Development of a spatially-structured model for stock assessment and TAC decision analysis for Australian abalone fisheries

H. Gorfine, B. Taylor, M. Cleland, M. Haddon, A. Punt,
D. Worthington & I. Montgomery

State Government Victoria
Department of Primary Industries

Australian Government
Fisheries Research and Development Corporation

Project No. 1999–116

Development of a spatially-structured model for stock assessment and TAC decision analysis for Australian abalone fisheries

H. Gorfine[1], B. Taylor[1], M. Cleland[2], M. Haddon[3], A. Punt[4],
D. Worthington[5] & I. Montgomery[6]

1. PIRVic, Marine & Freshwater Systems, PO Box 114, Queenscliff, VIC 3225, Australia (corresponding authors).
2. 19 Hood Cres, Arrowtown, New Zealand.
3. Tasmanian Aquaculture & Fisheries Institute, University of Tasmania, Marine Research Laboratories, Nubeena Cres, Taroona, TAS 7053, Australia.
4. School of Aquatic & Fishery Sciences, University of Washington, Box 355020, Seattle, WA 98195-5020, USA.
5. New South Wales Abalone Council, PO Box 263, Northbridge, NSW 2063, Australia.
6. Birdway Pty Ltd, PO Box 2049, Townsville, QLD 4810, Australia.

February 2005

Project Number 1999–116

Development of a spatially-structured model for stock assessment and TAC decision analysis for Australian abalone fisheries

H. Gorfine, B. Taylor, M. Cleland, M. Haddon, A. Punt, D. Worthington & I. Montgomery

Published by Primary Industries Research Victoria, Marine and Freshwater Systems, Department of Primary Industries, Queenscliff, Victoria, 3225.

DISCLAIMER

The authors do not warrant that the information in this book is free from errors or omissions. The authors do not accept any form of liability, be it contractual, tortious or otherwise, for the contents of this book or for any consequences arising from its use or any reliance placed upon it. The information, opinions and advice contained in this book may not relate to, or be relevant to, a reader's particular circumstances. Opinions expressed by the authors are the individual opinions of those persons and are not necessarily those of the publisher or research provider.

ISBN 1 74146 133 2

Preferred way to cite:
Gorfine, H., Taylor, B., Cleland, M., Haddon, M., Punt, A., Worthington , D. and Montgomery, I. (2005) Development of a spatially-structured model for stock assessment and TAC decision analysis for Australian Abalone fisheries. Final report to Fisheries Research and Development Corporation Project No. 1999-116. Primary Industries Research Victoria, Queenscliff.

Formatted/designed by Primary Industries Research Victoria Queenscliff

Printed by PIRVic Queenscliff, Victoria

Table of Contents

NON-TECHNICAL SUMMARY

1999/116 Development of a spatially-structured model for stock assessment and TAC decision analysis for Australian abalone fisheries.

Principal Investigator: **Harry Gorfine***

Co-Investigator: **Duncan Worthington**

Address*: **Primary Industries Research Victoria**
P.O. Box 114, Queenscliff, Vic 3225
Tel: (03) 5258 0111 Fax: (03) 5258 0270
Email: Harry.Gorfine@dpi.vic.gov.au

Objectives:

1. Develop an appropriate modelling framework to incorporate all relevant biological information into a spatially-structured assessment of the likely risk and performance of different future TACC for an abalone fishery.
2. Develop the necessary software to complete such an assessment, and modify it to incorporate differences among the states and species.
3. Evaluate the capabilities of the modelling framework.
4. Compile the necessary data and complete a preliminary assessment of each State's abalone fishery.

Non Technical Summary:

Length-based models are now used to assess many of the remaining viable abalone fisheries worldwide and the AbModeller package developed for Australian abalone fisheries takes this type of modelling to a new level of sophistication. This project brought together experienced fisheries scientists from a range of different disciplines to form a multi-discipline group, many with overlapping skill portfolios. Consensus was achieved in developing a biologically sensible conceptual framework that was used as the basis for producing the model software. Critical to the project's success was the need for the model to be an integrated product rather than a collage of crudely joined blocks of disparate computer code. Above all a professional quality product was sought that would lend itself to being used because of its accessibility whilst still retaining the robust statistical capabilities of a fully stochastic Bayesian model. This contrasts with most contemporary fisheries assessment models that are generally specific to a particular fishery and accessible only to highly trained and experienced modellers. The adoption of a MS Windows® GUI shell is an outstanding feature of the model not only because it makes it accessible to a broader range of fisheries scientists, but importantly because it substantially increases the efficiency of completing and tracking computing runs in large multiples. The GUI presents outputs in an easily interpretable format that reduces the time spent having to collate and summarise the results for presentation.

A testing and simulation capability has been built into the model with the ability to simulate and analyse results. This provides a powerful platform for ongoing testing that will continue beyond the duration of the project. Only by applying the model to various real assessment tasks can its limitations be fully revealed and appropriate modifications made in response. The AbModeller software and a User Manual have been transferred individual States so that each can conduct its own testing and its modellers can familiarise

themselves with the software in the context of their specific data holdings. The timing for adoption and the mode of application will vary depending State needs and readiness.

OUTCOMES ACHIEVED

- An abalone modelling software and installation package with detailed instruction manual has been distributed to fisheries biologists in abalone-producing states in Australia and to several researchers in Mexico and South Africa.
- The South Australian Research and Development Institute has used the modelling software in a preliminary assessment of the Southern Zone blacklip abalone resource and is considering expanding this application to other abalone resources within the South Australian fishery.
- Primary Industries Research Victoria has re-coded and re-developed its data sets in 2003 to be compatible with the new software. Implementation for fishery assessments is planned for 2005 to ensure that the transfer from the existing software does not change the current base case assessment for the Victorian fishery.
- Abalone fisheries modelling now includes the option of inputting estimates of habitat reduction and producing estimates of MSY. These are important elements of the NSW and Victorian abalone assessments.

A slow but progressive approach is required to maintain compatibility with previous model-based assessments in both NSW and Victoria and it will be several years before more tangible management outcomes become apparent

KEYWORDS: Abalone, Risk Assessment, Modelling**.**

Acknowledgments

PIRVic—Marine and Freshwater Systems acknowledges the substantial contribution from interstate collaborators.

This project would not have been possible without the considerable cooperation and support of the following organisations:

- Commonwealth – CSIRO Marine Research
- New South Wales – Fisheries Research
- South Australia – SARDI Aquatic Sciences
- Tasmania - TAFI
- Western Australia – WAMRL

FINAL REPORT

1999/116 Development of a spatially-structured model for stock assessment and TAC decision analysis for Australian abalone fisheries.

Background

Australia is the world's largest producer of wild abalone and annually exports almost 5000 tonnes to mostly Asian markets. Although all States manage their fisheries by catch quotas, and either have or are in the process of developing fishery independent monitoring programs, they are deficient in their capacity to address fundamental policy questions such as:

1. Are increased catches sustainable?
2. Can the current catch be sustained at a lower level of fishing effort?

Abalone fisheries biologists and industry representatives from across Australia met during July 1998 to discuss strategies and priorities for future research. The meeting endorsed the pursuit of collaborative proposals aimed at establishing closer links between stock assessment and management of Australia's abalone fisheries. Interstate collaboration not only brings together a variety of expertise, it also facilitates the pooling of information and resources unavailable to any individual State operating on its own.

This project in particular was given top priority because of its focus on determining sustainable catch quotas for each State’s fishery. Recent changes in total allowable commercial catches (TACC) to optimise production from their respective fisheries have highlighted the urgency for a widely accepted model to examine the potential risks involved within a biologically defensible framework. Such a model should be applicable in all abalone-producing States, limited only by their data and management objectives.

Need

Several aspects of the biology and ecology of abalone complicate the assessment of their fisheries. Perhaps most importantly, the short-distance dispersal of larval abalone leads to dis-aggregated populations with a limited ability to re-populate depleted areas. Further, the ability of commercial divers to differentially exploit spatially-separated sub-populations allows them to maintain catch rates despite reduced stock abundance. To date, such spatial structure has not been incorporated in any framework used for TACC advice. This is despite the availability of spatially-structured programs monitoring changes in the abundance of abalone of different sizes that have been completed within Victoria, South Australia and New South Wales for several years and are proposed in the other States. Such programs provide the necessary information with which to make spatially-structured assessments of the likely risk and performance of different future TACC.

In common with many other exploited species, abalone exhibit high variability in their population parameters across spatial scales ranging from individual reef complexes to management zones spanning several hundred kilometres of coastline. Stochastic approaches to modelling are required to accommodate the heterogeneity in parameter data. This variability also means that individual data sets tend to be idiosyncratic representations of specific sampling locations. Collectively the samples are mostly insufficient in number and geographic distribution to be considered typical of each State’s abalone resources as a whole. The application of Bayesian prior probability techniques provides a means of addressing this issue in fisheries modelling.

Objectives

1. Develop an appropriate modelling framework to incorporate all relevant biological information into a spatially-structured assessment of the likely risk and performance of different future TACC for an abalone fishery.
2. Develop the necessary software to complete such an assessment, and modify it to incorporate differences among the states and species.
3. Evaluate the capabilities of the modelling framework.
4. Compile the necessary data and complete a preliminary assessment of each State's abalone fishery.

Methods

Meetings, workshops and reviews of model framework

A steering committee was formed among the abalone producing States with representation from each State’s fisheries research institute (Assoc Prof M. Haddon - chair, Drs K. Friedman, H. Gorfine, R. Officer, S. Shepherd & D. Worthington) and a consultant from CSIRO/University of Washington (Res Prof A. Punt). Other participants included Drs R. Day (Univ Melb), K. Rodda & Y. Xiao, B. Taylor, C. Blount, R. Chick, C. Dixon & K. Hodgson. Dr Ian Montgomery joined the committee in July 2000 following his appointment by PIRVic to develop the model and write the modelling code. Staff changes at SARDI & WAMRL resulted in Drs S. Mayfield and A. Hart becoming involved during 2002. Over several interstate meetings, this committee discussed and planned strategies for the conduct of the project, developed a generally agreed conceptual modelling framework, and modified milestones where necessary to produce a workplan that addressed the stated objectives within the context of individual State priorities.

Following general agreement on a conceptual model framework, subsequent State-based meetings were held to refine this framework and to draft model specifications. The first meeting at PIRVic involved Andre Punt, and two following meetings were held in Tasmania with Andre Punt and Malcolm Haddon participating. Informal discussions were also held with Yongshun Xiao and Rick McGarvey at SARDI in South Australia. The model specifications were drafted to include current features of both the Victorian and New South Wales assessment models, but also specified additional features that accommodated seasonality, density dependent selectivity and cryptic habitat. The model development process was broadly separated into assessment and simulation testing phases.

The simulation testing phase was reviewed by the steering committee in August 2002 following extensive discussions between the project team and the States and was redefined to reflect the State needs and priorities. The assessment model was to be modified to generate sets of test data. The steering committee developed an extensive list of questions that it believed should be addressed through simulation modelling and a list of priorities was drafted. These included:

- Initial basic testing of the model against itself to recover parameters.
- Effects of mis-specification of priors (such as mortality and recruitment curve steepness).
- Different dynamics in the cryptic population compared to the non-cryptic.
- Comparing pooled with un-combined data.
- Fleet dynamics - by varying catch distribution through time via either explicit time series or setting catch proportional to abundance.
- Influence of serial depletion & the ability to detect serial depletion
- Assessment strategy – not running a survey or assessment every year.
- The loss of habitat, loss of population or loss of fishing grounds.
- The interaction between serial depletion and fleet dynamics.

It was recognised that although a number of these questions could be answered by this project, many would need to be investigated by future projects.

A final meeting was held in 2003 to review the model, consider future testing and help NSW prepare to adopt the model for their forthcoming annual assessment and pave the way for future use of the model in SA and WA.

Data collection and collation

The availability of data in each State was tabulated and discussed. The committee concluded that the transfer of these data to PIRVic for application in prototype models was both impractical and unnecessary for model development. Their preferred approach was to progressively transfer the code to individual States so that each State could conduct its own testing and its modellers could become familiar with the software in the context of their specific data holdings. The timing for this was to vary depending on progress with model development and State needs and readiness.

Test data from a range of sources, sufficient for model testing, were collected. It was anticipated that further samples of data would be collected on an as-needed basis as the model was progressively adapted for each State. Indeed, it was clear at this stage that trialing and implementation in some States would be limited by various deficiencies in the types and quantities of data available in each instance.

Model development

Coding of the deterministic prototype-model was completed by December 2000 and the model tested with samples of data. The model fitted to all abundance data types that were currently available (Fishery dependent: CPUE & commercial catch size-structure (l-f); fishery independent: relative abundance & size structure (l-f)). The first phase of development of the assessment model reflected the need for an assessment model for Tasmania. Transfer of the modelling code and documentation for a prototype model to Tasmania was completed in January 2001 and the model was successfully run at TAFI.

Development of the model continued with the implementation of fitting to multiple weighted growth matrices. This was tested by fitting to three matrices (corresponding to slow, medium and fast growth) with two weighting parameters. Fitting to two matrices with one weighting parameter proved to provide the same flexibility more efficiently, so the three matrix option was eliminated. Growth matrices generated by software based on Troynikov (1997) and on Bardos (2004) have been used. The results indicated that rate of growth was a fundamental feature of the model. The importance of selecting an appropriate way of representing growth led to the incorporation of a robust alternative growth estimation module (Bardos, 2004), and its addition to the software CD package.

The next feature involved alternative fitting to growth within the model. To provide flexibility, it was decided that the general growth model of Schnute (1981) with a gamma distribution of growth increments (as implemented by Worthington (1997)) would be incorporated. This model is sufficiently general that it can be made to behave like other commonly used growth models (eg. Von Bertalanffy and Gompertz) and provides for choice among alternative probability density functions (e.g. Gamma and Log-normal).

Specifications for emergence from cryptic habitat and redistribution or movement in response to fishing were reviewed. The original specifications allowed for density-dependent emergence of abalone from cryptic habitat. It was anticipated that it would be difficult to distinguish between emergence and redistribution, so a single density-dependent movement function was implemented first. This only made a distinction between visible and hidden populations. All recruitment related to the hidden population and all fishing and survey work related to the visible population. The model fits to an ogive to model the emerged population. The parameters for the width of the ogive and the size at 50% transition can depend on the degree of depletion of the visible population to allow for density-dependent effects.

Adding capability in the model code to handle many reefs and several levels of aggregation (areas, zones and states) accommodated the spatial dimension. This structural complexity mandated the use of a database for storage and retrieval and the use of graphing tools for the effective presentation of results. Consequently, a key aspect among the specifications was the definition of the architecture and software components of the system. A decision was made to use C++ as the computing language in which the model code would be written. This was largely to ensure compatibility with the adoption of AD Model Builder® software that would be used for parameter estimation. It also facilitated use of the MS Windows® environment in which to run the model.

Data modelling was also undertaken to provide the basis for database design and consequently a consultant built an MS Access® database to support both abalone research and modelling.

Agreement was reached among the States on priorities for development of the final form of the assessment model. These included:

- fitting to growth;
- managing size-structured data;
- separation of the population into hidden and visible components;
- selectivity ogives for each of the fishing sectors;
- simulation within the ADMB framework.

The procedures that map size-structured data to the size composition estimated in the model were rewritten to make them more flexible and robust. The width of the size classes in the data does not have to be the same as the width used in the model, nor do the ranges of sizes have to match.

Selectivity ogives, instead of knife-edge selection, were implemented for the three fishing sectors (commercial, recreational and unreported). A logistic ogive and a modified knife-edge ogive are available. The ogive parameters are currently calculated directly to avoid adding extra fitted parameters to the model, but the ability to estimate ogive parameters will be added to the model as required. The Victorian test values were based on the results of an unpublished study by Bruce Taylor.

The ability to switch off the fitting procedure in the assessment model was added to enable this model to be used for simulation testing. This allows the model to be run without parameter estimation in order to perform the life-cycle calculations for the desired time period (historical and/or projected) using supplied parameters and actual or simulated data. Simulation testing can therefore be done either within a command line environment using the ADMB framework or in the Windows/GUI/Database environment developed for the Simulation Model.

These enhancements have led to the Assessment Model exceeding the specifications initially defined for its development.

Evaluation of capability of modelling framework and software

A considerable amount of time during the final six months of the project was spent debugging the program code, and developing and integrating the GUI with multiple forms/screens to link the model to input and output databases. These links provided substantial increases in efficiency for data handling and tracking results from multiple runs used to test different fishery scenarios. The GUI allows the user to operate in a Windows®-based environment for the selection of modelling options, such as parameter specification, and the provision of diagnostic information about how well the model is performing with the input data and options selected. The output database is designed to assist reporting for decision making and communicating advice to management committees.

A testing and simulation capability has been built into the model with the ability to simulate and analyse results. This involves access queries, forms, modules to support this process - similar to that for analysing Bayesian outputs.

The model was subjected to basic simulation testing. The initial testing was run under the priorities of the steering committee meeting of June 2002, namely

Initial basic testing of the model against itself.

Effects of mis-specification of priors (this was initially done for mortality) and also for the weighting given to different observed indices of the population.

There is now a powerful structure in place for ongoing testing that will continue beyond the duration of this project. Some tests are required for future assessments in Victoria and will be reported to other States on an ongoing basis. Conduct of other testing that is computationally and time intensive is dependent on provision of further funding.

A long-term view taken by steering committee in August 2002 was to get the substantial modelling process in place, meeting a wide range of needs. Testing was seen as ongoing process, so a testing platform was put in place and preliminary testing started. The 2003 steering committee meeting of all States reinforced this, along with recommendation to seek further funding for an extended testing program.

Technology transfer

Technology transfer to each State occurred progressively throughout the project depending on the specific needs and their urgency with respect to State fishery assessments. This is an on-going process planned to extend beyond the life of this project with PIRVic continuing to take responsibility for limited product support after the final transfer. This will facilitate a process of continual refinement in terms of trouble-shooting and incorporating suggested improvements on a scale much larger than if PIRVic focussed solely on Victorian application of the model. It is anticipated that this additional real-world testing and improvement will provide long-term benefits to PIRVic and promote on-going dialogue and collaboration about abalone modelling nationally among State research institutes. These benefits will substantially outweigh the small maintenance costs involved.

All States have now been provided with copies of AbModeller software and its User Manual. Bruce Taylor has followed this dissemination with one-on-one discussions and guidance about the application of this technology. Most recently, SARDI researchers Stephen Mayfield and Thor Saunders visited the PIRVic Queenscliff Research Centre to work for several days with Bruce Taylor and Michael Cleland to familiarise themselves with the software package and apply it to their own data. During this visit the model was run successfully on a sample of data from the Southern Zone of the South Australian blacklip abalone fishery. Subsequently, SA has developed a preliminary assessment for the Southern Zone, using AbModeller (Mayfield et al. 2004).

A final interstate workshop was held (at Victorian State expense) in Melbourne during August 2003 to provide a briefing for each State to enable them to evaluate the AbModeller software and User Manual. PIRVic is prepared to provide limited assistance with incorporation of any future developments into the master software. Terry Hooper, DPI, is investigating the establishment of an FTP or download site where updated versions can be posted for public downloading.

Communication of project results

No scientific manuscripts have been drafted to date, although the model specifications and instructions on its use have been written in a User Manual appended to this report. Compilation of the detailed instruction manual has been a substantial but essential undertaking. It is anticipated that a manuscript based on model testing and application will eventually be prepared for publication in a scientific journal. As mentioned above, we also hope to make the software and manual available on an Internet website to allow ready access to periodic updates.

Results/Discussion

Length-based models are now used to assess many of the remaining viable abalone fisheries worldwide. Specifically, Australia (Worthington 1997, Gorfine et al. 2001), Mexico (2003), New Zealand (Breen et al. 2003) and South Africa (Plaganyi et al. 2001). In each instance the conceptual modelling framework and modelling process are substantially similar. Although the South African model purports to be age-structured, this is achieved via a deterministic conversion of length to age rather than independently estimating age. The commonality among models should come as no surprise given the limitations imposed on modelling by abalone population biology and research data acquisition. These models are not novel in the sense that they mostly utilise sets of mathematical equations derived from fisheries models generally. Indeed, in common with the model used in New Zealand (Breen et al. 2003), the model developed in this project has its roots in a model established by Punt and Kennedy (1997) for modelling rock lobster populations in Tasmania. The major difference among most of these models is how shell growth is accommodated. Different growth models and underlying relationships as well as methods for incorporating uncertainty are utilised. In recognition of the need for choice, the model we have developed provides a range of growth options as well as capacity for alternatives to be added in the future.

Breen et al. (2003) eloquently describe a rationale for selecting a length-based model to assess abalone stocks, salient features of model functions and an evaluation of their model's effectiveness when applied to stocks in the New Zealand abalone fishery. The main technical aspects of their account could be equally applied to the model featured in this project report, so rather than elaborate here the reader is referred to their paper.

For this project the results comprise three main elements i) the conceptual modelling framework (specifications), ii) the modelling software package including an instruction manual and iii) results from preliminary testing of the model to provide an evaluation of its performance and limitations. A complete description of specifications, including model equations, is provided in Appendix III Part 2. Appendix III Part 3 contains the instruction manual for the software and part 4 contains the Software Design Specification. Extensive, but by no means exhaustive, test results from basic simulations are provided in Appendix III Part 1. The software and installation programs, documentation and a sample database are on the accompanying compact disc.

A Stock Assessment and TAC decision analysis model for Australian abalone fisheries (AbModeller)

Fundamental to the decision of the State abalone fisheries scientists to support a national abalone assessment model was the development of a single, agreed model that incorporates individual State requirements and encourages the sharing of modifications and enhancements. This minimised the need to reinvent the wheel continually, and all States can benefit from additional work in any other State. A very positive outcome of the project has been the sharing between the States and a commitment to an on-going modelling process.

A substantial product was needed that was flexible, covered many needs, was easily used by a broad range of users and delivered outputs to satisfy a range of management requirements. This led to the decision to use an approach that linked the model to a GUI - database.

An output database was chosen that had the capacity to meet the needs of spatially diverse fisheries, with potentially many assessable regions and many scenarios requiring testing (based on assessment group or management requests). It was also chosen to deliver outputs focussed on management plan information requirements.

The design of the software, and the time spent on developing a robust and easy-to-use product that would focus on requirements of management, was paramount.

Strength of this approach lies in its capacity to efficiently run multiple assessments and multiple outputs. Ease of use for non-modellers was amply demonstrated by the South Australian preliminary assessment experience. A spatially diverse abalone fishery has different demands to most other fisheries. The Victorian experience with conducting multiple assessments supported the decision to build an integrated package. Victoria will move from six modelling regions to nine during the near future. NSW has six regions and runs multiple alternative scenarios. Tasmania has a huge coastline and potentially many assessable regions. Provision of an input database and a user-friendly GUI for data and parameter selection was a very effective way of dealing with multiple alternative input data sets.

Although the software was designed to handle complex assessment tasks efficiently, it cannot eliminate the high computing overheads associated with multiple scenarios across a large number of regions at the same time. Overseas institutions with Cray computers, such as NIWA in New Zealand, have a considerable advantage in this area.

Evaluation of capability of modelling framework and software

A structure for ongoing testing is in place in the software. The results of the completed testing are discussed in Appendix III, part 1.

Technology transfer for State stock assessments

An important point is the limitation imposed by inadequate data, either due to its insufficiency or bias. No matter how technically sophisticated or biologically accurate the model, there is no solution to this situation other than redressing the data deficiency. As far as possible, the model was rendered sufficiently flexible to utilise the alternative types of data held in each State.

As the project progressed two issues became abundantly clear, that initially only Victoria and NSW had adequate time series of abundance data for fitting the modelled estimates of biomass; and that all States suffered from insufficient sampling to provide data necessary to run the model at the small scale at which abalone populations effectively function (referred by some as a metapopulation scale). The latter reflects the inappropriately coarse scale of management that all fisheries agencies use for managing abalone resources and the poor level of State research and assessment funding made available relative to the value of the resource. These issues are important in relation to future reviews of research strategies and development of research projects.

Benefits

Benefits for assessment and management of all Australian abalone fisheries will accrue as the software developed during this project is progressively adopted to produce estimates of biomass that enable selection of sustainable harvesting strategies. The software is accessible to a broad range of users, and competent programmers will be able to modify its source code to suit their specific requirements. Those abalone-producing States yet to embrace modelling now have access to a 'ready to use' tool that provides a comprehensive range of functions and outputs. Agencies already engaged in modelling abalone fisheries can choose to transfer to a standardised well-documented software package that will facilitate the succession of the next generation of specialists dedicated to abalone fisheries modelling.

Easy up-loading of inputs from databases and generation of tabulated outputs that can be easily interpreted by assessment biologists and fisheries managers, will substantially improve the efficiency of assessment reporting and management decision-making processes. The software incorporates a robust simulation capability that will enable exploration of existing datasets to test their limitations and identify the types of data that should be collected in the future.

Further Development

Although the formal project has concluded, development of the model has not ceased and should be viewed as an ongoing process of continual improvement. Only by applying the model to various real assessment tasks can its limitations be fully revealed and appropriate modifications made in response. It is critical that research institutions do not make the mistake of considering the job to have been done and consequently fail to provide sufficient financial resources for maintenance and further development. The modelling process is dynamic and constantly evolving in response new insights and novel technology. There seems to be a tendency among some managers to avoid confronting the reality that models are complex tools that must be used within the context of the limitations imposed by current knowledge and data availability. As these limitations are overcome during the progress of time, opportunities for new applications will arise.

Continual development is an important outcome of this project. Version 1.3, which will provide MSY calculations, convergence tests, selectivity estimation, length dependent mortality and fitting to individual seasonal standard errors in abundance, is due for release late during 2004.

The Software Design Specification is included as Appendix III, Part 4 and is released on the software CD in the documentation folder. There are several avenues of model modification or development, allowing for differing levels of experience, or differing requirements. These are outlined in the Software Design Specification in more detail but can be summarised as:

- An ADMB template version that can be readily modified is included as AbaloneModel.tpl.

- Victoria is developing a system of using the GUI to develop input files and test the data & starting parameters, then using a "simple" text-output version (equivalent to the template version) to run under Linux on a fast computing facility. Text outputs from the model are input into a results database using an MS-Access form. Additional inputs can be read into the modified program using a secondary input file (without need to modify the GUI). The code is included as ModelConsole.

