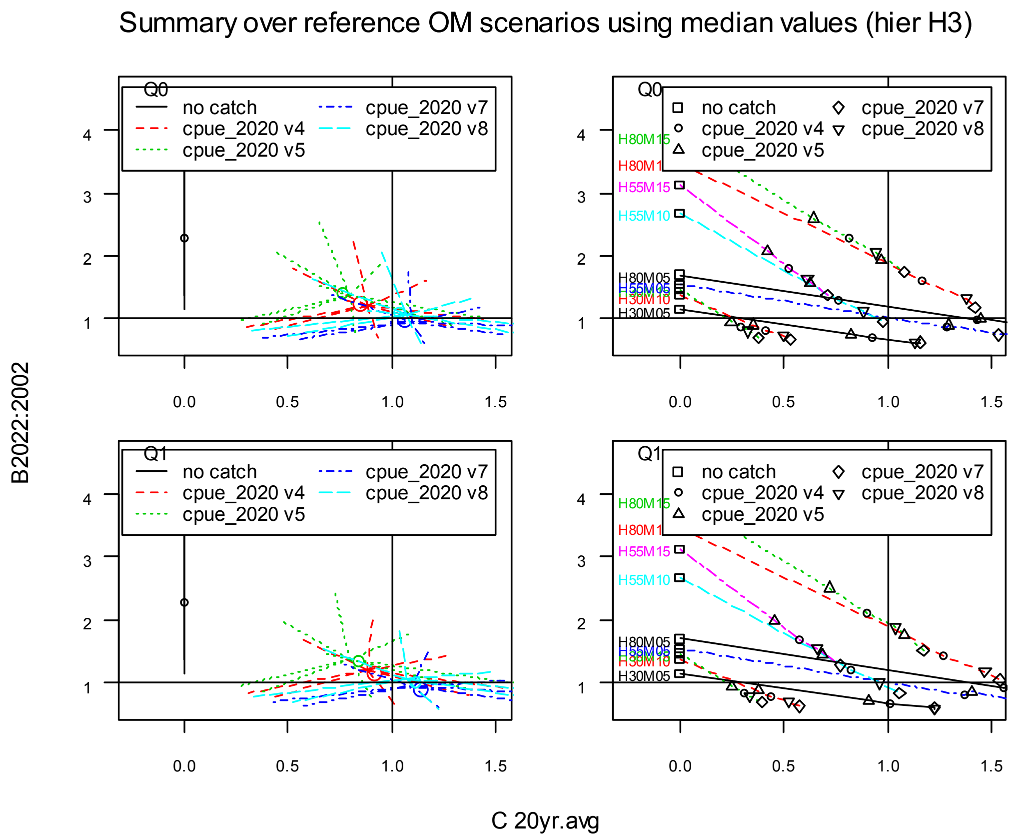


Figure CPUE_2020 2. Performance of 'CPUE_2020' with respect to robustness criteria.

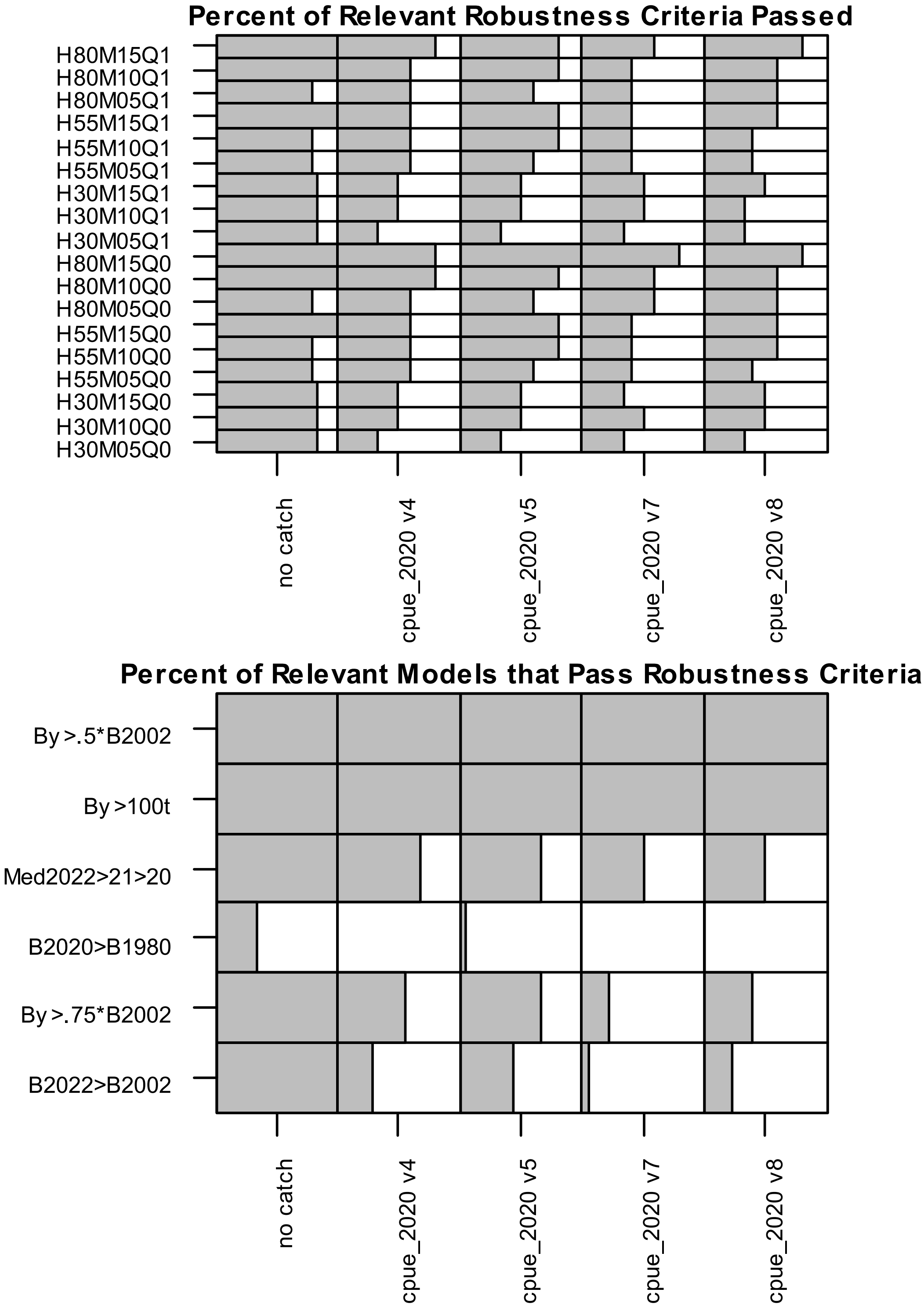
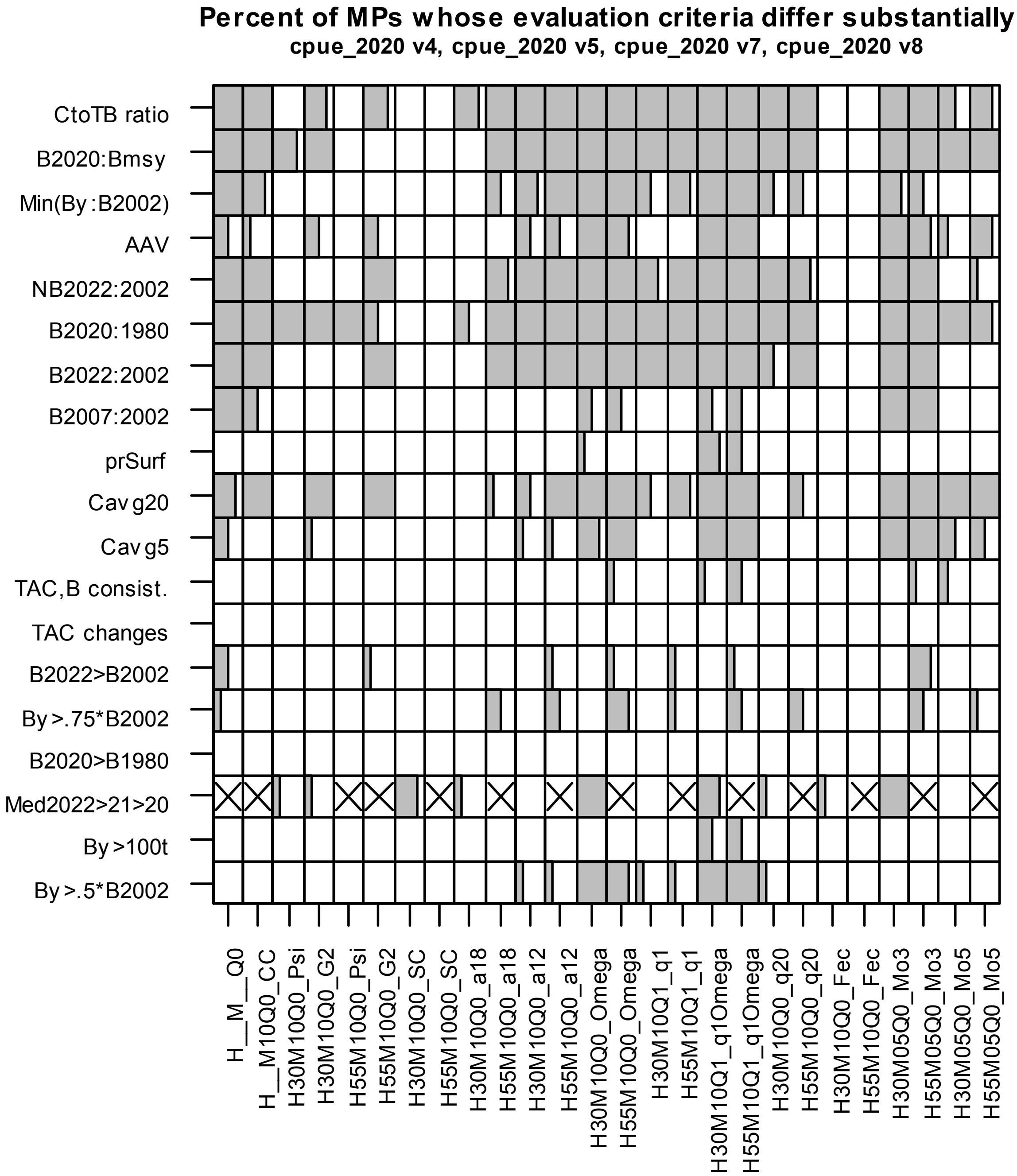


Figure CPUE_2020 3. Robustness tests for 4 versions of 'CPUE_2020' under hierarchy 3.



A4. Fox (Fox production model)

A4.1. Description of rule

A4.1.1. Overview

This decision rule fits a Fox surplus production model to the nominal LL1 CPUE. The parameter estimates of *r* and *k* are used to compute MSY and B_{MSY}. The rule sets the TAC to a certain fraction of the exploitation ratio ($F_{MSY} = MSY / B_{MSY}$). This fraction is determined based on the estimated ratio B_y / B_{MSY}.

A4.1.2. Mathematical description

$$TAC_{y+1} = B_y \left(M_y * F_{MSY} \right)$$
$$M_y = \eta \left(B_y / B_{MSY} - 1 \right) + \beta$$

A4.1.3. Versions (tuning parameter values)

Version	β	η
1	0.9	0.5
2	0.9	1.0
3	0.9	1.5
4	0.5	0.5
5	0.5	1.5

A4.2. Performance of rule

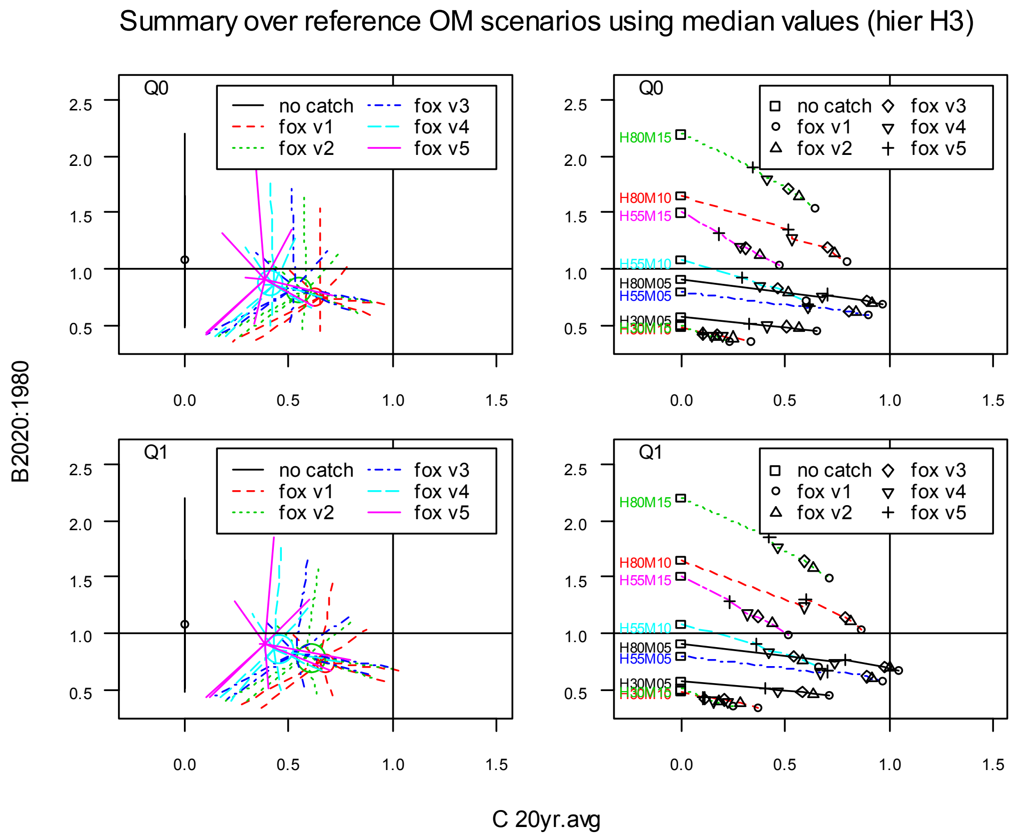
The current estimates of stock status are well below B_{MSY} and, subsequently, the rule reduces the TAC in the early years of the simulations. The catch reduction is constrained by the maximum allowable change of 3000 MT.

This rule is very conservative and performs well in terms of the biomass indicators. The management objective to rebuild the stock to 1980 levels by 2020 is achieved for all OM scenarios that achieve the objective under no catch. However, the catch is unnecessarily reduced for productive OM scenarios.

Figure 1 summarizes the performance of 'Fox' with respect to the catch/biomass trade-off, and Figure 2 summarizes its performance with respect to the robustness criteria. Figure 3 shows how sensitive this decision rule is to the various robustness models that are being explored. Overall this decision rule tends to show less sensitivity to the robustness trials than many of the other rules, perhaps as a result of its conservative nature.

A potential improvement to this rule would be to use the *r* parameter estimate to adjust TAC accordingly. "Fox_pe" is an attempt at implementing such a "learning" rule.

Figure Fox 1. Performance of 'Fox' with respect to catch and biomass statistics.



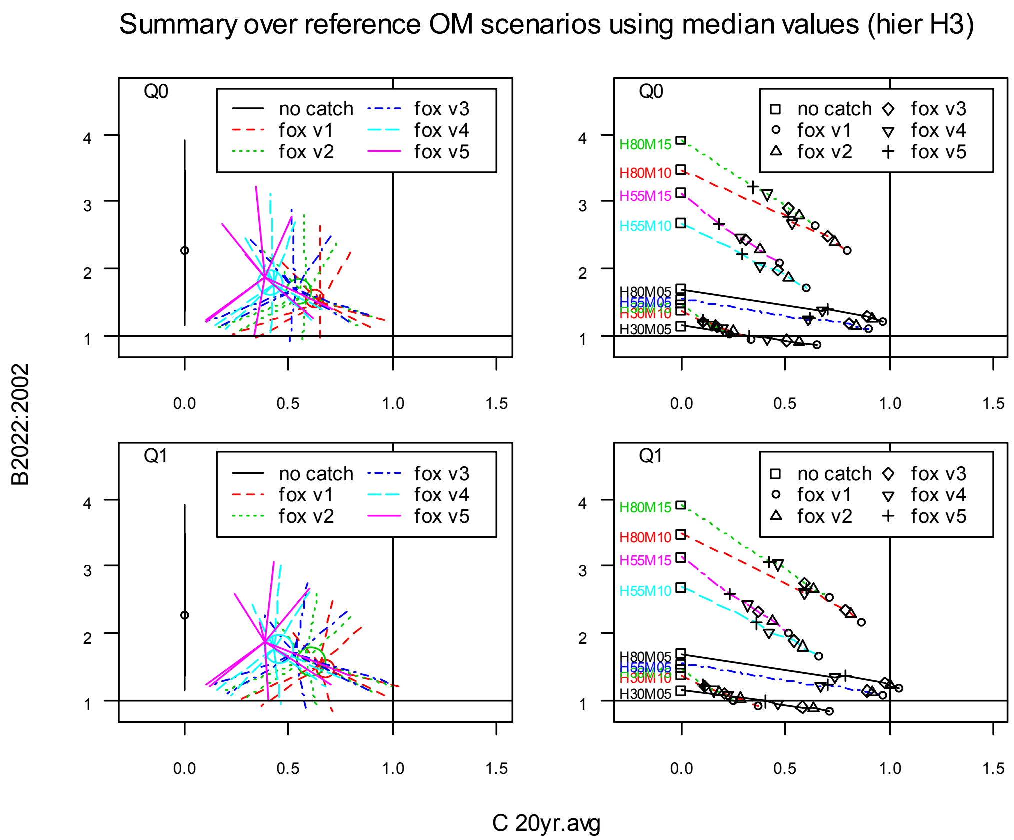
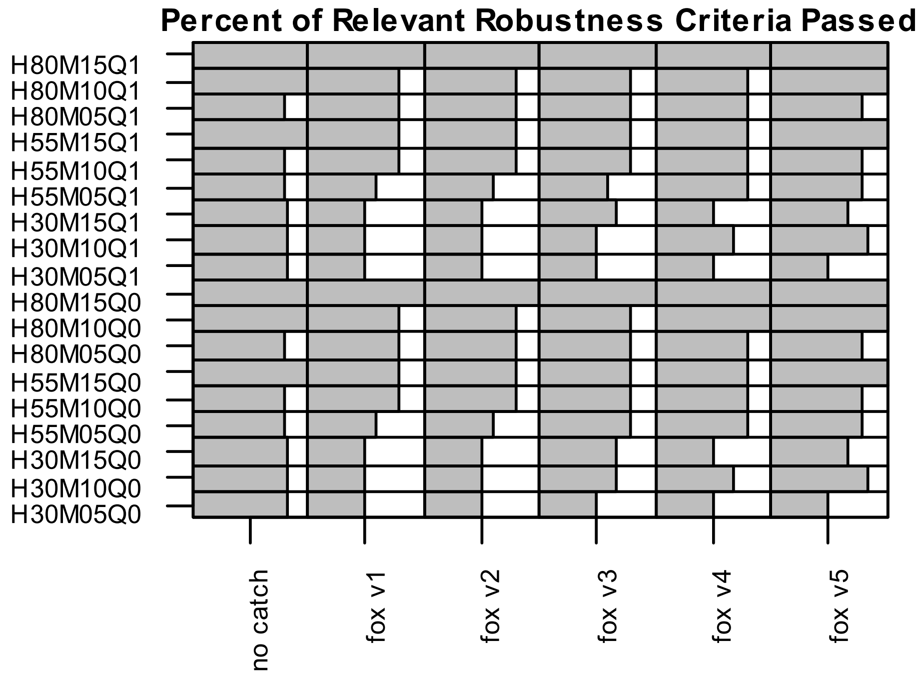


Figure Fox 2. Performance of 'Fox' with respect to robustness criteria.



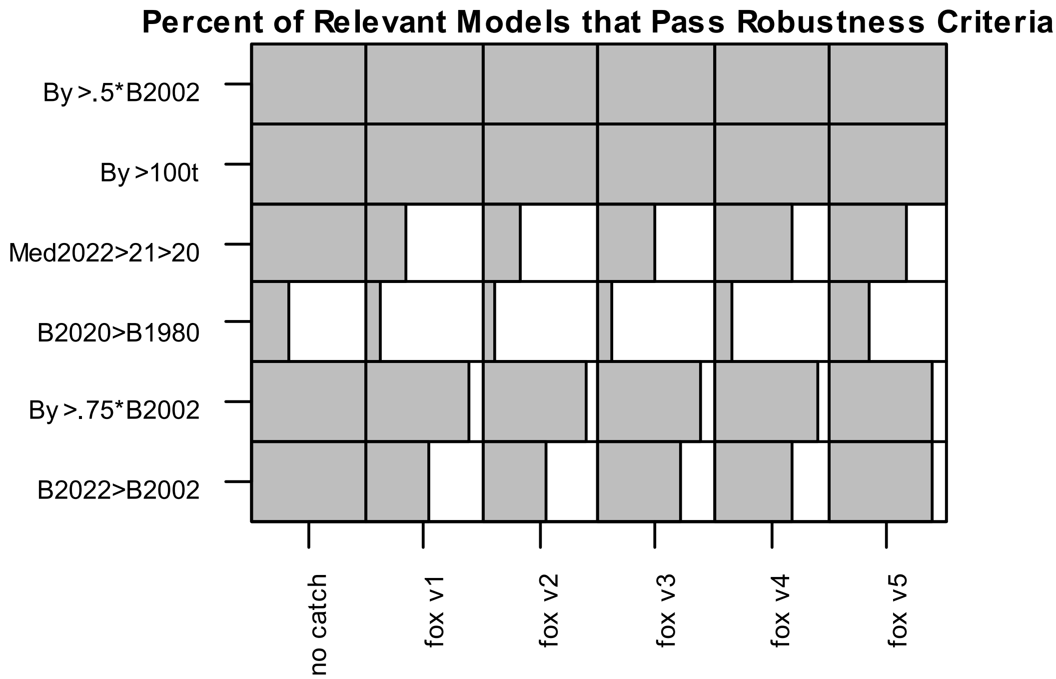
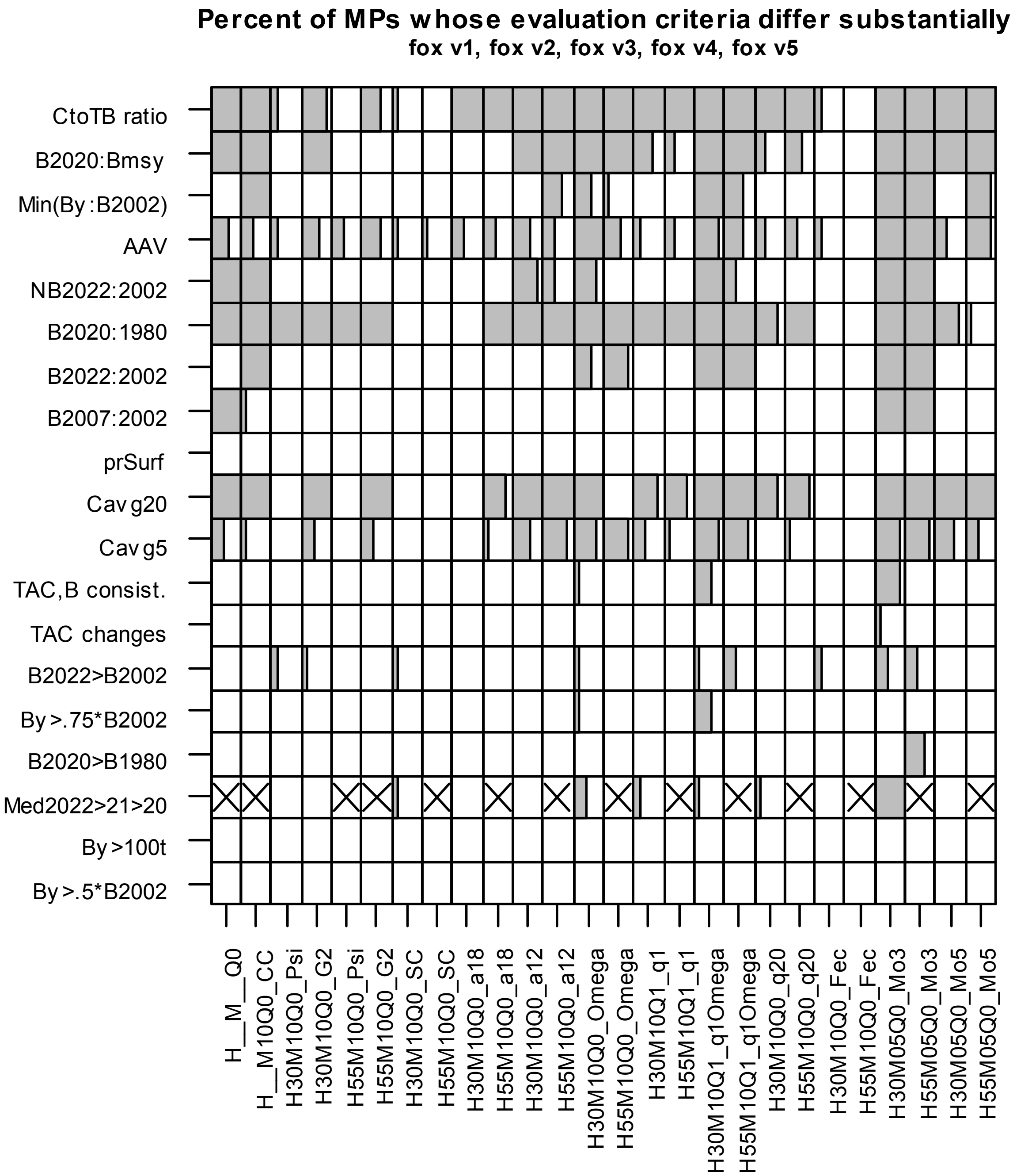


Figure Fox 3. Robustness tests for versions 1 to 5 of 'Fox' under hierarchy 3.



A5. Fox_pe

A5.1. Description of rule

A5.1.1. Overview

This decision rule is a variation of the rule 'Fox', which attempts to improve the performance of 'Fox' by adjusting the TAC according to the productivity of the stock (assumed to be directly related to the parameter r). The parameter estimates of r and k are used to compute MSY and B_{MSY}. The rule sets the TAC in year y using the MSY value times the ratio $\left(B_y / B_{MSY}\right)^{\delta}$, where δ depends on the estimated r value — δ is intended to make the biomass ratio smaller/larger as r decreases/increases (indicating a less/more productive stock). The exact details are given in the mathematical description below.

A5.1.2. Mathematical description

$$TAC_{y+1} = \alpha MSY \left(\frac{B_y}{B_{MSY}} \right)^{\delta},$$

$$\delta = \begin{cases} (r/r^*)^{\theta} & \text{if } B_y / B_{MSY} > 1 \\ (r^*/r)^{\theta} & \text{if } B_y / B_{MSY} \leq 1 \end{cases}$$

where α, r* and θ are tuning parameters of the decision rule.

Note that in all of the situations we examined, B_y / B_{MSY} was less than 1, with a typical ratio being about 0.4. A typical value of r was around 0.5, but it ranged from 0.1 up to 0.8.

A5.1.3. Versions (tuning parameter values)

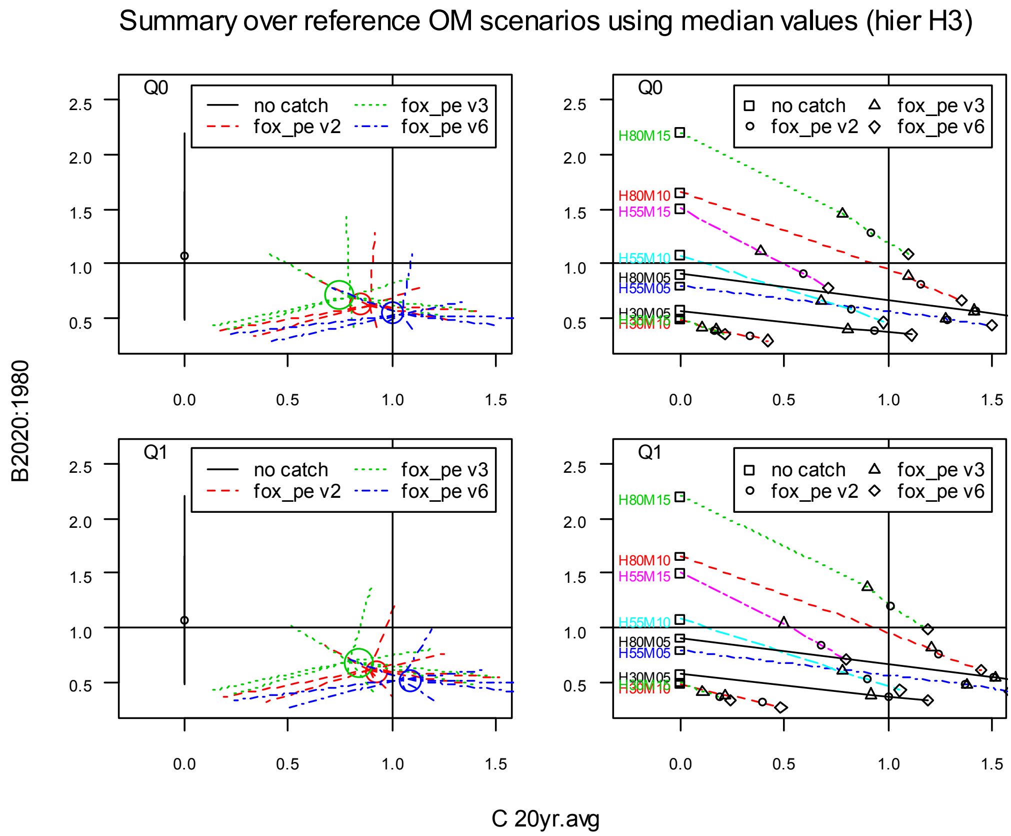
Version	α	r*	θ
1	1.5	0.7	0.7
2	1.5	0.7	1.0
3	1.5	0.7	2.0
4	1.5	0.9	0.7
5	1.5	0.9	1.0
6	2.0	0.7	1.0

A5.2. Performance of rule

'Fox_pe' is fairly successful at using the estimated r parameter as an indication of stock productivity and adjusting the TAC accordingly. This can be seen in Figure 1 by the fact that, in general, catches are greatly reduced in low productivity scenarios but are quite high in high productivity scenarios. The tuning parameters of this rule can be used to adjust the position of the results in terms of the catch/biomass trade-off. Versions 2, 3 and 6 were considered to give best performance while still showing the range of performance that can be attained. As such, these 3 versions are shown in all of the figures. The performance of these 3 versions is very similar with respect to the robustness criteria (Figure 2), even though they are quite different in terms of the catch/biomass trade-off (Figure 1).

Figure 3 shows how sensitive 'Fox_pe' is to the various robustness models that are being explored. This decision rule gives a similar picture in general to most of the other rules.

Figure Fox_pe 1. Performance of 'Fox_pe' with respect to catch and biomass statistics.



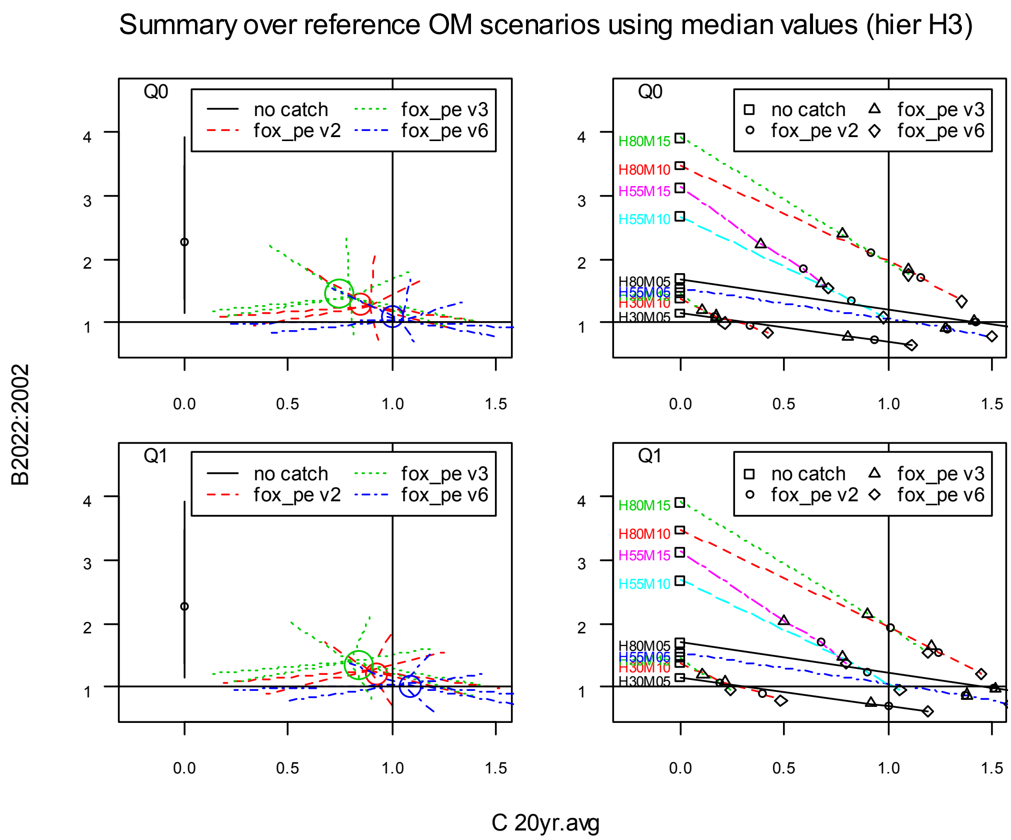
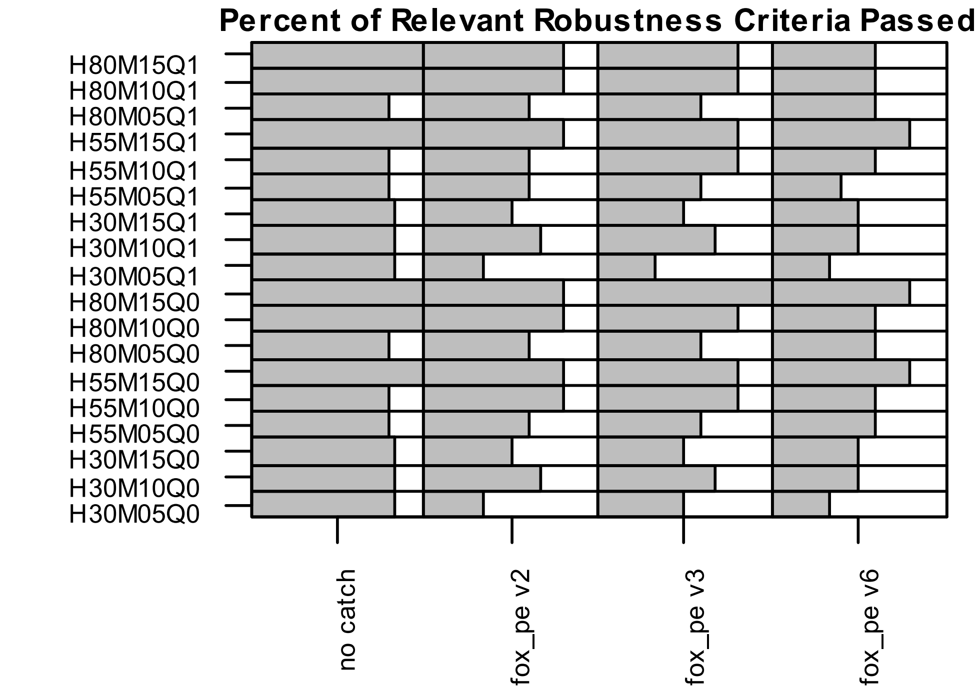


Figure Fox_pe 2. Performance of 'Fox_pe' with respect to robustness criteria.



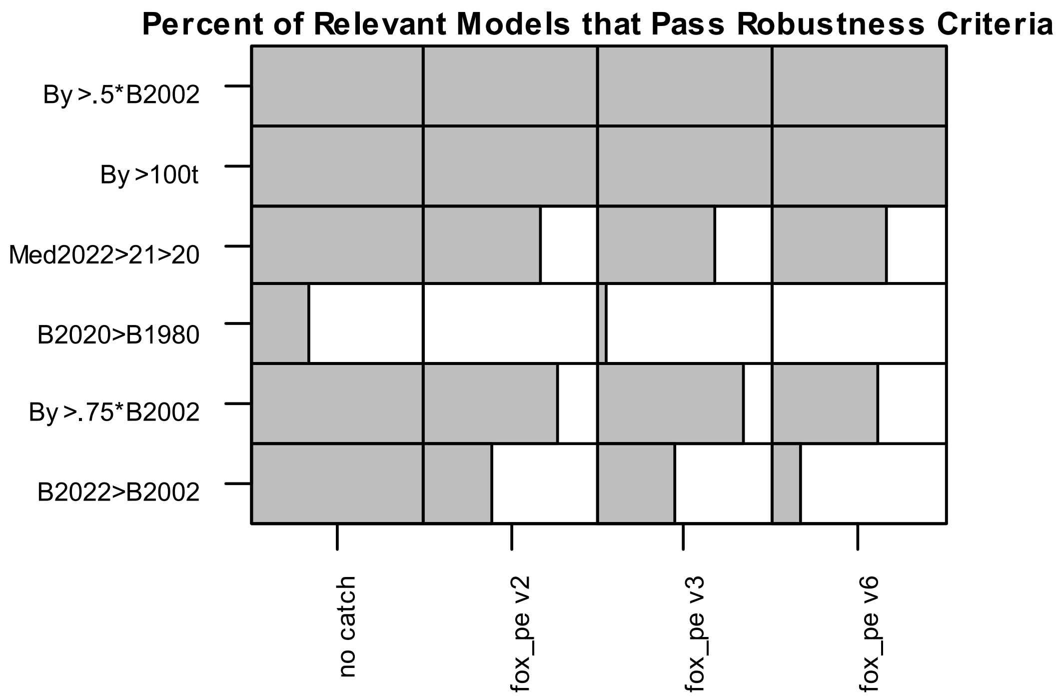
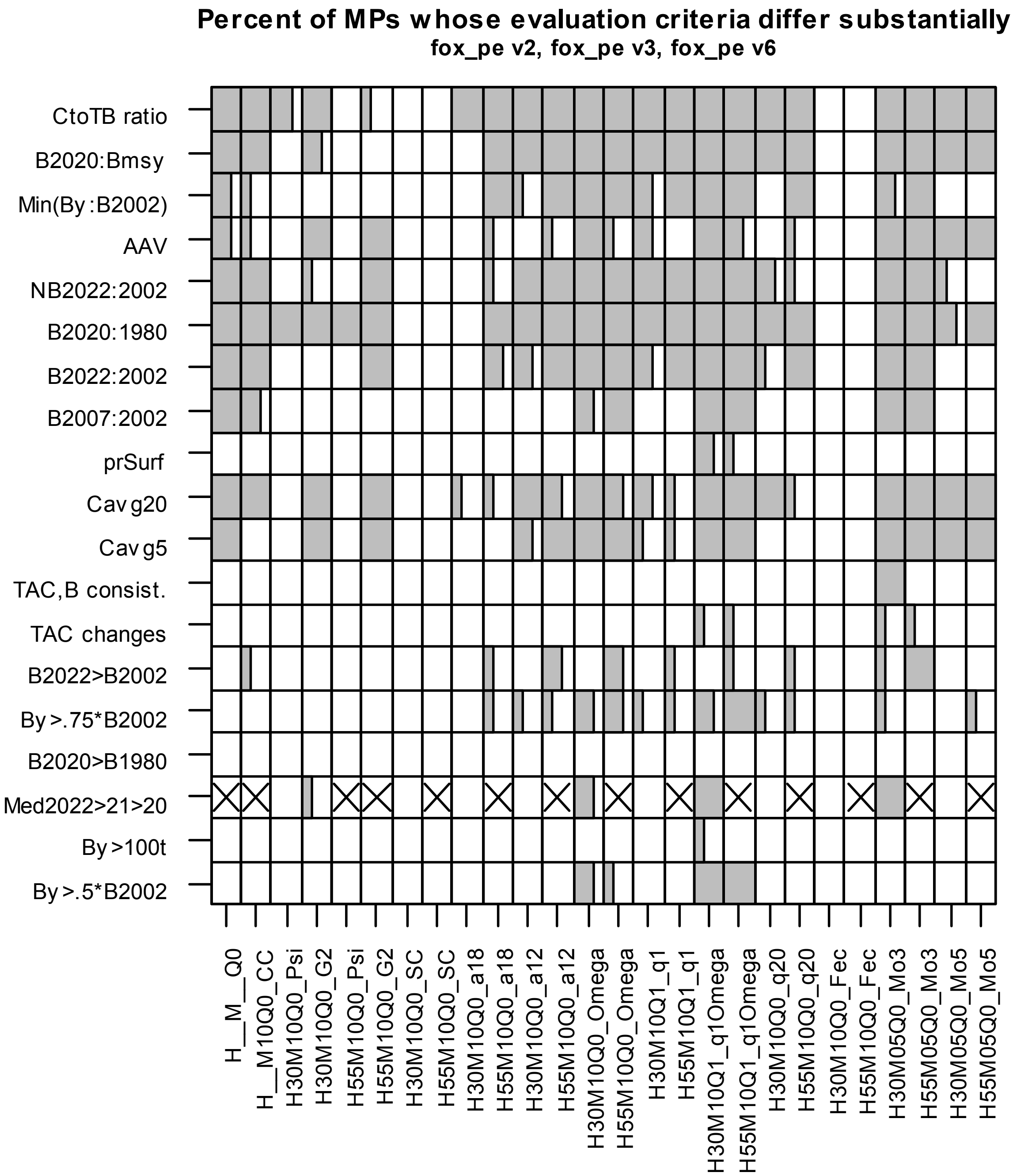


Figure Fox_pe 3. Robustness tests for 3 versions of 'Fox_pe' under hierarchy 3.



A6. Fox_cpue

A6.1. Description of rule

A6.1.1. Overview

The TAC is adjusted from the previous year using the trend in the log(nominal CPUE) over the last *n* years in conjunction with the estimated *r* parameter from fitting a Fox surplus production model to the nominal LL1 CPUE. The TAC is increased when the trend in the CPUE is positive, and it is decreased when the trend in the CPUE is negative. The amount which it is increased or decreased depends (in part) on the estimated *r* value, with the assumption that *r* is positively related to productivity of the stock. In particular, when the slope is positive we allow for greater increases in the TAC as *r* increases, but when the slope is negative we impose greater decreases in the TAC as *r* decreases. The exact details are given in the mathematical description below.

A6.1.2. Mathematical description

Let λ_n be the slope of the regression of log(CPUE) vs. time over the last *n* years.

$$TAC_{y+1} = \alpha TAC_y \left(1 + \beta \lambda_n\right)^{\delta},$$

$$\delta = \begin{cases} r / r^* & \text{if } \lambda_n > 0 \\ r^* / r & \text{if } \lambda_n \le 0 \end{cases}$$

where n, α, β, and r^* are all tuning parameters of the decision rule. Note that a typical value of *r* is around 0.5, but it can range from around 0.1 up to 0.8, and that the slope value ranges roughly from –0.04 to 0.08.

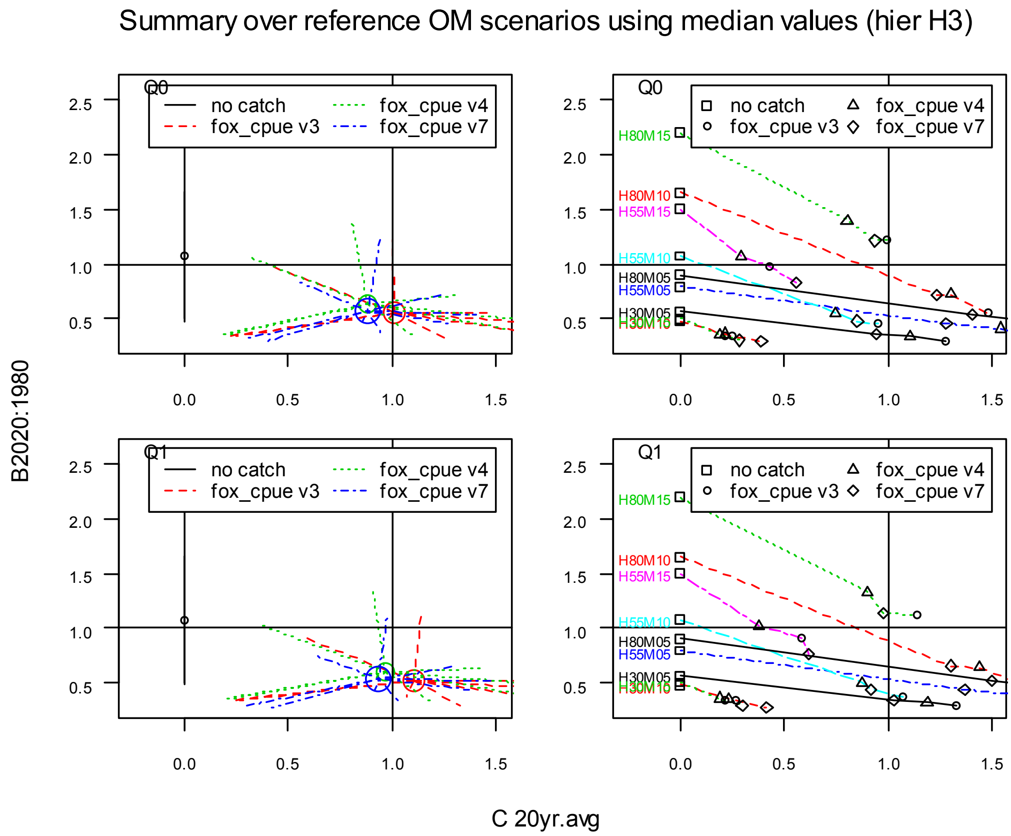
A6.1.3. Versions (tuning parameter values)

Version	n	α	β	r^*
1	5	1.0	10	0.7
2	10	1.0	10	0.7
3	10	0.9	10	0.7
4	10	0.9	10	1.0
5	10	0.9	5	0.7
6	5	0.9	5	0.7
7	5	0.87	5	0.7

A6.2. Performance of rule

'Fox_cpue' is an attempt at combining two rules to take advantage of the positive aspects of both 'CPUE' and 'Fox', the former of which tends to be too aggressive in taking catches and the latter which tends to be too conservative. This attempt at a composite rule appears to have been quite successful. As can be seen in Figure 1, catches are reduced substantially in low productivity scenarios but are quite high in high productivity scenarios. Versions 3, 4 and 7 were considered to give best performance, so only these 3 versions are shown in the figures to avoid clutter. Figure 3 shows the sensitivity of 'Fox_cpue' to the various robustness models that are being explored.

Figure Fox_cpue 1. Performance of 'Fox_cpue' with respect to catch and biomass statistics.



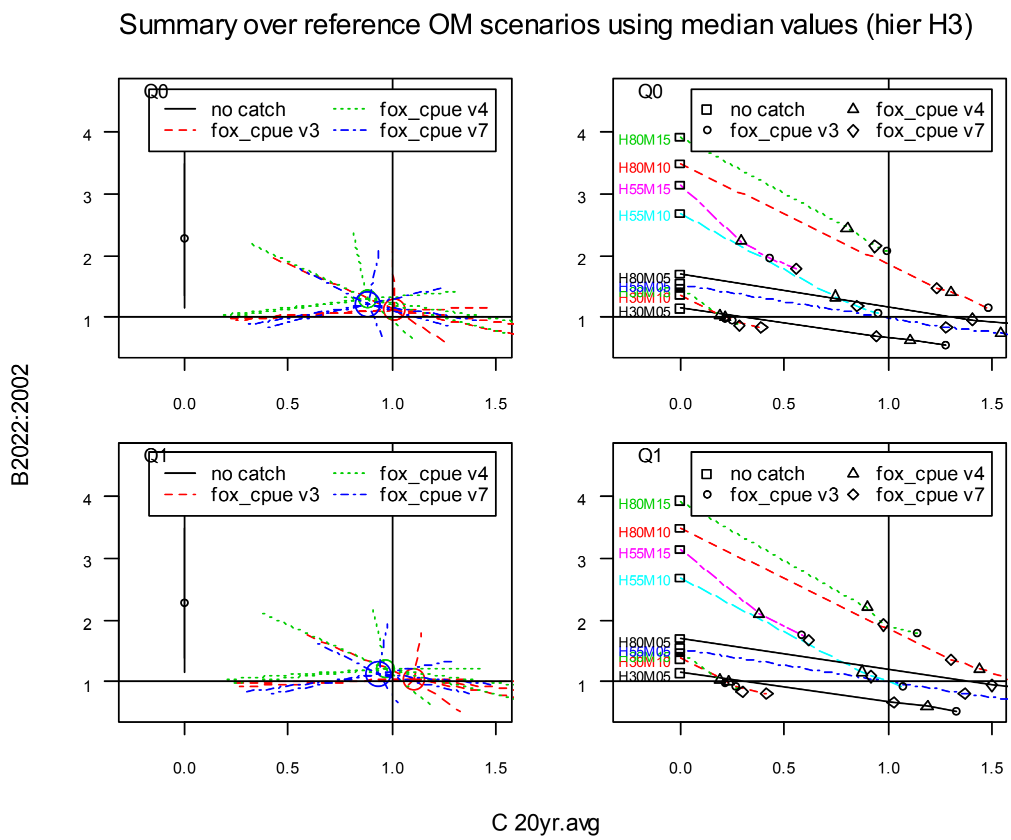


Figure Fox_cpue 2. Performance of 'Fox_cpue' with respect to robustness criteria.