- The model (abalone life history) is confined to one file (AbaloneModel.cpp or the template), so that model changes, as opposed to GUI or database changes, can be relatively straightforward for a user with ADMB or basic C++ experience.

One of the main areas in which the model could be modified would be to simulate spatial allocation of catch in response to local productivity. At the moment the model can be run separately for a series of adjacent geographic regions or statistical blocks at a particular scale. However, there is no way of using the modelled outcomes from one or more of these blocks to test the likely effects on future outcomes from other blocks, other than to manually re-set and run the model for these other blocks. It would be interesting to see would happen if this was automated so that, according to a set of 'criteria for fishing behavioural responses', effort was automatically re-distributed in response to the relative biomass estimates among blocks. This kind of modification would not be trivial and is akin to incorporating stock migration patterns into fisheries models.

PIRVic has undertaken to maintain the model and as such will be a central repository for receiving feedback about its performance and potential improvements, and maintaining a common version of the software. This information will be periodically disseminated to interested users. However, any substantial modifications or development to meet particular agency needs in the future will be the responsibility of that agency.

Planned Outcomes

The AbModeller software and a User Manual have been transferred to individual States so that each can conduct its own testing and its modellers can familiarise themselves with the software in the context of their specific data holdings. The timing for adoption and the mode of application will vary depending State needs and readiness.

The current situation regarding technology transfer and assessment by each State is:

- South Australia used the model when data became available after the completion of the project. The abalone team, rather than specialist modellers, rapidly developed a preliminary assessment, along with multiple alternative scenarios, making good use of the model selection and alternative data set selection capabilities in the GUI.
- Victoria re-coded and redeveloped its data sets in 2003. It was decided to use the pre-existing model for the 2003 fisheries assessment, but there is an expectation from the Fisheries Assessment Group and the Abalone Fishery Committee that the national model, along with updated data sets and extra modelling regions, will be used in 2004.
- NSW has the technology. A major induction session was held during August 2003 to support the technology hand-over. The capacity to include habitat reduction was added at NSW request from that meeting, and MSY calculations are currently being added at request of NSW and Victoria. NSW has expressed a desire to change over to the national model when the time and modelling personnel become available.
- Tasmania: FRDC data projects are underway. Suitable data for the model had not been available to date, but is expected from these projects.
- WA has had an earlier version installed and demonstrated, and they have the current version. They intend to trial the model before the end of 2004

Conclusion

This project brought together experienced fisheries scientists from a range of different disciplines to form a multi-discipline group, many with overlapping skill portfolios. The initial objective was to develop an appropriate modelling framework to incorporate all relevant biological information into a spatially structured assessment of the likely risk and performance of different future TACs for an abalone fishery. It was intended that this would provide the basis for the second objective of developing the software necessary to complete such an assessment. Inevitably there were differences in opinion about how the development of the model should proceed that ranged from choice of programming language to components of the biological framework. Critical to the project's success was the need for the model to be an integrated product rather than a collage of crudely joined blocks of disparate computer code. Without consensus among the participants this would have been an unrealistic expectation.

Above all we wanted a professional quality product that would lend itself to being used because of its accessibility whilst still retaining the robust statistical capabilities of a fully stochastic Bayesian model. A corollary to the user friendliness or simplicity of access in the final product was the relatively high level of sophistication of the language chosen and the complexity of computer code written in this language. This contrasts with most contemporary fisheries assessment models that are generally specific to a particular fishery and accessible only to highly trained and experienced modellers. The adoption of a MS Windows® GUI shell is an outstanding feature of the model not only because it makes its accessible to a broader range of fisheries scientists, but importantly because it substantially increases the efficiency of completing and tracking computing runs in large multiples. The GUI presents outputs in an easily interpretable format that reduces the time spent having to collate and summarise the results for presentation.

The remaining two objectives were to test and evaluate the capabilities of the modelling framework and apply the model for preliminary assessments of each States abalone fisheries. These seemingly straightforward objectives were necessarily subordinate to the first two objectives. However, their attainment presented several obstacles, most prominently the lack of availability of some data types required to run the model. Only NSW and Vic had all types of data necessary to conduct model-based assessments. This was identified at an early stage of the project and the steering committee resolved that the pursuit of the final objective should reside with individual States.

A testing and simulation capability has been built into the model with the ability to simulate and analyse results. Extensive testing and basic simulations were completed using Victorian data, yet more work is required to comprehensively evaluate the model. However, there is now a powerful structure in place for ongoing testing that will continue beyond the duration of this project. Only by applying the model to various real assessment tasks can its limitations be fully revealed and appropriate modifications made in response.

The adoption of the model by States other than Victoria has been slower than anticipated as a consequence of competing research and assessment priorities. South Australian researchers have successfully conducted a trial application on their Southern Zone blacklip abalone fishery. NSW still intends to transfer to using AbModeller as soon as is practicable and Western Australia is now in a position to trial the model during the latter half of 2004. Tasmania conducted some successful but limited early trials and is currently awaiting outcomes of other FRDC projects before embarking upon an expansion of their current fishery monitoring that would support future modelling.

It is our contention that the model we have developed exceeds the initial expectations for a robust and flexible item of software that can be readily adapted and applied to a varying State needs and abalone species with different population parameters.

References

Bardos, D.C. 2004. Probabilistic Gompertz model of irreversible growth. *Bulletin Mathematical Biology (*accepted*)*.

Breen, P.A., S.W. Kim & N.L. Andrew. 2003. A length-based Bayesian model for the New Zealand abalone *Haliotis iris*. *Mar. & Freshwater Res*., 54: 619–634.

Gorfine, H.K., B.L. Taylor & T.I. Walker. 2001. Triggers and targets: What are we aiming for with abalone fisheries models in Australia? *J. Shellfish Res*., 20: 803–811.

Mayfield, S., Saunders, T., Dixon, C.D. & Ward, T.M. 2004. Southern Zone Abalone (*Haliotis rubra* and *H. laevigata*) Fishery. Fishery assessment report to PIRSA Fisheries. South Australian Fisheries Assessment Series. Publication No. RD04/0091. 102pp.

Plagányi, É.E., D.S. Butterworth & A. Brandão. 2001. Toward assessing the South African abalone Haliotis midae stock using an age-structured production model. *J. Shellfish Res*., 20: 813–828.

Punt, A.E. & R.B. Kennedy. 1997. Population modelling of Tasmanian rock lobster, *Jasus edwardsii*, resources. *Mar. & Freshwater Res*., 48: 967–980.

Schnute, J.T. 1981. A versatile growth model with statistically stable parameters. *Can. J. Fish. & Aquatic Sc*., 38: 1128–1140.

Schnute, J.T. & L.J. Richards. 1990. A unified approach to the analysis of fish growth, maturity, and survivorship data. *Can. J. Fish. & Aquatic Sc*., 47: 24–40.

Sierra-Rodríguez, P., M. Muciño-Díaz & A. Vélez-Barajas. 2003. Confronting uncertainty and risk in reference points used in the Mexican abalone stock assessment: a study case with *Haliotis fulgens* in Bahía Tortugas. Oral presentation abstract from 5th International Abalone Symposium. p. 97. Program & Abstracts. Ocean University of China, Qingdao. Unpublished. 211 pp.

Troynikov, V.S. 1998. Probability density functions useful for parameterization of heterogeneity in growth and allometry data. *Bull. Math. Biol*., 60: 1099–1122.

Worthington, D.G. 1997. Demography and dynamics of the population of blacklip abalone, *Haliotis rubra*, with implications for management of the fishery in NSW, Australia. Ph.D. thesis, Macquarie University, Sydney, Australia.

Appendix 1: Intellectual Property

The intellectual property will be shared between PIRVic and FRDC.

Appendix 2: Staff

Name	Project Role	Position	Organisation
Harry Gorfine	Principal Investigator	Abalone Project Manager	PIRVic—Marine & Freshwater Systems
Bruce Taylor	Modeller	Fisheries Modeller(from February 2002)	PIRVic—Marine & Freshwater Systems
Ian Montgomery	Modeller	Abalone Modeller(July 2000 – February 2002)	PIRVic—Marine & Freshwater Systems
Michael Cleland	Contract Programmer	Private Consultant	Self-employed. 10 McNaught Ave, Point Lonsdale, Victoria, Australia, 3225.
David Bardos	Modeller	Private Consultant	Ecological Dynamics, 126 Amess St, Carlton North, Victoria, Australia, 3054.

Appendix 3: Other relevant material

PART 1. Model testing

PART 2. AbModeller Specifications

PART 3. AbModeller— User Manual version 1.2.1

PART 4. AbModeller— Software Documentation

PART 5. CD Software package: Software, Model code, User Manual and Documentation and sample database

Part 1: Model testing

A powerful testing capability has been built into the model with the ability to simulate and analyse results. This involves MS Access® queries, forms, and modules to support this process - similar to that for analysing Bayesian outputs.

The functionality and reliability of the model was evaluated using basic simulation testing. This initial testing was conducted in accordance with the priorities of the steering committee meeting of June 2002, specifically,

- Internal consistency of the model in recovering original parameter values whilst allowing for error in the observed data.
- Effects of mis-specification of priors (this was initially done for mortality).
- Alternative weighting given to different observed indices of the population.

The simulation structure of the model will facilitate ongoing testing beyond the duration of this project. Results from future testing in Victoria will be reported to other States on an ongoing basis, however the conduct of other State-specific testing will depend on further funding, because it can be both computationally intensive and time consuming.

Method

First model test - allowing for error in the observed data

Data sets

Two different catch data sets, each covering 36 years of fishing, were used. The catch data sets, and model parameters, were similar to those for the western region and part of the eastern region of Victoria, as were the assumed number of population length frequency samples in each year. These were used both with and without assumed stochastic recruitment

The abalone model was used to generate a data set (data generation model), consisting of fishery independent abundance of under-size and legal sized abalone, and independent length frequency (Fig. 1). The abundance data was generated for 10 years and the length frequency data for 12 or 14 years.

Monte Carlo simulation

The model was then fitted by minimising the negative log(likelihood), firstly against the generated data, then using a Monte Carlo simulation of 100 data sets with lognormal errors added to the "observed" data (catch, fishery independent abundance and population length frequency) (Fig. 1).

Observation error

The coefficients of variation (CV) for the error terms added to abundance and length frequency were taken from the differences between observed and fitted values from the 2002 Victorian regional stock assessments over a range of regions, namely of 0.1 (minimum CV) and 0.2 (maximum CV) for abundance and 0.2 and 0.3 for length frequency. The CV values for the error terms in catch history were taken as 0.05 and 0.2. One hundred Monte Carlo simulations were run for each of the following five combinations:

- minimum CV on the errors for catch, abundance and length frequency,
- maximum CV on the errors for catch, abundance and length frequency,
- maximum CV on the errors for catch, minimum on the others,
- maximum CV on the errors for abundance, minimum on the others,
- maximum CV on the errors for length frequency, minimum on the others.

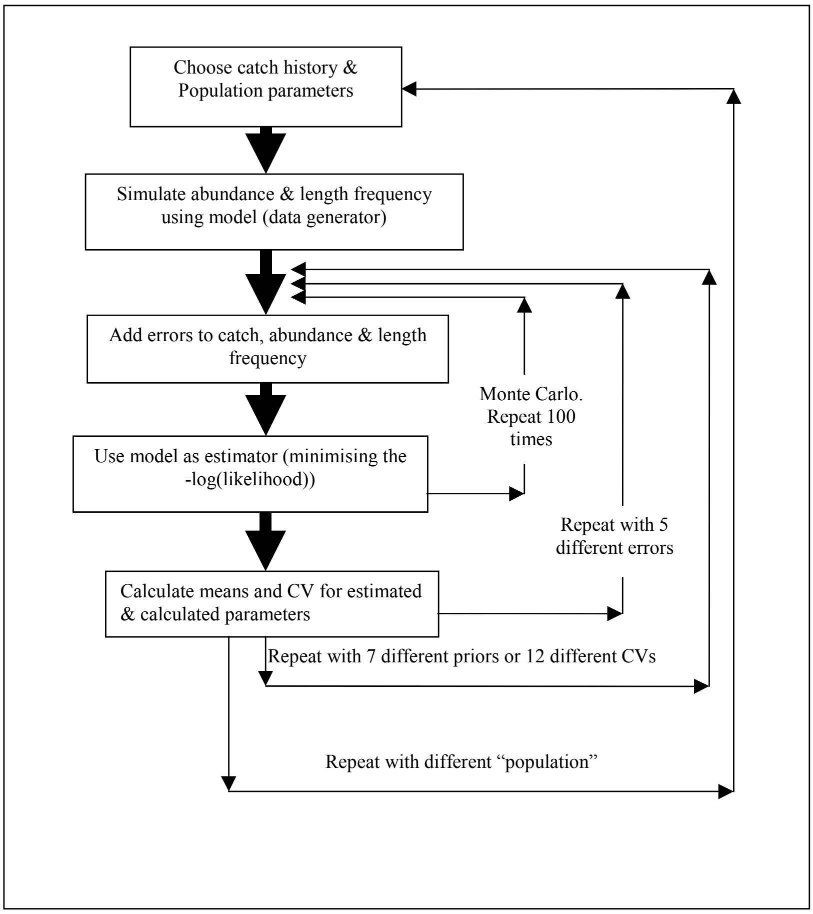

Choose catch history & Population parameters
Simulate abundance & length frequency using model (data generator)
Add errors to catch, abundance & length frequency
Use model as estimator (minimising the -log(likelihood))
Calculate means and CV for estimated & calculated parameters
Monte Carlo. Repeat 100 times
Repeat with 5 different errors
Repeat with 7 different priors or 12 different CVs
Repeat with different "population"

Figure 1. Flow diagram of the model simulation process

Model fitting

When testing for observation error, the likelihood function for independent abundance used the standard deviation of the data, and the independent length frequency likelihood function used an effective sample size of 30 for all the simulations. This sample size was chosen by minimising the contribution from length frequency to the likelihood function, taken from an initial test run. The model was fitted for combinations of the growth matrix and mortality being fixed, having a lognormal prior with a CV of 0.1 or being freely estimated (uniform prior). These were:

- fixed growth and fixed mortality
- fixed growth and lognormal prior on mortality
- fixed growth and fully estimated mortality
- lognormal prior on growth and fixed mortality
- lognormal prior on growth and lognormal prior on mortality
- fully estimated growth and fully estimated mortality
- fully estimated growth and fixed mortality

Fitted parameters

The relative deviation of the mean from the original value and the CV for each set of 100 simulated data, for each of the 35 simulation-estimation combinations, for each of the 5 modelling scenarios was estimated (Table 1). The relative means provide a measure of the bias in the estimated values, and the CV gives a measure of their spread. The derived values of mature biomass output were:

- B_{future}/ $B_{current}$, where the current catch is projected for 15 years

- B_0, the mature virgin biomass
- $B_{current}$, the mature biomass in the current year
- $B_{current}/B_0$,

and the estimated parameter values were:

- log (R_{init}) , the log of the initial recruitment value in the Beverton Holt recruitment equation
- R_{init}, the initial recruitment value in the Beverton Holt recruitment equation, calculated from log (R_{init})
- steepness, h, in the Beverton Holt recruitment equation
- mortality
- g_w, the weighting of the growth matrices (the growth matrix used is the weighted sum of two given matrices)

Projections of relative biomass ($B_{future}/B_{current}$) for constant catch scenarios are often used under management plans (such as in Victoria and NSW) as an important part of the quota setting process. Mature $B_{current}/B_0$ and $B_{current}$ are measures of current population size. The estimated values, and R_{init} , are measures of the estimator models ability to re-estimate the original values. Five scenarios were tested for two different catch histories and for mis-specified mortality and the resulting values were used in the data generation model (Table 1). The catch data sets were similar to those for the western region in Victoria and the central part of the eastern region of Victoria, as were the number of population length frequency samples assumed.

This computer intensive simulation process took the equivalent of more than 47 days of electronic processing time.

Table 1. Population parameters modelled for Test 1 (5 observation-error scenarios) and Test 2 (alternative parameter weighting. (*M* = natural mortality, Sc = scenario).

	Test			
Parameter	**1**			**2**
	Scenarios			
	1 & 2	**3 & 4**	**5** **(*M* mis-specified)**	**(fitting tested)**
R_{init} (million)	8.0	8.4	8.0	8.0
log (R_{init})	15.89	15.94	15.89	15.89
Steepness, h	0.6	0.6	0.6	0.6
Mortality	0.2	0.2	0.12 (0.2)	0.2
Growth weighting, g_w	0.5	0.9	0.5	0.8
L_{50} (mm)	102	102	102	102
W_{idth} (mm)	17	17	17	17
Stochastic recruitment in operational model	Sc#1 No Sc#2 Yes	Sc#3 No Sc#4 Yes	Yes	Yes
Catch data set	Western	Eastern	Western	Western

When mortality is mis-specified, but then freely estimated, it is only the starting value that is mis-specified. These results were included in the tables for comparative purposes.

Second model test – weights on observed indices

For the second series of tests, the effect of changing the relative weighting of the fitted data sets (fishery independent abundance and population length frequency) were tested. The catch and population parameters were those of case 2 above (with western catch history and stochastic recruitment). The method used was as above, with lognormal priors on mortality and growth.

Three sets of weighting were used for fitting the independent abundance, namely CV of 0.01, 0.1 and 0.5, and four sample sizes were used for effective sample sizes for independent length frequency, namely 10, 30, 50 and 100, to produce 12 combinations of outputs.

Results

Observation error

For the first series of tests, scenarios 1–5, the following estimates of mature biomass and population parameters were obtained.

$B_{15}/B_{current}$

The mean of the projected relative biomass for each of the 174 sets of simulations is in the range of ± 13% of the value of $B_{15}/B_{current}$ in the operational model. With mortality mis-specified, all the means were within ± 4%. The CVs ranged from 9–31% (Table 2.1).

B_0

For model 1 data (with and without stochastic recruitment), the means of B_0 for each of the simulation runs were 14–24% below the operational value. For model 2 data and no stochastic recruitment, or with growth fixed, the means were all less than 12% of the original value, but with stochastic recruitment and growth freely estimated, or with large priors on mortality and growth, the means can be much larger than the original, and the CVs generally were large (Table 2.2).

$B_{current}$

Generally, mean values of the sets of simulations were 40–60% lower than the original values, and CVs were 12–40%. As with B_0, for model 2 data and with stochastic recruitment and growth freely estimated, or with large priors on mortality and growth, the means can be much larger than the original, and the CVs generally were large (Table 2.3).

$B_{current}/B_0$

The mean values from simulation sets were generally 35–60% below the original operational value. The large differences sometimes found in $B_{current}$ and B_0 cancel out here, though the sets that had large values have larger CVs (Table 2.4).

$\log(R_{init})$

A very good estimator. Differences in the mean were small, as were CVs (Table 2.5).

R_{init}

Once $\log(R_{init})$ is exponentiated, more variation can be seen (Table 2.6).

Steepness

For the western based data sets, the mean is generally 15–20% above the original value, with CVs of 6–19%. For eastern data sets, steepness is generally –2 to 5% of the original, with small CVs for no assumed stochastic recruitment, and 9–19% with stochastic recruitment. When mortality is freely estimated, the deviations of the means from the original were higher, and the CVs were mainly 13–32%. When mortality is mis-specified, the estimates of h were at or near the upper bound (large deviation and small CV) (Table 2.7).

Table 2.1 Mature $B_{15}/B_{current}$

	Model 1 No stochastic recruitment	Model 1 Stochastic recruitment	Model 2 No stochastic recruitment	Model 2 Stochastic recruitment	Model 1 Mortality mis-specified
Value from operational	99 %	93 %	109 %	99 %	93 %
Estimated (as percentage of operational	100 %	104 %	100 %	111 %	102 %

Mean estimated value from each set of 100 simulated data (as percentage difference to operational value)
(simulation 0 has no data error)

Estimation	Simulation data error						Simulation data error						Simulation data error						Simulation data error						Simulation data error					
run	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5
1	0	-9	-10	-9	-9	-9	5	1	-2	1	-2	2	0	-5	-14	-9	-10	-5	5	5	-7	4	-6	3	9	-1	0	1	-1	1
2	1	-9	-10	-9	-9	-8	2	1	-3	-1	-2	1	1	-5	-13	-8	-9	-5	23	5	-9	1	-8	3	-4	0	-1	2	-2	0
3	0	-6	-7	-4	-6	-6	-3	3	1	0	2	3	1	4	0	-1	2	3	28	11	2	5	3	9	-2	3	2	1	2	3
4	0	-9	-10	-9	-9	-8	14	2	-3	1	-2	1	0	-5	-13	-9	-9	-4	5	9	-1	6	-1	11	8	0	0	1	-2	-1
5	0	-9	-10	-9	-9	-9	7	2	-3	1	-2	1	0	-5	-13	-9	-9	-4	5	11	2	7	4	12	8	-3	-2	2	-4	
6	0	-9	-10	-9	-9	-8	1	1	-5	0	-3	1	-1	-5	-12	-8	-8	-4	4	7	-4	5	-3	6	-4	0	0	2	-2	0
7	0	-7	-9	-6	-6	-7	2	-1	-2	-1	-2	1	0	4	0	0	1	4	4	13	7	9	7	9	3	-1	-2	0	-1	1
Data	Nil	All	All	Catch	Abund	LF	Nil	All	All	Catch	Abund	LF	Nil	All	All	Catch	Abund	LF	Nil	All	All	Catch	Abund	LF	Nil	All	All	Catch	Abund	LF
error		small	large	large	large	large		small	large	large	large	large		small	large	large	large	large		small	large	large	large	large		small	large	large	large	large

CV of estimated values from each set of 100 simulated data
(as percentage of mean value)

Estimation	Simulation data error					Simulation data error					Simulation data error					Simulation data error					Simulation data error				
run	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	11	16	16	10	12	13	19	15	15	14	19	26	24	20	20	23	31	24	29	23	11	17	18	9	10
2	11	16	16	10	12	14	19	17	16	14	19	26	24	20	20	23	33	28	32	23	10	19	17	10	10
3	11	16	13	9	10	11	16	19	12	11	17	23	23	15	18	19	29	25	24	21	11	15	16	11	11
4	11	16	16	10	12	13	19	15	15	14	16	24	22	18	17	19	25	20	25	17	11	17	18	9	12
5	10	16	16	9	11	12	19	17	13	13	16	24	22	18	17	21	29	22	28	20	11	20	19	9	
6	11	16	16	10	12	13	19	17	16	15	16	24	22	18	17	19	28	21	26	19	10	17	17	9	10
7	10	15	14	9	11	17	21	20	18	14	14	22	20	14	14	18	25	21	21	20	18	22	18	13	14

Data error (cv)	Small	Large
Catch	0.05	0.20
Abundance	0.10	0.20
Length freq	0.20	0.30

Mortality, growth fit	Large
CV on the	0.20
prior	0.20
	0.30

Table 2.2 Mature B_0 (,000 tonnes)

	Model 1 No stochastic recruitment	Model 1 Stochastic recruitment	Model 2 No stochastic recruitment	Model 2 Stochastic recruitment	Model 1 Mortality mis-specified
Value from operational model	5.4	5.4	7.5	7.5	5.4
Estimated value (as percentage of operational value)	100 %	97 %	100 %	104 %	96 %

Mean estimated value from each set of 100 simulated data (as percentage difference to operational value)
(simulation 0 has no data error)

Estimation	Simulation data error type						Simulation data error type						Simulation data error type						Simulation data error type						Simulation data error type					
run	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5
1	0	-23	-20	-20	-23	-23	-4	-22	-19	-20	-22	-22	0	-12	-7	-7	-12	-12	11	-8	-2	0	-4	-8	-2	-10	-8	-8	-10	-10
2	0	-23	-20	-20	-23	-23	-1	-22	-18	-18	-21	-21	0	-12	-7	-7	-11	-12	-2	-6	4	-1	-2	-6	-12	-11	-9	-10	-10	-11
3	0	-21	-18	-21	-19	-21	-8	-17	-14	-14	-15	-17	0	1	4	0	3	1	7	6	11	7	9	7	-1	-17	-14	-14	-16	-17
4	0	-23	-20	-20	-23	-23	2	-22	-19	-20	-22	-22	0	-9	-4	-4	-9	-9	4	25	16	0	13	24	1	-10	-8	-8	-10	-10
5	0	-24	-22	-22	-24	-24	-4	-24	-21	-22	-24	-24	0	-9	-4	-4	-9	-9	4	131	91	44	117	71	-1	-12	-9	-10	-12	
6	0	-23	-20	-21	-23	-23	-5	-22	-18	-18	-21	-21	3	-9	-4	-4	-8	-9	1	36	41	38	0	-3	-12	-11	-9	-10	-11	-11
7	0	-22	-21	-25	-20	-22	-2	-18	-17	-15	-18	-19	1	4	7	4	5	4	2	368	294	452	330	208	-2	-17	-16	-16	-18	-19
Data error	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large

CV of estimated values from each set of 100 simulated data
(as percentage of mean value)

Estimation	Simulation data error type					Simulation data error type					Simulation data error type					Simulation data error type					Simulation data error type				
run	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	4	7	7	4	4	6	10	9	7	12	2	6	6	3	3	5	8	25	17	5	2	5	4	2	3
2	4	8	7	4	4	6	12	11	9	8	2	6	6	3	3	17	36	15	15	10	3	6	6	2	4
3	7	9	7	6	7	7	11	12	10	8	5	7	9	2	5	11	8	15	6	16	8	11	12	10	8
4	4	7	7	4	4	10	10	12	7	6	2	6	6	2	2	156	125	13	125	163	2	5	4	3	2
5	4	7	7	4	4	4	8	7	6	4	2	6	6	2	2	188	188	179	190	194	2	5	4	2	
6	4	7	7	4	4	5	11	13	8	16	2	6	6	3	2	289	280	284	16	12	3	6	6	3	3
7	8	9	8	7	8	13	16	19	11	9	3	6	7	2	3	246	261	228	254	285	14	16	19	7	9

Data error (cv)	Small	Large
Catch	0.05	0.20
Abundance	0.10	0.20
Length freq	0.20	0.30

Mortality, growth fit	Large
CV on the	0.20
prior	0.20
	0.30

Table 2.3 Mature $B_{current}$ (,000 tonnes)

	Model 1 No stochastic recruitment	Model 1 Stochastic recruitment	Model 2 No stochastic recruitment	Model 2 Stochastic recruitment	Model 1 Mortality mis-specified
Value from operational model	2.5	3.1	3.2	4.1	3.1
Estimated value (as percentage of operational value)	100 %	93 %	100 %	98 %	76 %