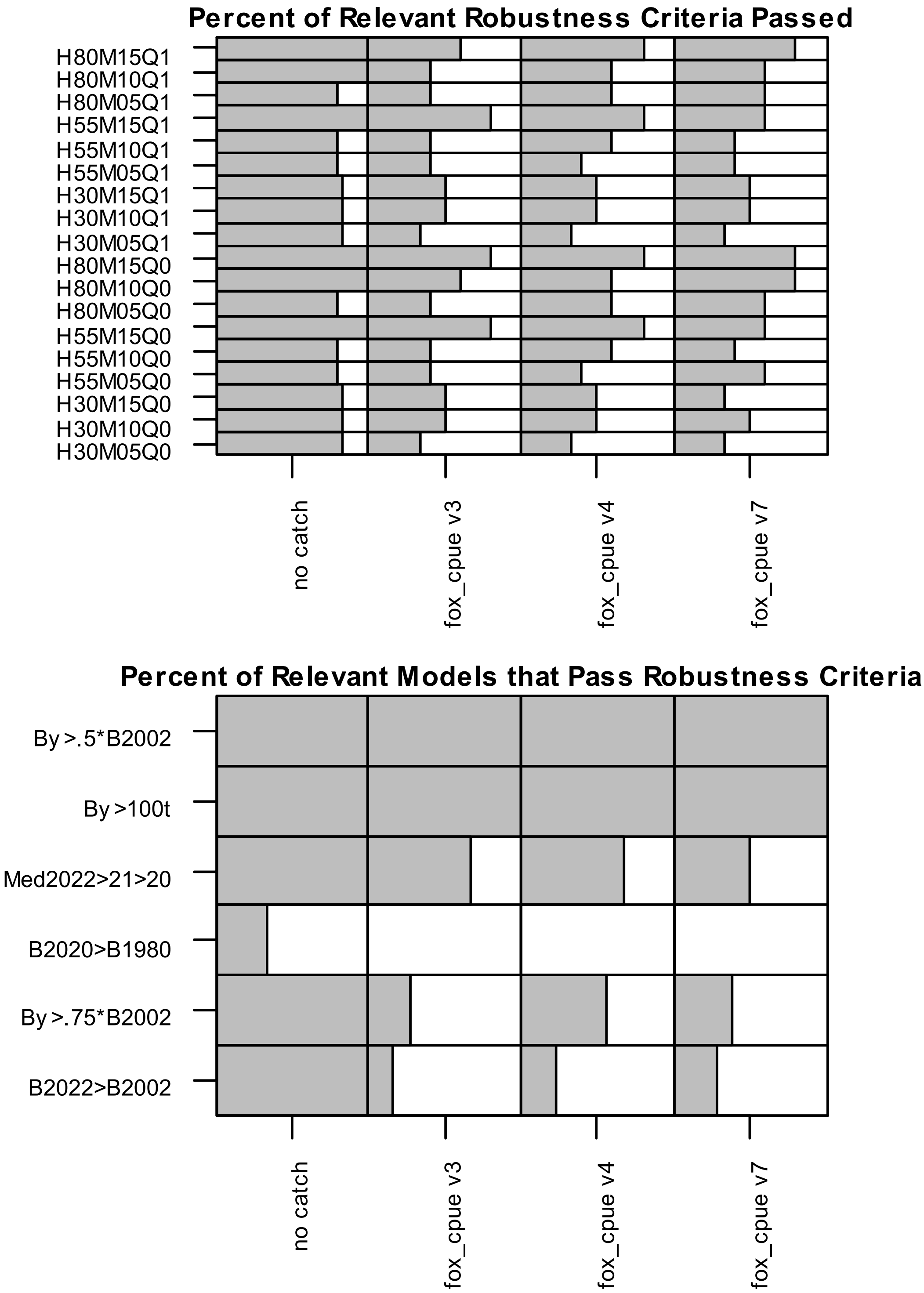
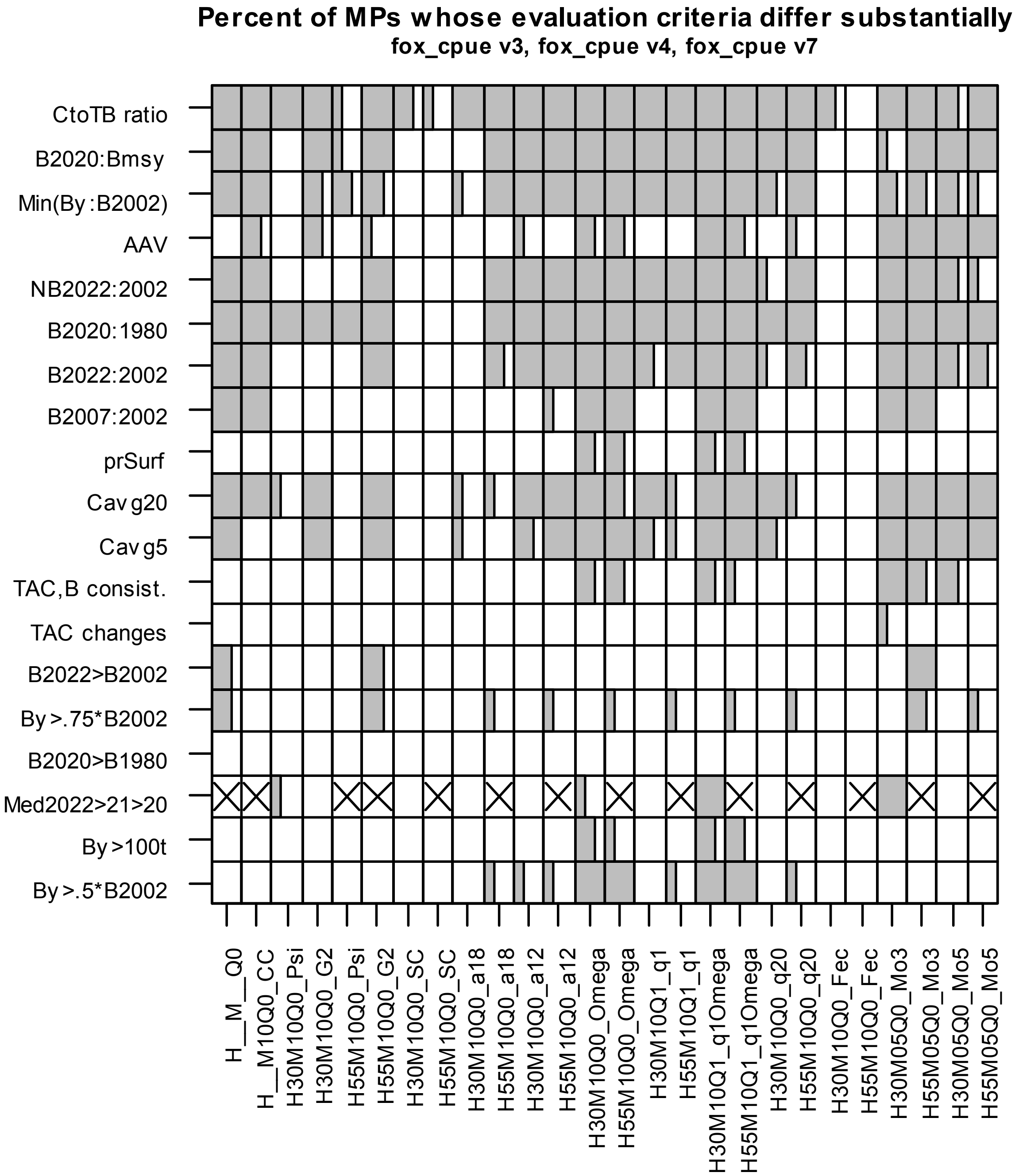


Figure Fox_cpue 3. Robustness tests for 3 versions of 'Fox_cpue' under hierarchy 3.



A7. ACRLRT - Aggregate CPUE with Rebuilding Lag and Historical Rebuilding Target (the MP formerly known as “Stinky”).

A7.1. Description of rule

ACRLRT is a very simple rule that uses age-aggregated longline CPUE to aim for a target biomass based on historical CPUE (some point between a minimum of current level and maximum of 1980 level), and consists of the following features:

- ACRLRT assumes that mass-based age-aggregated CPUE is a reasonable relative abundance index to benchmark historical biomass targets (eg 1980 and 2002) and also describe future biomass on the same scale.

- the rule may contain up to three distinct periods with different behaviour: 1) an initial period of constant TAC, 2) an intermediate period of deliberate rebuilding in which TAC decreases may be relatively large but TAC increases can only be small unless the 1980 rebuilding objective is achieved, and 3) a final period where TAC changes can be relatively large in either direction. The number of periods and when they are active are determined by control parameters.

- the magnitude of TAC changes is always a fixed proportion of current TAC (constrained to the maximum agreed TAC change).

A7.2. Performance of rule

The resulting performance of several versions of ACRLRT (Table ACRLRT-1) applied to the SBT operating models shows the typical trade-off between risk and yield (Fig. ACRLRT-1). The MP does exhibit some effectiveness in more aggressively harvesting the productive scenarios, and reducing catches in the unproductive scenarios as indicated by the “flattened starburst”. However, there is obvious room for improvement in the average performance, in that every version either substantially exceeds the 1980 target in some scenarios and/or results in 2022 biomass below current levels in other scenarios. However, until performance objectives are better defined, it is not clear what will be gained by further exploration.

A7.3. C++ code for ACRLRT

The ACRLRT rule is represented with the following simple and fairly self-documented C++ code:

```
TACTmp=quota(current_yr-1);  //by default set TAC to last value

       //hold TAC constant in initial phase
if(current_yr<firstRebuildYear) TACTmp=TACInitial;

B2002Index =       (merged_biomass_CPUE(2000)
                 + merged_biomass_CPUE(2001)
                 + merged_biomass_CPUE(2002))/3;

BCurrentIndex =    (merged_biomass_CPUE(current_yr-2)
```

```
                    + merged_biomass_CPUE(current_yr-3)
                    + merged_biomass_CPUE(current_yr-4))/3;

  B1980Index=       (merged_biomass_CPUE(1979)
                    + merged_biomass_CPUE(1980)
                    + merged_biomass_CPUE(1981))/3;


  recoveryTarget = B2002Index
                 + EndTarget*(B1980Index-B2002Index);

        // one-sided TAC adjustments in rebuilding phase
if(current_yr>=firstRebuildYear && current_yr <=lastRebuildYear){
   if(BCurrentIndex<B2002Index)         TACTmp=0.6*quota(current_yr-1);
   if(BCurrentIndex>recoveryTarget)     TACTmp=1.2*quota(current_yr-1);
}

        // fairly symmetrical TAC adjustments in final phase
if(current_yr>lastRebuildYear){
  if(BCurrentIndex<0.9*recoveryTarget) TACTmp=0.6*quota(current_yr-1);
  if(BCurrentIndex>1.1*recoveryTarget) TACTmp=1.4*quota(current_yr-1);
}

TAC=TACTmp;
TAC=constrain(TAC, current_yr-1); // constrain change in TAC if req'd

//End
```

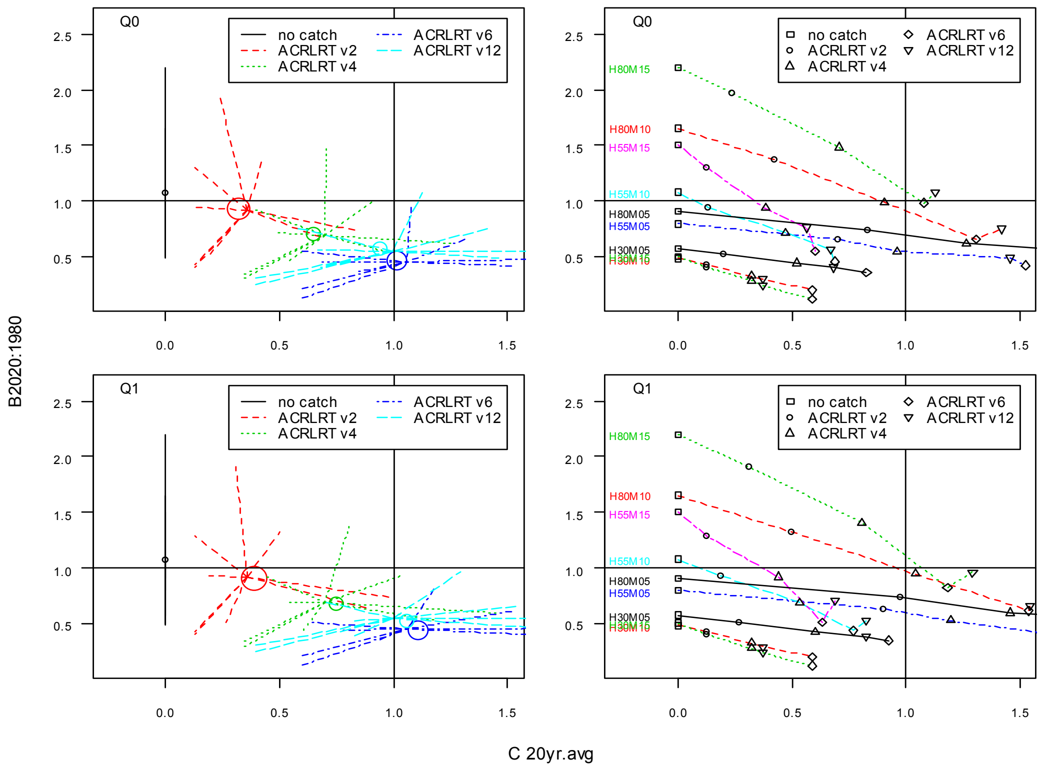
Table ACRLRT 1. Control parameters in MP versions tested:

	TACInitial	firstRebuildYear	lastRebuildYear	endTarget	max TAC change
v1	15385	2000	2000	1	3000
v2*	15385	2000	2000	0.5	3000
v3	15385	2000	2000	0.25	3000
v4* **	15385	2000	2017	1	3000
v5	15385	2000	2017	0.5	3000
v6* **	15385	2000	2017	0.25	3000
v7	15385	2007	2007	1	3000
v8	15385	2007	2007	0.5	3000
v9	15385	2007	2007	0.25	3000
v10	15385	2007	2017	1	3000
v11	15385	2007	2017	0.5	3000
v12*	15385	2007	2017	0.25	3000

*summary results plotted for Hierarchy 3 in Figs. ACRLRT-1,2
** robustness “tick test”results in Fig. ACRLRT-3

Fig. ACRLRT-1. Representative MP performance summary.

Summary over reference OM scenarios using median values (hier H3)



Summary over reference OM scenarios using median values (hier H3)

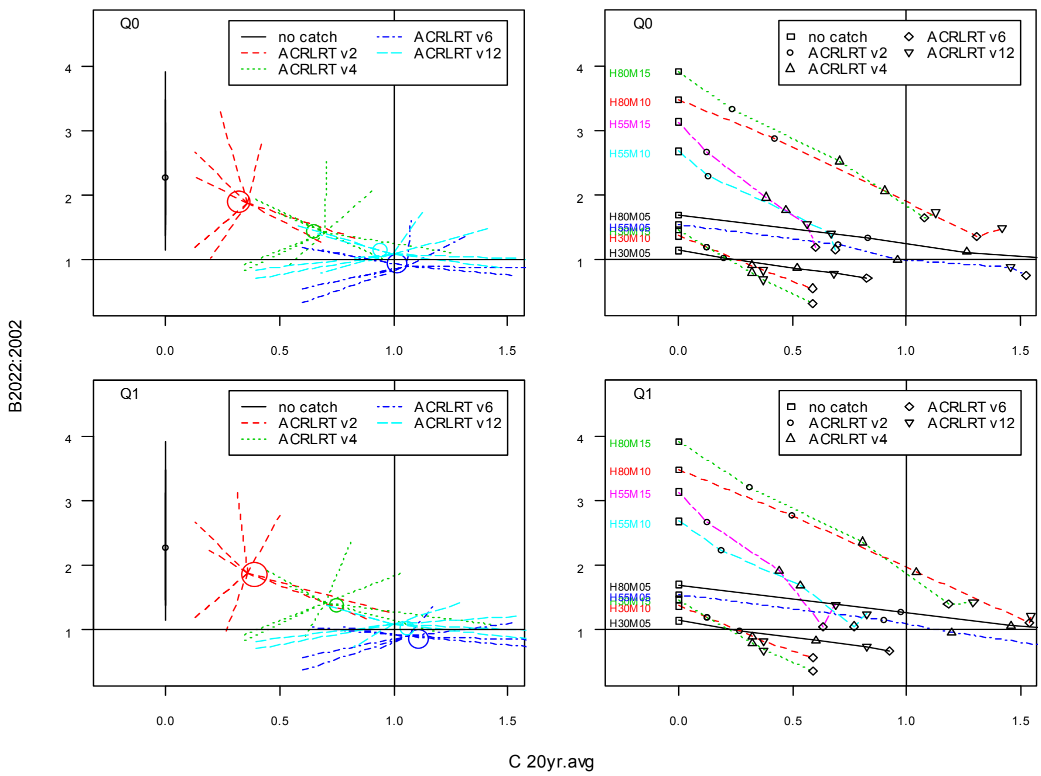


Fig. ACRLRT-2. Representative MP performance summary against robustness criteria.

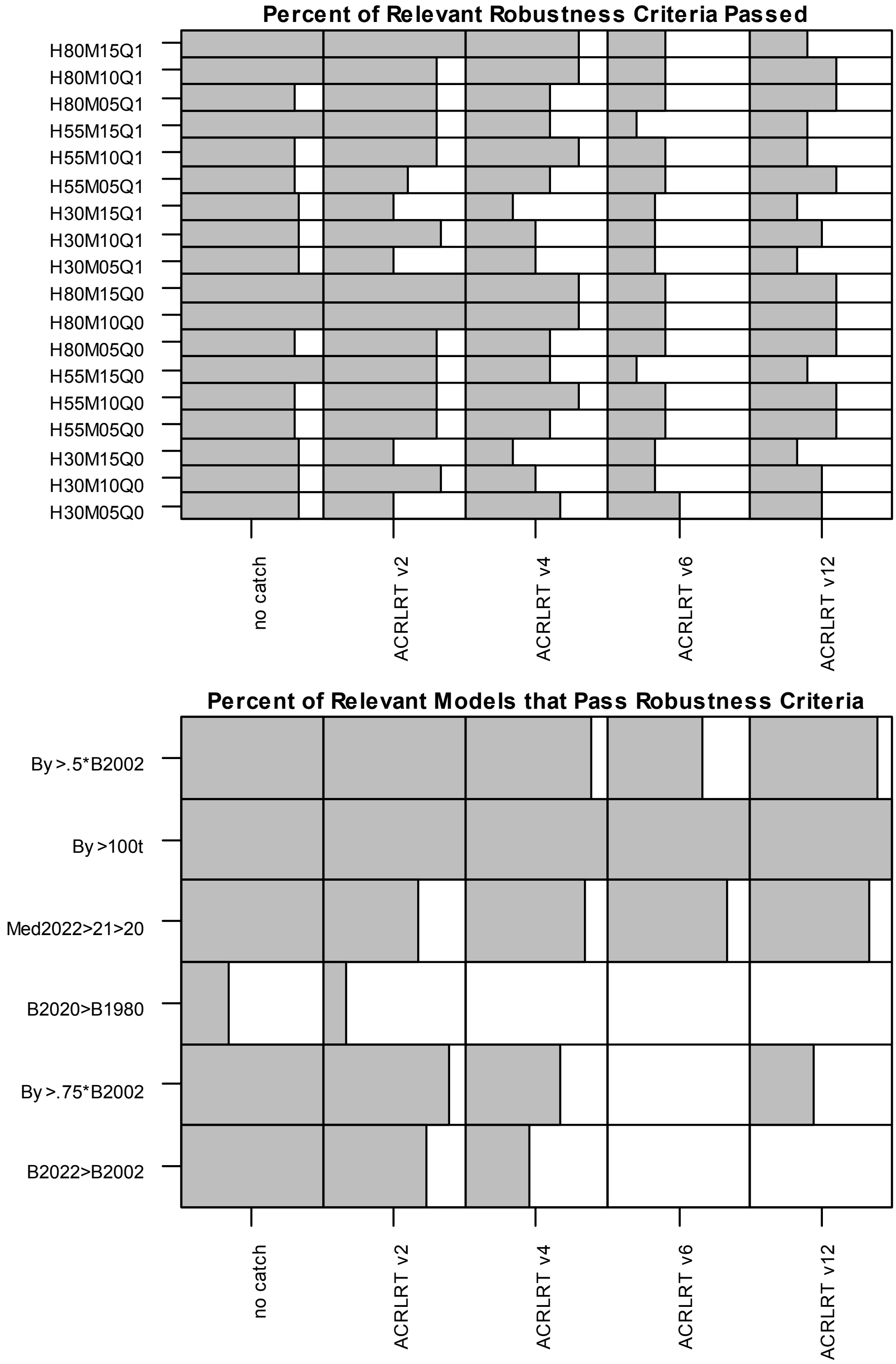
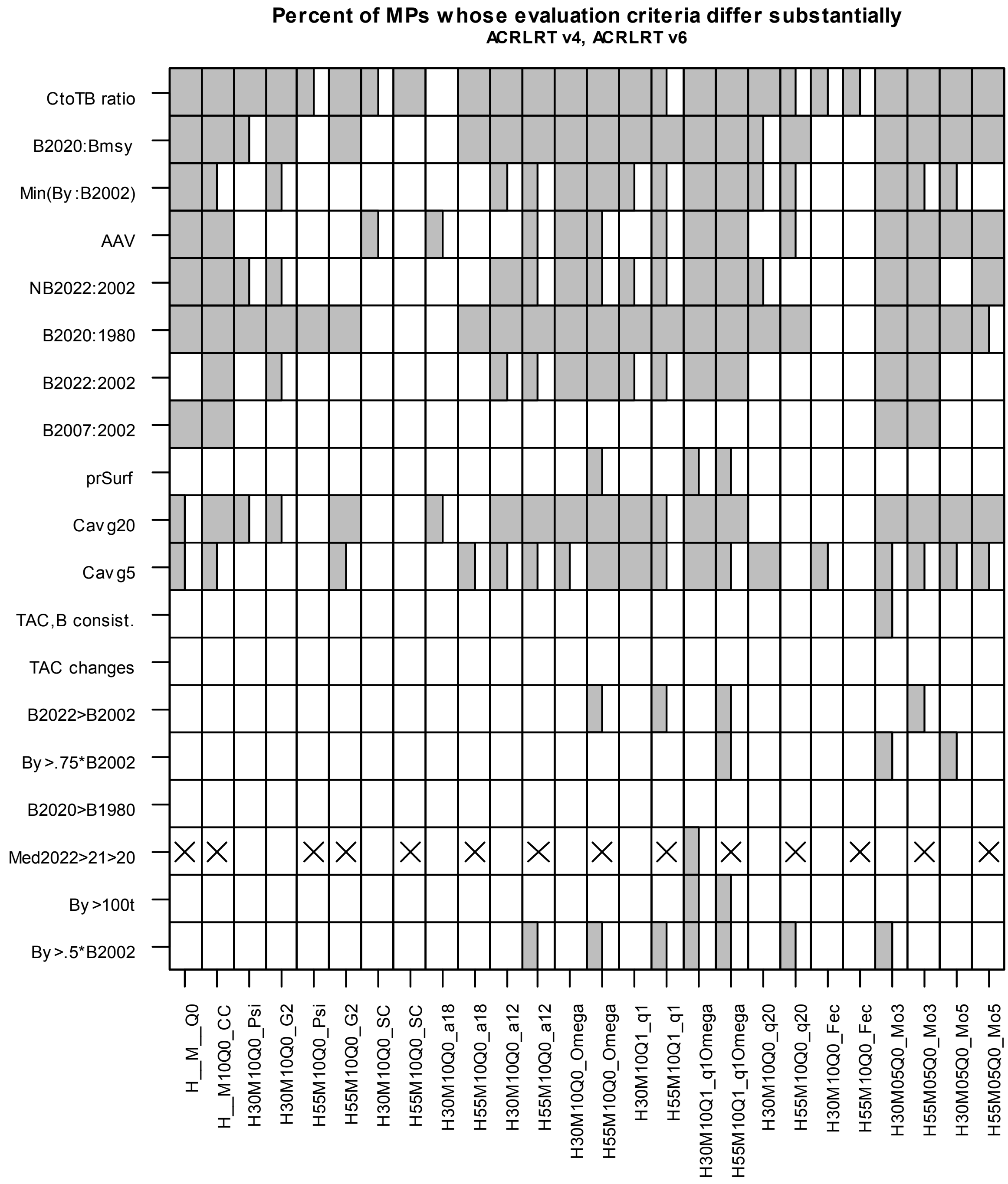


Fig. ACRLRT-3. Summary of representative performance of MP against robustness trials ("tick tests") used to refine selection of operating models.



A8. ASCURE - Age-Structured CPUE with Unabashed Reverse Engineering.

A8.1. Description of rule

ASCURE is similar to the MP ACRLRT, with two major differences: 1) longline catch-at-age data are used along with CPUE such that TAC decisions can consider the age-structure, and 2) MSY and depletion are estimated from early CPUE observations, under the assumption that the range of operating models are effectively assessments that span a range of uncertainty with a range of predictions about future abundance and age structure that will allow CPUE to rapidly distinguish scenarios. A number of potentially important considerations are raised in this latter point, relevant not only to ASCURE, but all MPs (and/or the operating model conditioning process). Key points in the MP include:

- the rule may contain up to three distinct periods with different behaviour: 1) an initial period of constant TAC, 2) an intermediate period of deliberate rebuilding in which quota cuts may be relatively large but increases can only be small unless the final rebuilding objective is estimated to be achieved, and 3) a final period where TAC changes can be relatively large in either direction.

- TAC changes are restricted to discrete increments related to the maximum agreed TAC change.

- mass-based CPUE is partitioned into approximate age classes: 1) ages 4-5 (recruitment), 2) ages 6-9 (immature), 3) and 10+ (approximate SSB)

- There may be CPUE-based rebuilding targets for recruitment and/or spawning biomass relative to 2002, 1980 or years immediately preceding the current year

- MSY and stock depletion (B2002/B0) are estimated from mean CPUE indices over the period 2003-2005. This involves reverse engineering in the sense that known operating model characteristics were used to define the regression relationship used in the estimation.

- In the first couple years, a constant baseline TAC is potentially incremented or decremented depending on current CPUE and historical CPUE-based reference points. Beginning in 2007, the baseline TAC is set as an additive linear combination of estimated MSY and estimated B2002/B0, and then potentially incremented or decremented depending on current CPUE and historical CPUE-based reference points as in the preceding years.

ASCURE estimates MSY and B2002/B0 (the dependent variables) using linear regression of the known Maximum Posterior Density values provided with (and inferred from) the documentation distributed with the sbtProj4 MP code. The original intent was to try using an artificial neural network for making inferences about stock productivity, but the initial exploration suggested that this offered no obvious advantage over linear regression as tested. The independent variables consisted of 3 averages:

CPUE(2003-2005, recruitment),
CPUE(2003-2005, immature),
CPUE(2003-2005, approximate SSB),

taken from the 18 baseline deterministic OM scenarios (Hierarchy 1) at three constant TAC levels, and 196 random samples from the stochastic OM scenarios (Hierarchy 3). The simple relationships adopted are not the result of an exhaustive analysis (other combinations were nearly equivalent, including different combinations of CPUE-based predictors and the mean TAC if longer time series were used). Based on these 250 "training data" we found that B2002/B0 could be predicted rather well (R-squared ~0.65; P<<0.01) and MSY somewhat less well (R-squared = 0.35; P<<0.01). The actual MPD values ranged from 9300-29000 t for MSY and 0.13 – 0.34 for B2002/B0. The joint residuals for the training data are shown in the following figure:

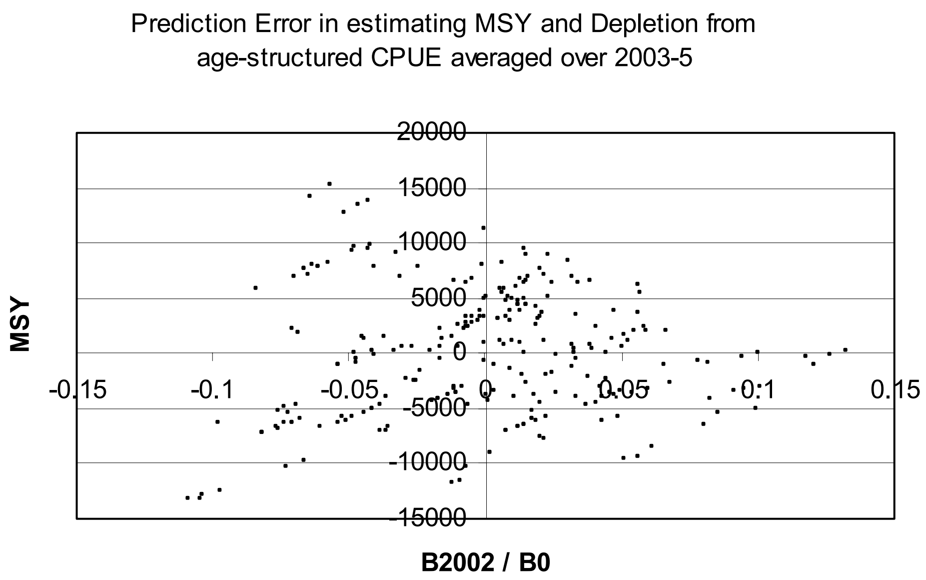


Fig. Estimation error from the regression model estimating SBT MSY and 2002 stock depletion using linear regression of simulated CPUE on known operating model values.

ASCURE has many features that could probably be improved if management objectives could be defined (and the whole concept was deemed acceptable). The control parameter combinations tested are listed in Table ASCURE-1.

A8.2. Performance of rule

ASCURE shows a certain degree of constructive feedback control (Fig. ASCURE-1) as demonstrated by the reasonably "squashed starburst", but it is not clear that knowing MSY and depletion with the current level of reliability actually adds to the effectiveness of the MP, as other MPs probably perform at least as well. This may be because ASCURE is not using the information very effectively, or perhaps the other candidate MPs have managed to extract similarly distinguishing information out of the data with or without explicitly realizing it. Without clarification of management objectives, we are not sure what will be gained by further testing. But ASCURE raises a number of issues for consideration:

1) We refer to the estimation of MSY and B2002/B0 as "reverse engineering", and recognize that this violates the spirit of Management Procedures because the actual operating model values (used in the regression) should never be known. However, we question whether it is actually substantially different than the implicit reverse engineering involved with probably all MPs. The impetus for exploring this direction comes from the observation that Butterworth and Mori (CCSBT MP/03/04/12) seemed to be able to discriminate productive from unproductive operating models within a very few years using age-aggregated production models (presumably other candidate MPs were also able to do this, but authors did not explicitly illustrate the fact). We thought it might be worth trying to directly find out where the discriminating information was coming from, rather than designing an MP that may or may not be making effective use of this information by chance. Since the operating model is essentially a stock assessment model that is designed to provide plausible descriptions of the status of the SBT stock, the dynamics encompassed by the operating models should provide a reasonable portrayal of the uncertainty in the real stock dynamics. If we can combine our prior perceptions of the stock status with the near-future CPUE to rapidly distinguish the stock productivity, this should help us to design an effective MP. But if the ASCURE approach is unacceptable – what would be acceptable? Would it be acceptable to apply a similar stock assessment model to the historical data, map out the key uncertainties, and calibrate the relationship between CPUE and MSY (or other useful management quantities) to that one instead? Would it be appropriate to use the explicit calibration to find the discriminating information in the simulated data, develop an MP that makes an equivalent discrimination of scenarios without actually relying on the explicit relationships (and then perhaps pretend that the explicit calibration never existed). Presumably it would be completely acceptable to come up with the latter MP by chance, but maybe by explicitly admitting what we are attempting to do, it might short circuit a lot of trial and error, or put us down a path that we would not have considered otherwise.

2) Perhaps the biggest concern that ASCURE has brought to our attention is the fact that we do seem to be able to extract a lot of information from very little data. As discussed at the April 2003 MP Workshop, this is surprising given that many years of stock assessments have had so much trouble distinguishing the stock productivity. It has been argued that this is in part due to the uninformative "one-way trip" nature of the stock trajectory in the past. However, some of us think it more likely that the nature of the stock-recruitment and abundance-CPUE relationships are unrealistically informative in the operating models. Again, this is a concern for all the MPs, not just ASCURE.

3) One of the problems with reverse engineering is that an MP may perform well for the operating models with which it was designed, but perform poorly in intermediate or alternative scenarios. This effect was observed with ASCURE to some degree, in that attempts to restrict the number of scenarios to include

in the training data, could result in poor estimation of MSY and depletion (eg if data from high and low steepness scenarios were included, the intermediate steepness cases were well estimated, however, if data from only high and low M were included, the intermediate M scenarios were rather poorly estimated). However, the inability to perform well on intermediate scenarios is potentially a problem with any candidate MP, and measures should be taken to minimize this risk.

A8.3. C++ code for ASCURE

The decision rule is defined by the following fairly self-documented C++ code:

```
TACTmp=quota(current_yr-1);  //by default

  //in initial years, risk averse MSY and depletion are assumed
if(  (current_yr<2005)  ){
  B2002oB0Hat = 0.13;      //depletion estimate
  MSYHat      = 13000;     //MSY estimate

}

    //beginning in 2007, use regression estimators for MSY and depletion
if(current_yr >= 2007){
    B2002oB0Hat = 0.062079104
                  + 5191.062548*(CPUE_by_age(2003,1)
                         + CPUE_by_age(2004,1)
                         + CPUE_by_age(2005,1))/3
                  - 4864.969951*(CPUE_by_age(2003,2)
                         + CPUE_by_age(2004,2)
                         + CPUE_by_age(2005,2)) /3
                   + 23979.42646*(CPUE_by_age(2003,3)
                         + CPUE_by_age(2004,3)
                         + CPUE_by_age(2005,3)) /3;

    MSYHat        = 12238.39205
                  + 1426534794*(CPUE_by_age(2003,1)
                         + CPUE_by_age(2004,1)
                         + CPUE_by_age(2005,1)) /3
                  -127119979.8*(CPUE_by_age(2003,2)
                         + CPUE_by_age(2004,2)
                         + CPUE_by_age(2005,2)) /3
                   -1797454673*(CPUE_by_age(2003,3)
                         + CPUE_by_age(2004,3)
                         + CPUE_by_age(2005,3)) /3;
}

  //initial TAC constant
if(current_yr<firstRebuildYear) TACTmp=TACInitial;

  //subsequently, baseline TAC is function of MSY and initial depletion
  //UREProp and InitialProp are suer-defined control parameters
else TACTmp = UREProp*MSYHat + initialProp*B2002oB0Hat;


      RecCurrent             =             0.25*(CPUE_by_age(current_yr-2,1)+
CPUE_by_age(current_yr-3,1)+                     CPUE_by_age(current_yr-4,1)+
CPUE_by_age(current_yr-5,1)  );  //current age 4-5 index
      SSBCurrent             =             0.25*(CPUE_by_age(current_yr-2,3)+
CPUE_by_age(current_yr-3,3)+                     CPUE_by_age(current_yr-4,3)+
CPUE_by_age(current_yr-5,3)  );  //current SSB index
      NBCurrent                  =         0.25*(CPUE_by_age(current_yr-2,2)+
CPUE_by_age(current_yr-3,2)+                     CPUE_by_age(current_yr-4,2)+
CPUE_by_age(current_yr-5,2)  );  //current immature index
```

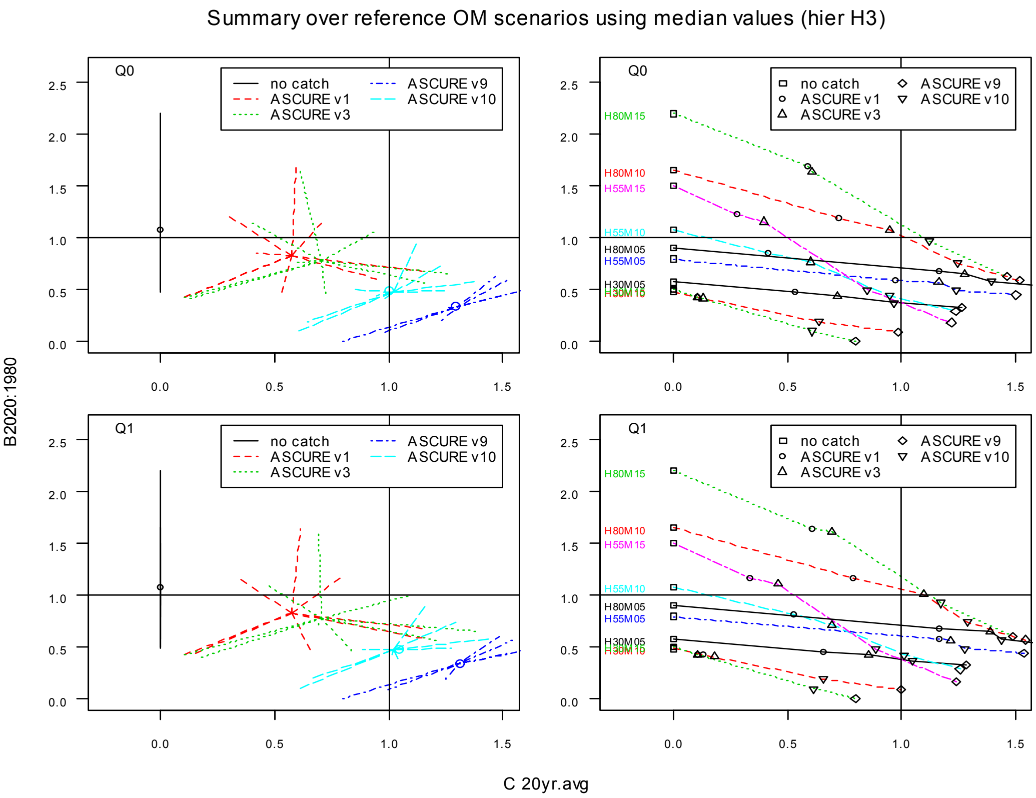
```
      NBRecent                     =       0.25*(CPUE_by_age(current_yr-5,2)+
CPUE_by_age(current_yr-6,2)+                    CPUE_by_age(current_yr-7,2)+
CPUE_by_age(current_yr-8,2)  );
      Rec1980            =   0.25*(CPUE_by_age(1978,1)+   CPUE_by_age(1979,1)+
CPUE_by_age(1980,1)+ CPUE_by_age(1981,1)  );
      SSB1980            =   0.25*(CPUE_by_age(1978,3)+   CPUE_by_age(1979,3)+
CPUE_by_age(1980,3)+ CPUE_by_age(1981,3)  );

      RecInit            =   0.33*(CPUE_by_age(1993,1)+   CPUE_by_age(1994,1)+
CPUE_by_age(1995,1)  );
      SSB2000            =   0.33*(CPUE_by_age(2000,3)+   CPUE_by_age(1999,3)+
CPUE_by_age(1998,3)  );

      // SSB recovery objective between B2002 and B1980
SSBTarget=SSB2000 + politicalEndTarget*(SSB1980 - SSB2000);
      //   Recruitment   recovery   objective   between   2002   and   1980
RecTarget=Rec1980 + politicalEndTarget*(Rec1980 - RecInit);


      double quotaUnit=maxChange/3;

             //rebuilding phase
      if(current_yr>=firstRebuildYear && current_yr <=lastRebuildYear){

        //  CUT quota if recruitment lower than initial
        if (RecCurrent < 0.9*RecInit) TACTmp=TACTmp -quotaUnit;

        //  also cut quota if incoming immatures are dropping
        if (NBCurrent<0.9*NBRecent) TACTmp=TACTmp -quotaUnit;

        //  also cut quota if SSB lower than initial
        if (SSBCurrent<0.9*SSB2000) TACTmp=TACTmp -quotaUnit;


        //   During Rebuilding, Raise TAC more if currentRec and SSB exceed
target levels
          if((RecCurrent   >   RecTarget)   &&(SSBCurrent   >   SSBTarget))
TACTmp=TACTmp +quotaUnit;

        //   During Rebuilding, Raise TAC more if currentRec and SSB exceed
1980 levels

        if(SSBCurrent > 0.9*SSB1980) TACTmp=TACTmp + quotaUnit;
        if(RecCurrent > 0.9*Rec1980) TACTmp=TACTmp + quotaUnit;


      }

       //third phase: aggressively move toward recovery target
      if(current_yr>lastRebuildYear){

        //SSB-Based Recovery Target
        if(SSBCurrent   >   1.1*SSBTarget)   TACTmp=quota(current_yr-1)   +
2*quotaUnit;
        if(SSBCurrent   <   0.9*SSBTarget)   TACTmp=quota(current_yr-1)   -
3*quotaUnit;

        //recruitment-based recovery target
        if(RecCurrent   >   1.1*RecTarget)   TACTmp=quota(current_yr-1)   +
2*quotaUnit;
        if(RecCurrent   <   0.9*RecTarget)   TACTmp=quota(current_yr-1)   -
3*quotaUnit;

      }

  TAC=constrain(TAC);  //apply maximum and minimum TAC change restrictions

//End
```

Table ASCURE-1. Control parameters used in initial MP testing.

	first/last RebuildYear	EndTarget	UREProp / InitialProp	K3	max. TAC change
v1*	2001/1	0.25	0.5 / 1.5	1	3000
v2	2001/1	0.25	1 / 1	1	3000
v3*	2001/1	0	0.5 / 1	1	3000
v4	2007/17	0.25	0.25 / 0.25	1	3000
v5	2007/17	0.25	0.5 / 0.5	1	3000
v6	2007/17	0.25	0.5 / 1	1	3000
v7	2007/25	0.25	0.5 / 0.5	1	3000
v8	2000/25	0.25	0.5 / 0.5	1	3000
v9*	2000/25	0.25	1 / 1	1	3000
v10*	2007/17	0	0.5 / 1.5	1	3000
v11	2007/17	0.25	0. / 2	1	3000
v12	2007/17	0	1 / 0	1	3000

* summary results from these versions are illustrated (Hierarchy 3)

Fig. ASCURE-1. Representative summary performance.



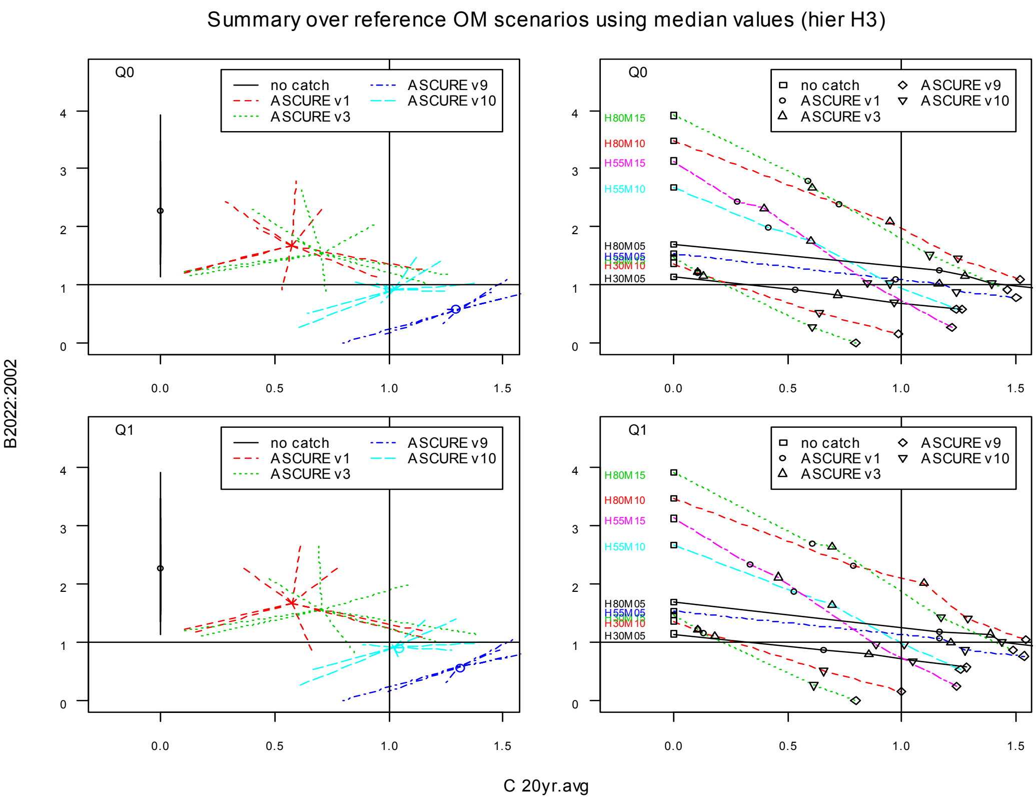
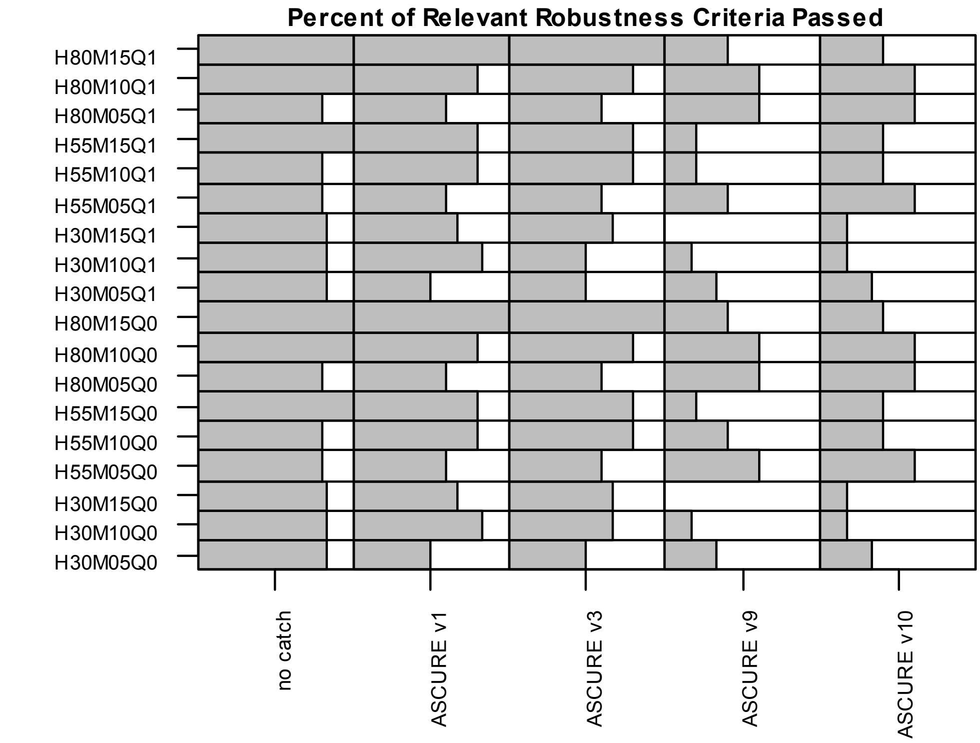


Fig. ASCURE-2. Summary of representative MP performance against robustness criteria (Hierarchy 3).



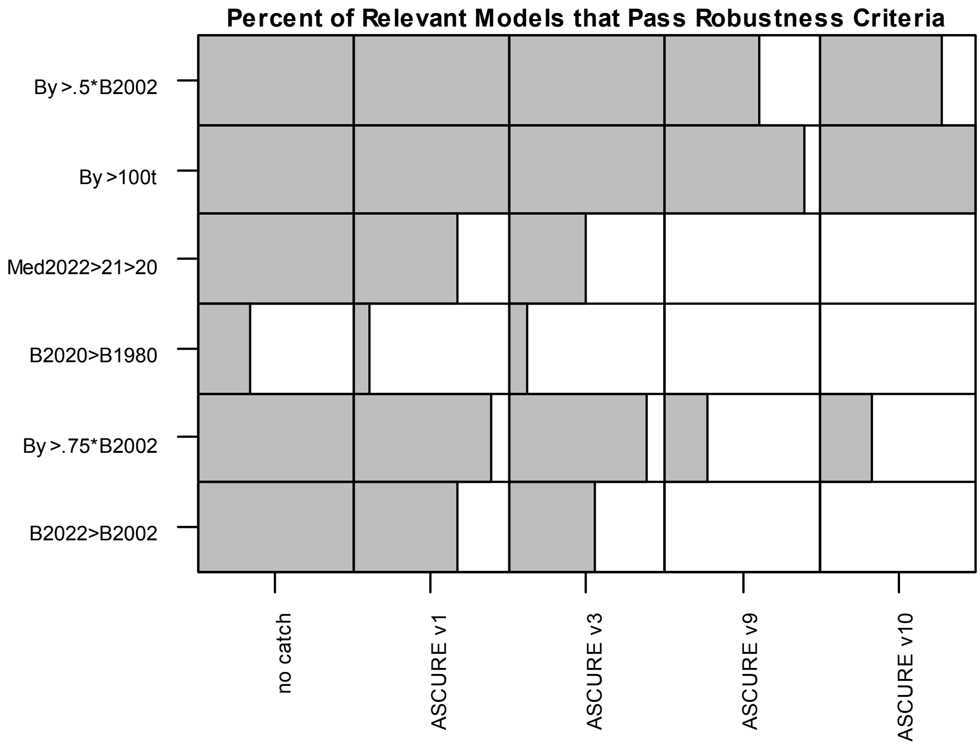
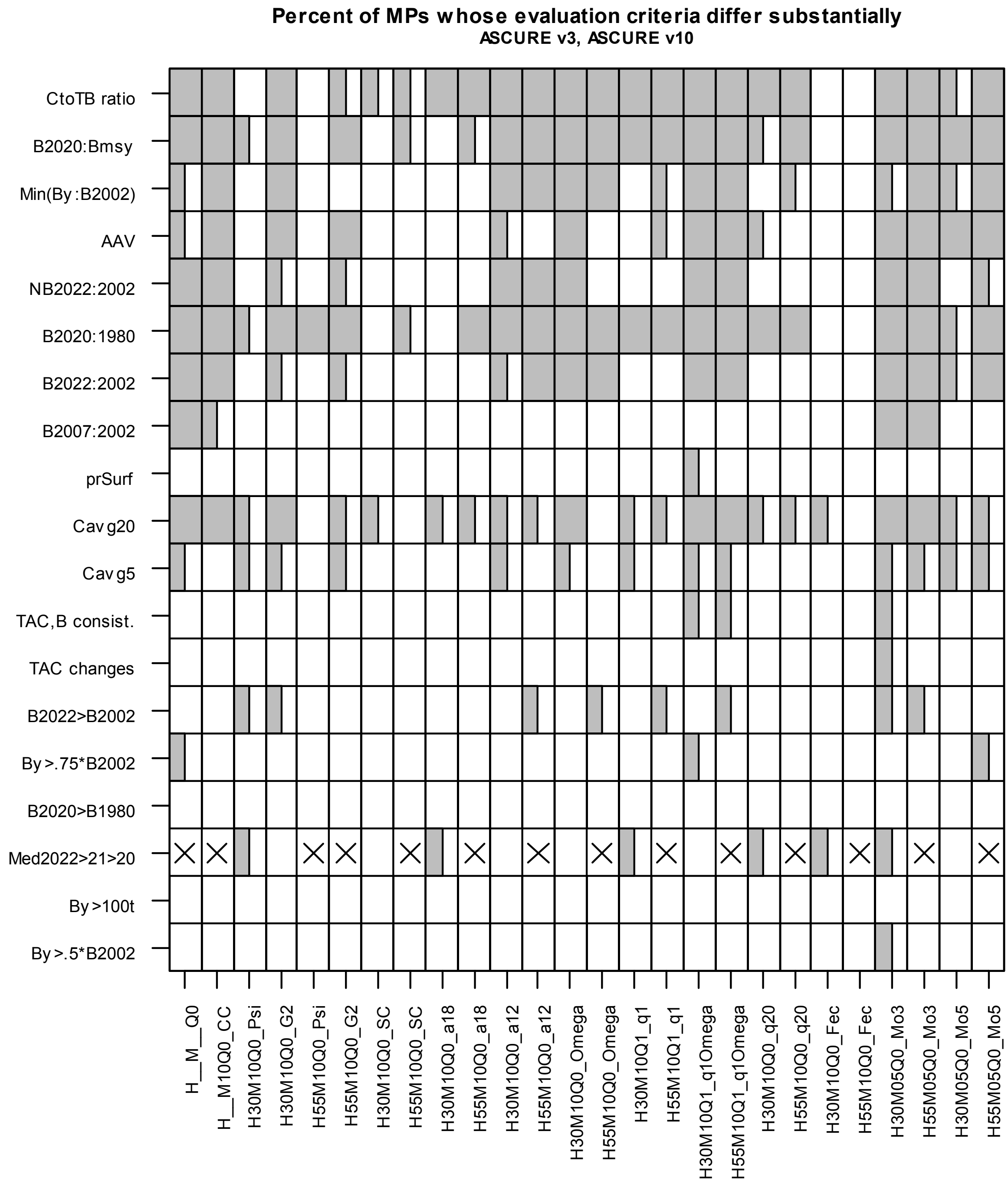


Fig. ASCURE-3. Results of robustness "tick tests" used to assist in the final operating model selection.