Mean estimated value from each set of 100 simulated data (as percentage difference to operational value)
(simulation 0 has no data error)

Estimation	Simulation data error type						Simulation data error type						Simulation data error type						Simulation data error type						Simulation data error type					
run	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5
1	0	-52	-49	-49	-52	-51	1	-55	-52	-54	-56	-53	0	-53	-50	-49	-54	-53	-3	-54	-51	-39	-49	-55	-40	-62	-61	-61	-62	-61
2	0	-52	-49	-49	-52	-51	-9	-55	-51	-53	-55	-54	0	-54	-50	-49	-54	-53	-9	-52	-37	-47	-48	-53	-16	-62	-60	-61	-62	-61
3	0	-47	-42	-27	-54	-44	-15	-50	-45	-31	-55	-50	0	-58	-55	-52	-59	-58	-9	-58	-57	-49	-61	-54	-15	-50	-46	-30	-56	-49
4	0	-53	-49	-50	-53	-51	0	-55	-53	-51	-56	-55	1	-47	-43	-43	-47	-46	2	2	-23	-36	-20	-10	-36	-62	-62	-61	-63	-61
5	0	-59	-56	-56	-59	-58	-2	-61	-59	-59	-61	-60	1	-47	-43	-43	-47	-46	2	114	62	10	92	42	-35	-67	-66	-65	-67	
6	1	-53	-49	-50	-53	-51	-11	-56	-53	-54	-57	-53	-2	-47	-43	-43	-48	-46	2	10	16	13	-43	-44	-16	-62	-62	-61	-63	-61
7	1	-57	-49	-39	-61	-55	-10	-57	-52	-37	-62	-56	0	-54	-50	-48	-55	-53	2	471	358	593	409	241	-9	-57	-50	-39	-62	-56
Data error	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large

CV of estimated values from each set of 100 simulated data
(as percentage of mean value)

Estimation	Simulation data error type					Simulation data error type					Simulation data error type					Simulation data error type					Simulation data error type				
run	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	14	16	14	16	14	17	23	18	20	43	14	17	14	17	14	31	41	123	85	25	10	14	12	14	13
2	14	16	14	16	14	14	22	18	28	19	14	17	14	17	14	56	173	85	56	31	15	18	16	17	16
3	28	37	33	23	36	42	53	49	40	43	17	20	20	19	17	51	66	80	58	133	43	55	48	39	42
4	14	17	14	17	14	42	19	58	19	18	12	16	13	15	12	233	213	63	214	255	13	14	13	15	11
5	16	18	16	19	16	18	19	17	20	18	12	16	13	15	12	229	242	271	242	258	12	16	14	16	
6	14	17	14	17	14	16	19	20	18	53	12	16	13	15	12	503	491	474	52	46	16	18	16	18	17
7	25	32	28	23	27	40	45	43	31	44	15	18	18	17	15	294	325	262	310	373	40	48	43	38	44

Data error (cv)	Small	Large
Catch	0.05	0.20
Abundance	0.10	0.20
Length freq	0.20	0.30

Mortality, growth fit	Large
CV on the	0.20
prior	0.20
	0.30

Table 2.4 Mature $B_{current}/B_0$

	Model 1 **No stochastic recruitment**	**Model 1** **Stochastic recruitment**	**Model 2** **No stochastic recruitment**	**Model 2** **Stochastic recruitment**	**Model 1** **Mortality mis-specified**
Value from operational model	0.5	0.6	0.4	0.5	0.6
Estimated value (as percentage of operational value)	100 %	96 %	100 %	95 %	80 %

Mean estimated value from each set of 100 simulated data (as percentage difference to operational value)
(simulation 0 has no data error)

Estimation	Simulation data error type						Simulation data error type						Simulation data error type						Simulation data error type						Simulation data error type					
run	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5
1	0	-38	-35	-36	-38	-36	5	-42	-40	-42	-44	-40	0	-47	-46	-45	-48	-47	-13	-50	-50	-44	-49	-51	-39	-57	-58	-57	-58	-57
2	0	-38	-35	-36	-38	-36	-8	-43	-40	-41	-44	-42	0	-47	-46	-45	-48	-47	-7	-51	-47	-48	-48	-50	-5	-57	-57	-57	-57	-57
3	0	-32	-29	-6	-43	-28	-8	-40	-36	-18	-47	-39	0	-58	-56	-52	-61	-58	-15	-60	-61	-53	-64	-59	-14	-40	-36	-18	-47	-38
4	0	-39	-36	-37	-38	-37	-2	-43	-41	-40	-43	-42	0	-42	-40	-40	-43	-41	-3	-32	-40	-38	-38	-40	-36	-58	-59	-57	-59	-57
5	0	-45	-43	-43	-45	-44	2	-48	-47	-47	-49	-47	0	-42	-40	-40	-43	-41	-2	-34	-36	-41	-39	-37	-34	-62	-62	-61	-63	
6	0	-39	-36	-36	-39	-37	-6	-44	-42	-43	-45	-42	-4	-42	-41	-40	-43	-41	1	-43	-41	-39	-44	-43	-4	-58	-58	-57	-58	-57
7	0	-44	-35	-18	-50	-42	-9	-47	-41	-24	-53	-45	0	-56	-53	-50	-57	-55	0	-48	-49	-40	-50	-48	-8	-47	-40	-26	-53	-46
Data error	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large

CV of estimated values from each set of 100 simulated data
(as percentage of mean value)

Estimation	Simulation data error type					Simulation data error type					Simulation data error type					Simulation data error type					Simulation data error type				
run	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	11	19	14	15	12	15	23	17	20	22	12	18	15	16	12	28	36	53	44	25	9	14	11	12	11
2	11	19	14	15	12	14	22	20	21	17	12	18	15	16	13	27	57	47	45	25	14	18	16	17	15
3	31	39	33	26	37	42	49	48	40	44	20	24	25	21	20	41	61	53	50	62	43	51	46	40	44
4	11	19	14	15	12	22	21	33	19	16	11	17	14	15	11	56	37	44	43	30	10	14	11	13	10
5	12	18	15	15	13	15	20	16	18	15	11	17	14	15	11	49	49	47	46	36	10	15	12	14	
6	11	19	14	15	12	14	22	19	18	24	11	17	14	15	11	32	46	43	34	30	15	18	15	18	16
7	30	35	30	25	31	42	45	43	33	45	17	22	23	19	18	66	62	67	72	62	41	47	43	36	44

Data error (cv)	Small	Large
Catch	0.05	0.20
Abundance	0.10	0.20
Length freq	0.20	0.30

Mortality, growth fit	Large
CV on the	0.20
prior	0.20
	0.30

Table 2.5 In(R_{init})

	Model 1 No stochastic recruitment	Model 1 Stochastic recruitment	Model 2 No stochastic recruitment	Model 2 Stochastic recruitment	Model 1 Mortality mis-specified
Value from operational model	15.9	15.9	15.9	15.9	15.9
Estimated value (as percentage of operational value)	100 %	99 %	100 %	100 %	96 %

Mean estimated value from each set of 100 simulated data (as percentage difference to operational value)
(simulation 0 has no data error)

Estimation	Simulation data error type						Simulation data error type						Simulation data error type						Simulation data error type						Simulation data error type					
run	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5
1	0	-2	-1	-1	-2	-2	0	-2	-1	-1	-2	-2	0	-1	0	0	-1	-1	1	-1	0	0	0	-1	-6	-6	-6	-6	-6	-6
2	0	-2	-1	-1	-2	-2	-1	-2	-1	-1	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-3	-6	-6	-6	-6	-6
3	0	-1	0	5	-4	0	-3	-2	-1	3	-4	-2	0	-6	-5	-4	-7	-6	-1	-5	-5	-3	-6	-5	-2	-2	-2	3	-3	-2
4	0	-2	-1	-2	-2	-2	0	-2	-1	-1	-2	-2	0	0	0	0	0	0	1	0	1	0	0	0	-5	-6	-6	-6	-6	-6
5	0	-3	-2	-2	-3	-3	-1	-3	-2	-2	-2	-3	0	0	0	0	0	0	1	1	1	1	1	1	-5	-7	-7	-7	-7	
6	0	-2	-1	-1	-2	-2	-2	-2	-2	-2	-2	-2	0	0	0	0	0	0	1	0	0	0	0	0	-3	-6	-6	-6	-6	-6
7	0	-2	0	3	-5	-2	-1	-3	-2	2	-4	-3	0	-6	-5	-4	-6	-6	1	-3	-4	-2	-4	-4	-1	-3	-2	2	-5	-3
Data error	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large

CV of estimated values from each set of 100 simulated data
(as percentage of mean value)

Estimation	Simulation data error type					Simulation data error type					Simulation data error type					Simulation data error type					Simulation data error type				
run	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
2	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	1	2	1	1	1	0	0	0	0	0
3	5	5	5	4	5	6	5	6	4	6	3	2	4	1	3	4	3	4	3	4	6	6	6	5	6
4	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	2	1	1	1	2	0	0	0	0	0
5	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	4	3	2	4	3	0	0	0	0	
6	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	2	2	2	1	1	0	0	0	0	0
7	5	5	4	4	5	5	5	6	3	5	2	2	3	1	2	6	5	7	6	5	5	5	6	3	5

Data error (cv)	Small	Large
Catch	0.05	0.20
Abundance	0.10	0.20
Length freq	0.20	0.30

Mortality, growth fit	Large
CV on the	0.20
prior	0.20
	0.30

Table 2.6 R$_{init}$ (milllion)

	Model 1 No stochastic recruitment	Model 1 Stochastic recruitment	Model 2 No stochastic recruitment	Model 2 Stochastic recruitment	Model 1 Mortality mis-specified
Value from operational model	8.0	8.0	8.4	8.4	8.0
Estimated value (as percentage of operational value)	101 %	85 %	100 %	104 %	58 %

Mean estimated value from each set of 100 simulated data (as percentage difference to operational value)
(simulation 0 has no data error)

Estimation	Simulation data error type						Simulation data error type						Simulation data error type						Simulation data error type						Simulation data error type					
run	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5
1	0	-23	-20	-20	-23	-23	-4	-22	-19	-20	-22	-22	0	-12	-7	-7	-12	-12	11	-8	-2	0	-4	-8	-59	-63	-62	-62	-63	-63
2	1	-23	-20	-20	-23	-23	-17	-23	-20	-20	-22	-23	0	-13	-8	-8	-13	-13	-16	-10	0	-7	-10	-9	-39	-62	-61	-61	-62	-62
3	1	25	34	164	-31	38	-34	21	30	157	-17	18	0	-55	-55	-33	-64	-54	-9	-43	-50	-16	-49	-39	-30	23	31	158	-17	20
4	0	-24	-21	-21	-24	-24	2	-23	-20	-21	-22	-23	1	0	6	6	1	0	12	20	17	9	13	20	-56	-63	-62	-62	-63	-63
5	0	-33	-31	-31	-33	-33	-10	-33	-30	-31	-33	-33	1	0	6	6	1	0	11	78	59	32	72	44	-56	-66	-65	-65	-67	
6	3	-24	-21	-20	-24	-24	-24	-24	-21	-22	-24	-25	-3	-2	4	4	-2	-1	8	5	13	9	-2	0	-38	-63	-61	-61	-63	-62
7	3	-10	22	103	-41	-7	-21	-8	12	112	-40	0	0	-58	-56	-38	-61	-58	10	30	9	63	21	6	-19	-11	15	101	-43	-4
Data	Nil	All	All	Catch	Abunc	LF	Nil	All	All	Catch	Abunc	LF	Nil	All	All	Catch	Abunc	LF	Nil	All	All	Catch	Abunc	LF	Nil	All	All	Catch	Abunc	LF
error		small	large	large	large	large		small	large	large	large	large		small	large	large	large	large		small	large	large	large	large		small	large	large	large	large

CV of estimated values from each set of 100 simulated data
(as percentage of mean value)

Estimation	Simulation data error type					Simulation data error type					Simulation data error type					Simulation data error type					Simulation data error type				
run	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	4	8	7	4	4	6	10	9	7	12	2	6	6	3	3	5	8	25	17	5	2	5	4	2	3
2	4	8	7	4	4	7	10	10	18	10	3	6	6	3	3	11	57	19	18	10	7	8	7	7	7
3	81	94	61	85	93	128	128	87	137	126	59	53	67	23	62	137	121	125	197	129	129	132	87	136	124
4	4	7	7	4	4	12	10	13	7	7	2	6	6	2	2	100	73	12	75	100	3	5	5	3	3
5	5	8	8	5	5	4	8	8	6	5	2	6	6	2	2	137	127	110	134	129	3	6	7	4	
6	4	8	7	4	4	8	12	13	11	8	2	6	6	3	2	82	76	80	14	14	7	7	7	7	7
7	77	74	51	79	76	112	105	80	91	118	38	34	67	23	38	201	214	180	214	206	115	108	82	90	119

Data error (cv)	Small	Large
Catch	0.05	0.20
Abundance	0.10	0.20
Length freq	0.20	0.30

Mortality, growth fit	Large
CV on the	0.20
prior	0.20
	0.30

Table 2.7 Steepness (h)

	Model 1 No stochastic recruitment	Model 1 Stochastic recruitment	Model 2 No stochastic recruitment	Model 2 Stochastic recruitment	Model 1 Mortality mis-specified
Value from operational model	0.6	0.6	0.6	0.6	0.6
Estimated value (as percentage of operational value)	100 %	134 %	100 %	114 %	144 %

Mean estimated value from each set of 100 simulated data (as percentage difference to operational value)
(simulation 0 has no data error)

Estimation	Simulation data error type						Simulation data error type						Simulation data error type						Simulation data error type						Simulation data error type					
run	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5
1	0	15	15	14	15	15	41	19	20	20	18	21	0	5	0	2	4	4	-10	2	-2	1	-1	1	57	57	54	56	57	58
2	0	15	15	14	15	16	25	19	19	18	18	20	1	5	1	2	4	5	45	1	-3	0	0	0	53	58	54	56	57	57
3	0	18	17	-11	40	18	38	35	34	12	42	36	1	49	48	32	55	48	15	41	43	26	47	38	15	35	37	14	43	35
4	1	15	15	15	15	16	34	20	19	21	19	21	1	2	-2	-1	1	2	8	5	-2	2	0	2	49	58	54	56	57	57
5	1	18	17	17	18	18	50	22	21	22	21	22	1	2	-2	-1	1	2	10	8	1	1	4	5	57	58	55	57	58	
6	-1	15	15	14	16	16	31	20	18	19	18	20	-6	3	-1	0	2	3	17	3	-1	3	-1	1	54	58	55	56	57	58
7	-1	25	11	-11	42	24	19	32	23	2	40	35	0	54	52	41	56	54	14	48	48	42	52	45	22	33	25	5	44	36
Data error	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large

CV of estimated values from each set of 100 simulated data
(as percentage of mean value)

Estimation	Simulation data error type					Simulation data error type					Simulation data error type					Simulation data error type					Simulation data error type				
run	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	5	14	12	7	6	10	19	16	13	12	4	8	7	5	4	12	16	13	15	12	3	5	4	3	1
2	5	14	12	7	6	11	20	16	14	12	4	8	7	5	4	9	16	13	15	10	2	6	4	3	2
3	32	30	28	21	31	24	22	31	20	23	15	14	24	6	16	20	15	27	13	21	25	21	31	20	23
4	5	14	12	7	6	10	18	15	13	11	4	8	7	5	4	15	16	14	16	14	2	5	4	3	3
5	4	11	10	5	5	6	15	12	9	8	4	8	7	5	4	19	19	14	19	16	0	6	2	1	
6	5	14	12	7	6	10	19	16	13	12	4	8	7	5	4	10	16	14	11	11	1	5	4	3	1
7	29	31	29	20	29	26	26	31	20	26	9	10	20	6	9	15	12	19	9	16	26	26	31	18	25

Data error (cv)	Small	Large
Catch	0.05	0.20
Abundance	0.10	0.20
Length freq	0.20	0.30

Mortality, growth fit	Large
CV on the	0.20
prior	0.20
	0.30

Weighting on the growth matrices (g_w)
Small deviations of the means when a prior is on the growth weight, or for eastern data sets. Large deviations and CVs for western data and freely estimated growth weight (Table 2.8).

Mortality
There is a large deviation and CV for freely estimated mortality, and smaller deviations of the mean, and CVs, when there is a prior on the mortality (Table 2.9).

Second model test – weights on observed indices

For the second series of tests, changing the relative weighting on fitted data sets produced the following estimates of mature biomass and population parameters.

$B_{15}/B_{current}$
For medium and large CV (01. and 0.5) on abundance fits, and N_{lf}=30, 50 or 100, the deviations of the means from the starting value were ≤ 9%, with CVs of 5–14% for N_{lf}=100 rising to 11–24% for N_{lf}=30. For fitting with small CV on abundance, deviations of the means were in the range 9–26% with CVs of 14–38% (Table 3.1).

B/B_0 , $B_{current}$ and B_0
The smallest deviations of the means were for N_{lf}=100 and medium and large CV on abundance fits, then progressively increasing as N_{lf} becomes smaller. Similarly, the CV increases as N_{lf} decreases (Table 3.1).

negative log likelihood
The smallest CVs were for medium and large abundance CVs (Table 3.1).

log(R_{init}) and R_{init}
The smallest deviations of the means were for N_{lf}=100 and medium and large CV on abundance fits, then progressively increasing as N_{lf} becomes smaller. Similarly, the CV increases as N_{lf} decreases (Table 3.2).

Steepness
Poorly estimated, with means and CVs marginally improving with smaller $N_{lf.}$ (Table 3.2).

Mortality
With the prior on mortality, the mean of the simulated data sets were very close to the original value, with very little CV, for N_{lf}=100 or 50 and medium and large CV on abundance fits (Table 3.2).

Growth (g_w)
Small deviations of the mean and small CVs for medium and large CV on abundance fits (Table 3.2).

Discussion and Conclusions

Observation error

The main outcomes from the first series of tests were:

- $B_{15}/B_{current}$ (mature) is one of the better performing variables investigated. This is reassuring, as it is one of the most important management decision-making indicators.
- The population variables $B_{0,}$ $B_{current}$ and $B_{current}/B_0$ were not reliably replicated in this first series of tests. The model generally fits log(R_{init}) well, but not so R_{init}.
- The model estimates mortality, growth and the steepness term in the Beverton Holt equation poorly. Where there is some prior information for estimates of these parameters, this should be used.
- There were differences in the model’s response to inputting the western and eastern data sets.

Weighting observed indices

In this test of alternative weighting, the means of the estimated parameters were found to be dependent on the choice of fitting values. For the data set chosen, there was no simple best fit from the values tested. The best fits to the model parameters appeared to be for N_{lf}=50 or 100 and medium or large CV (0.1 or 0.5) for abundance. Consequently, it is important to test the fitting to other indices, test with alternative data sets and test for $N_{lf} > 100$, prior to applying the model in a specific fishery assessment.

In the best cases, the population parameters fitted reasonably well, but growth and steepness fitted poorly.

Future research directions

Future exploration of the model's capabilities and limitations should encompass the following:

- The fitting process should be further tested for larger values of N_{lf} .
- The basic testing (similar to scenarios 1–5) should be repeated with larger N_{lf} and specified (medium or large) abundance CV.
- The TAC setting and risk assessment is based on 70–80% probabilities from a Bayesian MCMC process, whereas the simulations presented here are from maximum likelihood solutions (which may align with median values from Bayesian posteriors). Despite being extremely computer intensive, further testing should be done for Bayesian analyses. These will typically require 4–6 hours per simulation, compared with about 3 minutes per maximum likelihood solution.
- Prior information, including parameter estimates that are external to the model, should be used wherever possible.

Due to the vast amount of fits generated, the graphed abundance and length frequency fits were not analysed as part of this study.

In the Methods section we mentioned that the steering committee meeting of June 2002 identified a preliminary list of priorities for testing. To re-iterate, these included:

1. Initial basic testing of the model against itself.
2. Effects of mis-specification of priors (such as mortality and recruitment curve steepness).
3. Different dynamics in the cryptic population compared to the non-cryptic.
4. Comparing pooled with un-combined data.
5. Fleet dynamics - by varying catch distribution through time via either explicit time series or setting catch proportional to abundance.
6. Influence of serial depletion & the ability to detect serial depletion
7. Assessment strategy – not running a survey or assessment every year.
8. The loss of habitat, loss of population or loss of fishing grounds.
9. The interaction between serial depletion and fleet dynamics.

The first round for testing points 1 and 2 have been completed, and these now need to be extended in the light of the first set of tests. Points 3 and 4 can be tested using the current test platform. Point 5 needs the model to be regionalised. Some functionality for testing linear loss (point 8) is now in the model and alternate functions can be added.

However, Points 5 and 6 require a substantial undertaking that would have to be supported by additional funding if it was to be completed successfully. It also seems that having developed the modelling product, resources available to each State for implementation are limited and value could be added to the project outcomes through provision of specific funding for this purpose.

Table 2.8 Weight for first growth matrix

	Model 1 No stochastic recruitment	Model 1 Stochastic recruitment	Model 2 No stochastic recruitment	Model 2 Stochastic recruitment	Model 1 Mortality mis-specified
Value from operational model	0.5	0.5	0.9	0.9	0.5
Estimated value (as percentage of operational value)	103 %	88 %	100 %	104 %	109 %

Mean estimated value from each set of 100 simulated data (as percentage difference to operational value)
(simulation 0 has no data error)

Estimation	Simulation data error type						Simulation data error type						Simulation data error type						Simulation data error type						Simulation data error type					
run	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5
1																														
2																														
3																														
4	0	-8	-7	-6	-8	-8	0	-5	-6	-6	-6	-6	1	9	9	9	9	9	6	1	4	8	4	4	41	-5	-5	-5	-5	-5
5	0	-91	-91	-91	-91	-91	-44	-91	-91	-89	-91	-91	1	9	9	9	9	9	6	-4	0	3	-1	-1	53	-90	-82	-74	-90	
6	11	-8	-6	-6	-8	-8	-18	-6	-4	-4	-5	-5	-4	9	9	9	9	9	7	5	6	3	7	7	0	-6	-6	-6	-6	-6
7	11	-91	-91	-91	-91	-91	-20	-91	-91	-91	-91	-91	0	9	9	9	9	9	7	-3	0	-6	-3	2	-23	-91	-90	-91	-91	-91
Data	Nil	All	All	Catch	Abunc	LF	Nil	All	All	Catch	Abunc	LF	Nil	All	All	Catch	Abunc	LF	Nil	All	All	Catch	Abunc	LF	Nil	All	All	Catch	Abunc	LF
error		small	large	large	large	large		small	large	large	large	large		small	large	large	large	large		small	large	large	large	large		small	large	large	large	large

CV of estimated values from each set of 100 simulated data
(as percentage of mean value)

Estimation	Simulation data error type					Simulation data error type					Simulation data error type					Simulation data error type					Simulation data error type				
run	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1																									
2																									
3																									
4	1	1	1	1	1	6	5	8	2	4	1	1	1	1	1	24	17	6	15	18	5	7	5	5	6
5	106	102	102	105	105	56	54	105	56	56	1	1	1	1	1	34	28	23	30	30	150	189	182	152	
6	2	2	2	2	2	3	6	8	5	4	1	1	1	1	1	15	11	17	11	8	3	6	4	1	3
7	117	114	115	116	117	82	82	81	89	81	1	1	1	1	1	33	27	36	32	24	78	112	77	81	78

Data error (cv)	Small	Large
Catch	0.05	0.20
Abundance	0.10	0.20
Length freq	0.20	0.30

Mortality, growth fit	Large
CV on the	0.20
prior	0.20
	0.30

Table 2.9 Mortality

	Model 1 No stochastic recruitment	Model 1 Stochastic recruitment	Model 2 No stochastic recruitment	Model 2 Stochastic recruitment	Model 1 Mortality mis-specified
Value from operational model	0.2	0.2	0.2	0.2	0.1
Estimated value (as percentage of operational value)	100 %	93 %	100 %	98 %	124 %

Mean estimated value from each set of 100 simulated data (as percentage difference to operational value)
(simulation 0 has no data error)

Estimation	Simulation data error type						Simulation data error type						Simulation data error type						Simulation data error type						Simulation data error type					
run	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5
1																														0
2	0	0	0	1	0	0	-10	-1	-1	-1	-1	-1	0	-1	-1	0	-1	-1	-9	-2	-3	-3	-4	-2	36	2	2	3	1	2
3	0	17	17	74	-15	20	-17	4	7	54	-14	3	0	-40	-41	-24	-48	-40	-9	-37	-42	-21	-45	-35	38	73	76	157	45	73
4																														0
5																														
6	0	0	0	1	0	0	-11	-1	-2	-2	-2	-2	-1	-1	-1	-1	-1	-1	0	-4	-3	-3	-5	-3	36	2	2	3	1	2
7	0	7	25	72	-16	9	-10	0	12	57	-16	4	0	-45	-44	-33	-47	-45	0	-42	-45	-37	-46	-38	53	64	88	154	37	70
Data error	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large	Nil	All small	All large	Catch large	Abunc large	LF large

CV of estimated values from each set of 100 simulated data
(as percentage of mean value)

Estimation	Simulation data error type					Simulation data error type					Simulation data error type					Simulation data error type					Simulation data error type				
run	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1																									
2	0	0	1	0	0	5	6	6	5	6	0	1	1	0	0	9	10	11	13	9	4	4	5	4	5
3	46	47	34	43	49	59	55	49	54	58	33	29	41	13	35	49	43	52	51	52	60	57	49	54	57
4																									
5																									
6	0	0	1	0	0	4	7	9	8	7	0	1	1	0	0	12	11	11	12	8	4	5	5	4	5
7	47	43	31	42	46	57	51	49	40	58	21	20	37	13	21	31	19	43	36	45	57	53	50	39	57

Data error (cv)	Small	Large
Catch	0.05	0.20
Abundance	0.10	0.20
Length freq	0.20	0.30

Mortality, growth fit	Large
CV on the prior	0.20
	0.20
	0.30

Table 3.1 Fitting to abundance and length frequency

	$B_{15}/B_{current}$ No stochastic recruitment	B_0 (,000 tonnes) Stochastic recruitment	$B_{current}$ (,000 tonnes) No stochastic recruitment	$B_{current}/B_0$ Stochastic recruitment	-Log(likelihood) Stochastic recruitment
onal	0.93	5.4	3.1	0.57	
Estimated (as percentage of operational	107 %	80 %	58 %	72 %	

Mean estimated value from each set of 100 simulated data (as percentage difference to operational value)
(simulation 0 has no data error)

Estimation	Simulation data error						Simulation data error						Simulation data error						Simulation data error						Simulation data error					
run	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5
1	6	-18	-22	-16	-21	-15	-23	-5	137	65	69	-4	-56	15	457	199	258	14	-43	1	93	50	51	0	-1426	-679	973	-577	997	-670
2	10	-15	-26	-13	-20	-14	-24	-4	94	66	42	-4	-41	16	356	193	200	19	-23	9	90	50	53	11	-1395	-653	969	-392	1180	-736
3	11	-13	-25	-12	-25	-12	-19	2	88	68	47	2	-30	29	345	189	214	29	-14	15	96	48	64	16	-1380	-771	869	-742	909	-718
4	11	-9	-22	-9	-23	-9	-10	12	87	69	53	12	-14	48	326	183	228	51	-5	22	89	49	71	26	-1362	-824	866	-796	1260	-814
5	2	-7	-5	-5	-5	-8	-24	-17	-19	-17	-19	-18	-64	-62	-60	-60	-62	-63	-52	-54	-50	-50	-52	-55	-1430	-1026	-1135	-448	-823	556
6	8	1	-6	1	-4	2	-24	-21	-19	-18	-20	-21	-56	-56	-54	-54	-56	-53	-42	-43	-42	-43	-45	-42	-1365	-683	-399	-896	-947	-470
7	11	4	-5	2	-6	3	-22	-16	-11	-12	-14	-15	-35	-35	-32	-33	-36	-33	-17	-22	-22	-23	-24	-21	-1339	-1317	-1274	-1316	-1130	-1299
8	10	6	1	5	0	6	-10	-6	-1	-2	-4	-5	-13	-13	-10	-11	-14	-13	-4	-7	-8	-8	-10	-7	-1317	-1283	-1225	-1282	-1260	-1248
9	-3	-12	-10	-12	-16	-13	-23	-14	-16	-11	-12	-14	-65	-63	-61	-54	-63	-62	-55	-56	-53	-50	-57	-56	-1408	-1389	-303	-704	-1380	-1270
10	6	1	-5	-4	-2	-1	-22	-20	-17	-15	-20	-19	-53	-52	-50	-47	-54	-51	-40	-40	-39	-39	-42	-39	-1335	-1105	-905	-762	-1088	-1291
11	11	8	2	6	5	6	-19	-16	-10	-12	-15	-15	-30	-30	-29	-30	-31	-30	-14	-16	-20	-19	-18	-17	-1309	-1294	-1275	-1294	-1293	-1276
12	10	9	5	7	7	7	-10	-8	-2	-4	-7	-6	-13	-14	-13	-12	-14	-14	-4	-6	-10	-8	-7	-8	-1286	-1258	-1223	-1258	-1257	-1224
Data	Nil	All	All	Catch	Abund	LF	Nil	All	All	Catch	Abund	LF	Nil	All	All	Catch	Abund	LF	Nil	All	All	Catch	Abund	LF	Nil	All	All	Catch	Abund	LF
error		small	large	large	large	large		small	large	large	large	large		small	large	large	large	large		small	large	large	large	large		small	large	large	large	large

CV of estimated values from each set of 100 simulated data
(as percentage of mean value)

Estimation	Simulation data error					Simulation data error					Simulation data error					Simulation data error					Simulation data error				
run	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	25	37	21	34	34	84	115	93	138	83	163	128	122	171	165	56	54	42	67	57	1430	2300	2827	2397	1034
2	21	34	17	35	21	51	93	77	110	50	110	119	103	157	107	50	52	40	65	47	1198	1839	3484	3002	810
3	18	35	16	33	17	43	83	70	95	44	93	107	97	133	93	42	49	38	59	42	532	1620	635	1742	553
4	16	33	14	38	15	42	72	62	77	42	83	100	86	116	82	36	50	32	56	34	300	1270	420	4080	243
5	21	22	24	22	22	19	13	18	15	18	26	20	18	23	16	22	22	19	24	19	3635	1313	5150	4021	8732
6	13	24	16	22	13	9	10	14	9	17	18	18	15	18	35	14	23	16	21	16	3653	3349	1177	1352	3440
7	11	22	12	23	11	9	11	10	10	9	9	19	11	19	10	15	25	17	24	15	4	19	8	1442	6
8	7	13	7	14	7	7	9	9	8	7	8	17	9	16	9	12	20	14	20	12	5	17	6	14	10
9	21	19	19	26	22	20	15	22	20	21	29	33	55	26	46	19	29	35	23	27	28	5699	3214	35	1102
10	11	20	19	15	15	8	11	17	7	10	33	20	62	13	14	22	23	35	17	18	1214	2882	3146	1048	205
11	6	12	9	11	9	7	11	10	9	8	7	11	8	9	9	11	18	14	14	14	3	5	3	3	5
12	5	9	7	7	7	6	9	9	7	7	6	10	7	8	9	9	15	12	12	12	4	9	4	4	8

Data error (cv)	Small	Large
Catch	0.05	0.20
Abundance	0.10	0.20
Length freq	0.20	0.30

Table 3.2 Fitting to abundance and length frequency

	log(R_{init}) Stochastic recruitment	R_{init} (million) Stochastic recruitment	Steepness (h) No stochastic recruitment	Mortality Stochastic recruitment	Weight on first growth matrix Stochastic recruitment
Value from operational model	15.9	8.0	0.6	0.20	0.5
Estimated value (as percentage of operational value)	98 %	76 %	141 %	97 %	97 %

Mean estimated value from each set of 100 simulated data (as percentage difference to operational value)
(simulation 0 has no data error)

Estimation	Simulation data error type						Simulation data error type						Simulation data error type						Simulation data error type						Simulation data error type					
run	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5
1	-3	1	8	5	5	1	-35	49	440	232	260	49	30	18	35	39	19	18	-9	23	67	38	58	22	-1	-24	-36	-13	-35	-20
2	-2	1	7	5	4	1	-31	50	397	250	251	52	58	25	36	45	27	26	-5	23	64	37	55	23	2	-24	-38	-14	-37	-24
3	-2	2	8	5	5	1	-23	58	437	247	293	59	58	27	38	43	25	30	-3	21	71	35	62	21	1	-24	-42	-14	-47	-25
4	-1	2	7	5	6	2	-14	79	434	248	322	81	58	29	42	44	30	33	-2	22	65	34	58	23	-3	-29	-47	-20	-45	-29
5	-2	-3	-2	-2	-2	-3	-27	-32	-22	-25	-29	-32	20	16	13	15	15	16	-1	-9	-2	-5	-5	-8	-4	-5	-2	-3	-5	-4
6	-2	-2	-1	-2	-2	-2	-25	-25	-19	-22	-24	-25	24	19	16	19	17	21	0	-2	0	-2	-2	-2	-6	-4	-6	-3	-7	-3
7	-2	-1	-1	-1	-1	-1	-23	-16	-10	-12	-14	-15	58	35	27	28	29	35	0	0	1	0	0	0	-3	-3	-4	-2	-4	-3
8	-1	-1	0	0	0	0	-12	-8	-2	-3	-5	-7	58	42	31	35	31	39	-1	-1	0	0	0	-1	-3	-4	-4	-3	-5	-4
9	-2	-3	-2	-2	-3	-3	-26	-32	-26	-28	-34	-32	15	11	12	10	10	11	-2	-10	-6	-10	-13	-11	-7	-6	-5	-4	-7	-5
10	-2	-2	-1	-2	-2	-2	-23	-22	-19	-19	-22	-21	20	20	18	18	20	21	0	0	-1	-3	-1	-1	-4	-4	-4	-2	-4	-3
11	-1	-1	-1	-1	-1	-1	-20	-17	-11	-12	-16	-15	58	47	33	37	42	43	0	0	0	0	0	0	-3	-3	-3	-3	-3	-3
12	-1	-1	0	0	-1	-1	-12	-10	-4	-6	-9	-8	58	51	37	44	47	45	-1	-1	-1	-1	-1	-1	-4	-4	-4	-4	-4	-4
Data	Nil	All	All	Catch	Abunc	LF	Nil	All	All	Catch	Abunc	LF	Nil	All	All	Catch	Abunc	LF	Nil	All	All	Catch	Abunc	LF	Nil	All	All	Catch	Abund	LF
error		small	large	large	large	large		small	large	large	large	large		small	large	large	large	large		small	large	large	large	large		small	large	large	large	large

CV of estimated values from each set of 100 simulated data
(as percentage of mean value)

Estimation	Simulation data error type					Simulation data error type					Simulation data error type					Simulation data error type					Simulation data error type				
run	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	4	7	5	7	4	105	73	98	94	105	31	25	23	31	31	25	37	22	40	26	68	120	56	121	70
2	4	7	6	7	4	87	77	95	99	87	29	25	19	29	29	25	36	23	39	25	66	125	56	121	64
3	4	6	5	7	4	85	73	93	92	87	29	24	21	30	28	26	34	23	38	26	62	129	53	138	62
4	4	7	5	7	4	90	75	91	91	90	29	21	21	27	27	23	35	22	38	23	59	127	55	132	57
5	1	1	1	1	1	18	15	18	15	18	7	13	12	9	7	18	11	15	14	18	10	8	7	3	6
6	1	1	1	1	1	9	10	12	12	9	10	21	16	16	12	8	3	8	8	8	11	3	11	10	12
7	1	1	1	1	1	9	10	10	10	9	23	29	26	28	24	0	2	1	1	0	1	2	1	2	1
8	0	1	1	0	0	7	9	9	8	7	21	29	25	29	23	0	1	1	1	1	1	2	1	2	1
9	1	2	2	2	2	18	23	23	26	22	9	13	14	8	10	18	15	19	22	19	9	7	10	9	17
10	0	1	1	1	0	8	11	32	8	7	14	21	20	15	18	0	6	11	3	4	2	3	11	2	4
11	0	1	1	1	1	7	11	10	8	8	17	26	23	21	20	0	0	0	0	0	0	1	0	0	1
12	0	1	1	0	0	6	9	9	6	7	15	25	21	18	20	0	1	0	0	1	1	1	1	1	1

Data error (cv)	Small	Large
Catch	0.05	0.20
Abundance	0.10	0.20
Length freq	0.20	0.30

Part 2: AbModeller

Model Specifications

AbModeller

AbModeller is a length based abalone assessment model, developed in ADModelBuilder (Otter Research Ltd, 2000), to undertake a stock assessment and risk analysis. The model considers the population divided into small length-classes. The number of abalone in each length-class is followed over time.

Population dynamics model

This model allows for movement into and out of cryptic habitat, differences in natural mortality among size-classes and between animals in cryptic and non-cryptic habitat, and different selectivity patterns for each sector. Allowance is made for selectivity to vary between years. Recruitment to the model is assumed to be restricted to the cryptic habitat.

A "season" is a portion of a year. The number of seasons and duration of each season is to be specified by the States. A season is identified by sequential number within year number. It has a duration expressed as a proportion of a year (Δt) with $\sum \Delta t = 1$. During a season, natural mortality is assumed to occur throughout the season ($e^{-M\Delta t}$) while growth occurs as a transition at the end of the season.

A sector refers to a fishing sector, such as commercial, recreational and unaccounted or illegal. There can be several commercial sectors covering different fishing practices or selectivity series.

The number of abalone in length-class *l* at time *t*+1 is determined by applying growth, natural mortality, fishing mortality (catch) and recruitment to the number of abalone in each length-class less than or equal to length-class *l* (*l*' ≤ *l*) by the equations

$$N^{c}_{t,l,m} = \alpha_{t,l,m\text{-}1}\Big(\sum_{l'\le l} N^{c}_{t,l',m\text{-}1}\, e^{-M^{c}_{l'}}\Big[1-\sum_{i} F^{i}_{t,m\text{-}1}\, S^{i}_{t,l',m\text{-}1}\Big] X^{c}_{l,l',m\text{-}1} + \sum_{l'\le l} N^{\tilde{c}}_{t,l',m\text{-}1}\, e^{-M^{\tilde{c}}_{l'}}\, X^{\tilde{c}}_{l,l',m\text{-}1}\Big) \qquad (1a)$$

$$N^{\tilde{c}}_{t,l,m} = (1-\alpha_{t,l,m\text{-}1})\Big(\sum_{l'\le l} N^{c}_{t,l',m\text{-}1}\, e^{-M^{c}_{l'}}\Big[1-\sum_{i} F^{i}_{t,m\text{-}1}\, S^{i}_{t,l',m\text{-}1}\Big] X^{c}_{l,l',m\text{-}1} + \sum_{l'\le l} N^{\tilde{c}}_{t,l',m\text{-}1}\, e^{-M^{\tilde{c}}_{l'}}\, X^{\tilde{c}}_{l,l',m\text{-}1} + R_{t,l,m\text{-}1}\Big) \qquad (1b)$$

when we have proportions at length having moved into the non-cryptic habitat

where $N^{c}_{t,l,m}$ is the number of abalone in size-class *l* in non-cryptic habitat at the start of season *m* of year *t*,

$N^{\tilde{c}}_{t,l,m}$ is the number of abalone in size-class *l* in cryptic habitat at the start of season *m* of year *t*,

$\alpha_{t,l,m\text{-}1}$ is the probability that an animal in size-class *l* is in the non-cryptic habitat at the end of season *m-1*of year *t*,

$M_l^{c/\tilde{c}}$ is the instantaneous rate of natural mortality on animals in size-class *l* in the non-cryptic / cryptic habitat,

$X_{l,l',m\text{-}1}^{c/\tilde{c}}$ is the probability that an animal in the non-cryptic / cryptic habitat and size-class *l′* grows into size-class *l* at the end of season *m-1* (the size-transition matrix),

$F_{t,m\text{-}1}^{i}$ is the fully-selected exploitation rate by sector *i* (commercial, recreational, and illegal) during season *m-1* of year *t*,

$S_{t,l,m\text{-}1}^{i}$ is the selectivity of sector *i* on animals in size-class *l* during season *m-1* of year *t*, and

$R_{t,l,m\text{-}1}$ is the recruitment to size-class *l* at the end of season *m′* of year *t*.

Catches

Catches (in weight) are assumed to be available by sector, season and year. The fully-selected exploitation rate for each sector, $F_{t,m}^{i}$, is defined as the ratio of the catch to the exploitable biomass:

$$F_{t,m}^{i} = \frac{C_{t,m}^{i}}{B_{t,m}^{\mathrm{e},i}} = \frac{C_{t,m}^{i}}{\sum_{l} w_l \, S_{t,l,m}^{i} \, N_{t,l,m}^{c} \, e^{-M_l^c \Delta t}} \qquad (3)$$

where $C_{t,m}^{i}$ is the catch by sector *i* at the end of season *m* of year *t*,

$B_{t,m}^{\mathrm{e},i}$ is the exploitable biomass corresponding to sector *i* during season *m* of year *t*,

w_l is the mean weight of an animal in size-class *l*.

Growth

Three options are considered for the size-transition matrices, $\mathbf{X}^c$ and $\mathbf{X}^{\tilde{c}}$:

1. User-specified.
2. Weighted average of two user-specified size-transition matrices.

$$X^c = g_w X^1 + (1 - g_w) X^2$$

3. Estimated as part of the model fitting process.

$$X_{l,l',m}^{c} = \int_{\bar{l}-\Delta L/2}^{\bar{l}+\Delta L/2} I(\bar{l}', x)dx \qquad (4)$$

where $I(l',l\ )$ is the Schnute probability density function for an animal growing from size l' to size l,

$\bar{l}$ is the mid-point of size-class *l*, and

ΔL is the width of a size-class .

Weight as a function of size is given by:

$$w_l = a\,\overline{l}^{\,b} \tag{5}$$

where a , b are the parameters of the weight-size relationship.

Currently, the size-transition matrices, $\mathbf{X}^c$ and $\mathbf{X}^{\tilde{c}}$ are assumed to be the same.

Fishing selectivity

Fishing selectivity as a function of size-class is given by either the proportion selected

$$S^i_{t,l,m} = \begin{cases} 0, l < l_{\min,i} \\ p^i_{t,m} + (1 - p^i_{t,m})(l - l_{\min}) / (l_{\max} - l_{\min}), l_{\min,i} > l > l_{\max,i} \\ 1, l \geq l_{\max,i} \end{cases} \tag{6}$$

or as an ogive:

$$S^i_{t,l,m} = (1 + \exp(-\ell n19(\overline{l} - l^i_{50,t}) / \phi^i_t))^{-1} \tag{7}$$

where $\phi^i_{t,m}$ is the width of the selectivity ogive for sector *i* during season *m* and year *t*:

$l^i_{50,t}$ is the length of 50% selectivity for sector *i* during season *m* and year *t,*

and $l_{min,i}$ is the minimum length fished for sector i during season *m* and year *t,*

$l_{max,i}$ is the first length fully fished for sector i during season *m* and year *t,*

$p^i_{t,m}$ is the proportion of animals selected for sector *i* at $l_{min,t}$ during season *m* and year *t*. When $p^i_{t,m}$ =1, we are left with the step function.

The values for the parameters of the selectivity function are provided by the user. In the next implementation, they will be able to be estimated from the data.

Transition from cryptic to non-cryptic habitat

Emergence in equations 1, as a function of size-class, is given by:

$$\alpha_{t,l,m} = (1+\exp(-\ell n19(\overline{l} - l_{50,t,m})/\phi_{t,m}))^{-1} \tag{8}$$

where $\alpha_{t,l,m}$ is the probability of an animal in size-class *l* being in non-cryptic habitat at the end of season *m* of year *t,*

$\phi_{t,m}$ is the width of the emergence ogive during season *m* of year *t,*

$l_{50,t,m}$ is the size-at-50%- emergence during season *m* of year *t,*

Density dependent emergence is implemented as a change in $l_{50,t,m}$:

$$l_{50,t,m} = l^1_{50,t,m} + (l^2_{50,t,m} - l^1_{50,t,m})\frac{B^c_{t,m}}{B^c_{-\infty,m}} \tag{9}$$

where $l_{50,t,m}$ is the size-at-50%-emergence during season *m* of year *t,*

$l^2_{50,t,m}$ is the size-at-50%-emergence in a virgin state,

$l^1_{50,t,m}$ is the size-at-50%- emergence in the limit of zero population size, and

$B^c_{t,m}$ is the biomass:

$$B^c_{t,m} = \sum_l w_l \, N_{t,l,m} \tag{10}$$

The emergence ogive is used to redistribute the animals at each length at the end of each season and year.

Maturity

Maturity is specified as either a maturity ogive:

$$f_l = (1+\exp(-\ell n19(\overline{l} - l^{mat}_{50})/\phi^{mat}))^{-1} \tag{11}$$

where ϕ^{mat} is the width of the maturity ogive ,

and l^{mat}_{50} is the length at 50% maturity,

or as a pre-specified proportion mature for each length-class.

Recruitment processes

The recruitment by size-class, year and season is given by:

$$R_{t,l,m} = \omega_{l,m} \sum_{t' \le t} \theta_{(t-t')} \frac{(B^S_{t'} / B^S_{-\infty}) e^{\varepsilon_{t'}}}{\tilde{\alpha} + \tilde{\beta}(B^S_{t'} / B^S_{-\infty})} \qquad (12)$$

where

$\omega_{l,m}$ is the fraction of the recruitment that occurs to size-class l during season m (pre-specified by the user),

$\theta_{(t-t')}$ is the weighting from spawning in year *t′* to recruitment in year *t* (user specified); $\sum_{t' \le t} \theta_{(t-t')} = 1$,

$\tilde{\alpha}$, $\tilde{\beta}$ are the parameters of the stock-recruitment relationship (defined in terms of the "steepness" of the relationship, h, and the virgin recruitment, $R_{-\infty}$),

$$\tilde{\alpha} = \frac{(1-h)}{4hR_{-\infty}} \quad and \quad \tilde{\beta} = \frac{(5h-1)}{4hR_{-\infty}}$$

h is the steepness of the stock-recruitment relation so that

$$hR = \frac{0.2}{\alpha + 0.2\beta} \text{ and } R = \frac{1}{\alpha + \beta}$$

B^S_t is the mature biomass corresponding to the recruitment during year *t*:

$$B^S_t = \sum_l w_l\, f_l\, (N^c_{t',l,m(S)} + N^{\tilde{c}}_{t',l,m(S)}) \qquad (13)$$

f_l is the fraction of animals in size-class *l* that are mature (ogive or user-specified),

w_l is the weight at length *l*,

$m(S)$ denotes the season of spawning (user-specified), and

ε_t is the 'recruitment residual' for year *t*.

Initial conditions

The size-structure at the start of the first year of the projection period (which may not be the first year for which catches are available) is assumed to be that corresponding to deterministic equilibrium in the absence of fishing mortality.

Likelihood functions

Abundance (Independent Survey):

$$-\ln L = \sum_i \sum_t [\ln \sigma_i + \frac{1}{2\sigma_i^2}(\ln A_{i,t} - \ln(\hat{q}\sum_{l \in l_i} S_l^r N_{l,t}))^2]$$

where $N_{l,t}$ is the estimated number ,

$A_{i,t}$ is the observed abundance of size interval *i,*

S_l^r is the selectivity of the fishery-independent survey on animals in size-class l

σ_i is either a given standard deviation or the standard error from the data and

$\hat{q} = \exp[\frac{1}{n}\sum_y (\ln A_{i,t} - \ln(\sum_{l \in l'} S_l^r N_{l,t}))]$.

Abundance (CPUE):

$$-\ln L = \sum_t [\ln \sigma_i + \frac{1}{2\sigma_i^2}(\ln A_t - \ln(\hat{q}\sum_l (S_l^i N_{l,t}))^\gamma)^2]$$

where $N_{l,t}$ is the estimated number ,

A_t is the observed CPUE,

S_l^i is the selectivity of the fishing on animals in the commercial sector i

σ_i is the either a given standard deviation or the standard error from the data

and $\hat{q} = \exp[\frac{1}{n}\sum_y (\ln A_t - \ln(\sum_l (S_l^r N_{l,t}))^\gamma)]$.

Length frequency:

We use the robust likelihood function of Fournier *et al* (1998):

$$-\log L = -0.5\sum_t \sum_l \log[2\pi(\xi_{l,t} + 0.1/I)] - \sum_t I \log(\tau_t) + \sum_t \sum_l \log[\exp\{-\frac{(\rho_{l,t}^{obs} - \rho_{l,t}^{pred})^2}{2(\xi_{l,t} + 0.1/I)\tau_t^2}\} + 0.01]$$

where $\rho_{l,t}^{obs}$ is the observed proportion of animals in length class *l* and year *t,*

$\rho_{l,t}^{pred}$ is the predicted proportion of animals in length class *l* and year *t,*

$\xi_{l,t} = (1 - \rho_{l,t}^{obs})\rho_{l,t}^{obs}$,

$\tau_t^2 = 1/\min(S_t, N_{eff})$,

S_t is the sample size for the length frequency data for year *t,*

N_{eff} is the effective sample size , and

I is the number of length classes in the sample.

A multinomial likelihood is also available.

Recruitment residuals (lognormal prior):

$$-\ln L = -\sum_t (\sigma_t^r)^2 / \sigma_{\mathrm{Rec}}^2 .$$

where σ_{Rec} is specified.

Commercial data (average weight):

Fitting to average weight, $W_{t,m}^i$: $C_{t,m}^i / \sum_l F_{t,m}^i \, S_{t,l,m}^i \, N_{t,l,m}^c$ (log-normal) will be added in the next release.

Priors

Priors and bounds can be specified on a range of parameters in the model. Priors can uniform, normal or log-normal. These parameters are:

- Stochastic recruitment (normal only)
- Mortality
- Growth function parameters
- Steepness
- log(Recruitment in a virgin population)
- Emergence lengths (uniform only)

Penalties

Penalties can also be placed on the fishing exploitation rate to keep it below 1 (fully fished) for past years, or >>0 for the last year fished. Steepness and the weighting of the growth matrices (g_w) can have penalties applied near minimum or maximum values to help ensure a positive definite hessian matrix at maximum likelihood while testing.

References

Bardos, D.C. 2004. Probabilistic Gompertz model of irreversible growth. *Bulletin of Mathematical Biology* (accepted)

Breen, P.A., S.W. Kim & N.L. Andrew. 2003. A length-based Bayesian model for the New Zealand abalone *Haliotis iris*. *Mar. & Freshwater Res.*, 54: 619–634.

Fournier, D.A., Hampton, J., and Sibert, J.R. 1998. MUL:TIFAN-CL: a length-based age-structured model for fisheries stock assessment, with applications to South Pacific albacore, *Thunnus alalunga. Canadian Journal of Fisheries and Aquatic Sciences*, 55: 2105–16.

Otter Research Ltd. 2000. An introduction to AD Model Builder Version 5.0.1 for use in nonlinear modeling and statistics. (Otter Research Ltd: Nanaimo, British Columbia, Canada.)

Punt, A.E. & R.B. Kennedy. 1997. Population modelling of Tasmanian rock lobster, *Jasus edwardsii*, resources. *Mar. & Freshwater Res.*, 48: 967–980.

Schnute, J.T. 1981. A Versatile Growth Model with Statistically stable parameters. *Can. J. Fish. & Aquatic Sc.*, 38: 1128–1140.

Troynikov, V.S. 1998. Probability density functions useful for parametrization of heterogeneity in growth and allometry data. *Bulletin of Mathematical Biology*, 60: 1099–1121.

Worthington, D.G. 1997. Demography and dynamics of the population of blacklip abalone, Haliotis rubra, with implications for management of the fishery in NSW, Australia. Ph.D. thesis, Macquarie University, Sydney, Australia

Part 3: AbModeller User Manual Version 1.2.1

AbModeller

Abalone Modelling Software

User Manual Version 1.2.1

Not for citation without permission

No. 48

February 2005

Victoria
The Place To Be

AbModeller
Abalone Modelling Software
User Manual Version 1.2.1

Bruce Taylor [1] and Michael Cleland [2]

1. PIRVic, Marine & Freshwater Systems, PO Box 114, Queenscliff, VIC 3225, Australia (corresponding authors).
2. 19 Hood Cres, Arrowtown, New Zealand.

Not for citation without permission

February 2005

Department of Primary Industries

Published: Primary Industries Research Victoria,
Marine and Freshwater Systems
Department of Primary Industries, Queenscliff

General disclaimer
This publication may be of assistance to you but the State of Victoria and its employees do not guarantee that the publication is without flaw of any kind or is wholly appropriate for your particular purposes and therefore disclaims all liability for any error, loss or other consequence which may arise from you relying on any information in this publication.