A9. Kaltac (based on Kalman Filter 'assessment')

A9.1. Description of rule

A9.1.1. Overview

The 'kaltac' decision rule is identical to that presented at the CCSBT MP workshop in April 2003 (CCSBT MP0304-06 Appendix 10). Since a full description is given in that document, only a brief reminder is provided here. The assessment is based on a Kalman Filter which provides estimates of population numbers. The rule, however, uses the ratio of the current population size to the population size in 1980 that is in fact used by the rule. The rule aims to rebuild the population to the 1980 level by the end of the simulation period. A TAC is determined in such a way that the projected population size, after that TAC has been taken, would lie on a line connecting the population size in the last year of the assessment with the 1980 population level at the end of the simulation period. This is, however, done in terms of the ratios of population size in a given year to that in 1980. Note that the same logic is applied for cases where the current ratio is below 1 and for cases where it is above 1.

The assessment consists of fitting a Kalman filter to CPUE indices. A Kalman filter with 2 states representing 'recruits' (age 5) and 'adults' (ages 6 and older) is fitted to CPUE indices (in numbers). The catches in numbers and assumed survival rates are used to update the 'states' from one time-step to the next. Outputs are estimates of the states (number of recruits and adults) at each time step.

The rule, is designed to continually aim at being at the 1980 biomass at the end of the simulation period. Recall that the target is set in terms of SSB, whereas the Kalman filter works in terms of numbers. The rule makes two adjustments to obtain a proxy for SSB: first, 'adult' numbers are turned into numbers of age 10 and older by using the proportions at age in the input data; second, numbers of age 10+ are turned into biomass by using proportions at age and the length-weight equation to calculate an approximate mean weight of age 10+. Note that the choice of 10+ is arbitrary, and other ages could be tried. The product of the two adjustment factors, is applied to the estimates of adult numbers to obtain a 'proxy' spawning biomass. This is done for the last year in the assessment (current year -2) and for 1980.

The rule calculates the catch required in the current year to put the biomass ratio (in current year +1 to that in 1980) on a line connecting the population ratio at the end of the assessment (current year -2) and 1.0 (the target ratio for 2021).

The unknown TAC is calculated via the following set of equations where A* denotes the 'proxy' spawning biomass described above, where T is the last year in the simulations:

slope=(1-A*t-2/A*1980)/(T-(t-2))

intercept=1-slope.T

X = s.At-1+meanR - (t.slope+interc).A1980.(d1980/dt-2)

In the previous tests, aiming for the 1980 SSB appeared to perform reasonably, but in the new set of scenarios, it was informative also to test a version of the rule which aims for some proportion of the 1980 SSB (see below).

The rule has been implemented with a constraint on the absolute change in TAC allowed from year to year, as agreed at the CCSBT MP workshop in April 2003. A maximum change of 3000t and a minimum change of 100t were used.

A9.1.2. Rule versions (control parameter values)

The key differences between versions lie in assumptions about ratios in catchability between recruit and adults, and ratio between (process error variance) and (observation error variance), and the proportion of SSB1980 being aimed for. For all versions, the TAC is set every year, but for versions 2 and 3, the TAC is constrained to be at least at the level of the current catch (i.e. the TAC is fixed at current catch unless an increase is indicated) for several years (see below). The parameter governing how many years, is called “startfixyrs”.

Versions for kaltac are:

Version	qratio	Vratio	startfixyrs	proportion SSB1980
1	1	1	0	0.75
2	1	1	10	0.75
3	1	1	5	0.75

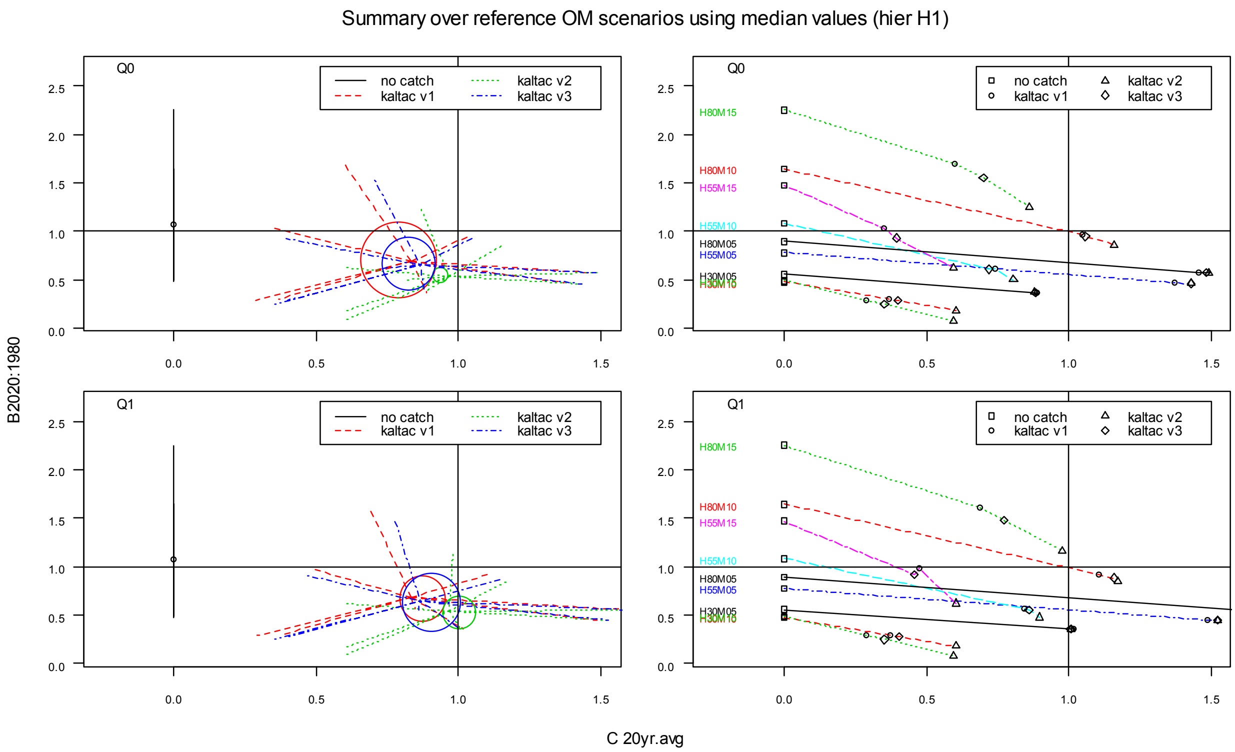
A9.2. Performance of Kaltac rule

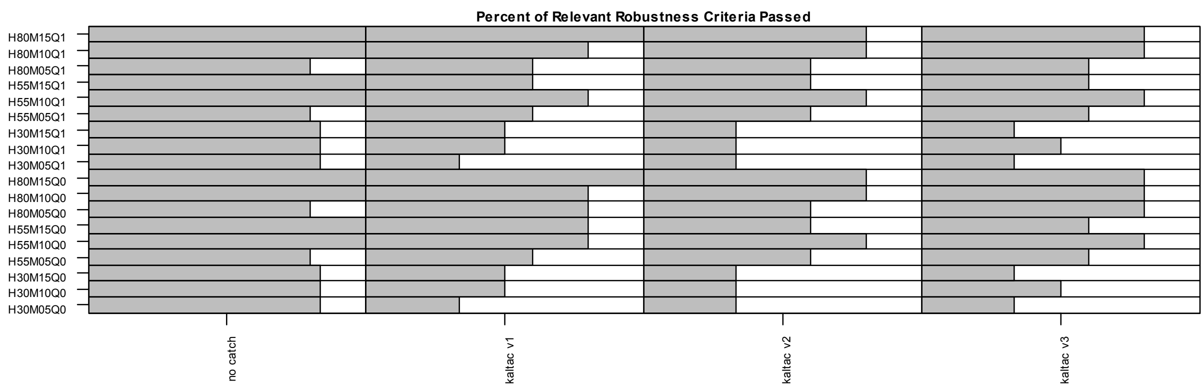
Results are presented in Figures Kaltac 1 to 3. Versions 1 and 3 have large interannual variance in the catch, and this will need some further attention. The rule still shows relatively poor performance for the high productivity scenarios, in the sense that it allows for less catch than it could.

There are several areas where improvements could be made, as noted in CCSBT MP 0304-06, Appendix 10. For example, different definitions of 'recruits' and 'adults', and alternative ways of determining a 'proxy' SSB could be tried.

The rule itself may need to change the 'target' it uses, if substantial changes to management objectives are made.

Figure Kaltac 1. Performance of kaltac; hierarchy H1 (plotted ito B2020:B1980)





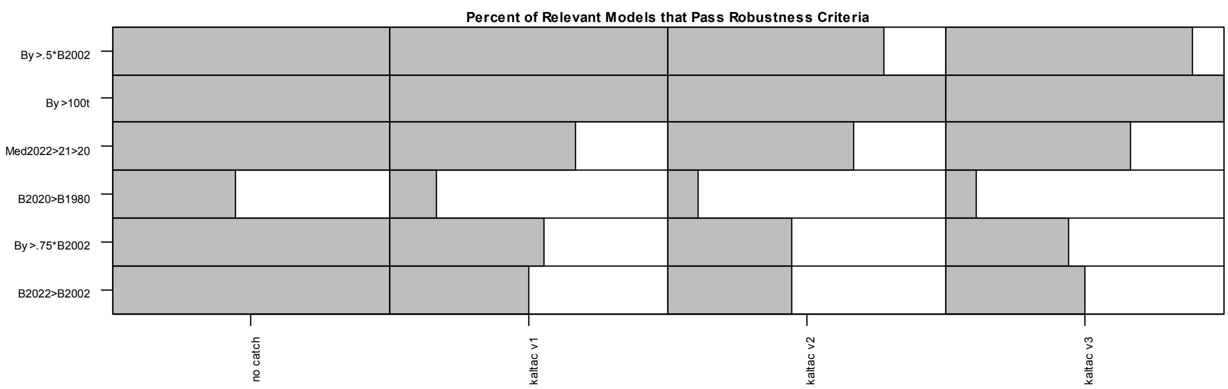


Figure Kaltac 2. Performance of kaltac, hierarchy 1 (plotted ito B2022:B2002)

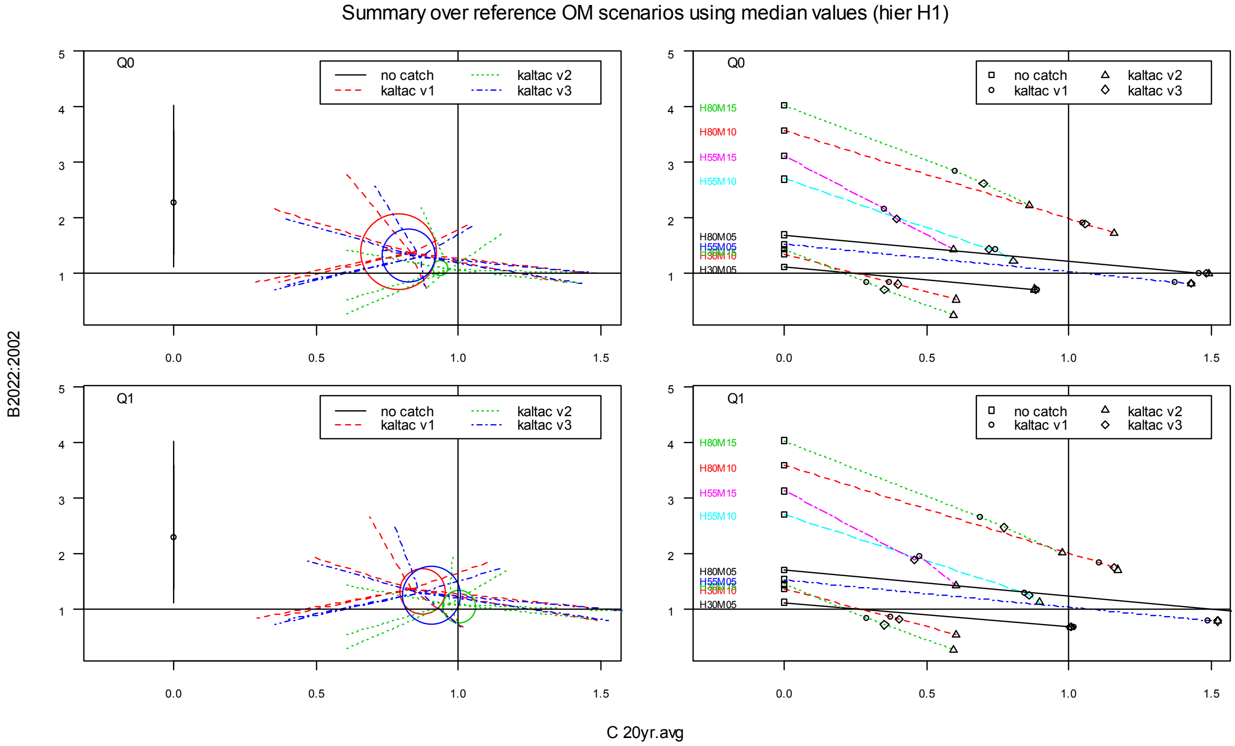
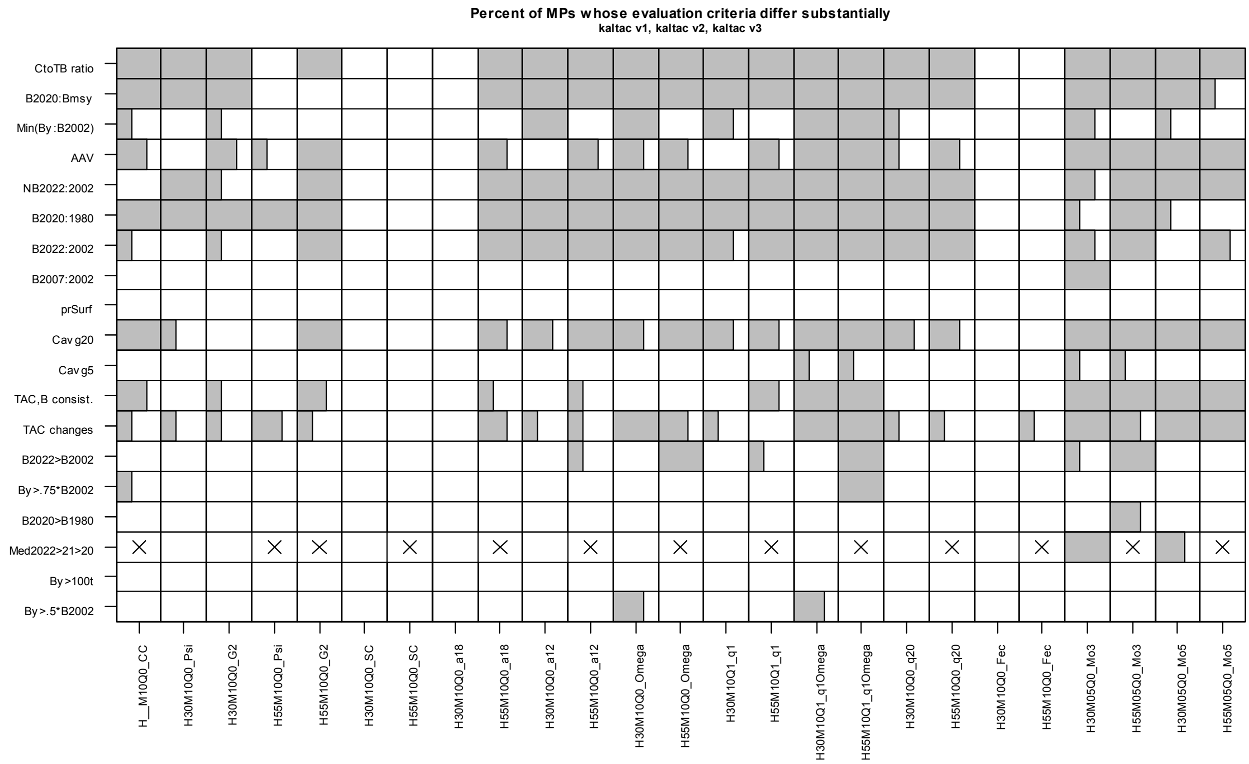


Figure Kaltac 3. Robustness trial evaluation



A10. VPA-SPR (VPA and spawner per recruit)

A10.1. Description of the rules

A10.1.1. Overview and description of assessment

This appendix presents details of two rules based on a VPA assessment and spawner per recruit considerations. First the assessment that underlies both rules is presented and thereafter each rule is described.

The basic assessment underlying both these rules is virtual population analysis (VPA), using the standard catch and population equations. The implementation used here is relatively simple. Only one parameter is estimated, namely the fishing mortality in the last (most recent) year, $F_{A,T}$, and the last true age class, A, (i.e. not the plus group), called the terminal F. The data are catches at age in numbers and longline CPUE at age in numbers, or overall CPUE (for a group of age classes, in numbers). For a given value of 'terminal F' it is possible to back-calculate the population numbers at each age. Other assumptions about fishing mortalities, required for calculation of the full matrix of numbers at age, include:

$$F_{A,t} = \alpha . F_{A-1,t}$$

where A is the oldest true age class, $A+$ is the plus group, t is any year prior to the last year, T is the last (most recent) year (i.e. $t<T$). The constant α can be estimated or treated as control parameters. In all cases presented here, they have been fixed at α=1.

For cohorts which are younger than the last true age class in year T, the fishing mortality is determined by assuming for age $A-1$:

$$F_{A-1,T} = 1/\alpha . F_{A,T}$$

For ages $A-2$ and younger in year T, assume that the relative selectivity at that age, $S_{a,T}$ is the average over J years, where J is treated as a control parameter:

$$S_{a,T} = \frac{1}{J}\sum_{j=1}^{J}\frac{F_{a,T-j}}{F_{A,T-j}}$$

and then let

$$F_{a,T} = S_{a,T} . F_{A,T}$$

Natural mortality is also treated as an input, and in all cases presented here, the following mortality vector was used: 0.4 0.367 0.333 0.3 0.267 0.233 0.2 0.175 0.15 0.125 0.1

Estimation is based on fitting to the age-specific CPUE by assuming that:

$$\ln(U_{a,t}) \sim N(\ln(q_a N_{a,t}), \sigma_a^{\ 2})$$

where

$U_{a,t}$ is the CPUE at age a in year t

q_a is the catchability at age a

$N_{a,t}$ is the numbers in the population at age a in year t

σ_a^2 is the variance for age a

When fitting to overall CPUE, assuming that:

$$\ln(U_t) \sim N(\ln(q\sum_{a=A1}^{A2} N_{a,t}), \sigma^2)$$

where the CPUE now the overall longline cpue in terms of numbers, and the population numbers at age are summed over a specified range of ages, *A1* to *A2* (cases presented here used ages 4-11; other ranges can, of course, be tried). The log-likelihood is easily constructed for the above assumptions. The catchabilities and variances can be determined analytically, so that the problem simplifies to a one dimensional maximisation of the likelihood, with respect to the terminal F. There are several other control parameters for the VPA which are listed in the next section.

After the optimisation, the numbers in the plus group can also be calculated, but there are several different approaches, all with some difficulties. The one implemented here is simple, and can probably be improved upon. First assume $F_{A+,T} = \beta.F_{A,T}$, where β has been fixed at the arbitrary value of 0.2 for runs presented here, but could be treated as a control parameter. (Some attempts at estimating were also made, but this did not prove very successful). The population numbers at age are then back-calculated using the catches in numbers rather than fishing mortalities. When the plus group is an old age (e.g. 30), then poor estimation of the plus group has relatively little effect on outputs compared to a case where the plus group is young, e.g. 15.

Outputs from the VPA are standard, i.e. numbers at age in each year, fishing mortality at age in each year. For a given set of weights at age and maturity at age, spawning biomass in each year can also be calculated.

A10.1.2. Description of 'virtualSPN'

This simple rule is based on spawner per recruit calculations and estimates of fishing mortality from the VPA. First, the SPN under F=0 (SPN0) is calculated. second, an F multiplier is determined such that the estimated F-vector in the last year of the VPA, multiplied by the multiplier would give a target ratio for SPN(F)/SPN0. The target ratio is treated as a control parameter for the rule.

The TAC is determined by forward projection of the population using an average recruitment for age 0 (this has almost no effect on the catch) and the F-multiplier obtained above. Since the most recent years are likely to be least certain and possibly quite noisy, recruitment for the the last 3 years are ignored in the calculation of the average and only the previous 5 years are used. For the same reason, the catch is determined over ages 2 and older. Also, for the current year-1, i.e. where there is a quota but no data, and the current year, the mean weight in the catch is assumed to be similar to that in the current year -2 (i.e. the last year in the assessment). Finally, the TAC is constrained to change by at least 100t and not by more than 3000t, as agreed at CCSBT MP workshop in April 2003.

If the quota were to go to zero in any year or sequence of years, there will be difficulties in running the VPA. Although such circumstances could, in principle, be dealt with in special ways, it has not yet been implemented in this version. Instead, a nominal catch of 100t is set in such a case. This issue will be commented on below under results.

A10.1.3. Versions of VirtualSPN (control parameter values)

The control parameters are:

- plus group age
- time window, i.e. number of years over which to fit the CPUE
- type of time window, i.e. whether the number of years for fitting stays the same or expands as more years of data are added
- number of years for average selectivity
- fit CPUE by age or overall
- ages for which age-specific cpue should be fitted (if age-based; if overall, ** check which ages included in sum
- weighting for each age in the fitting, if age-based cpue is used
- the % spawner per recruit (spn) to aim for

Equal weighting was used for all age-based CPUE fitting

	v1	**v2**	**v3**	**v4**	**v5**	**v6**
plus group age	30	30	30	15	20	30
time window (years)	10	20	30	20	30	30
window type	fix	fix	fix	fix	fix	fix
years for avg. selectivity	8	8	8	8	8	8
fit CPUE	overall	by age	by age	overall	overall	overall
CPUE ages		4-18	4-14			
spn %	30	30	10	30	30	30

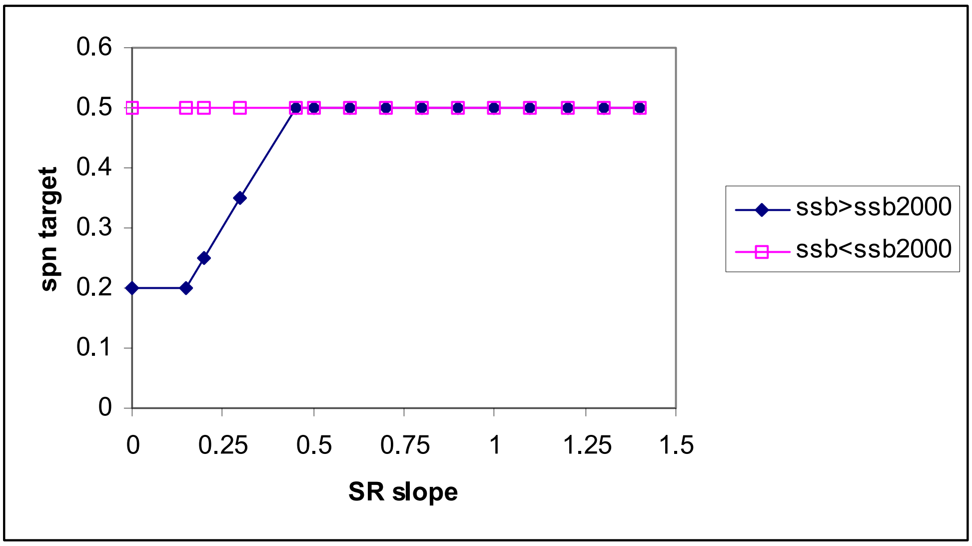
A10.1.4. Description of SPNadapt

The idea behind this rule is to adapt or change the target ratio (SPN(F)/SPN0) on the basis of the relationship between recruitment and SSB. The hope is that if the steepness, h, is high, there should be no or a very low slope in a regression of R on SSB, whereas a low steepness may lead to a relatively high positive slope between R and SSB.

The first step is to calculate the slope of a regression between R and SSB. I have chosen to use recruitment at age 3, versus SSB associated with that cohort (i.e. R(age=3,year=t) versus SSB(year=t-3). For simplicity of interpretation, both quantities are first scaled to the mean over the period considered. The recression is therefor between R/meanR and SSB/meanSSB. The number of years included in this calculation is a control parameter of the rule. Because scaled variables are used, a slope of 1 would go through the point (1,1) i.e. where R=meanR and SSB=meanSSB. A steep slope/high value for slope indicates that R decreases rapidly with decreasing SSB, and this is interpreted as a need for caution about letting SSB drop too low. Therefore, in this case the target SPN ratio is increased, which implies lowering fishing mortality. On the other hand, a low slope suggests little or no decline in R if SSB were to decrease, so the assumption is that the target SPN ratio can be pushed a little lower, implying an increase in fishing mortality. There should, however, be some limit on how low target should be, and arguably also on how high the target needs to be. These limits (lower SPN target and upper SPN target) are control parameters. The type of relationship between the estimated slope (called SRslope) and the SPN target, implemented in SPNadapt is illustrated in the figure below. The two final control parameters define where the change-points in the relationship are (inflection points in terms of the value of the slope: inflect1 and inflect2).

Although the upper and lower target levels (0.2 and 0.5 in the figure above) are control parameters, sensible choices can be informed by notions about acceptable or reasonable ratios of spawner per recruit based on meta-analyses for example.

Figure showing spn target as a function of the value of the SR slope - described in the text



The last control parameter, 'check SSB', is intended to minimise declines in and allow for faster rebuilding of SSB. If 'check SSB' is 'yes' then the SPN target is only adjusted when current SSB is above SSB in 2000, e.g.:

if current SSB > SSB in 2000 then SPN is taken from the dark solid line

if current SSB $\leq$ SSB in 2000 then SPN is taken from the light line (ie set at the upper target)

If 'check SSB' is no, then SPN is adjusted irrespective of the value of SSB, i.e. taken from the dark solid line in the example above.

Once the target SPN has been determined from the above procedure, the same steps are followed to determine an F multiplier, then a TAC and finally to constrain the TAC.

A10.1.5. Versions of SPNadapt (control parameter values)

The first 6 control parameters are identical to thoses for VirtualSPN. The others are particular to SPNadapt, namely the lower and upper SPN target % (e.g. 10-30%), the lower and upper inflection points in terms of the value of the slope, the number of years over which the slope is calculated and whether the target spn is changed only when current SSB > SSB in 2000 (check SSB is 'yes').

Note that Equal weighting has been used for all ages in the cpue-by-age VPA's.

	v1	**v2**	**v3**	**v4**	**v5**	**v6**	**v7**	**v8**	**v9**
plus group age	30	30	30	30	15	30	30	30	30
Time window (years)	30	30	30	30	20	30	30	30	30
window type	fix	fix	fix	fix	fix	fix	fix	fix	fix
years for avg. selectivity	8	8	8	8	8	8	8	8	8
fit CPUE	by age	by age	overall	overall	overall	overall	by age	by age	by age
CPUE ages	4-14	4-10					4-14	4-10	4-20
spn %	10-30	10-30	10-30	10-40	20-40	10-30	10-30	10-30	10-30
inflections	0.25 0.65	0.25 0.65	0.25 0.65	0.15 0.45	0.15 0.45	0.25 0.65	0.25 0.65	0.25 0.65	0.25 0.65
years for slope	30	30	30	30	30	30	30	30	30
Check SSB	no	no	yes	yes	yes	yes	yes	yes	yes

A10.2. Performance of VirtualSPN

Initial sets of runs which explored the sensitivity of performance to the control parameters highlighted the following issues:

a) assessments on age-based CPUE appear to have larger AAV and are less conservative (higher catches, lower biomass for the same target SPN) than ones based on overall cpue

b) the assessment is more stable (i.e. changes in the estimated Fterminal, and hence overall population level, from one year to the next are smaller) when more years are used for averaging selectivities

c) longer time windows appear to perform better in the assessment, though one might anticipate that bigger changes in catchability could lead to better performance of shorter time windows

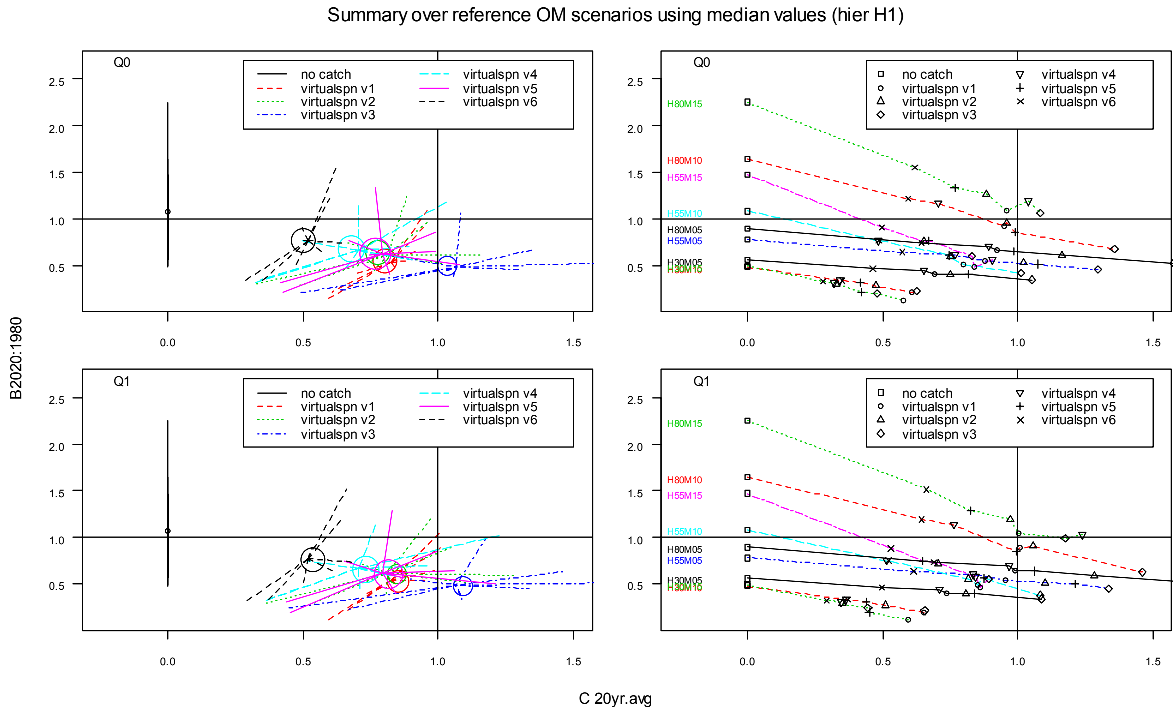
d) an old age for the plus group appears to perform better in the assessment.

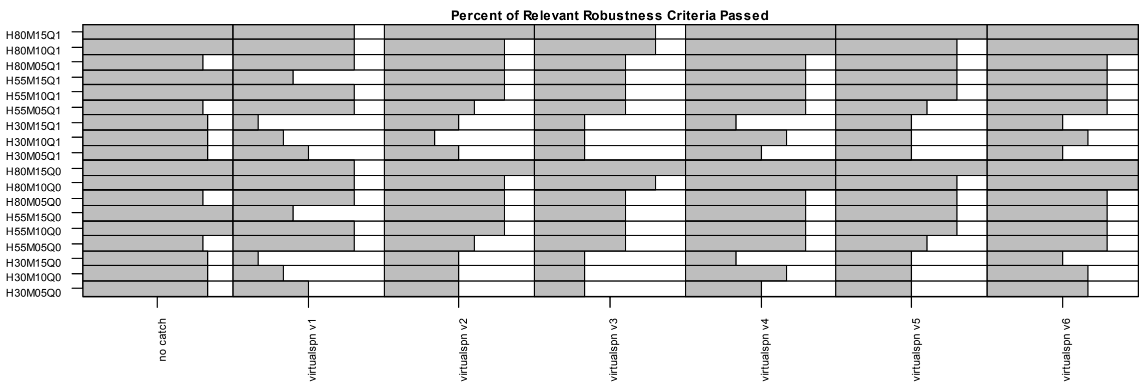
Recall that this approach makes no attempt to 'learn' whether the stock is productive or not, and the performance bears this out. With a conservative target (e.g. 30%), the high productivity scenarios lead to lower than necessary catches and with a less conservative target (e.g. 10%) the extent of biomass rebuilding for the lower productivity scenarios is poor. See figures virtualSPN 1 and 2 below.

A10.3. Performance of SPNadapt

This rule is subject to the same sensitivities in the assessment as virtualSPN. Figures SPNadapt 3 and 4 show that SPNadapt can also be 'tuned' to span a wide range of catch-biomass trade-offs. Figure SPNadapt 5 shows performance of versions 7 and 8 under hierarchy 3. Results of robustness trials are presented in the main text.

Figure virtualSPN 1. Performance of 'virtualSPN', hierarchy H1 (plotted ito B2020:B1980).





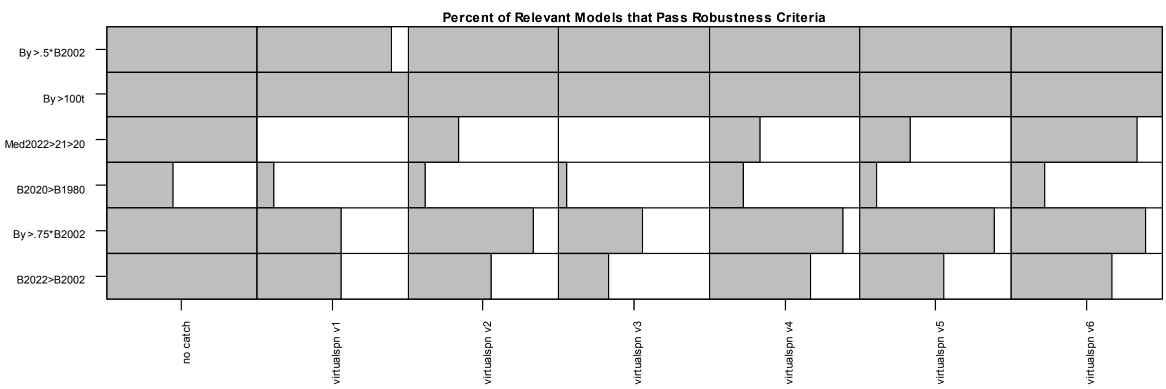


Figure virtualSPN 2. Performance of 'virtualSPN', hierarchy H1 (plotted ito B2022:B2002).

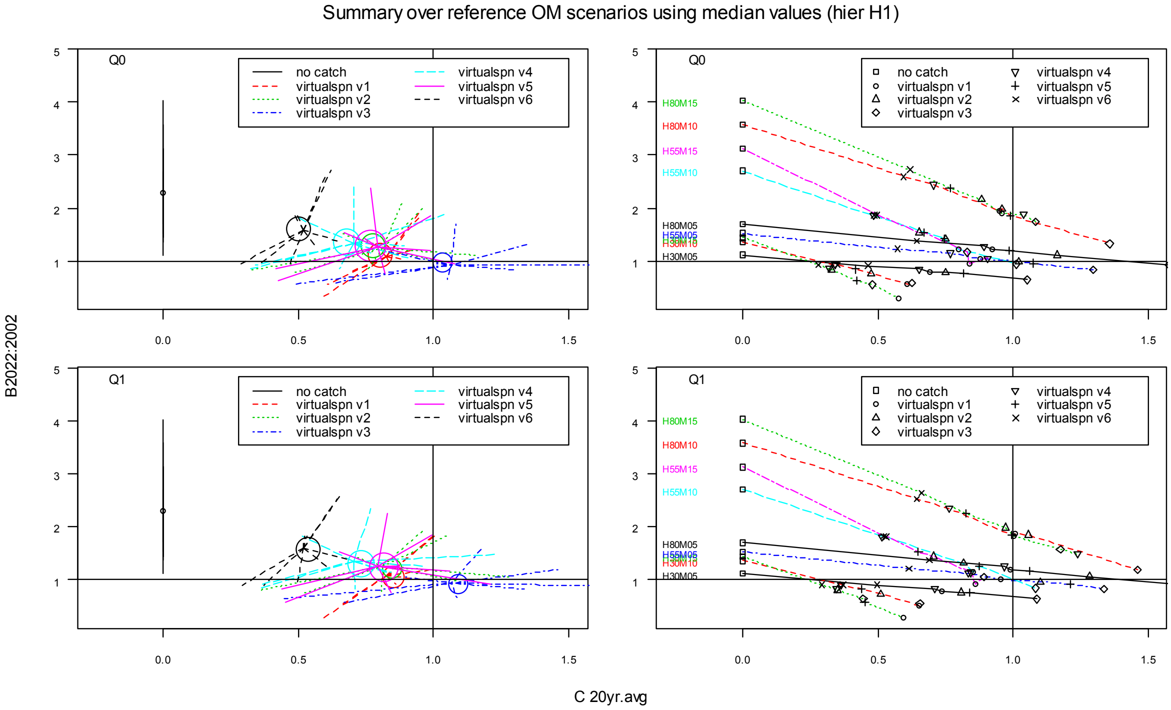


Figure SPNadapt 3. Performance of 'SPNadapt', hierarchy H1. (plotted ito B2020:B1980)

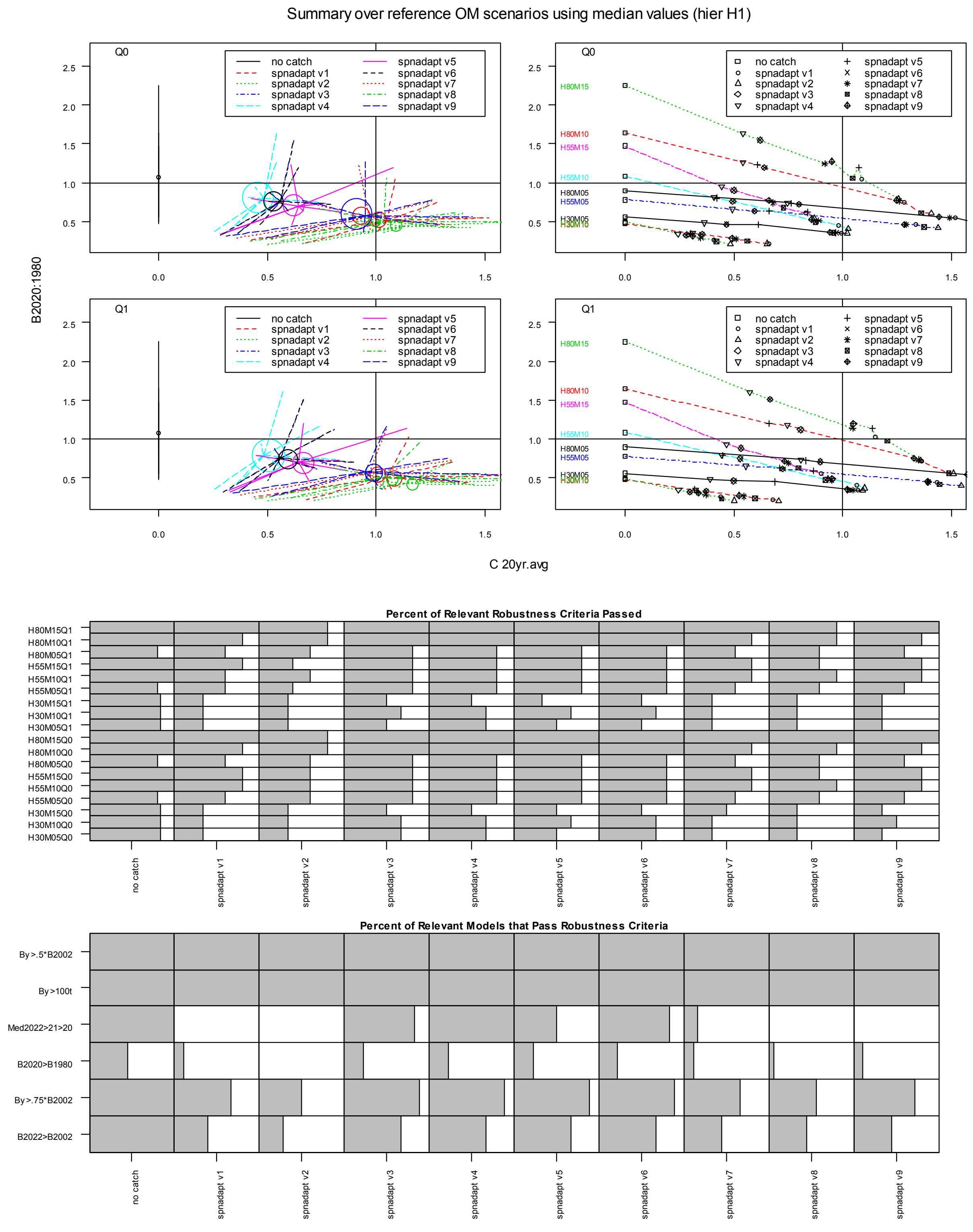


Figure SPNadapt 4. Performance of 'SPNadapt'. hierarchy H1. (plotted ito B2022:B2002)

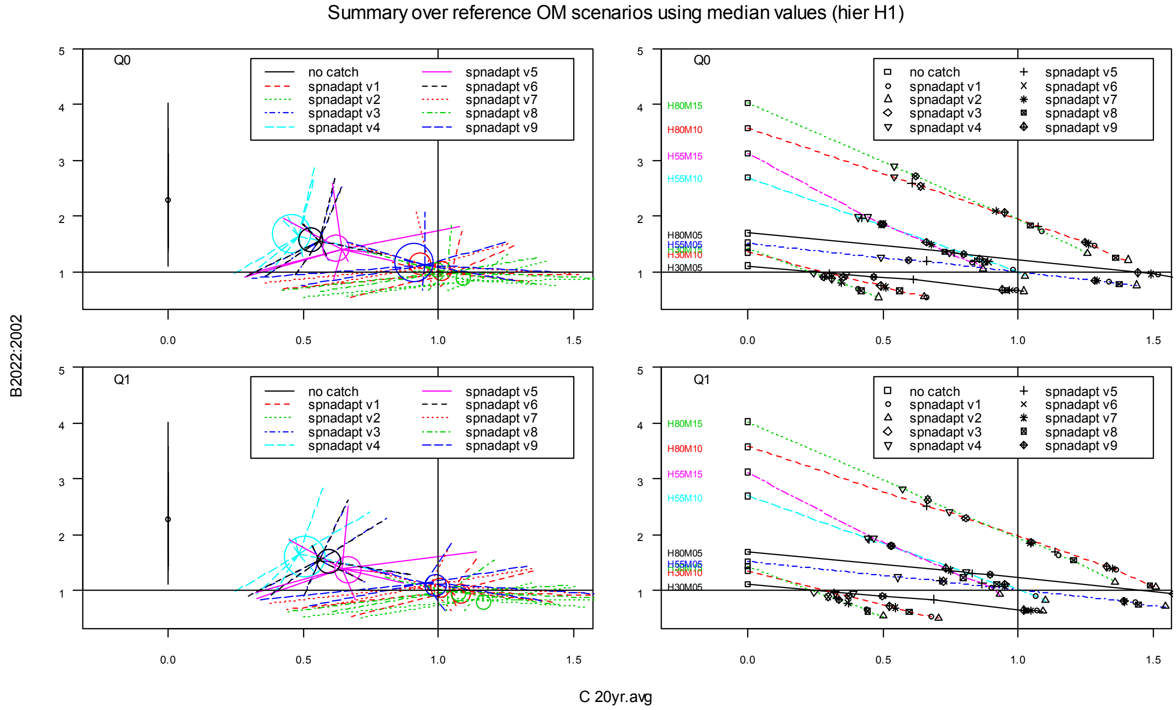


Figure SPNadapt 5. Performance of SPNadapt versions 7 and 8, under hierarchy 3 (plotted ito B2020:B1980)

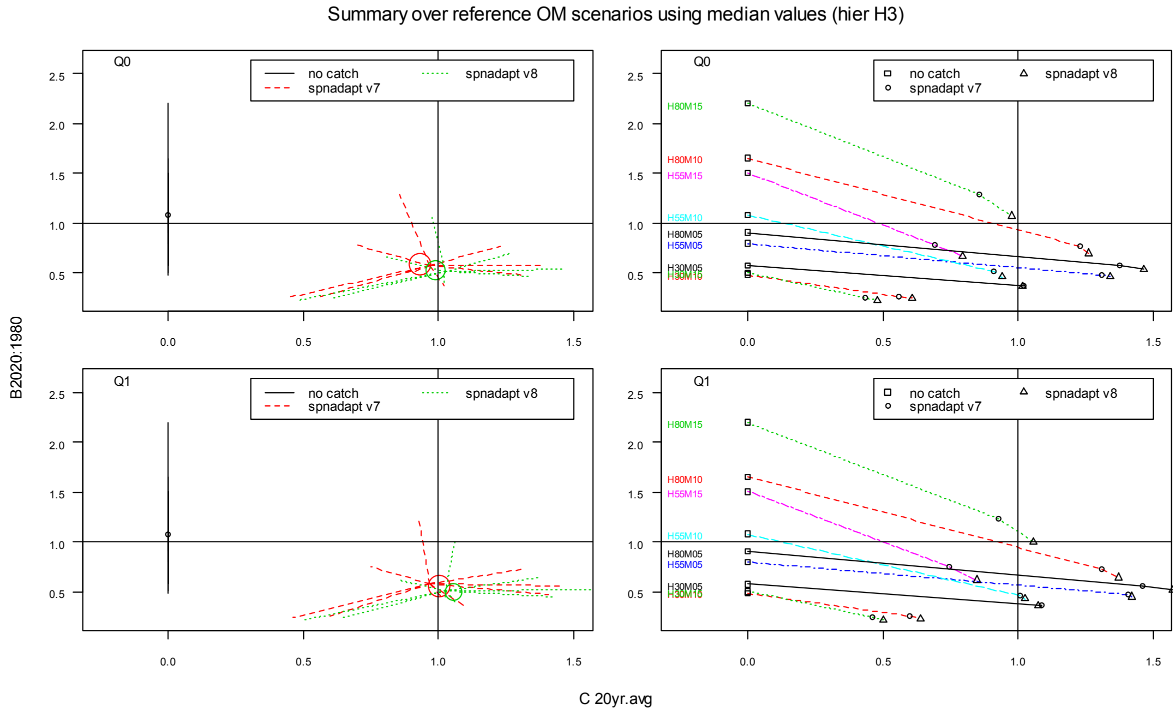


Figure SPNadapt 6. Robustness tests for versions 1 to 9 under hierarchy 1.