Authorised by the Victorian Government, Weeroona Parade, Queenscliff 3225

Printed by PIRVic Queenscliff, Victoria.

Preferred way to cite:
Taylor, B. and Cleland, M. (2005) AbModeller Abalone Modelling Software User Manual Version 1.2.1. Marine and Freshwater Systems Internal Report No. 48. Primary Industries Research Victoria, Queenscliff.

Table of Contents

Getting Started

What is AbModeller?

AbModeller is a graphical, user-friendly, size-structured model for abalone stock assessment. It combines a familiar Windows GUI with a powerful AD Model Builder® application.

The model and software was developed through a national collaborative project led by Primary Industry Research Victoria (PIRVic, formerly the Marine and Freshwater Resources Institute, MAFRI), in Victoria, Australia. Research institutes in all abalone-producing Australian states participated in this project funded by the national Fisheries Research and Development Corporation (FRDC).

AbModeller consists of three parts:

- A stock assessment and risk analysis program (ModelEngine.exe) developed using AD Model Builder® (Otter Research Ltd., 2000) that combines the model parameter estimation and projections for risk analysis and policy evaluation.
- A user-friendly GUI (AbModeller.exe) for entering model options, viewing the model outputs and controlling the modelling process.
- MS Access® databases for model inputs, output, analysis and reporting.

Features of this package are:

- Easy to use.
- Automated time saving process for data entry.
- Availability of outputs for analysis.

It is requested that users acknowledge any application of this software in their work with the following citation:

Taylor, B. L. and Cleland, M. (2005) AbModeller Abalone Modelling Software, Version 1.2.1 - User Manual. Marine and Freshwater Systems Internal Report No. 48. PIRVic, Queenscliff, Victoria.

Installation

You cannot run AbModeller from the CD; you must install the components onto your hard drive and run the software from that drive. AbModeller comes with its own setup program for both itself and any necessary Windows database libraries.

To install AbModeller

- Restart Windows, and do not start any other applications.
- Insert the AbModeller CD into your CD drive. The AbModeller install program should run. If it doesn't, run setup.exe by double clicking on the icon in Windows Explorer.
- In the AbModeller Setup dialog box, click `NEXT` whenever you are ready to proceed to the next panel. Follow the on-screen instructions to move past the introduction and to accept the license agreement (if present).
- To change the destination folder for the AbModeller files, click `BROWSE` , and use the browser to locate a folder or name a new one (eg C:\MyAbModeller).
- Click `NEXT` to accept the default program group and shortcuts.
- Click `INSTALL` to begin the installation. When the process is complete, a message tells you that AbModeller is installed.
- The Database libraries comprising DAO, ADO and Jet are installed automatically. Follow the instructions by clicking NEXT where appropriate.
- Restart the computer.
- Start AbModeller from the installed desktop icon or from the Windows `Start-Programs`.

System Requirements

Hardware Requirements

AbModeller is a memory intensive processor-bound application designed to run on the latest powerful processors. A 2.4 GHz Pentium 4 or above is required. It can take longer on slower machines

The more RAM, the better and a minimum of 256 megabytes is recommended. More memory will improve the performance of the program.

Large databases can be produced from model runs.

The forms in the GUI have been designed for at least 1024 x 768 dpi resolution.

Software Requirements

AbModeller has been designed for the Windows 2000® Operating System. Windows XP® may also be used. Earlier Windows versions have not been thoroughly tested and may not be supported.

Microsoft Access® is recommended for additional charting and is required to run the analysis and reporting procedures. Access 2000 or above is recommended, and is the default. Access 97 has been trialled. To use Access 97, use the sample Access 97 databases, or convert the Access 2000 databases to Access 97. Access 2000 was used for software development.

Support and Software Enhancement

Support

The model, code and reporting databases are supplied only in the current format. The PIRVic Abalone Assessment Project will provide support for this version of AbModeller to each State agency responsible for abalone assessment programs. PIRVic will coordinate and disseminate future enhancements to this model, including future Victorian developments, to promote on-going dialogue and collaboration about abalone modelling nationally among state research institutes.

Contact details for the PIRVic Abalone Assessment Project are:

PIRVic Abalone Assessment
PO Box 114,
Queenscliff, Vic, 3225
Phone: 03 5258 0111
Fax: 03 5258 0270
E-mail: Bruce.Tayor@dpi.vic.gov.au

The GUI programming and access functionality were designed and coded by database consultant, Michael Cleland. His contact details are:

Michael Cleland
Programmer and Database Designer
10 McNaught Street
Pt Lonsdale, Vic, 3225
Phone: 03 5258 1204 or 0418 477 718
E-mail: mcleland@bigpond.net.au

AD ModelBuilder® and ADMB® documentation are available from their website at http:\\www.otter-rsch.com or from:

Otter Research Ltd.
P.O. Box 2040
Sidney, B.C., V8L 3S3, Canada
Phone : 250-655-3364   Fax: 250-655-3364
E-mail: otter@otter-rsch.com

Model Description

The model is described in the accompanying 'AbModeller — an abalone stock assessment and risk assessment model. Model description'

Modelling Process

Overview

AbModeller is designed as a 3 stage process

- Initial estimation – A maximum likelihood estimation of the parameters of the model
- MCMC – Deriving Bayesian posteriors of the derived parameters
- Evaluation – projecting the model under a range of possible harvest strategies using the uncertainty from the MCMC process. The results from this are then available for analysis in the MS-Access database and consideration in the management process.

A set by step "Quick Guide" to AbModeller is found on page 22.

Fisheries modelling

A fuller discussion of fisheries models and the Bayesian modelling process can be found in the literature. Some examples are the ADMB Manual (Otter Research, 2000), the Users Manual for Coleraine (Hilborn et al, 2001) and BAYES-SA - Bayesian Stock Assessment Methods in Fisheries (Punt and Hilborn, 2001).

References and Resources

Bardos, D.C. (2004). Probabilistic Gompertz model of irreversible growth. *Bulletin of Mathematical Biology* (accepted)

Gelman, A, Carlin, J.B., Stern H.S. and Rubin, D.B., (1995). Bayesian Data Analysis. (Chapman and Hall, New York)

Hilborn, R., Maunder, M., Parma, A., Ernst, B., Payne, J. and Starr, P. (2001). Coleraine: A generalized Age-Structured Stock Assessment Model: User's Manual Version 2.0. Fisheries Research Institute, University of Washington, SAFS-UW-0116 (see http://www.fish.washington.edu/Publications/pdfs/0116.pdf)

Punt, A and Hilborn, R. (2001) BAYES-SA - Bayesian Stock Assessment Methods in Fisheries - User's Manual, 2001, FAO Computerized Information Series (Fisheries) - No.12 (available from http://www.fao.org/ or see http://www.fao.org/DOCREP /005/Y1958E/y1958e00.htm#Contents)

Otter Research Ltd. (1994). AUTODIF - A C ++ Array Language Extension with Automatic Differentiation for Use in Nonlinear Modelling and Statistics. (Otter Research Ltd, British Columbia, Canada)

Otter Research Ltd. (2000). An introduction to AD Model Builder Version 5.0.1 for use in nonlinear modelling and statistics. (Otter Research Ltd, British Columbia, Canada).

Initialising Input data

Overview

The AbModeller process simplifies the control of the input data. Input data sets are stored in an MS Access® database and the appropriate data are selected for the model. Data can be simply pasted in from MS Excel®, or drawn from another database. These include Catch, Abundance, Length Frequency and Growth. The model input database tables must conform to the field structure as in the sample provided. Extra fields may exist, but no field may be missing from each of the tables.

Although only one model input database is active at any time, the user may select an alternative. To select a new modelling database, use the Options menu.

Alternatively the scenario field present in all tables, allows the modeller the flexibility to maintain separate instances of like data. For example, two scenarios may be with and without standardisation of the input data. Alternatively it may be two catch scenarios, dry and wet weights.

The tables of data and their contents follow.

Modelling database

Version

Contains the version number of the modelling database.

Region

This is a list of the geographic modelling regions, and associated data sets, available for modelling.

Catch

Sector based Catch history for the regions. The fields in the table are Region, Sector, Scenario, Year, Season and Catch. Fishing sectors include commercial, recreational, unaccounted (poached, undeclared, other). Up to 5 sectors can be used in the model, each with their own selectivity.

Abundance

Regional and seasonally based Independent Abundance data grouped into class size categories. The Class field describes the minimum size of each of the abundance length classes (or of the abundance length class). The fields in the table are Region, Class, Scenario, Year, Season and Abundance.

CPUE

Regional and seasonally based CPUE data. The fields in the table are Region, Year, Season, Scenario and CPUE.

Length Frequency (Commercial and Independent)

Regional and seasonally based Length Frequency data. The fields in both the Commercial Length Frequency and the Independent Length Frequency tables are Region, Season, Scenario, Year, Length and Frequency.

Growth Description

Describes the growth matrices. The fields in the table are KeyNumber and Description. These appear as choices in the selections of options for a model run. The growth matrices are not needed if Schnute growth is used.

Growth

Growth matrices in columnar form. The fields in the table are Number, Row, Column, Value and Season. Column is the start length, Row is the final length and Value is the probability of growing from the start to the final length. For annual growth, Season is 1.

Software to generate growth matrices from tagging recapture data is included with the installation CD.

Run Info (defaults for defined runs)

The set of options and values that describe a run of the model are referred to as the Run Info. This table provides the set of defaults for the Run Info items created in the GUI. These can be reset from within the GUI. Initially, they can be copied from the sample data set.

Model input file

Data Input and control of the estimation and MCMC processes comes primarily from the input data file (modelengine.dat) file. Further control parameters are sent via the command line interface.

The modelengine.dat file is generated from the values entered in the Run Info Editor (subsequently stored in the results database) as well as from model input database queries built form user defined Run Info items.

For example, parameter estimates are direct inputs whereas Catch history is generated from a query derived from the user's entry of region, start and end year. These same values are used by other queries to produce the data for Length Frequency and Abundance observations.

Starting AbModeller

Introduction

The data selection and options for modelling (the Run Info) are defined from a series of pages in the AbModeller GUI.

Starting a model

Assessment Database Menu - Selecting modelling and Results databases

Change Modelling (input) database

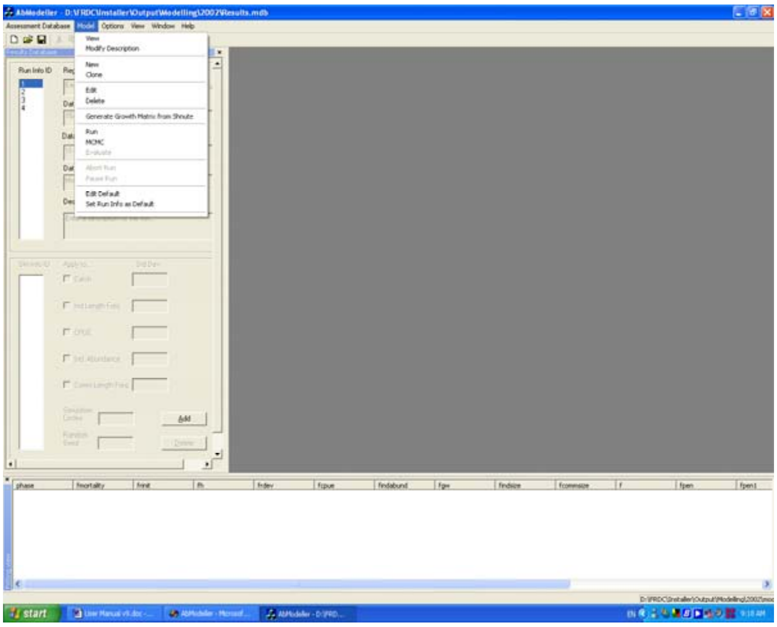
From Assessment in the menu, select Change Modelling Database to define or browse for an input database. Provided for use before a results database has been opened.

Results (output) database

From Assessment in the menu, select New to create a new output database or Open to open an existing results database. If no modelling database has been set, one will be asked for at this stage.

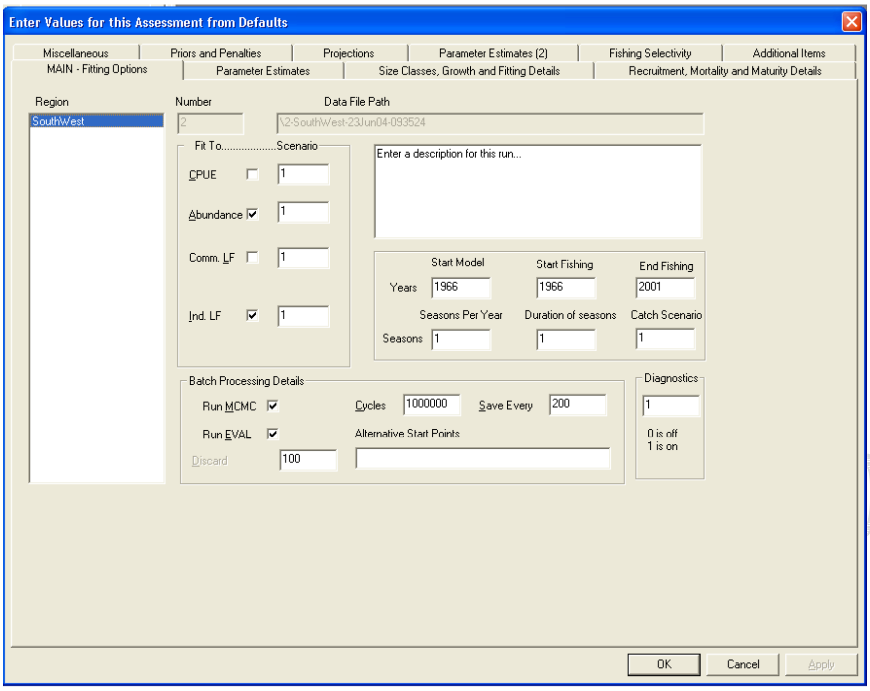
Model Menu – Selecting or defining a model

From Model in the menu, select New, Edit or Clone an existing model definition.



Defining a model - Parameters and options

In the GUI, the options for the model and values of the set parameters, the initial values and the bounds of the estimated parameters and pointers to data in the input database are set. These are found on ten pages that define the model run. These pages start when either **New** or **Clone** is selected from the **Model Menu**, or if you choose to **Edit** an existing run.



Region: SouthWest

Number: 2

Data File Path: \2-SouthWest-23Jun04-093524

Fit To.................Scenario

CPUE □ 1

Abundance ☑ 1

Comm. LF □ 1

Ind. LF ☑ 1

Enter a description for this run...

	Start Model	Start Fishing	End Fishing
Years	1966	1966	2001
	Seasons Per Year	Duration of seasons	Catch Scenario
Seasons	1	1	1

Batch Processing Details

Run MCMC ☑ Cycles 1000000 Save Every 200

Run EVAL ☑ Alternative Start Points

Discard 100

Diagnostics 1

0 is off
1 is on

MAIN – Fitting Options

Region

Select a modelling region from the list from the Modelling database.

Number

Run Info number (assigned by program), a unique number for a set of options, values and parameters defining one run of the model.

Data File Path

Path of the ADMB input data file. This is a subdirectory of the directory that contains the Results database and the subdirectory name is assigned by AbModeller. Note that the actual data file name is, by default, ModelEngine.dat, in line with the name of the ADMB application.

Fit to

The observed data used for fitting the model (select from Commercial CPUE, independent survey abundance, Commercial Length Frequency and independent survey Length Frequency).

Scenario

Choose an alternative history for each of the fitted items as well as Catch.

Diagnostic Level

Diagnostic >= 1 prints out progressive outputs to the GUI. Generally used for initial model (maximum likelihood) testing. The model is cyclic and the diagnostic level controls how many cycles it takes before a diagnostic output is generated.

Diagnostic = 0 turns off the running diagnostic output. Generally switched off for batch and MCMC runs.

Years

Start model year = start burn-in period to stabilise the length structure of the virgin population.

Start fishing year = start of catch data = year after last year of virgin population.

End fishing year = end of catch data = last year before projections.

Seasons

Seasons per year = number of seasons in a year.

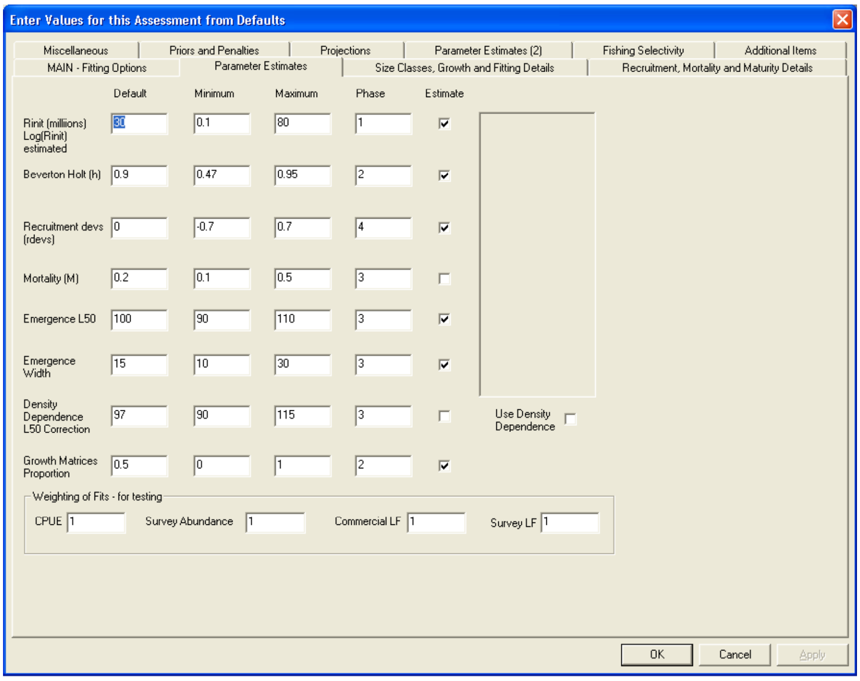
Duration as fraction = list of the lengths of each season. This can be in months, weeks or fractions of a year. The program converts them to fractions of a year. The default is equal seasons.

Batch processing details

Assigns the default number of runs for the MCMC and evaluation (eval) phases primarily for batching of runs. The number of cycles to discard (*the burn in*) is also stored in the Run Info table, used not by AbModeller but in the Access reporting feature during the analysis process.

These can also be redefined when MCMC or evaluate are selected from the "Model" menu. To define alternative MCMC start points, enter a space separated list of modifier values to augment the parameter estimates from the ML fit solution. For each item in the list the MCMC routine will be performed.

Estimates.



	Default	Minimum	Maximum	Phase	Estimate
Rinit (millions) Log(Rinit) estimated	30	0.1	80	1	☑
Beverton Holt (h)	0.9	0.47	0.95	2	☑
Recruitment devs (rdevs)	0	-0.7	0.7	4	☑
Mortality (M)	0.2	0.1	0.5	3	□
Emergence L50	100	90	110	3	☑
Emergence Width	15	10	30	3	☑
Density Dependence L50 Correction	97	90	115	3	□
Growth Matrices Proportion	0.5	0	1	2	☑

Use Density Dependence □

Weighting of Fits - for testing

CPUE 1 Survey Abundance 1 Commercial LF 1 Survey LF 1

Columns - overview

Default = Initial value if the parameter is estimated, or set value if it is not estimated.

Minimum = minimum value if the parameter is estimated (otherwise, ignored).

Maximum = maximum value if the parameter is estimated.

Phase = the phase in which the parameter is first estimated.

Estimated = tick if the parameter is to be estimated, the alternative is the parameter is fixed at the default value.

Log (Rinit)

The number of millions of abalone recruited in a virgin stock. Used in the Beverton Holt recruitment equation (R_{init}). Log(R_{init}) is the estimated parameter. If no other prior is set, the prior is uniform on log(R_{init}).

Beverton Holt (h) - Steepness

The steepness coefficient, h, in the Beverton Holt recruitment equation.

Recruitment devs

The log normal stochastic component of recruitment for the years fished. This is a vector. If different values are required for each year, these are entered on the Recruitment and Maturity page.

Mortality

Mortality of emerged population. Population in cryptic habitat is the same unless set differently in the Miscellaneous page. See also Length Dependent Mortality.

Emergence (L_{50})

Length of 50% emerged. Used in emergence ogive. Applied at beginning of each year

Emergence Width

L_{95} – L_{50}. Difference in length between 95% and 50% emerged. Used in emergence ogive. Applied at beginning of each year.

Density Dependence L_{50} correction

Only used if "Use Density Dependence" checked. Gives value of L_{50} in a virgin population, and Emergence (L_{50}) becomes L_{50} for depleted population.

Growth Matrix Proportion

When using growth matrices, two matrices are used as upper and lower bounds for growth. This parameter (G_w) is the proportion of the first matrix added to (1-G_w) of the second matrix. If a single matrix is used, the same matrix can be entered twice and G_w fixed or the 'first matrix' only can be selected.

Weighting of Fit

Weights the log likelihood values from each of the fitted abundances and length frequencies. These can be used for coarse rapid changes while getting the model running but should be reset to 1.

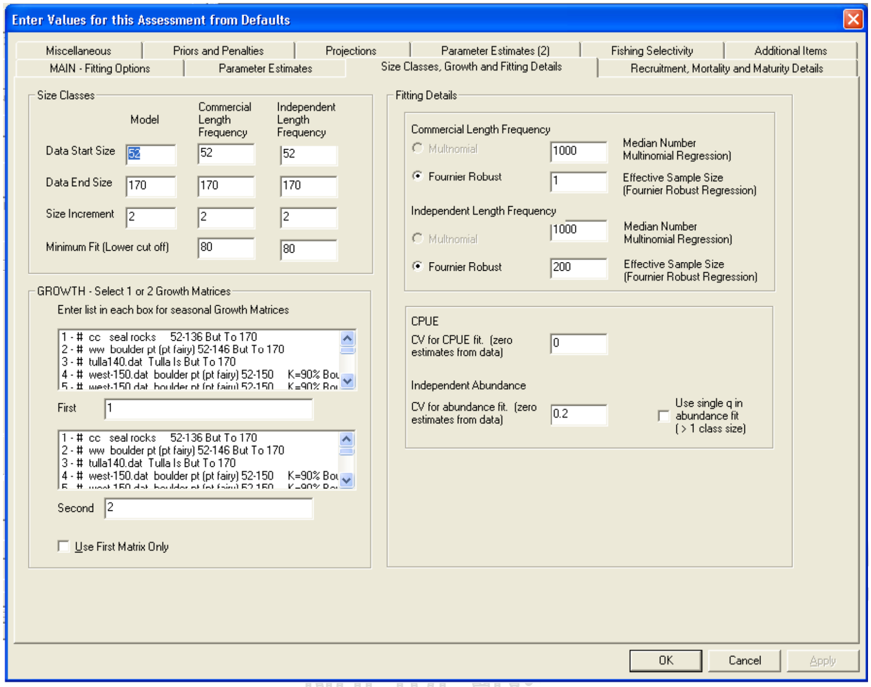
Use Density Dependence

See 4.2.8 above.

Use Growth Matrices

Use weighted sum of two (2) supplied growth matrices. The alternative is using Schnute growth (set in Schnute growth page). See 4.2.9 above.

Growth, Size Class and Fitting.



Size Classes

	Model	Commercial Length Frequency	Independent Length Frequency
Data Start Size	52	52	52
Data End Size	170	170	170
Size Increment	2	2	2
Minimum Fit (Lower cut off)		80	80

GROWTH - Select 1 or 2 Growth Matrices

Enter list in each box for seasonal Growth Matrices

```
1 - #  cc   seal rocks     52-136 But To 170
2 - #  ww  boulder pt (pt fairy) 52-146 But To 170
3 - #  tulla140.dat  Tulla Is But To 170
4 - #  west-150.dat  boulder pt (pt fairy) 52-150     K=90% Bou
5 - #  west-150.dat  boulder pt (pt fairy) 52-150     K=90% Bou
```

First 1

```
1 - #  cc   seal rocks     52-136 But To 170
2 - #  ww  boulder pt (pt fairy) 52-146 But To 170
3 - #  tulla140.dat  Tulla Is But To 170
4 - #  west-150.dat  boulder pt (pt fairy) 52-150     K=90% Bou
5 - #  west-150.dat  boulder pt (pt fairy) 52-150     K=90% Bou
```

Second 2

□ Use First Matrix Only

Fitting Details

Commercial Length Frequency

○ Multnomial 1000 Median Number Multinomial Regression)

● Fournier Robust 1 Effective Sample Size (Fournier Robust Regression)

Independent Length Frequency

○ Multnomial 1000 Median Number Multinomial Regression)

● Fournier Robust 200 Effective Sample Size (Fournier Robust Regression)

CPUE

CV for CPUE fit. (zero estimates from data) 0

Independent Abundance

CV for abundance fit. (zero estimates from data) 0.2

□ Use single q in abundance fit (> 1 class size)

Model

Smallest length, largest length and length increment for the model (in mm).

Minimum fit sets to zero, any class size frequencies, below a defined value.

Lengths for Length Frequency (Commercial and Independent)

The smallest and largest sizes in the Commercial and Survey length data, and the width of the length classes in this data.

First and Second Growth Matrices

When using growth matrices, two matrices are used as upper and lower bounds for growth. This parameter (G_w) is the proportion of the first matrix added to ($1-G_w$) of the second matrix.

If a different growth is to be used for each season, enter the number of the growth matrix to be used in each season in order.

Use First Growth Matrix Only

Select this option for use with a single growth matrix. This effectively turns off fitting to G_w, the growth matrix weight.

Length Frequency fitting (Commercial & Independent)

Allows a choice of Fournier robust likelihood function, or Multinomial likelihood. The effective sample size in the robust likelihood function needs to be set if the Fournier robust likelihood function is used. The assumed median size is set if multinomial fitting is used. The minimum size that is fitted is defined.

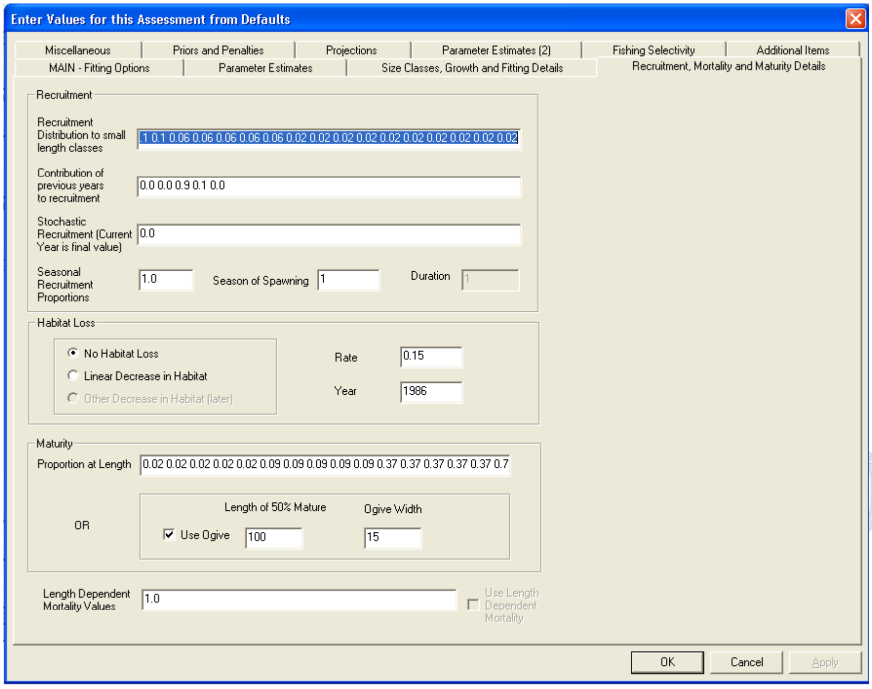
Abundance Fit q value

If abundance data defines two distinct classes, select whether q is taken from the geometric mean of the q from the individual classes.

Abundance Fit CV and CPUE

Select a defined CV for survey abundance and CPUE, or generate from data.

Recruitment Mortality and Maturity Data



Recruitment

Recruitment Distribution to small length classes: .1 0.1 0.06 0.06 0.06 0.06 0.06 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02

Contribution of previous years to recruitment: 0.0 0.0 0.9 0.1 0.0

Stochastic Recruitment (Current Year is final value): 0.0

Seasonal Recruitment Proportions: 1.0 Season of Spawning: 1 Duration: 1

Habitat Loss

● No Habitat Loss
○ Linear Decrease in Habitat
○ Other Decrease in Habitat (later)

Rate: 0.15

Year: 1986

Maturity

Proportion at Length: 0.02 0.02 0.02 0.02 0.02 0.09 0.09 0.09 0.09 0.09 0.37 0.37 0.37 0.37 0.37 0.7

OR

Length of 50% Mature Ogive Width

☑ Use Ogive 100 15

Length Dependent Mortality Values: 1.0 ☐ Use Length Dependent Mortality

Omega - Recruitment Distribution

The relative number of the recruited abalone population allocated to each length class, stating from the smallest. Any values unspecified are assumed to be 0. If they do not add to 1, they are converted to proportions.

Contributions of previous years to recruitment

The relative contribution to recruitment from each year's spawning biomass, starting with the current year and going back in time. If they do not add to 1, they are converted to proportions.

Initial Recruitment (from current year backwards)

The stochastic recruitment values can be set to be different for each year. They can be entered here as a list. If the number of values in the list is less than the number of years fished, the missing years are assumed to be at the start of the time series and to be zero.

Seasonal recruitment proportions

The proportion of the total annual recruited abalone allocated to season.

Season of Spawning

The season in which spawning is assumed to occur (used in calculation of spawning biomass).

Habitat Loss

The Habitat Loss function allows the modeller to use a declining recruitment function to simulate a period in which the abalone habitat was diminished. Currently only a linear decrease is available.

Maturity

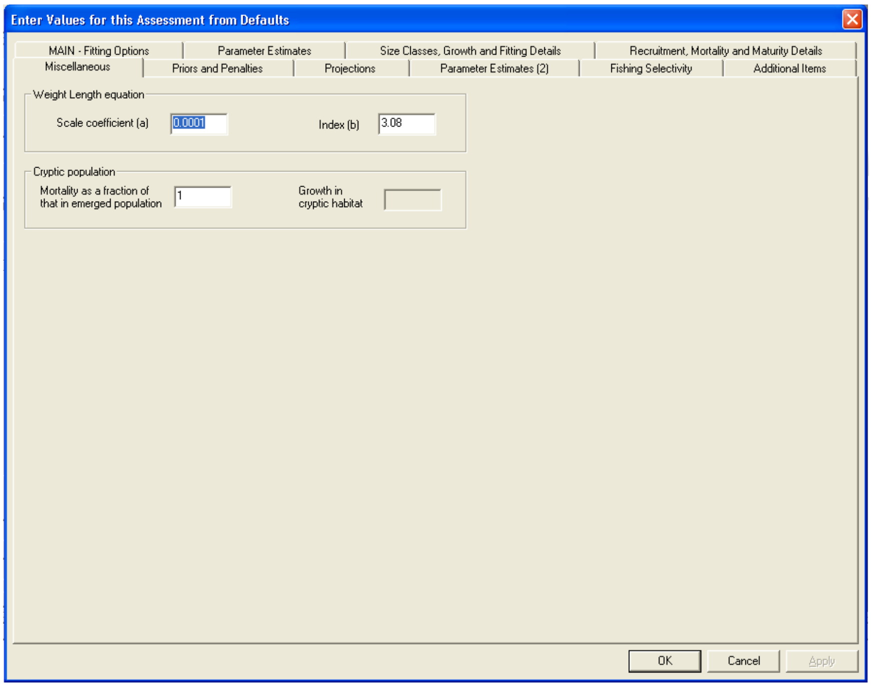
This is either

- An ogive where L_{50} and L_{width} (i.e. L_{95}-L_{50}) are specified or
- A list of the proportion mature at each length. If the list contains less than the number of length classes, the (larger) missing ones are assumed to be 1.

Length Dependent Mortality

Mortality at Length uses the distribution specified in the edit box on this form. The distribution is scaled relative to the minimum value, which is set equal to the mortality parameter on the main parameters page. The mortality parameter can be either set or estimated (with or without priors).

Miscellaneous



Weight Length equation

Scale coefficient (a) 0.0001 Index (b) 3.08

Cryptic population

Mortality as a fraction of that in emerged population 1 Growth in cryptic habitat

Weight size parameters

a is the coefficient and b is the index in the weight-size relationship where

weight = a * (length)^b

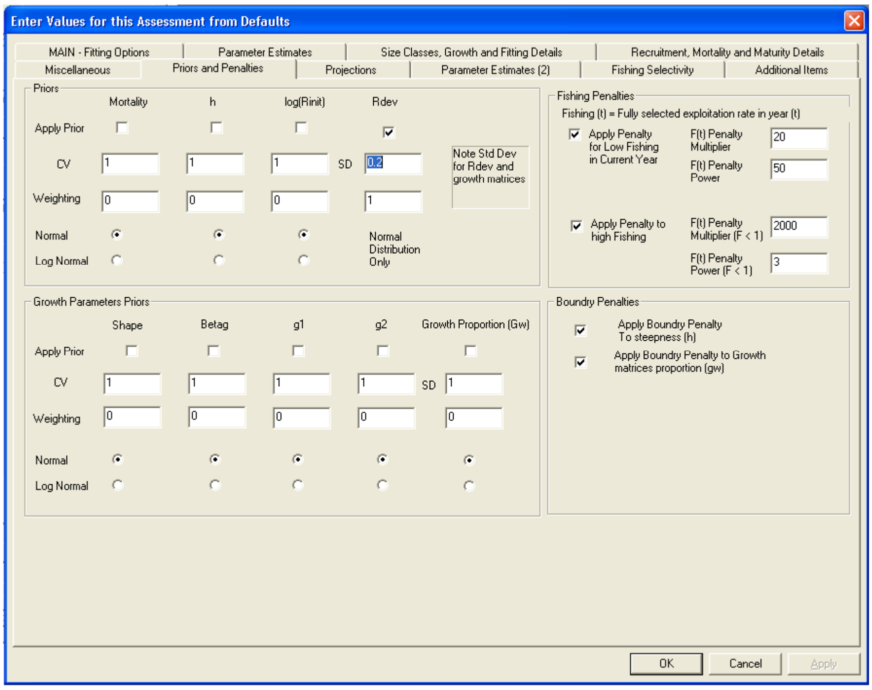
(weight is in kg and length is in cm)

Cryptic population component

Mortality – mortality in the cryptic component can be set as a fraction of the mortality in the emerged population. The default is 1.

Growth – the proportion of the growth matrix parameter (G_w) assumed for the cryptic population. The default is 1. This is currently not available.

Priors and Penalties



Priors

	Mortality	h	log(Rinit)	Rdev
Apply Prior	□	□	□	☑
CV	1	1	1	SD 0.2
Weighting	0	0	0	1
Normal	●	●	●	Normal Distribution Only
Log Normal	○	○	○	

Note Std Dev for Rdev and growth matrices

Growth Parameters Priors

	Shape	Betag	g1	g2	Growth Proportion (Gw)
Apply Prior	□	□	□	□	□
CV	1	1	1	1	SD 1
Weighting	0	0	0	0	0
Normal	●	●	●	●	●
Log Normal	○	○	○	○	○

Fishing Penalties

Fishing (t) = Fully selected exploitation rate in year (t)

☑ Apply Penalty for Low Fishing in Current Year

F(t) Penalty Multiplier: 20

F(t) Penalty Power: 50

☑ Apply Penalty to high Fishing

F(t) Penalty Multiplier (F < 1): 2000

F(t) Penalty Power (F < 1): 3

Boundry Penalties

☑ Apply Boundry Penalty To steepness (h)

☑ Apply Boundry Penalty to Growth matrices proportion (gw)

Priors.

A lognormal prior is on the stochastic recruitment deviations. The standard deviation for this is set here.

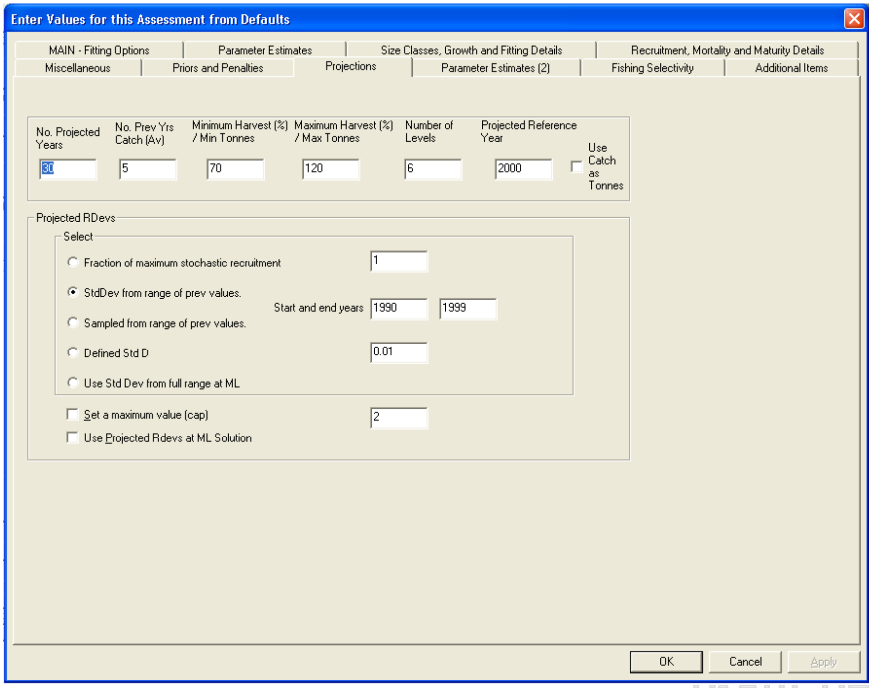
Mortality, steepness (h) and log(Recruitment in a virgin stock) can have normal priors set about the initial estimate values. The CV for each of these is set here. They are switched off with the tick boxes, or with a weighting of 0 (default).

Log normal priors can be chosen as an alternative by checking the log normal prior checkbox .

Penalties for testing.

Choose whether to apply boundary penalties to Beverton-Holt steepness (h), growth weighting Gw and fishing exploitation rate, F.

Projections and Reference Points



No. Projected Years	No. Prev Yrs Catch (Av)	Minimum Harvest (%) / Min Tonnes	Maximum Harvest (%) / Max Tonnes	Number of Levels	Projected Reference Year	
30	5	70	120	6	2000	□ Use Catch as Tonnes

Projected RDevs

Select

○ Fraction of maximum stochastic recruitment 1

● StdDev from range of prev values.

Start and end years 1990 1999

○ Sampled from range of prev values.

○ Defined Std D 0.01

○ Use Std Dev from full range at ML

□ Set a maximum value (cap) 2

□ Use Projected Rdevs at ML Solution

Projection (Years and reference catch)

- Number of years to project after the last year fished.

- The reference catch is taken as the average of the last n years fished, where n is specified here. (n = 1 projects the last years catch into the future)

- The projected reference year is the index year for the Biomass Indicators BRef/B0 etc. output in the results summary tables.

Projected harvest levels.

The model is projected at a range of future catches. These are set as a percentage of the reference catch (above). The lowest level, the highest level and the total number of levels are specified here.

Projected recruitment deviations

The options for the stochastic recruitment deviations in projected years are

- type 1: stochastic log normal, with standard deviation as fraction of the maximum bound for the fished years in the estimation process
- type 2: standard deviation taken from the range of (past) years specified.
- type 3: recruitment deviations sampled from the range of (past) years specified.
- type 4: stochastic log normal, with standard deviation given
- type 5: standard deviation taken from the full range of (past) years

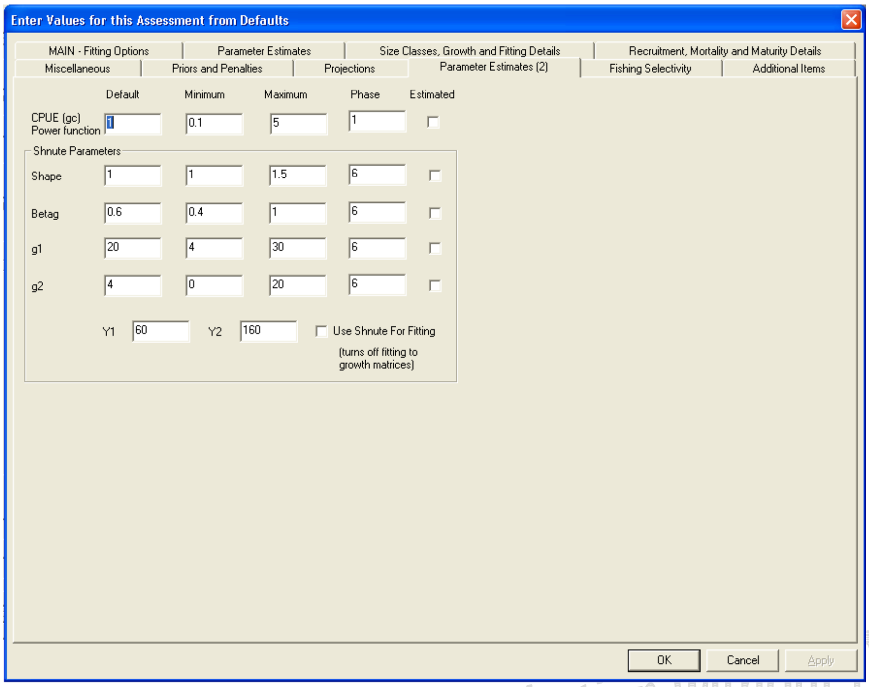
In addition, a maximum value can be specified. If the stochastic value is greater, then it is truncated.

At maximum likelihood fit, projected recruitment deviations can be selected, or left out.

Note that Biomass in each year as a fraction of the biomass in each reference year can be calculated from the full biomass series in the Results database. Reports are provided in the sample database.

Parameter estimation (2) for Schnute and CPUE

These follow those on the Parameter Estimates page.



	Default	Minimum	Maximum	Phase	Estimated
CPUE (gc) Power function	1	0.1	5	1	□
Shnute Parameters					
Shape	1	1	1.5	6	□
Betag	0.6	0.4	1	6	□
g1	20	4	30	6	□
g2	4	0	20	6	□

Y1 60 Y2 160 □ Use Shnute For Fitting

(turns off fitting to growth matrices)

Columns

Default = Initial value if the parameter is estimated, or set value if not estimated.

Minimum = minimum value if the parameter is estimated (otherwise, ignored).

Maximum = maximum value if the parameter is estimated (otherwise, ignored).

Phase = the phase in which the parameter is first estimated (otherwise, ignored).

Estimated = tick if the parameter is to be estimated, the alternative is the parameter is fixed at the default value.

Schnute growth parameters

Shape, Beta G (β_G), g1, g2 can be estimated.

Lengths y1 and y2 need to be specified.

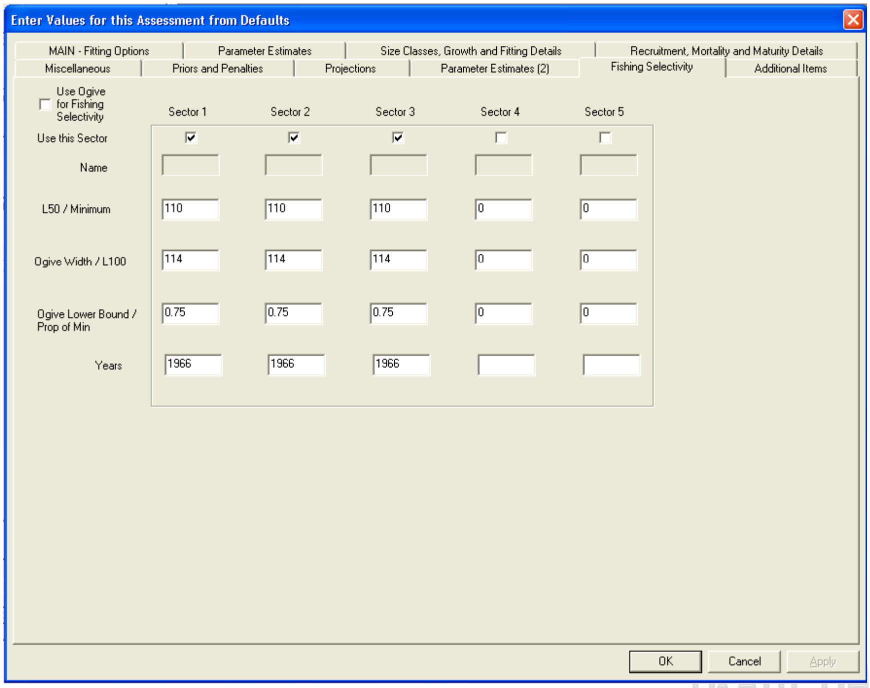
Use Schnute for fitting (growth)

This needs to be ticked if Schnute growth equations are to be used.

CPUE power function

Index in the commercial CPUE fitting (g_c).

Fishing Selectivity



☐ Use Ogive for Fishing Selectivity

	Sector 1	Sector 2	Sector 3	Sector 4	Sector 5
Use this Sector	☑	☑	☑	☐	☐
Name					
L50 / Minimum	110	110	110	0	0
Ogive Width / L100	114	114	114	0	0
Ogive Lower Bound / Prop of Min	0.75	0.75	0.75	0	0
Years	1966	1966	1966		

Sectors

Fishing sectors include commercial, recreational, unaccounted (illegal, undeclared, other).

Selectivity

Fishing selectivity can be set either as

- an ogive - set L50 (length at which 50% are selected) and L_{width} (L95-L50) and with an optional lower bound.

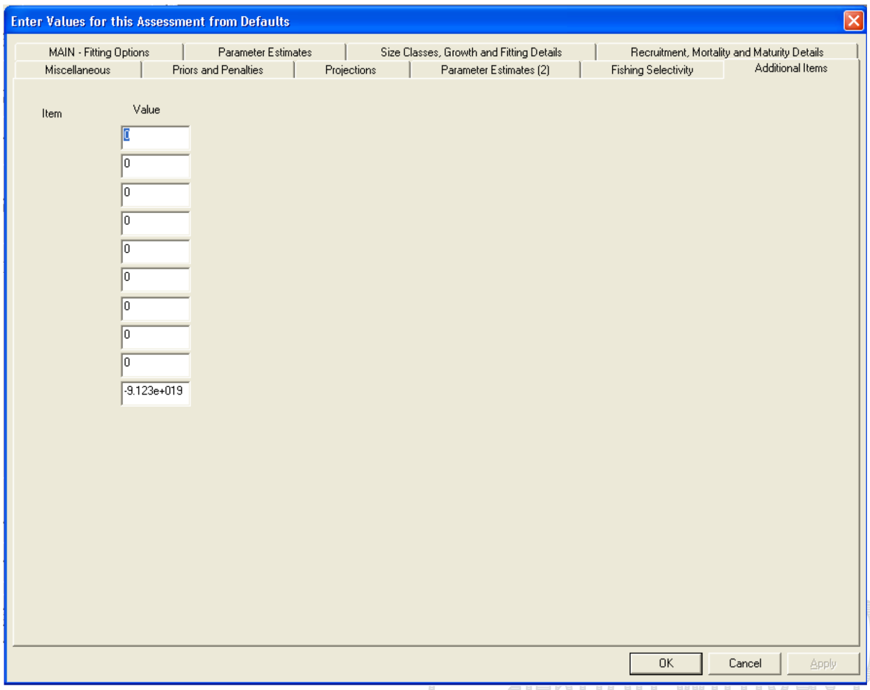
- linear increase – set minimum length (eg LML), first length fully selected and proportion selected at minimum length.

Tick ogive checkbox to select ogive.

Multiple fishing selectivity's can be specified by entering a list if values in each field. The same number of values is required in each field. The year is the first year that the specified fishing selectivity is applied, and it remains until a further year is encountered. The first value is applied back to the first year if it is greater then the first year of fishing

Additional Items

Used to define up to 10 additional numeric floating-point items to the model. Typically used by modellers to test new functionality in the model engine before making changes definite in the GUI.



Item	Value
	0
	0
	0
	0
	0
	0
	0
	0
	0
	-9.123e+019

Running AbModeller

Introduction

Definition: Here we use model data for contents of input database and model specifications for GUI input.

AbModeller supports the modelling process of data definition and review through model estimation, the MCMC process and evaluation.

Model specifications are collected through ten input forms and saved to an MS Access® database. This *results* database is also the repository for outputs from the estimation and MCMC processes.

From the input data in the input db and from the specifications of the model, the traditional Ad Model Builder® text input file (modelengine.dat) is generated. The DOS based ModelEngine process gets kick-started from the GUI and is given access to the input file to initialise in the usual way. Additional arguments are sent as command line parameters.

The user has three primary run options.

- A simple run. Produces estimated and derived parameters and biomass estimates at maximum likelihood solution.
- MCMC – Same as 1 above but also performs MCMC for defined number of cycles and produces set of saved parameter values (needs variance-covariance matrix, ModelEngine.cor, from first run to start). There is the option to set the MCMC process to repeatedly start from alternative start points.
- Evaluation. Available only after a successful MCMC run has occurred. Takes saved parameter file from MCMC above as input. Produces data sets of yearly biomass estimates and parameter values for each of

the saved parameter sets. Makes forward projections based on varying harvest strategies (levels of constant harvest).

During a run, most of the GUI menu actions are disabled, however the engine may be paused (or terminated) allowing the user to view the progress of the fitting in a scroll-able window. Run output is dispatched to both the GUI display and, when necessary, saved into the results database.

When the run completes, the GUI again becomes fully available. The user may then view the output of the selected run option or go on with further run types.

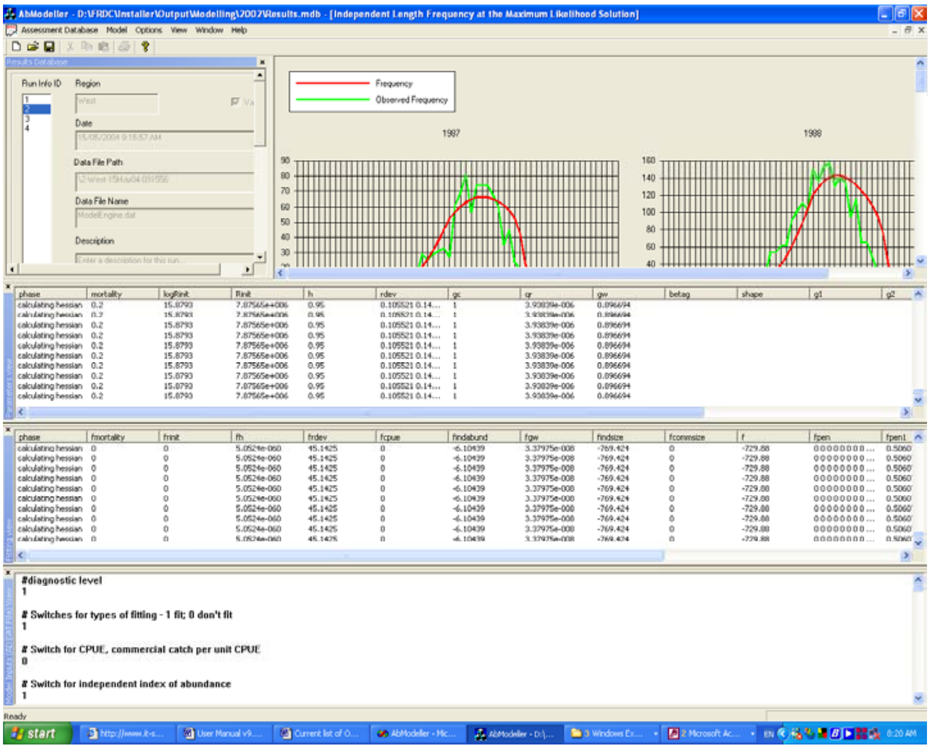
GUI Elements

Workspace

The Workspace comprises the full set of views.

Figure 1. Shows:

1. Top Left – Run Info View and Run Info Menu Options
2. Top Right – Chart of Length Frequency Fit
3. Bottom – Parameter Summary
4. Two Middle – Parameter and Fitting View



AbModeller - D:\FRDC\Installer\Output\Modelling\2002\Results.mdb - [Independent Length Frequency at the Maximum Likelihood Solution]

Run Info ID: 1, 2, 3, 4

Region: West

Date: 15/05/2004 9:15:57 AM

Data File Path: \2-West-15May04-091556...