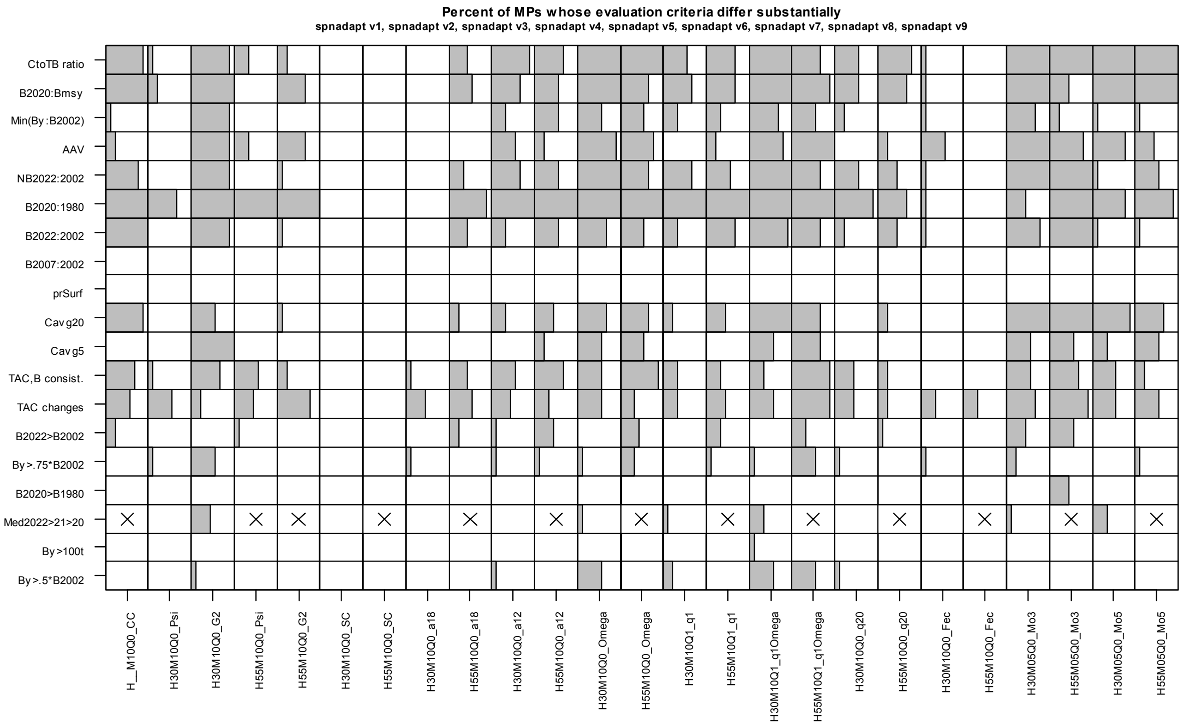


Figure 1. Summary performance of a selection of MP's. On the vertical axis is the biomass performance statistic B2020:B1980. On the horizontal axis is average catch over the 20-year projection period, *where each vertical line represents current catch level (i.e. catch in 2002, which was 15385.7 MT)*. The plotting symbols at the ends of the rays show the median performance of the MP's (over 100 runs). A different symbol is used for each model scenario (as indicated). The centroid of the star, plotted as an open circle, is the average across models, where the radius of the circle is proportional to the average AAV statistic across models. The top panel is for the 9 'Q0' reference operating model scenarios and the bottom panel is for the 9 'Q1' operating model scenarios

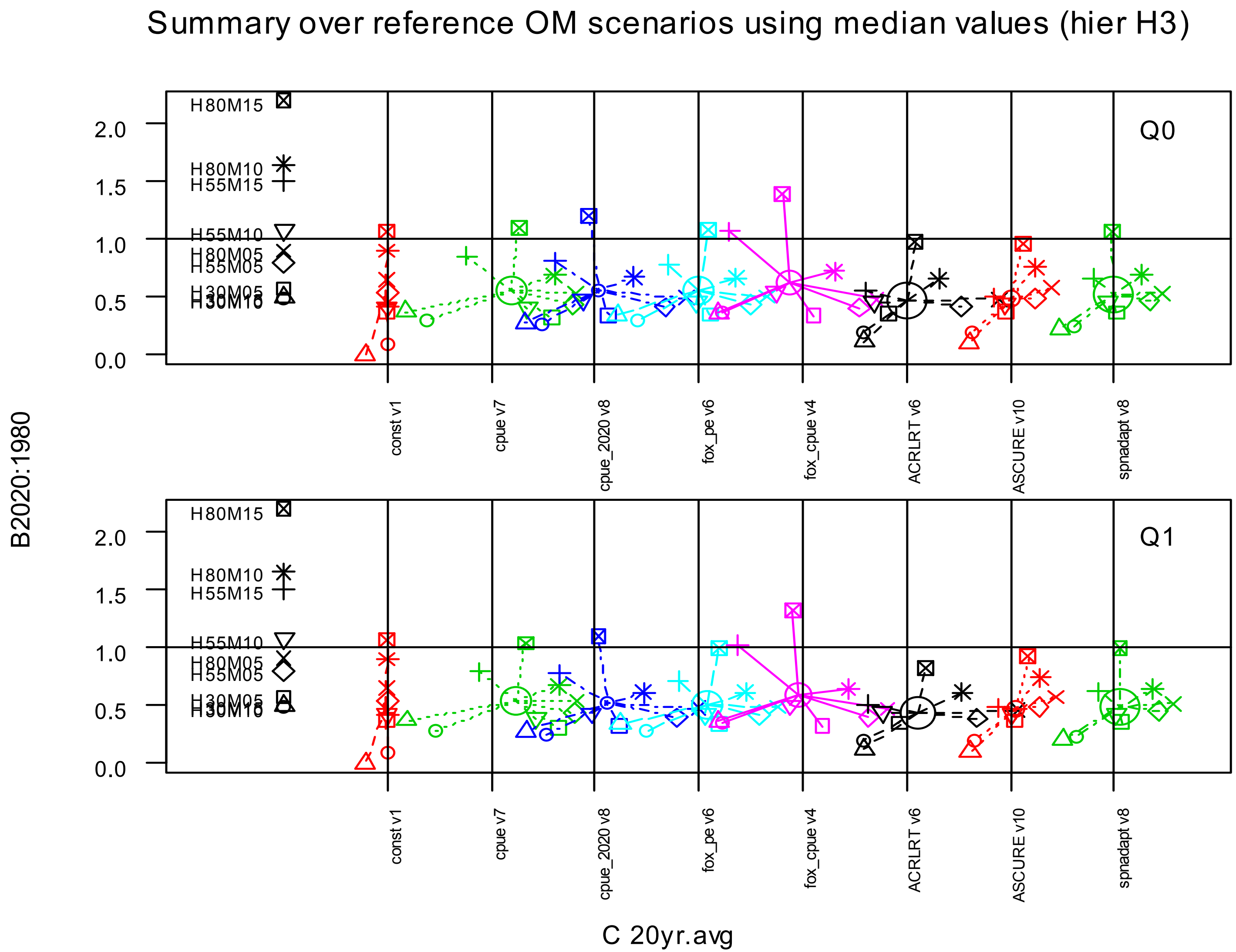


Figure 2. Summary performance of a selection of MP's. On the vertical axis is the biomass performance statistic B2022:B2002. On the horizontal axis is average catch over the 20-year projection period, *where each vertical line represents current catch level (i.e. catch in 2002, which was 15385.7 MT).* The plotting symbols at the ends of the rays show the median performance of the MP's (over 100 runs). A different symbol is used for each model scenario (as indicated). The centroid of the star, plotted as an open circle, is the average across models, where the radius of the circle is proportional to the average AAV statistic across models. The top panel is for the 9 'Q0' reference operating model scenarios and the bottom panel is for the 9 'Q1' operating model scenarios

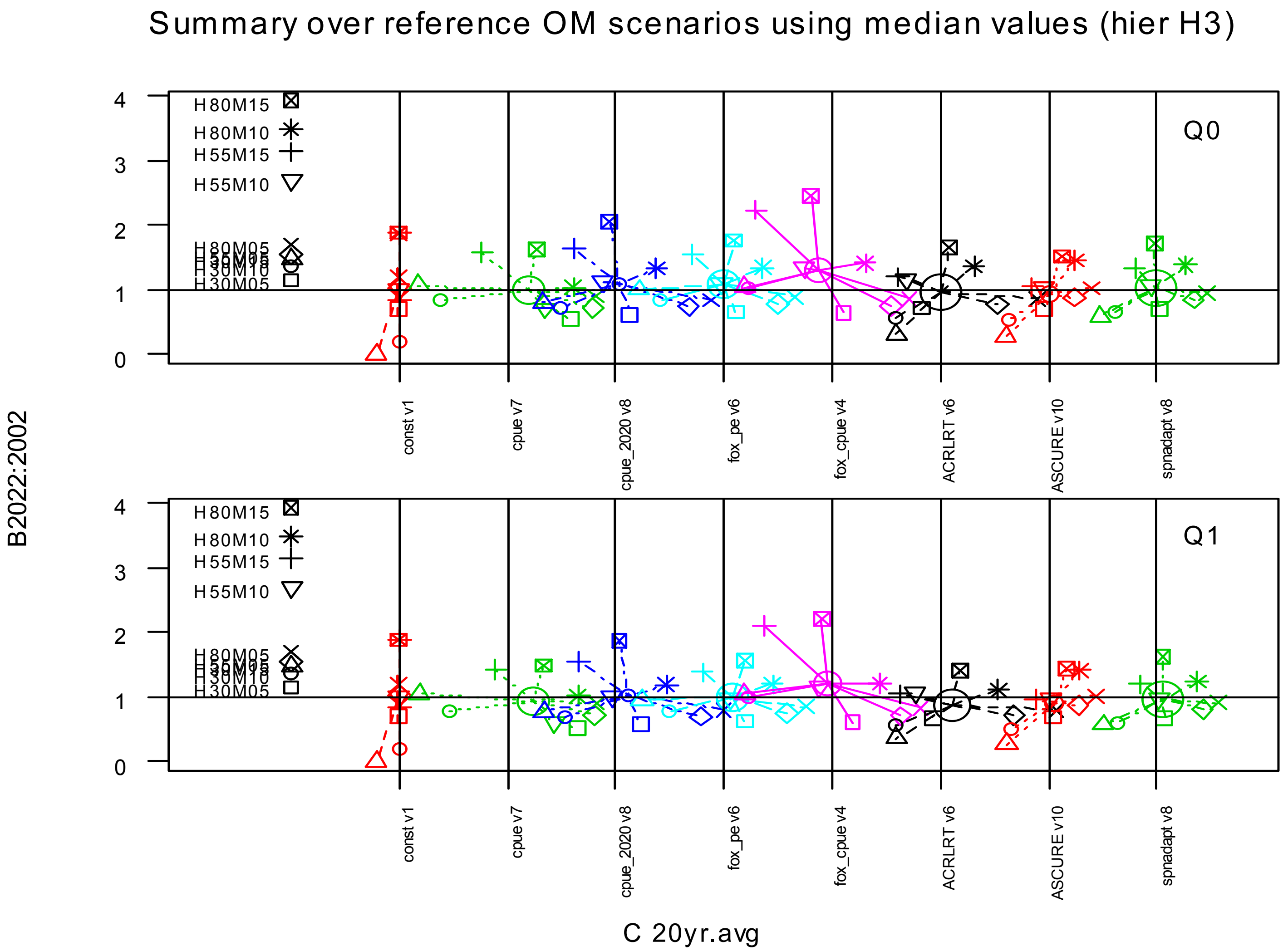


Figure 3. Summary performance of a selection of MP's. On the vertical axis is the biomass performance statistic B2022:B2002. On the horizontal axis is average catch over the 20-year projection period, *where each vertical line represents current catch level (i.e. catch in 2002, which was 15385.7 MT).* The plotting symbols at the ends of the rays show the lower 10th percentiles of the catch and biomass statistics for the MP's (over 100 runs). A different symbol is used for each model scenario (as indicated). The centroid of the star, plotted as an open circle, is the average across models, where the radius of the circle is proportional to the average AAV statistic across models. The top panel is for the 9 'Q0' reference operating model scenarios and the bottom panel is for the 9 'Q1' operating model scenarios

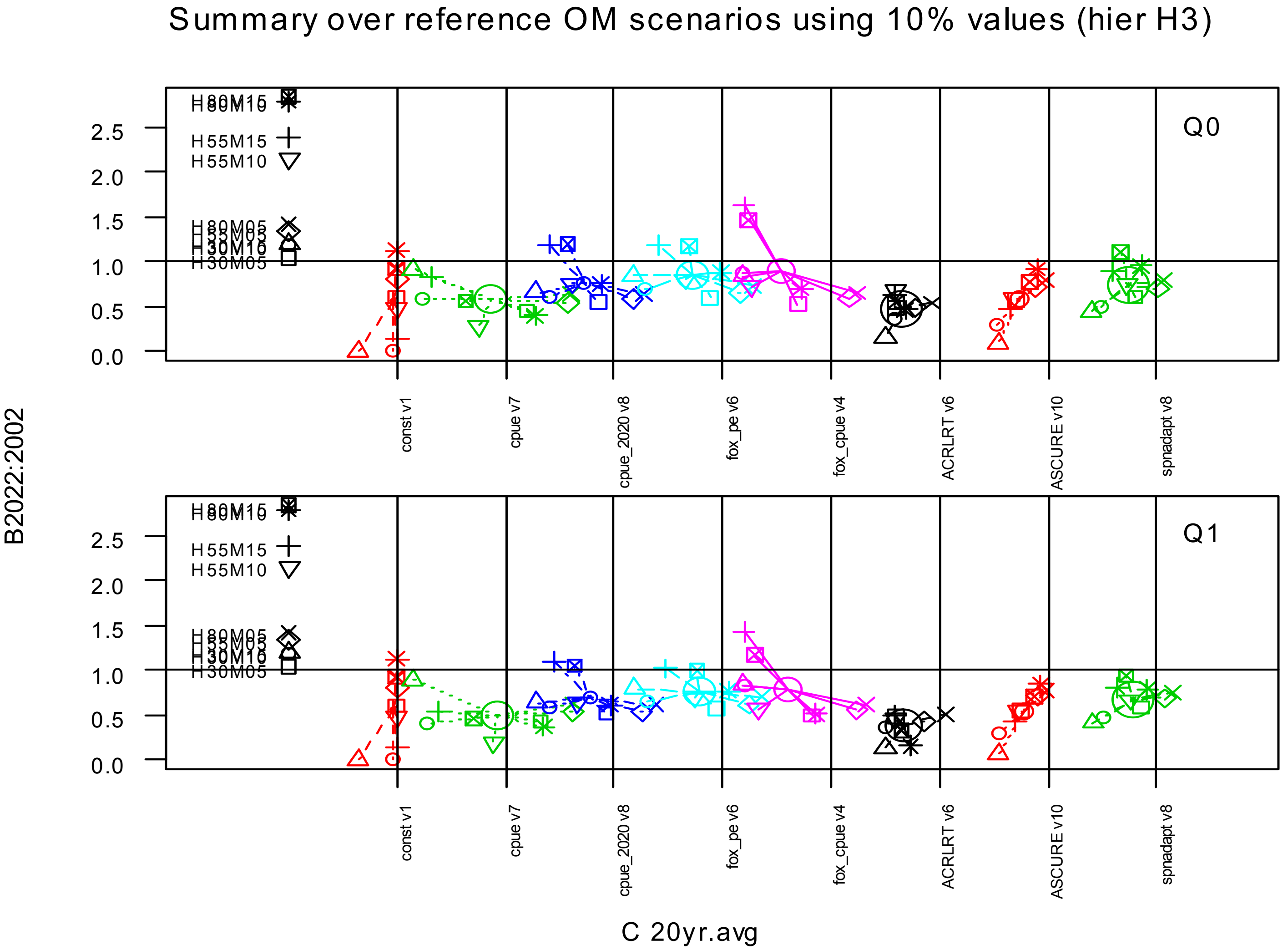


Figure 4. Comparison of robustness criteria, TAC-, catch-, and biomass-related performance statistics (over 100 runs) for a selection of MP's using a low productivity operating model scenario. For the robustness criteria (the top left-hand panel) a black circle indicates that the robustness criteria was not passed, and an 'X' indicates it is not relevant. The two TAC-related statistics are plotted as percent occurrences. For the other performance statistics, the median value is shown as a circle with an error bar extending from the 10th to 90th percentile.

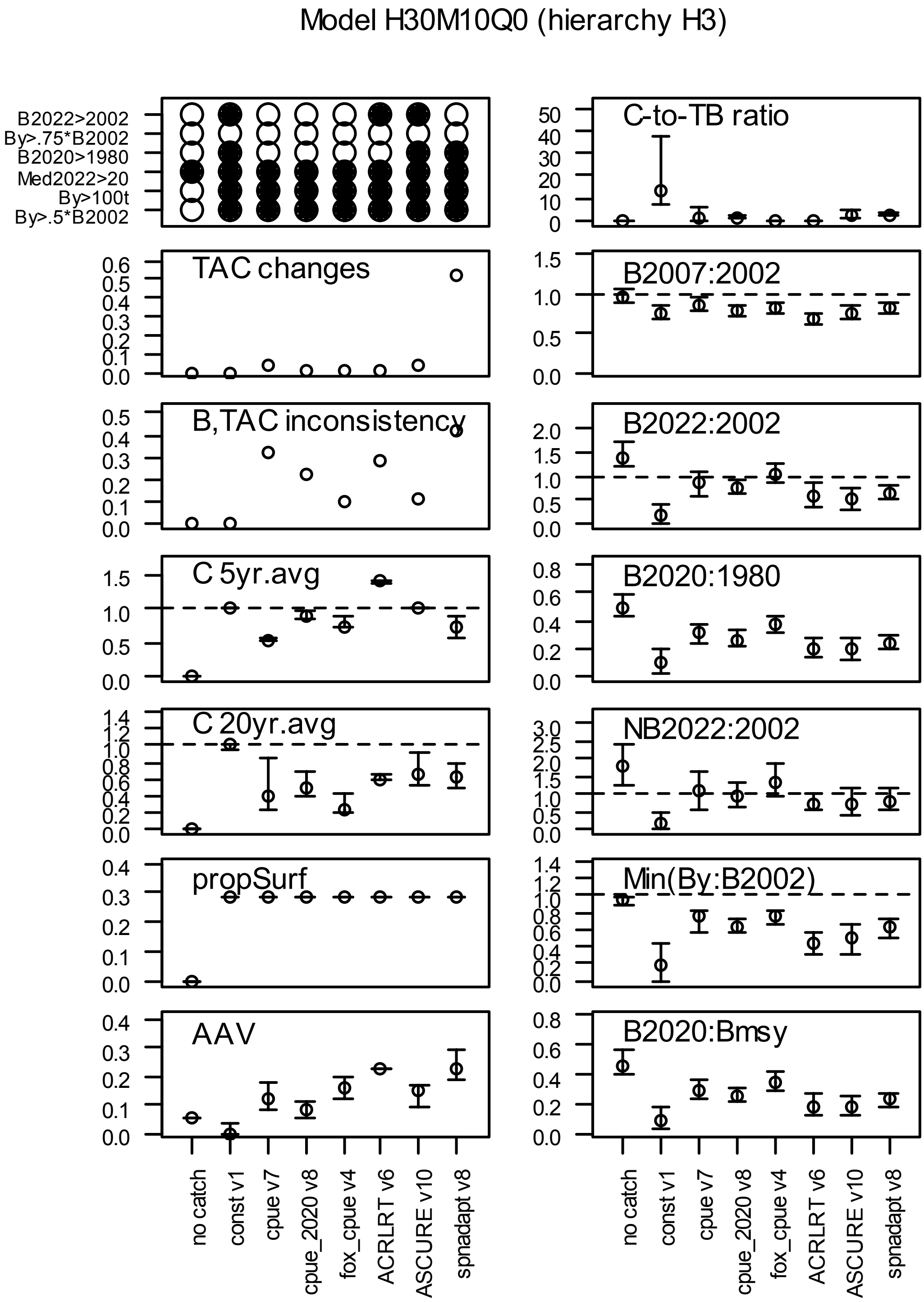


Figure 5. Comparison of robustness criteria, TAC-, catch-, and biomass-related performance statistics (over 100 runs) for a selection of MP's using an intermediate productivity operating model scenario. For the robustness criteria (the top left-hand panel) a black circle indicates that the robustness criteria was not passed, and an 'X' indicates it is not relevant. The two TAC-related statistics are plotted as percent occurrences. For the other performance statistics, the median value is shown as a circle with an error bar extending from the 10th to 90th percentile.

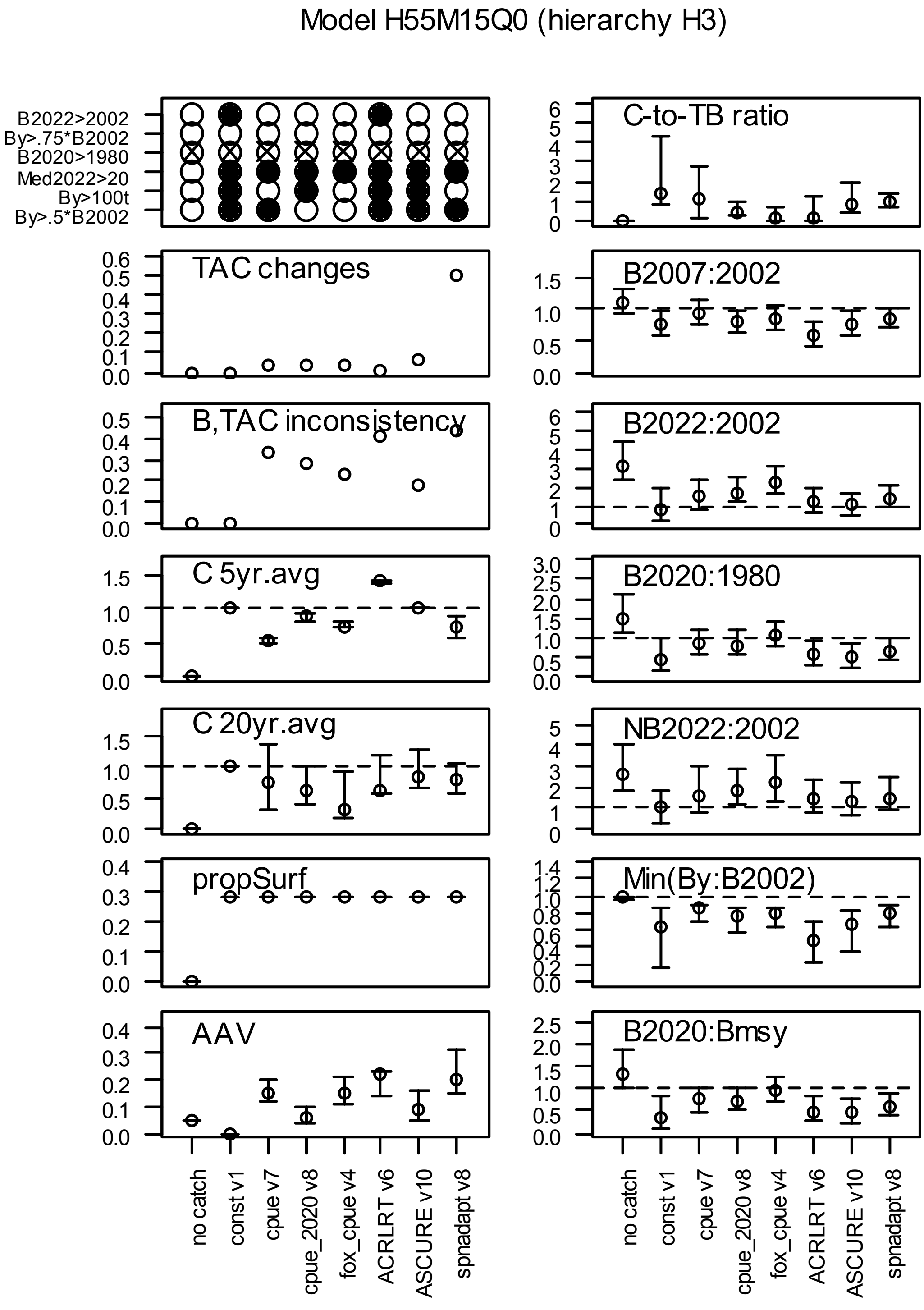


Figure 6. Comparison of robustness criteria, TAC-, catch-, and biomass-related performance statistics (over 100 runs) for a selection of MP's using a high productivity operating model scenario. For the robustness criteria (the top left-hand panel) a black circle indicates that the robustness criteria was not passed, and an 'X' indicates it is not relevant. The two TAC-related statistics are plotted as percent occurrences. For the other performance statistics, the median value is shown as a circle with an error bar extending from the 10th to 90th percentile.