Data File Name: ModelEngine.dat

Description: Enter a description for this run...

phase	mortality	logRinit	Rinit	h	rdev	gc	gr	gw	betag	shape	g1	g2
calculating hessian	0.2	15.8793	7.87565e+006	0.95	0.105521 0.14...	1	3.93839e-006	0.896694				
calculating hessian	0.2	15.8793	7.87565e+006	0.95	0.105521 0.14...	1	3.93839e-006	0.896694				
calculating hessian	0.2	15.8793	7.87565e+006	0.95	0.105521 0.14...	1	3.93839e-006	0.896694				
calculating hessian	0.2	15.8793	7.87565e+006	0.95	0.105521 0.14...	1	3.93839e-006	0.896694				
calculating hessian	0.2	15.8793	7.87565e+006	0.95	0.105521 0.14...	1	3.93839e-006	0.896694				
calculating hessian	0.2	15.8793	7.87565e+006	0.95	0.105521 0.14...	1	3.93839e-006	0.896694				
calculating hessian	0.2	15.8793	7.87565e+006	0.95	0.105521 0.14...	1	3.93839e-006	0.896694				
calculating hessian	0.2	15.8793	7.87565e+006	0.95	0.105521 0.14...	1	3.93839e-006	0.896694				
calculating hessian	0.2	15.8793	7.87565e+006	0.95	0.105521 0.14...	1	3.93839e-006	0.896694				
calculating hessian	0.2	15.8793	7.87565e+006	0.95	0.105521 0.14...	1	3.93839e-006	0.896694				

phase	fmortality	frinit	fh	frdev	fcpue	findabund	fgw	findsize	fcommsize	f	fpen	fpen1
calculating hessian	0	0	5.0524e-060	45.1425	0	-6.10439	3.37975e-008	-769.424	0	-729.88	0 0 0 0 0 0 0 0 0 ...	0.5060
calculating hessian	0	0	5.0524e-060	45.1425	0	-6.10439	3.37975e-008	-769.424	0	-729.88	0 0 0 0 0 0 0 0 0 ...	0.5060
calculating hessian	0	0	5.0524e-060	45.1425	0	-6.10439	3.37975e-008	-769.424	0	-729.88	0 0 0 0 0 0 0 0 0 ...	0.5060
calculating hessian	0	0	5.0524e-060	45.1425	0	-6.10439	3.37975e-008	-769.424	0	-729.88	0 0 0 0 0 0 0 0 0 ...	0.5060
calculating hessian	0	0	5.0524e-060	45.1425	0	-6.10439	3.37975e-008	-769.424	0	-729.88	0 0 0 0 0 0 0 0 0 ...	0.5060
calculating hessian	0	0	5.0524e-060	45.1425	0	-6.10439	3.37975e-008	-769.424	0	-729.88	0 0 0 0 0 0 0 0 0 ...	0.5060
calculating hessian	0	0	5.0524e-060	45.1425	0	-6.10439	3.37975e-008	-769.424	0	-729.88	0 0 0 0 0 0 0 0 0 ...	0.5060
calculating hessian	0	0	5.0524e-060	45.1425	0	-6.10439	3.37975e-008	-769.424	0	-729.88	0 0 0 0 0 0 0 0 0 ...	0.5060
calculating hessian	0	0	5.0524e-060	45.1425	0	-6.10439	3.37975e-008	-769.424	0	-729.88	0 0 0 0 0 0 0 0 0 ...	0.5060
calculating hessian	0	0	5.0524e-060	45.1425	0	-6.10439	3.37975e-008	-769.424	0	-729.88	0 0 0 0 0 0 0 0 0 ...	0.5060

```
#diagnostic level
1

# Switches for types of fitting - 1 fit; 0 don't fit
1

# Switch for CPUE, commercial catch per unit CPUE
0

# Switch for independent index of abundance
1
```

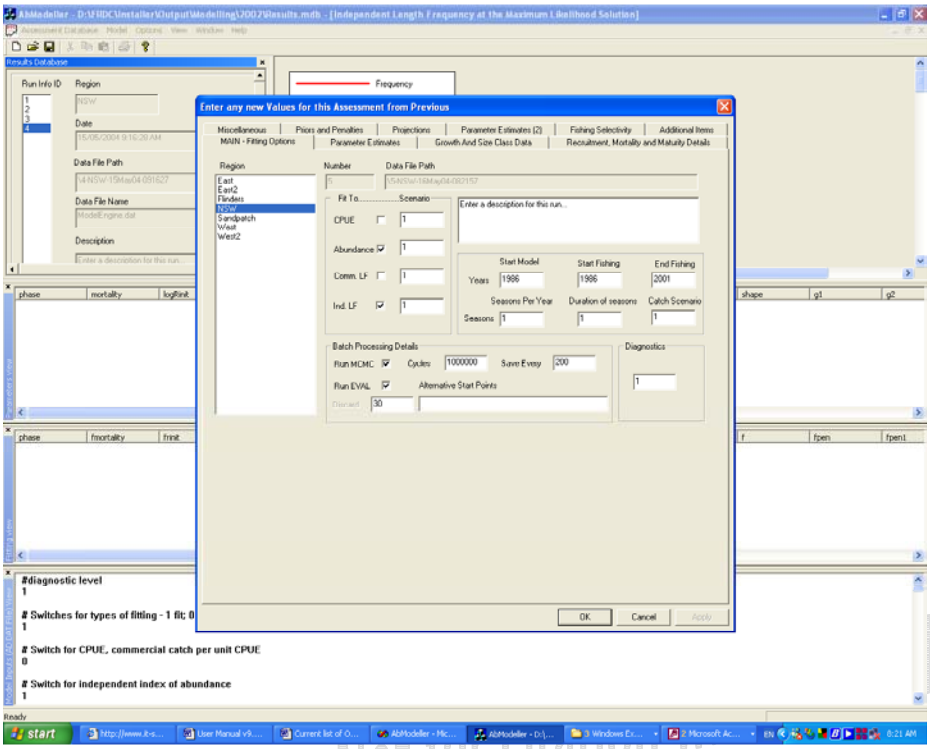
Ready

Run Info View

The Run Info View houses summary information pertaining to each of the Run Infos populated from the currently open results database. The user scrolls through available Run Infos to initiate edit or run tasks to be performed on the selected item.

Run Info Editor

The Run Info Editor allows the user to specify model data for a Run Info. The tabbed window groups the model data into appropriate areas.



Run Info ID: 1, 2, 3, 4
Region: NSW
Date: 15/05/2004 9:16:28 AM
Data File Path: \4-NSW-15May04-091627
Data File Name: ModelEngine.dat
Description: Enter a description for this run...

Enter any new Values for this Assessment from Previous

Region: East, East2, Flinders, NSW, Sandpatch, West, West2
Number: 5
Data File Path: \5-NSW-16May04-082157

Fit To.............Scenario
CPUE ☐ 1
Abundance ☑ 1
Comm. LF ☐ 1
Ind. LF ☑ 1

Enter a description for this run...

	Start Model	Start Fishing	End Fishing
Years	1986	1986	2001
	Seasons Per Year	Duration of seasons	Catch Scenario
Seasons	1	1	1

Batch Processing Details
Run MCMC ☑ Cycles 1000000 Save Every 200
Run EVAL ☑ Alternative Start Points
Discard 30

Diagnostics 1

#diagnostic level
1
Switches for types of fitting - 1 fit; 0
1
Switch for CPUE, commercial catch per unit CPUE
0
Switch for independent index of abundance
1

Parameter and Fitting Views

Windows are available to observe and analyse the model's progression. The Parameter View shows current and past values for each of the model parameters whilst the Fitting View depicts the movement of the individual components that make up the overall objective function value.

Combined with the ability to pause and restart the model engine process, these views provide the user with improved observation during the fitting process compared to a more conventional DOS based text output window. Future releases may provide further enhancements to these windows.

Summary

The Summary View shows the values of each of the parameters and derived parameters at the maximum likelihood solution.

Data File Viewer

A read only view of the underlying Data file created for a Run Info is shown in this window. Although hidden from the user, AbModeller still uses this file as the interface between the GUI and the model engine. When the engine is being used in stand-alone mode, a data file created from the GUI is probably the best starting point.

Charting Features

AbModeller provides a set of default charts including fits to Length Frequency and Abundance and a standard biomass depletion curve at the maximum likelihood solution. Since AbModeller uses a MS Access® database for its output, users are encouraged to use MS Access® to provide further chart options.

Additional built-in charts may become available in the future.

Run Info Features and Functions

Menu Selection

Run Info functions can be initiated from either the main menu or by right clicking on a Run Info item in the Run Info View. Obviously the action applies to the currently selected Run Info.

Default Settings

Defaults come from either the default Run Info table in the input (modelling) database or from hard-wired values in the application software if there is no Run Info table. It is also possible to set the defaults from an existing Run Info using Model – Set Run Info as Default from the menu or to edit the default values in the input database.

Note that some hard-wired values, or initial default values, may seem unsuitable since they reflect samples of the current Victorian assessment.

Create

Once a results database is established, it is possible to create new Run Infos. The editor is designed to categorise the myriad of data required for a single run. The Create action populates the fields with default values. The Run Info number is automatically generated by the database and cannot be selected.

After the Run Info has been accepted a new Run Info is added to the database. A subfolder in the location of the results database is created to house the wealth of data files soon to be generated by the estimation process. The following convention based on a time stamp provides uniqueness across the Run Infos.

<RunInfoNumber>-<region>-<DD><MMM><YY>-<HH><MM><SS>

eg. 8-SouthWest-06Feb03-091332

At this stage the ADMB® input data file (currently modelengine.dat) is generated and placed in the above folder.

Edit

Existing Run Infos may be edited, however all related fitting and projection data will be removed from the results database and the modelengine.dat file will be regenerated. A warning is given before the edit action is allowed to continue.

Clone

Clone creates a new Run Info from the currently selected one with the exception of the Run Info number, which gets automatically generated.

View

View only editor.

Delete

Existing Run Infos may be deleted along with all related fitting and projection data. A warning is given before the edit action is allowed to continue.

Modify Description

This simple feature allows the user to change the description of a Run Info without losing associated results data.

Data Checks

Although not exhaustive at this stage, AbModeller attempts to verify the user's data entry values. For example the start year cannot be greater than the final year of fishing Database errors such as a region without catch cannot be controlled, so model input data must be carefully scrutinised.

If an error occurs in the generation of a Run Info, a warning is issued to the user and it is flagged as invalid. Although the output folder is created, no input file is produced. Warnings produce an error message but are recoverable in most instances.

Other Model Run Options

Batch Runs

A batch processing feature is used to assist in unaccompanied extended sets of runs. There is the option to override the MCMC and Eval values that are specified in the Run Infos.

Command Line

ModelEngine may be used as a stand-alone process through a command line interface with batch processing achieved with DOS-like batch files.

Components of the command line are as follows-

```
<EXE_NAME> <NUMBER> <DAT_PATH> <DATABASE_PATH> <RUNINFO_NUM>
<BATCH_JOB_FLAG> <DIAGNOSTIC_FLAG> <OPTION1>
< ARGUMENT1> < OPTION2> <ARGUMENT1>
```

- EXE_PATH: ModelEngine.exe name and path
- NUMBER: can be any valid integer
- DAT_PATH: Full path to and name of modelengine.dat file
- DATABASE_PATH: Full path and name of output (results) database
- RUNINFO_NUM: must be a valid run info number in the results database
- BATCH_JOB_FLAG: 0, if not a batch job
- DIAGNOSTIC _FLAG: 0, for no diagnostics to the GUI
- OPTION1: first admb option eg -mcmc. These options are not set if estimating only.
- ARGUMENT1: first admb argument eg number of mcmc cycles
- OPTION2: second admb option eg -mcsave
- ARGUMENT2: second admb argument eg number for mcsave

Examples of this (using Run info number 8) are:

```
C:\AbModeller\ModelEngine.exe 721128 C:\AbModeller\Modelling\8-SouthWest-14Mar03-
130332\modelengine.dat  C:\AbModeller\Modelling\Results.mdb 8 0 0
```

```
C:\AbModeller\ModelEngine.exe 721128 C:\AbModeller\Modelling\8-SouthWest-14Mar03-
130332\modelengine.dat  C:\AbModeller\Modelling\Results.mdb 8 0 0 -mcmc 1000000 -mcsave 200
```

Results Output Database

The results database stores user run information and associated run outputs. To open an existing results database select Open from the Assessment menu. To create a new results database select New from the Assessment menu. Once a results database is open, model Run Info items can be created edited and deleted using the GUI. To close a results database, select Close from the Assessment menu item.

The structure of the database is not particularly complicated apart from the link between the Run Info table and each of the run output tables. The Run Info table has many fields, each of which corresponds to a value entered in the Run Info editor.

Tables in the output (results) database.

There are several tables in the database that can be categorised as follows:

Run Info

Contains fields that completely define a model run.

Version

Contains the version number of the database.

Abundance

Yearly observed and estimated Independent Abundance data grouped into class size categories.

CPUE

Yearly observed and estimated CPUE data.

Length Frequency

CommercialLengthFrequency - Yearly observed and estimated Commercial Length Frequency data.

IndependentLengthFrequency - Yearly observed and estimated Independent Length Frequency data.

ModelInfo

ModelInfo – For each fit, the positive definiteness is stored.

The next items have independent tables for both MCMC and the Maximum Likelihood

Biomass Tables

Biomass –Yearly values of Biomass for the years fished.

LegalBiomass –Yearly values of the biomass selected by sector 1 for the years fished. This is the biomass available to the Commercial divers if sector 1 defined as such.

Projected Biomass Tables

ProjectedBiomass – Projected Yearly values of Biomass at various Harvest strategies.

ProjectedLegalBiomass – Projected Yearly values of Biomass selected by sector 1 at various Harvest strategies.

Summary data for projected legal and mature biomass ratios at 15 and 30 years into the future are placed into a summary table.

Parameter and General Value Tables

General – Derived parameters of interest and Objective function values.

Parameters – Values of parameters.

Likelihoods - All of the fitting values are stored in a separate table.

Rdev

Contains fields that completely describe the estimated stochastic recruitment (R_{dev}) values.

MCMC tables

There are equivalent tables for the output from a MCMC run;

ParametersMCMC
GeneralMCMC
LikelihoodsMCMC
BiomassMCMC
LegalBiomassMCMC
BiomassSummaryMCMC
RdevMCMC
ProjectedBiomassMCMC
ProjectedLegalBiomassMCMC

A Quick Guide to running AbModeller

Steps in the Modelling Process

1. Set up AbModeller from the CD
2. Establish a working directory, modelling and results databases and the location of the two required executable files (AbModeller.exe and ModelEngine.exe). The results database can either be an existing database, or a new one generated from the GUI.
3. Add data into modelling database
4. Growth matrices can be generated from tagging data if required using the accompanying Bardos growth package (Bardos, 2004).
5. Open modelling and results databases
6. Define a model, either new or by cloning an existing one
7. Run maximum likelihood estimate with **Model** - **Run**
8. The development of the parameters are available in **View** - **Parameters**, **Function Values**. The model runs in a console (DOS) window.
9. "Press any key" in console (DOS) window to exit run of model
10. The graphs in the GUI (**View - Length Frequency fit**, **Abundance fit** and **Biomass Depletion** menu items) are useful for investigating the initial fit. The length frequency fits are useful checks on the choice of growth and mortality.
11. To move from the estimation to the MCMC stage, the correlation matrix (ModelEngine.cor) is needed. The MCMC cannot run if a message that the "hessian is not positive definite" is generated at the end of the estimation stage.
12. Check length of MCMC run, and number saved, in **Model – Edit**. Diagnostics (0) turns off the outputs to AbModeller, and runs MCMC faster.
13. Run **Model – MCMC** (also runs Estimate on the saved values)

14. Analyse data (sample analysis and reporting)

Troubleshooting

Limitations

- Not to be used for general modelling.
- Fixed to 5 sectors.

Known Bugs

- Deleting first RunInfo will cause subsequent RunInfo to have an ID as 2 (correct functionality) but the data file path will suggest an ID of 1. (incorrect functionality) This is not repeated for a SimInfo since the SimID is not an auto-incremented field.
- Use abort with caution. Database may become inconsistent. Transaction based updates is not implemented.
- Group radio boxes (eg. Priors distributions) are not disabled when in Run Info Viewing Mode.
- Ignore message at start of model running in console(DOS) window which says "warning -- calling gradcalc when no calculations generating derivative information have occurred"
- Simulation procedures are still to be documented.
- There is a restriction

Current Developments

- Individual seasonal, or annual, CVs for Independent Abundance or CPUE.
- Multinomial fitting for Length Frequency
- MSY estimates
- Length dependent mortality (relative to the estimated or set value)

Sample data

Sample Data

Sample input and results databases are included. A working copy, based on draft Victorian data, is made available as a sample. These are in both Access 97 and Access 2000 formats.

Reporting

Introduction

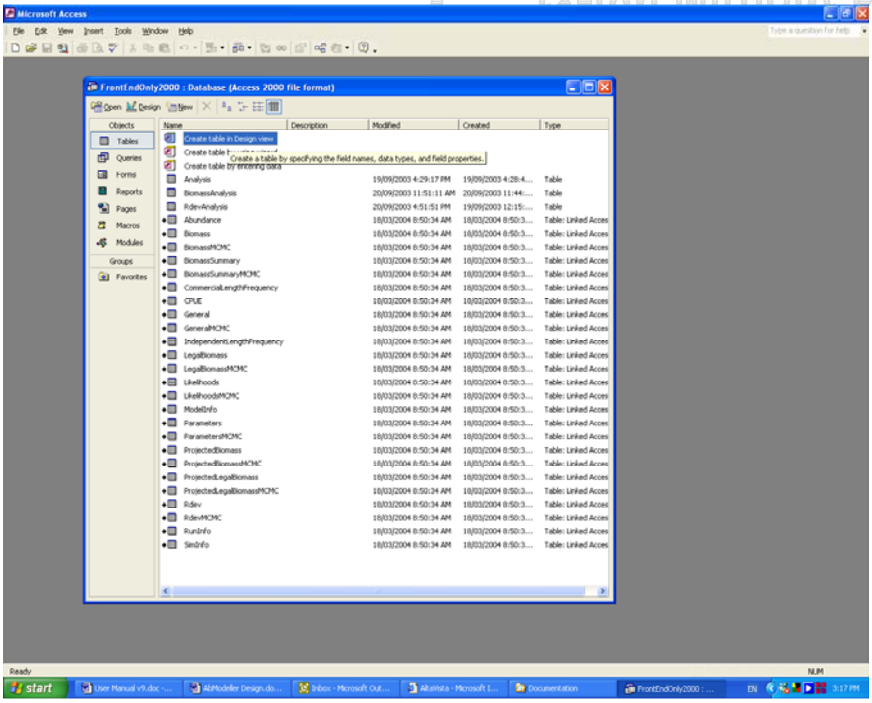
One of AbModeller's main features is the MS Access database interface. This gives the user the opportunity to produce management oriented reports on many of the fields of interest.

The reporting function has been offloaded into a separate Access database. A sample (ResultsAnalyser2000.mdb or ResultsAnalyser.mdb – for Access 97) is provided on the distribution and quite straight forward to use.

- First open the database and set up the links to the AbModeller database tables. Use the inbuilt function of Access to do this operation, and link in all tables from the results database. The catch table from the associated modelling database also needs to be linked.
- Use the button "Analyse Data" on the Access Form to trigger the block of Access Basic software for the MCMC analysis.
- Open the Sample reports using the "Run Report" and "Run MCMC Report" buttons on the same form to make ready for printing. Save them into the pdf format if you have the facility for easy distribution.

Analysis Process

Each field of every Results database table generated during the MCMC evaluation stage is analysed for the important statistical aspects including the 5, 20, 30, 70 and 95 percentiles, the medium, the mean and standard deviation.



Create table in Design view
Create table by using wizard
Create table by entering data
Create a table by specifying the field names, data types, and field properties.

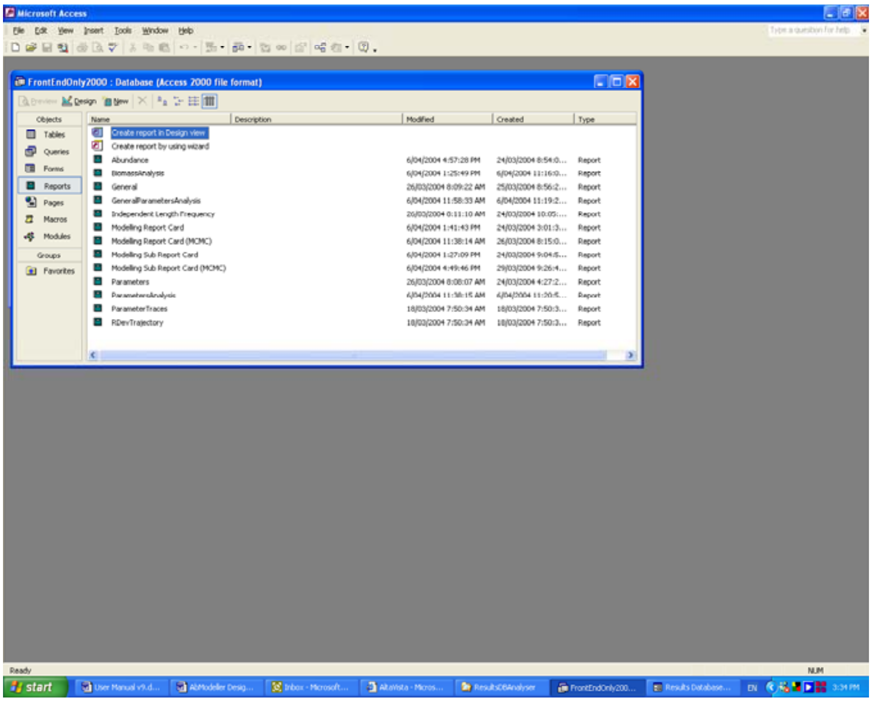
Name	Description	Modified	Created	Type
Analysis		19/09/2003 4:29:17 PM	19/09/2003 4:28:4...	Table
BiomassAnalysis		20/09/2003 11:51:11 AM	20/09/2003 11:44:...	Table
RdevAnalysis		20/09/2003 4:51:51 PM	19/09/2003 12:15:...	Table
Abundance		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
Biomass		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
BiomassMCMC		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
BiomassSummary		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
BiomassSummaryMCMC		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
CommercialLengthFrequency		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
CPUE		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
General		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
GeneralMCMC		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
IndependentLengthFrequency		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
LegalBiomass		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
LegalBiomassMCMC		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
Likelihoods		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
LikelihoodsMCMC		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
ModelInfo		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
Parameters		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
ParametersMCMC		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
ProjectedBiomass		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
ProjectedBiomassMCMC		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
ProjectedLegalBiomass		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
ProjectedLegalBiomassMCMC		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
Rdev		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
RdevMCMC		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
RunInfo		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces
SimInfo		18/03/2004 8:50:34 AM	18/03/2004 8:50:3...	Table: Linked Acces

The figure above shows the linked tables from the Results Database, with the obvious annotated arrow, along with the tables generated from the processing.

The two MCMC Biomass tables are dumped into BiomassAnalysis. The Analysis table contains the analysis of the Parameters, General and Likelihoods tables. The RDevAnalysis table completes the set, containing the statistics for each of the yearly recruitment deviations.

The general algorithm is to take each field of the table, group it by Run Info and then sort it by that field discarding the number of items defined in the corresponding Run Info. The statistical entries are read off and saved by moving down the ordered set by the various percentiles. A record is then written to the output table.

A results database typically contains multiple runs for different spatial areas sensitivity trials. The analysis groups the data by Run Info number hence the obvious correspondence between the number of runs in the Run Info table and the subsets of data in the generated analysis tables.



Name	Description	Modified	Created	Type
Create report in Design view				
Create report by using wizard				
Abundance		6/04/2004 4:57:28 PM	24/03/2004 8:54:0...	Report
BiomassAnalysis		6/04/2004 1:25:49 PM	6/04/2004 11:16:0...	Report
General		26/03/2004 8:09:22 AM	25/03/2004 8:56:2...	Report
GeneralParametersAnalysis		6/04/2004 11:58:33 AM	6/04/2004 11:19:2...	Report
Independent Length Frequency		26/03/2004 8:11:10 AM	24/03/2004 10:05:...	Report
Modelling Report Card		6/04/2004 1:41:43 PM	24/03/2004 3:01:3...	Report
Modelling Report Card (MCMC)		6/04/2004 11:38:14 AM	26/03/2004 8:15:0...	Report
Modelling Sub Report Card		6/04/2004 1:27:09 PM	24/03/2004 9:04:5...	Report
Modelling Sub Report Card (MCMC)		6/04/2004 4:49:46 PM	29/03/2004 9:26:4...	Report
Parameters		26/03/2004 8:08:07 AM	24/03/2004 4:27:2...	Report
ParametersAnalysis		6/04/2004 11:38:15 AM	6/04/2004 11:20:5...	Report
ParameterTraces		18/03/2004 7:50:34 AM	18/03/2004 7:50:3...	Report
RDevTrajectory		18/03/2004 7:50:34 AM	18/03/2004 7:50:3...	Report

Once the statistical data has been collated, the user can select from a rich set of management reports (see above) that incorporate both the model outputs from the MPD solution and from the analysis generated form the above process.

There are two complete reports in the Access Report Window

- “Modelling Report Card (MCMC)” and
- “Modelling Report Card”

the others being sub reports of the above.

Sample Report Outputs

To view the Sample reports, the reader should open the those in the sample database or alternatively view the pdf versions of the reports that accompany this distribution.

Modifications and Enhancements

System Documentation

A range of approaches to modifying and enhancing AbModeller are described in the accompanying document “AbModeller National Abalone Modelling Software, Version 1.2.1. – Software Documentation”.

This document covers three topics:

1. the **Software Architecture** – an introduction to the alternative versions of the code covering
   - the full code, including the model and its output (ModelEngine) and the GUI (Abmodeller)
   - ModelTextOutput which is a text output version of the model
   - a tpl based version

   These provide access to the code for programmers with varying levels of experience, and a range of needs. These three versions of the code are all included on the CD.

2. a Software Design Specification for the full GUI version, covering AbModeller and ModelEngine.