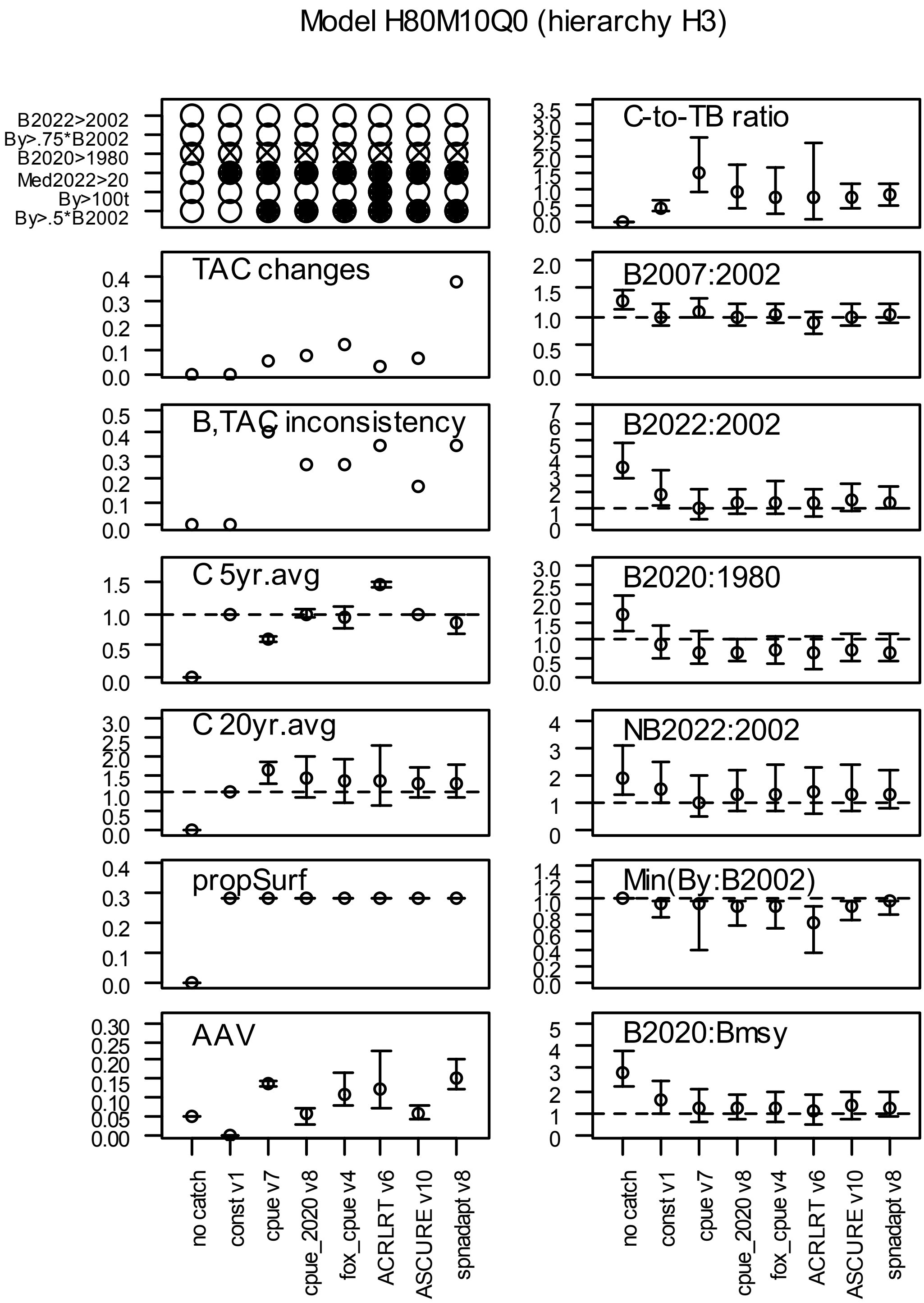


Figure 7. Summary performance of a selection of MP's with regard to the robustness criteria and the 18 reference case operating model scenarios.

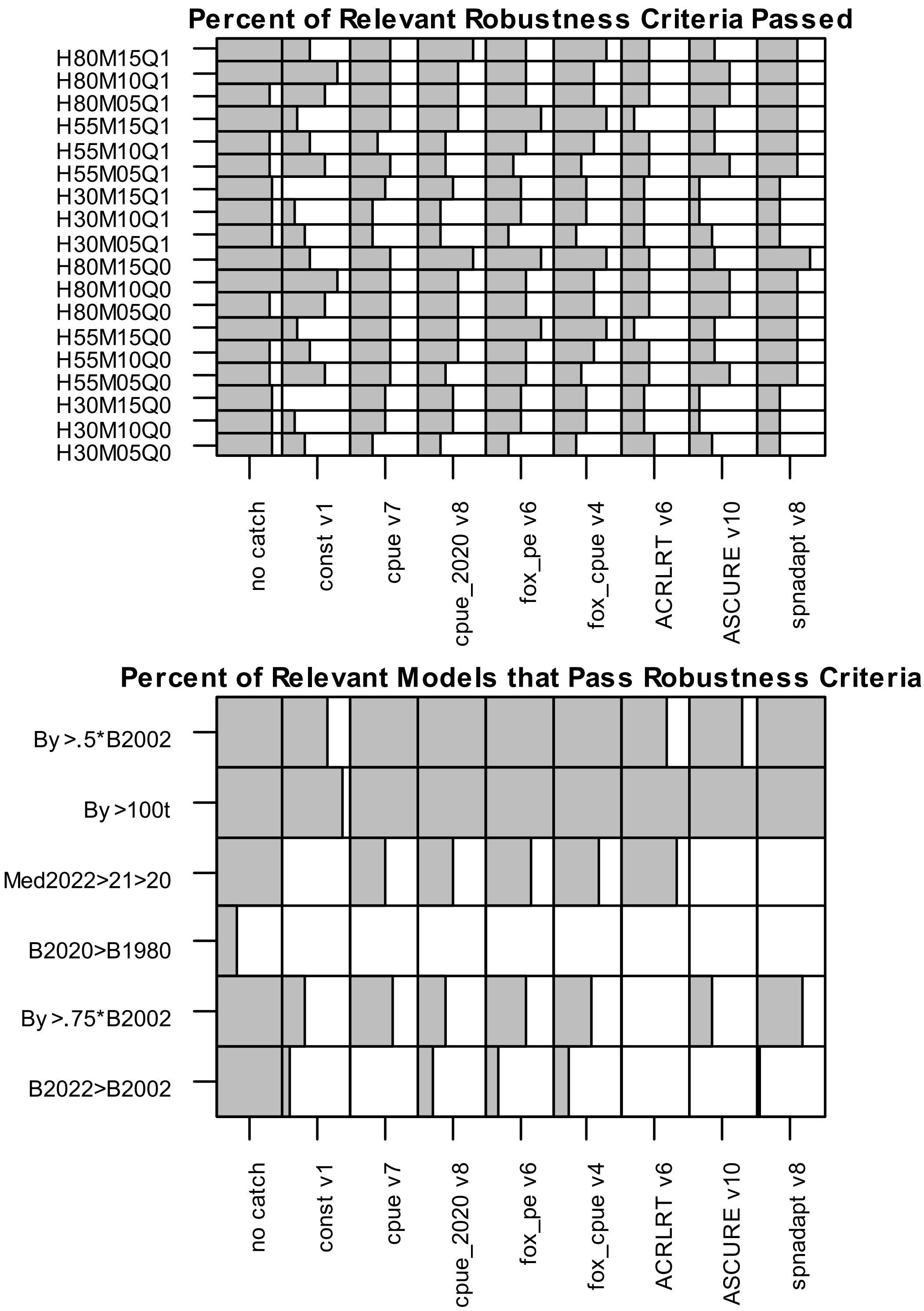


Figure 8. Summary results from the robustness operating model trials (i.e. "tick tests") over a selection of MP's. The results from each robustness trial were compared to the results from the most analogous reference case model to see if there was a substantial difference.

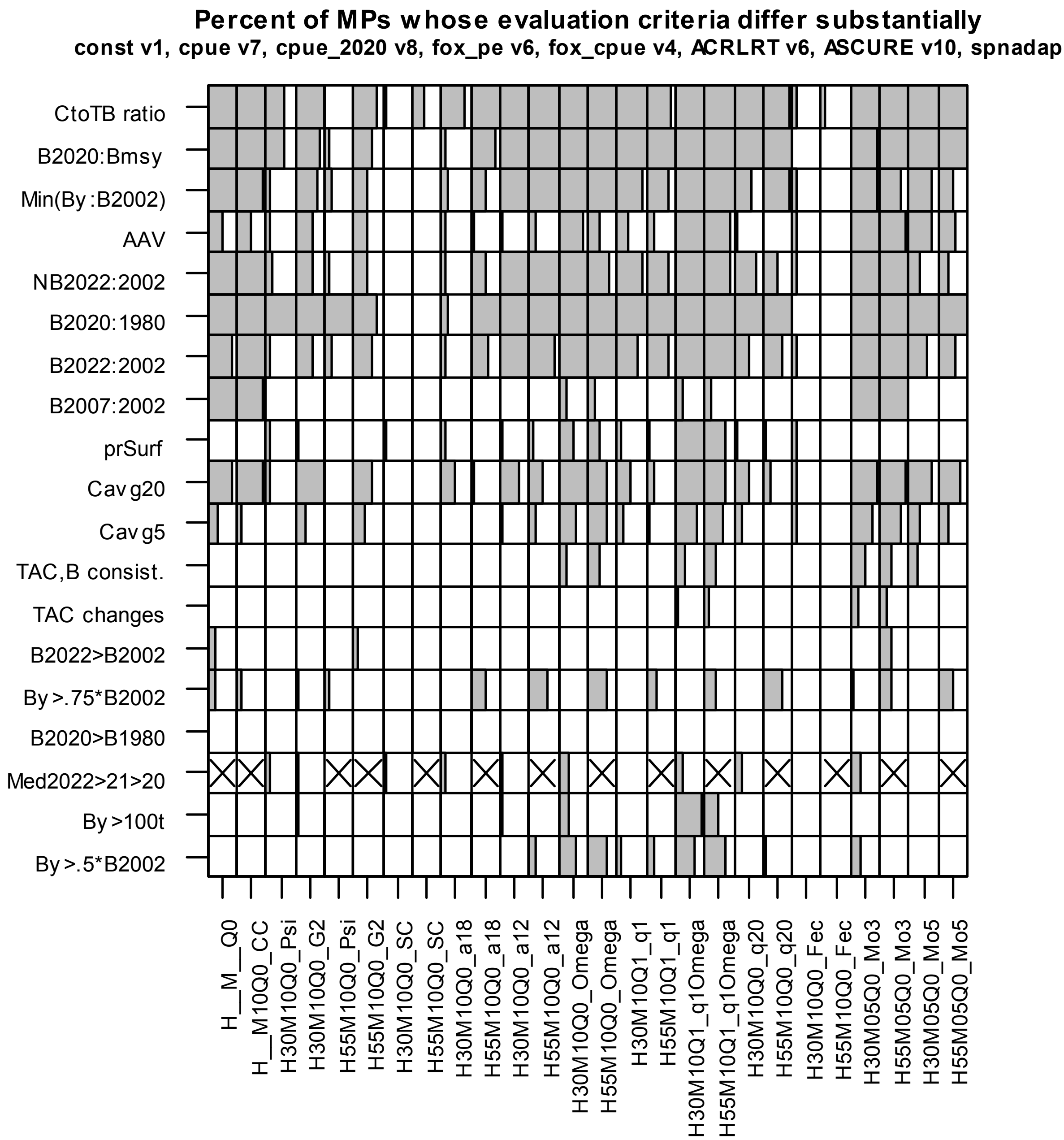
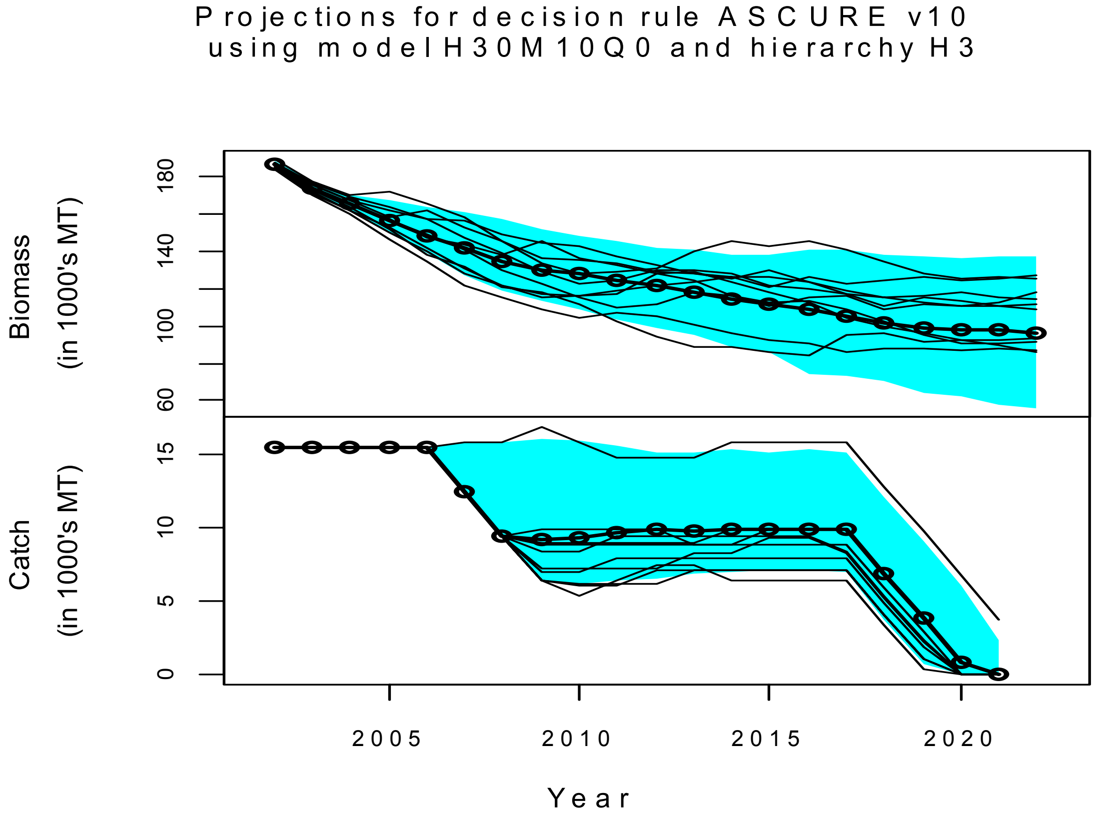


Figure 9. Catch and biomass trajectories for three MP's under a low productivity operating model scenario. The bold line with circles indicates the median over 100 runs. The other lines show 10 individual realizations, and the shaded area represents the 80% confidence interval.



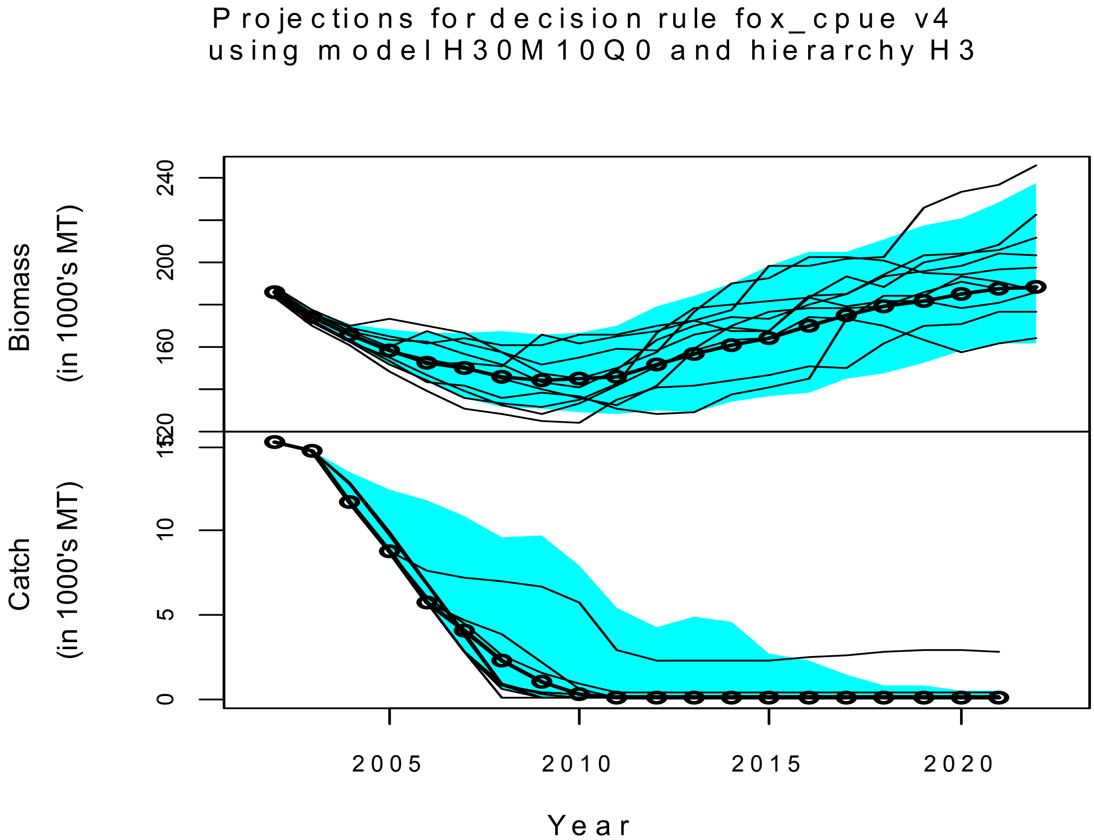


Figure 9 cont.

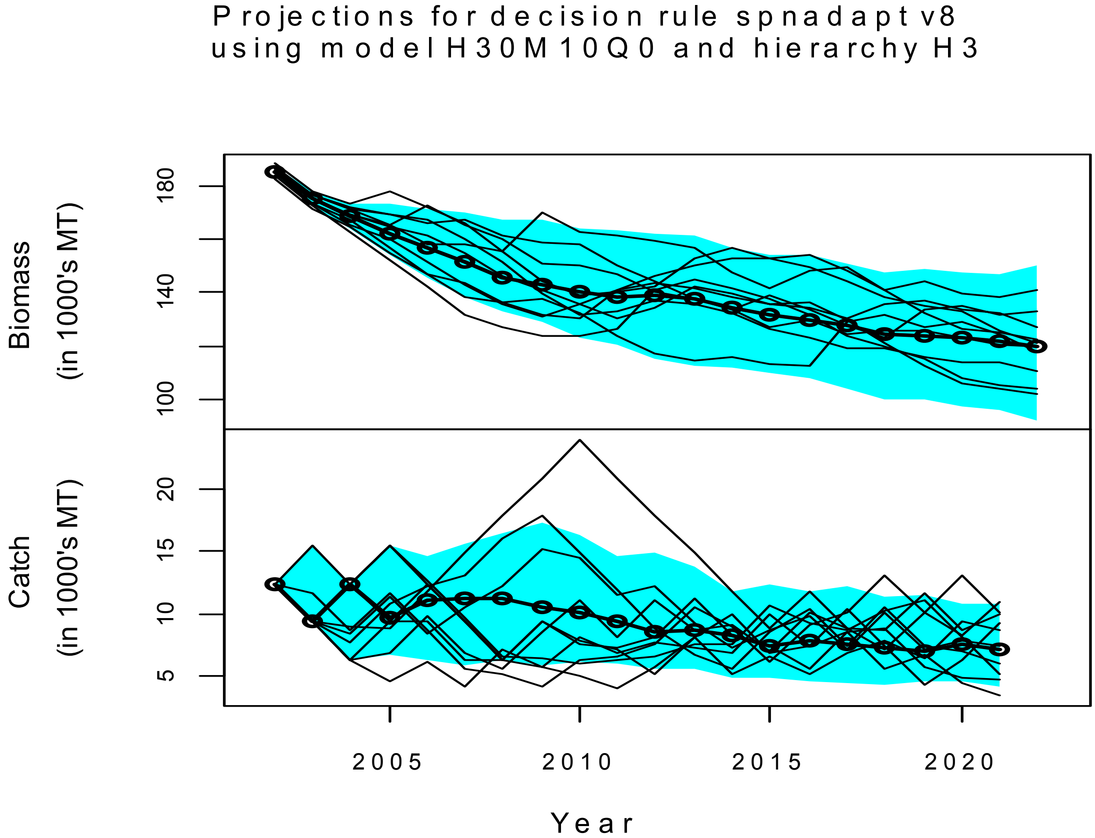
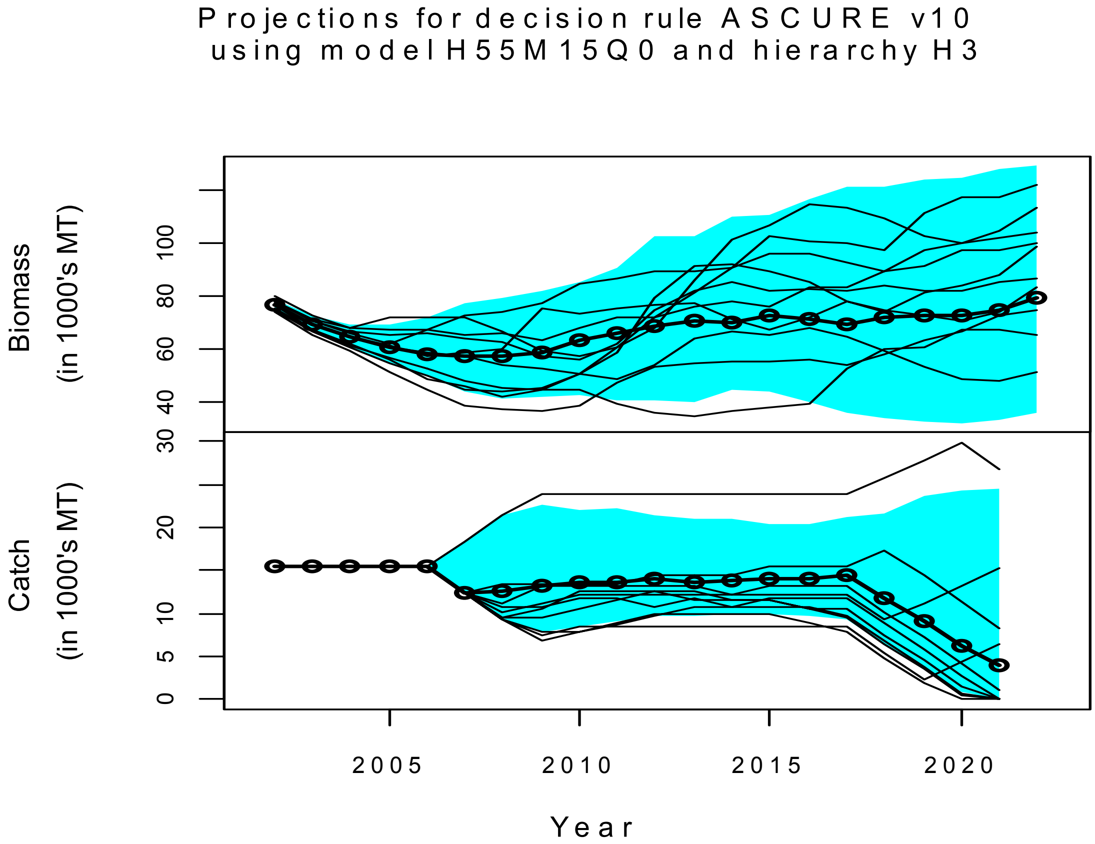


Figure 10. Catch and biomass trajectories for three MP's under an intermediate productivity operating model scenario. The bold line with circles indicates the median over 100 runs. The other lines show 10 individual realizations, and the shaded area represents the 80% confidence interval.



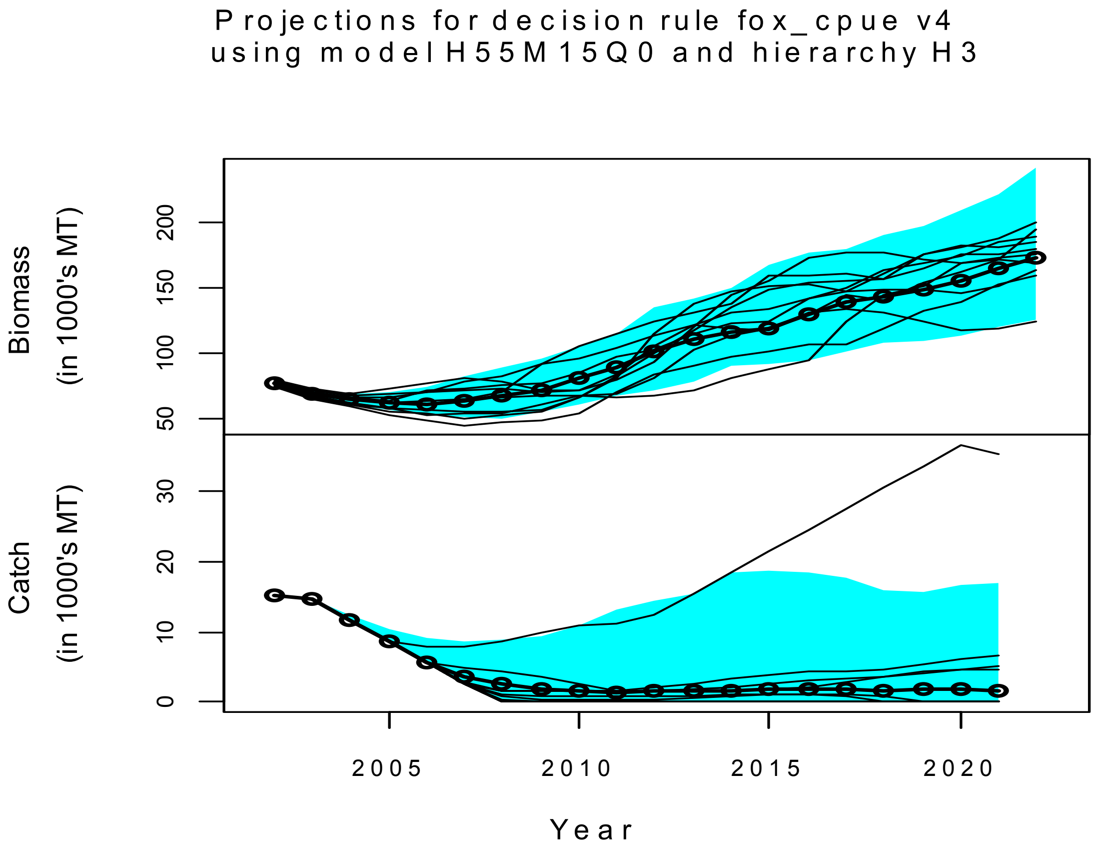


Figure 10 cont.