3. Access Analysis and Reporting - the ResultsAnalysis database

GLOSSARY

CPUE	Catch per unit effort (usually kg/day or kg/hour)
CV	Coefficient of Variation
GUI	Graphical User Interface
MCMC	Monte Carlo Markov Chain
Rdev	Stochastic recruitment deviation.
RunInfo	An identification number assigned to each set of model options and data specifications

Part 4: AbModeller Software Documentation

AbModeller Abalone Modelling Software, Version 1.2.1 – Software Documentation

Not for citation without permission

No. 49
February 2005

AbModeller Abalone Modelling Software, Version 1.2.1 – Software Documentation

M. Cleland[1] and B.L.Taylor[2]

1. 19 Hood Cres, Arrowtown, New Zealand.
2. PIRVic, Marine & Freshwater Systems, PO Box 114, Queenscliff, VIC 3225, Australia (corresponding author).

Not for citation without permission

February 2005

Department of Primary Industries

Published: Primary Industries Research Victoria,
Marine and Freshwater Systems
Department of Primary Industries, Queenscliff

General disclaimer
This publication may be of assistance to you but the State of Victoria and its employees do not guarantee that the publication is without flaw of any kind or is wholly appropriate for your particular purposes and therefore disclaims all liability for any error, loss or other consequence which may arise from you relying on any information in this publication.

Authorised by the Victorian Government, Weeroona Parade, Queenscliff 3225

Printed by PIRVic Queenscliff, Victoria.

Preferred way to cite:
Cleland, M. and Taylor, B. L. (2005) AbModeller Abalone Modelling Software, Version 1.2.1 – Software Documentation. Marine and Freshwater Systems Internal Report No 49. Primary Industries Research Victoria, Queenscliff.

Table of Contents

Introduction

This document covers three topics:

1. the **Software Architecture** – an introduction to the alternative versions of the code covering
   - the full code including the model and its output (ModelEngine) and the GUI (Abmodeller)
   - ModelTextOutput which is a text output version of the model
   - a tpl based version

   These provide access to the code for programmers with varying levels of experience, and a range of needs. These three versions of the code are all included on the CD.

2. a **Software Design Specification** for the full GUI version, covering AbModeller and ModelEngine.

3. **Access Analysis and Reporting** - the ResultsAnalysis database

Software Architecture

Alternative Software Packages

The software architecture is packaged in a way to help facilitate users with varying degrees of programming experience the opportunity to make changes. It is possible to make modifications or add new features with a *recipe* type approach (modelling existing code).

- **AbModeller**, the GUI**,** being a sophisticated application requires a higher level of knowledge about the workings of a Windows application, in particular the MFC (Microsoft Foundation Classes) which the GUI is built upon. The design specifications for this are detailed on page 3.

The **Model Engine** encapsulates the ADMB modelling process and adds a sophisticated command line interface as well as the database interfaces established by the AbModeller GUI. It also provides the application level, call-back mechanisms that enables AbModeller to control the models running behaviour.

- A **text output** version. This takes the Model Engine structure, but outputs the data to text files, rather than to a database. (The encapsulated ADMB modelling software module, provided by the source file AbaloneModel.cpp, is available as an independent package).

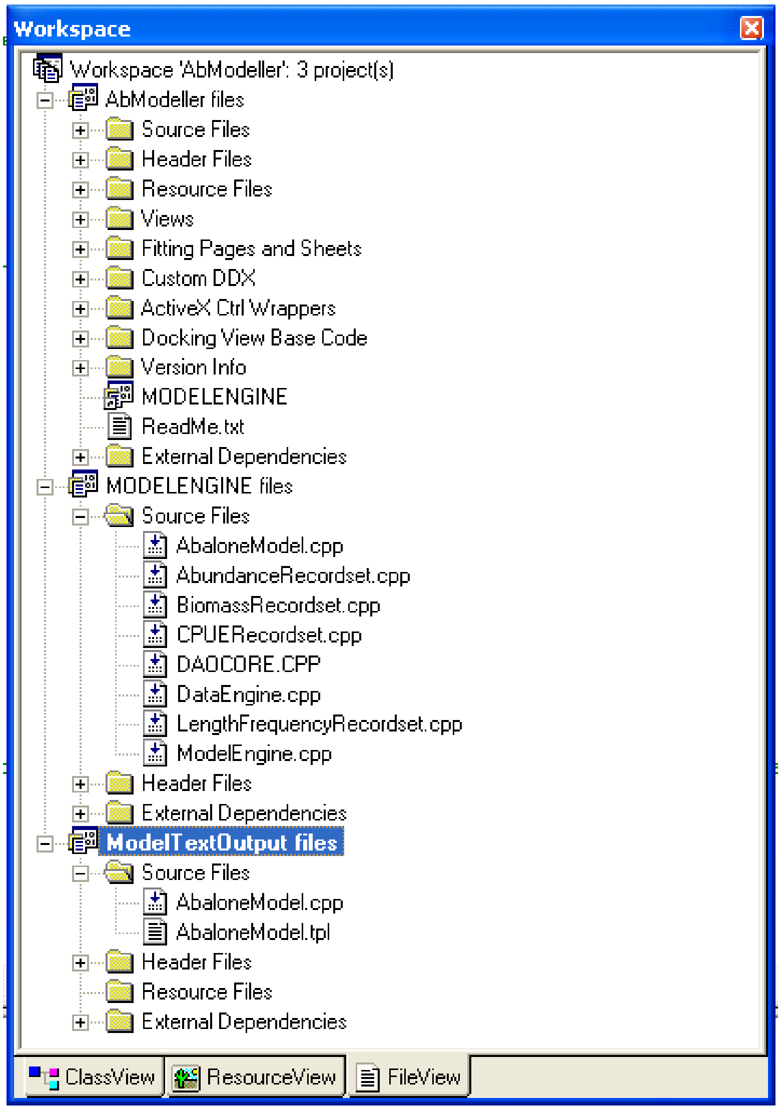
- A **tpl** version, the traditional ADMB tpl style format. Some non standard features are added in order to manage a single software module for the model across the three distinct applications.

AbModeller and Model Engine

The source code to the application is provided should the user want to make modifications or add new features to the GUI or the model engine. Although a high level of programming expertise is required, some programmers may become comfortable with the code, with the *recipe* approach to code modification, provided on request.

The Visual C++ workspace, shown below depicts AbModeller and ModelEngine projects within the software package. ModelEngine is seen as a dependency of AbModeller implying code changes to ModelEngine force a rebuild of the AbModeller application.

In fact, changes to the AbaloneModel data structure require the equivalent items to be placed somewhere within the relevant GUI input windows and added to the runinfo table of the results database.



```
Workspace
Workspace 'AbModeller': 3 project(s)
  AbModeller files
    Source Files
    Header Files
    Resource Files
    Views
    Fitting Pages and Sheets
    Custom DDX
    ActiveX Ctrl Wrappers
    Docking View Base Code
    Version Info
    MODELENGINE
    ReadMe.txt
    External Dependencies
  MODELENGINE files
    Source Files
      AbaloneModel.cpp
      AbundanceRecordset.cpp
      BiomassRecordset.cpp
      CPUERecordset.cpp
      DAOCORE.CPP
      DataEngine.cpp
      LengthFrequencyRecordset.cpp
      ModelEngine.cpp
    Header Files
    External Dependencies
  ModelTextOutput files
    Source Files
      AbaloneModel.cpp
      AbaloneModel.tpl
    Header Files
    Resource Files
    External Dependencies
ClassView | ResourceView | FileView
```

Figure 1. Visual C++ Workspace

Model Text Output

A requirement for a Linux version of the model drove the implementation of a text only Model Engine. Renamed the Model Text Output, this module is as close as you can get to a pure ADMB application. As with ModelEngine in the GUI application, the program has a shell program to handle the output (to a database in the full program, or to text files in this program). The shell program is in the ModelTextOutput subdirectory and the model component as described in the model specifications (AbaloneModel.cpp file) is in the ModelCore subdirectory. The outputs from this version are simple text files, unlike the database outputs from AbModeller. The output call from ModelTextOutput.cpp is similar to that in ADMB, with the addition of the pointer label.

The same software module is used, but compiled with the pre-processed symbol “TEXT_OUTPUT” to isolate the database calls in the model code. It would be expected that a programmer with a moderate knowledge of C++ should find working at this level ideal. In the GNU environment the same module is compiled using the “TEXT_OUTPUT” definition and linked against the ADMB libraries. Pre-processor definitions within the ADMB include files define any specific platform dependent calls from the Model Text application.

This current use of this implementation being developed in Victoria still uses the ModelEngine.dat data file generated from the AbModeller GUI as the input file. The GUI is still used to run the model to the test the initial values and the estimated parameters to the maximum likelihood stage. The linux version is simply used for the MCMC run, to take advantage of the processing power offered by the linux facility. The final estimation stage, with its large outputs, can be run either under linux, or from the standard AbModeller version. The text files can be input into access using the access “import file” feature (a form for this will be written in the Access Reporting database in a later update) so the Access reporting routines can still be used.

Additional data can be simply input into this version by defining an additional input file, as with standard ADMB.

ADMB TPL Version

A TPL version of the model is available for ADMB programmers who are used to the syntax of the TPL file. ADMB’s GLOBALS_SECTION pre-processor feature is used to include the additional source code into the model for completeness.

To isolate this level of programming from the other versions, a separate Visual C++ project and workspace is available in the distribution (in the TPLVersion subdirectory). The C++ source file from the TPL2CPP process is strictly for this version and should not be used as an input with any of the previous packages.

Design

Introduction

AbModeller consists of two executables: AbModeller.exe and ModelEngine.exe. AbModeller.exe is the GUI “front end” and ModelEngine.exe is a non-GUI “console application” which runs the model fitting and MCMC processes and generates results.

ModelEngine.exe launched by Abmodeller.exe when the user chooses to start a modelling run; it is not designed to be a standalone application or to be launched by a user, however it is possible with a knowledge of the command line options (see User Manual).

The two executables communicate with each other during a run using interprocess communication channels provided by the Windows operating system.

ModelEngine

The model engine component performs the following functions:

- It expands and encapsulates the traditional ADMB style console application with enhanced features and control via command line switching.
- Allows the model engine to run in a separate process space so any fatal errors in the engine do not affect the main AbModeller application. The independence of the model process from the GUI is due to a limitation in the ADMB libraries. It was discovered that certain ADMB class objects could not be instantiated more than once from the same process space.
- Writes the results of the modelling run to a database
- Returns the values of the model parameters to AbModeller so that they can be displayed to the user as they are generated.
- Handles fatal errors, ensuring the database is left in a consistent state when an error occurs
- Signals AbModeller at various stages of the run so that AbModeller’s user controls can be updated to reflect the progress of the run.

The model engine consists of the following key components:

- A C-style main() function. Abmodeller.exe builds a command line and passes it to the model engine. The contents of the command line specify the input data file for the modelling run, the database where the results are to be written and various switches that control the model process. The contents of the command line are extracted and passed to the various classes in the component.
- A block of "shared memory" is allocated by the model engine and made available to the GUI. This is an area of memory visible both to ModelEngine.exe and AbModeller.exe. The two executables use this area to pass results generated by the run back to AbModeller for display to the user.

CDataEngine class.

This class controls a connection to an MS-Access database where the results of each modelling run are stored. Microsoft's "Data Access Objects" (DAO) database access technology is used to interface to the database. Objects that represent the various tables in the database are opened when the run begins and are updated with new data as the modelling proceeds.

The CDataEngine class is also responsible for writing data into the shared memory area and sending a message to the AbModeller executable whenever new data is available. AbModeller removes the data from the shared memory and places it in various windows.

ADMB classes.

The ADMB Model Engine classes consist of model_parameters and model_data. These objects form the basis of the traditional ADMB model.

This ADMB model implementation interfaces to the Windows environment thru the call-back mechanism provided by the CDataEngine. A non-windows, text only version can use the same code by #defining its own version of the CDataEngine, providing the required set of interfaces the model relies on. A cleaner solution would have been to use an abstract CDataEngine interface.

AbModeller (The GUI)

AbModeller is a GUI application built using the Microsoft Foundation Classes (MFC) framework. MFC supplies a set of classes that provide a main window, child windows of various types, user interface widgets such as menus and dialogs, and wrapper classes for database access and for access to ActiveX components such as a charting control. Some familiarity with the design principles of MFC is required to perform any maintenance or enhancement of AbModeller.

The classes in AbModeller can be divided up logically into the following broad functional components:

- MFC core classes: The "boilerplate" classes that form the backbone of the application
- The dialog classes that support the input of model parameters and model run parameters
- The view classes that form the main output areas of the application – where modelling data is shown as a modelling run proceeds and where the results of past runs are graphed.
- The database classes that implement the interface to the MS-Access databases

Core MFC Classes

CAbModellerApp coordinates the creation of the CAbModellerDoc "document" class and the various windows that show views of the data in the document.

CAbModellerDoc is the workhorse class, in the tradition of MFC-style applications. This class represents both the application's data and its state. It performs the following key functions:

- It manages the database connections and table (or 'recordset") objects that interface to the MS-Access databases which provide the input data and hold the modelling run results.
- It implements handlers for user input from the application's menu.

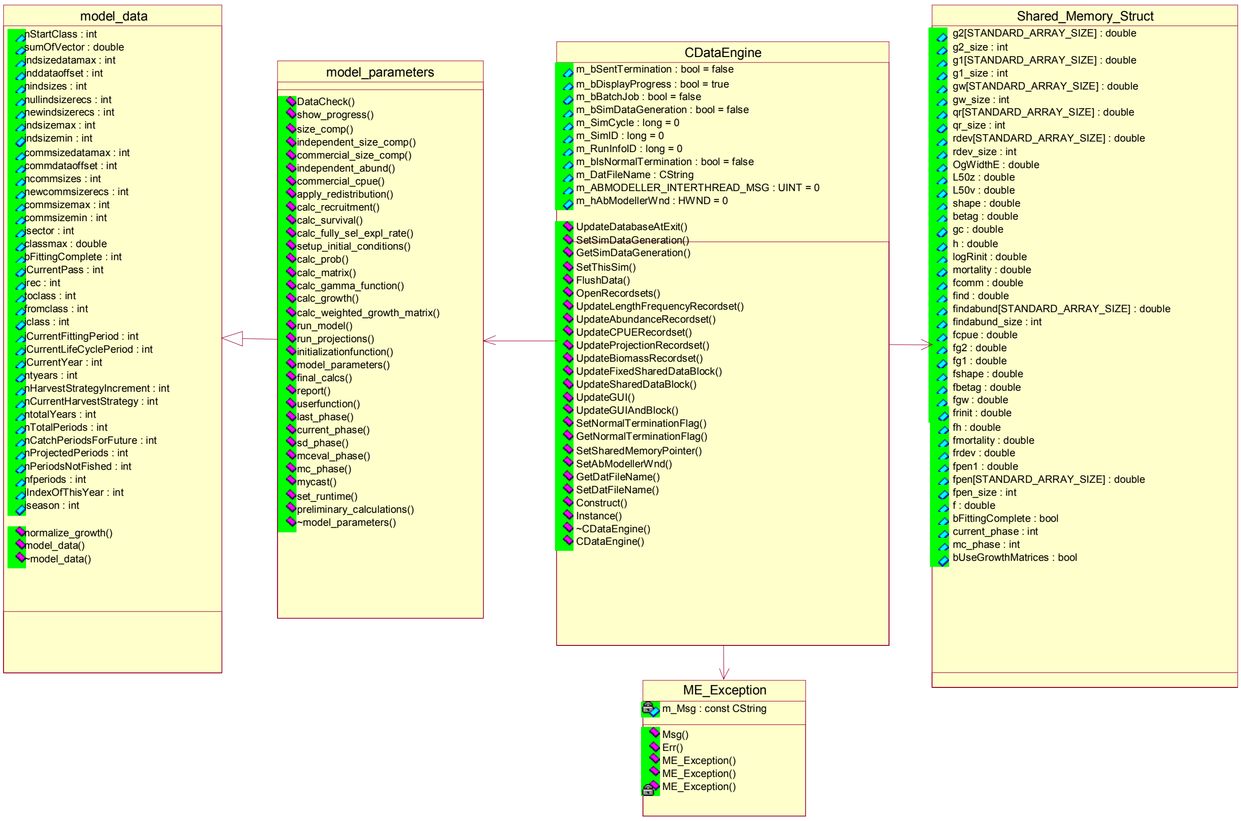


Figure 2 ModelEngine classes

- It manages the state of the application – enabling or disabling menu items as required indicating when a modelling run can be started and when a run is currently taking place. AbModeller's application state is determined by past user input and by messages passed back from any running ModelEngine executable.
- It opens the nominated modelling input database and also gathers user input for the parameters which control each modelling run. This data is written to a text file (*.DAT) on disk which the ModelEngine component consumes as input. The parameter settings for each modelling run are also saved in the Results database. Each run is represented by a data set called a "runinfo". This logical grouping of data appears as a table in the results database and as a "runinfo recordset" object within the code. Results databases can either be opened and updated with new modelling runs, or can be created new. Database errors returned by the MS-Access engine must also be handled.
- It manages some of the child windows that form the output area of the application. Because of the peculiarities of the MFC architecture, child windows that float within the main application are managed by the "document" class and child windows that are dockable are managed by the main application window ("CMainFrame") class. Since the dockable windows also show views of the data, this architecture does not reflect how the application really works. For this reason various functions exist to ensure that all child windows have references back to the document object.
- It launches instances of the various types of ModelEngine runs which include; initial fits to find the maximum likelihood solution, MCMC runs or MCMC runs that start from alternative points in the parameter space. Batch runs of combinations of the above can also be initiated.

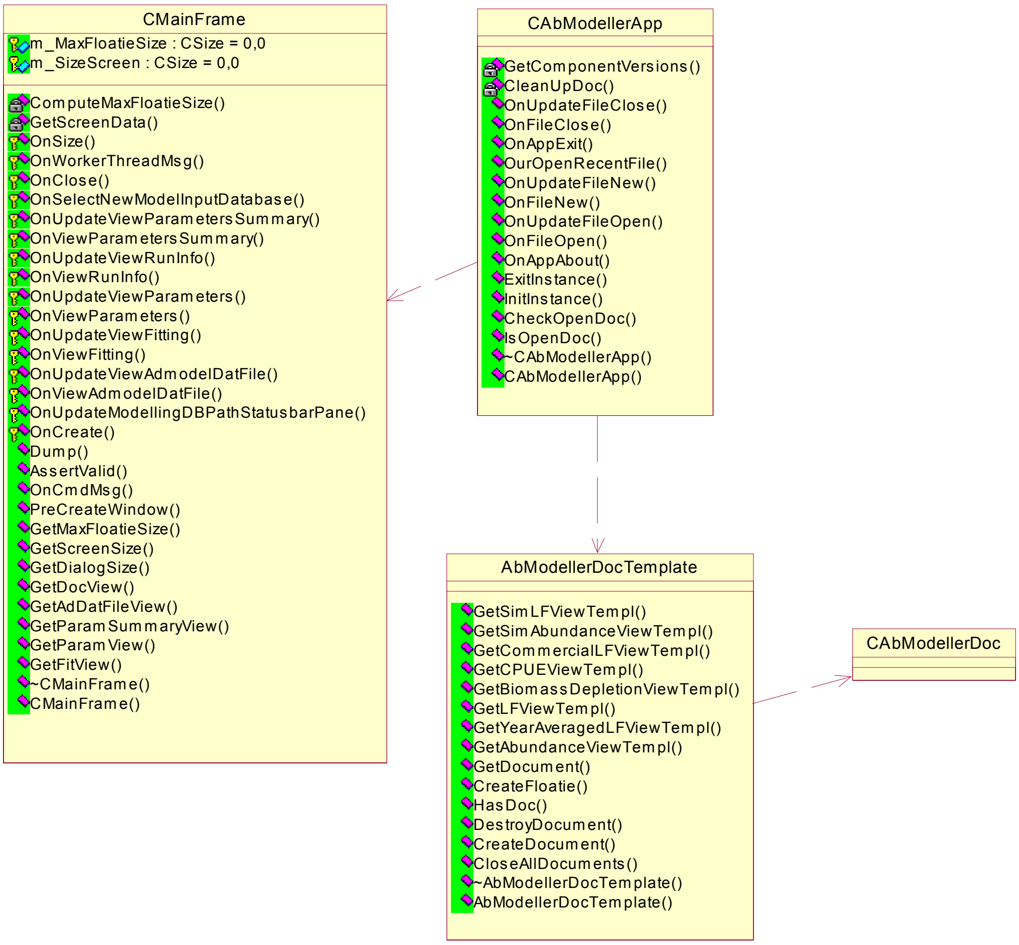


Figure 3. Core MFC Classes

CAbModellerDoc maintains a state machine for controlling the application state. The state machine is represented in the following diagram. The state is initialised in any of the menu handlers and changes to the next state when the ModelEngine terminates and alls back to AbModeller through the OnWorkerThread method.

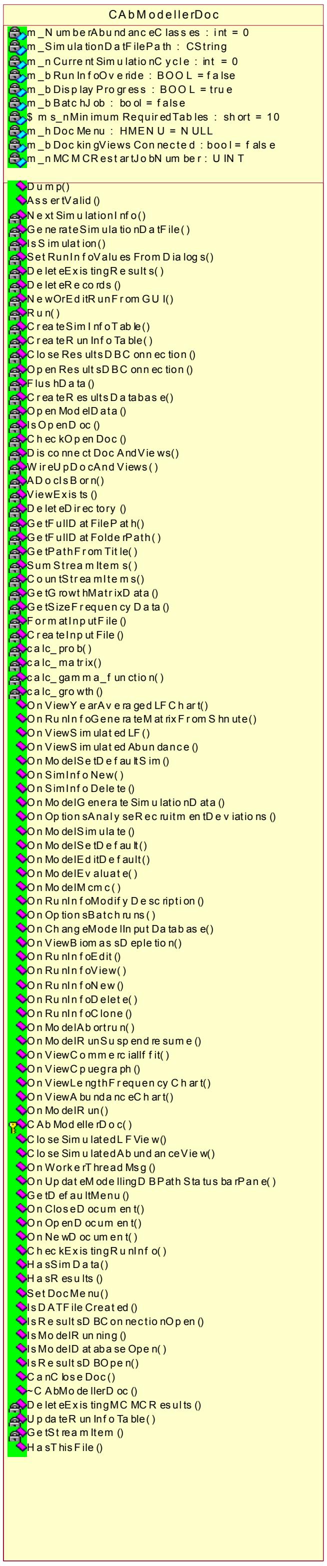


Figure 4.CAbModellerDoc

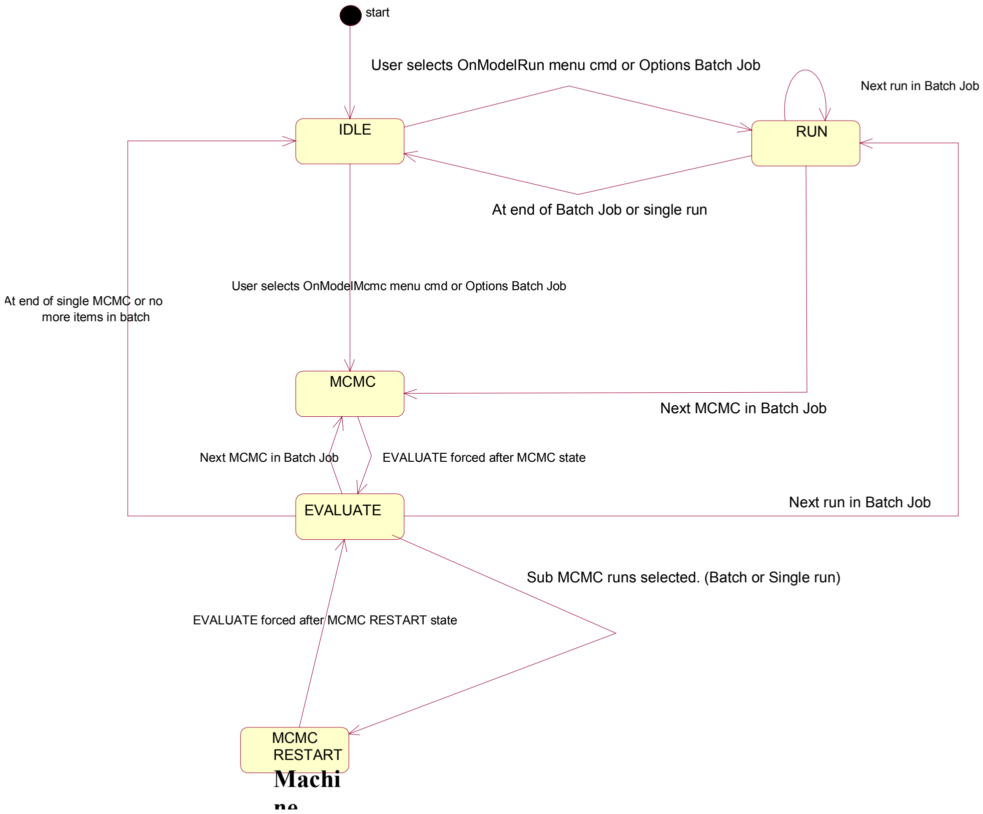

start
User selects OnModelRun menu cmd or Options Batch Job
Next run in Batch Job
IDLE
RUN
At end of Batch Job or single run
User selects OnModelMcmc menu cmd or Options Batch Job
At end of single MCMC or no more items in batch
MCMC
Next MCMC in Batch Job
Next MCMC in Batch Job
EVALUATE forced after MCMC state
EVALUATE
Next run in Batch Job
Sub MCMC runs selected. (Batch or Single run)
EVALUATE forced after MCMC RESTART state
MCMC RESTART
Machi ne

Figure 5: State Machine

Dialog Classes

A set of classes exist which perform the same basic function: extracting "runinfo" run parameters from the database and filling dialog boxes with that data. The data may be presented as a read-only view or presented for user modification. If the data can be modified then the database must be updated when the dialog is closed. MFC provides a consistent method for the transfer of data between user interface widgets in a dialog box and variables that hold the data; this "DataExchange" mechanism is mostly auto-generated when the dialog box is designed.

However some of these classes do need to manipulate the data during its transfer between database and dialog. Mostly this manipulation consists of simple string manipulation.

Some of these classes are called "page" because they are presented to the user as pages of a tabbed dialog. This is a standard MFC framework idiom where the tabbed dialog is called a "property sheet" and the tab contents are called "property pages"

View Classes

AbModeller presents two basic types of views:

- A set of dockable windows that show text-based information such as the modelling data as it is generated
- A set of "floating" child windows that contain one or more graphs showing data from past modelling and model runs.

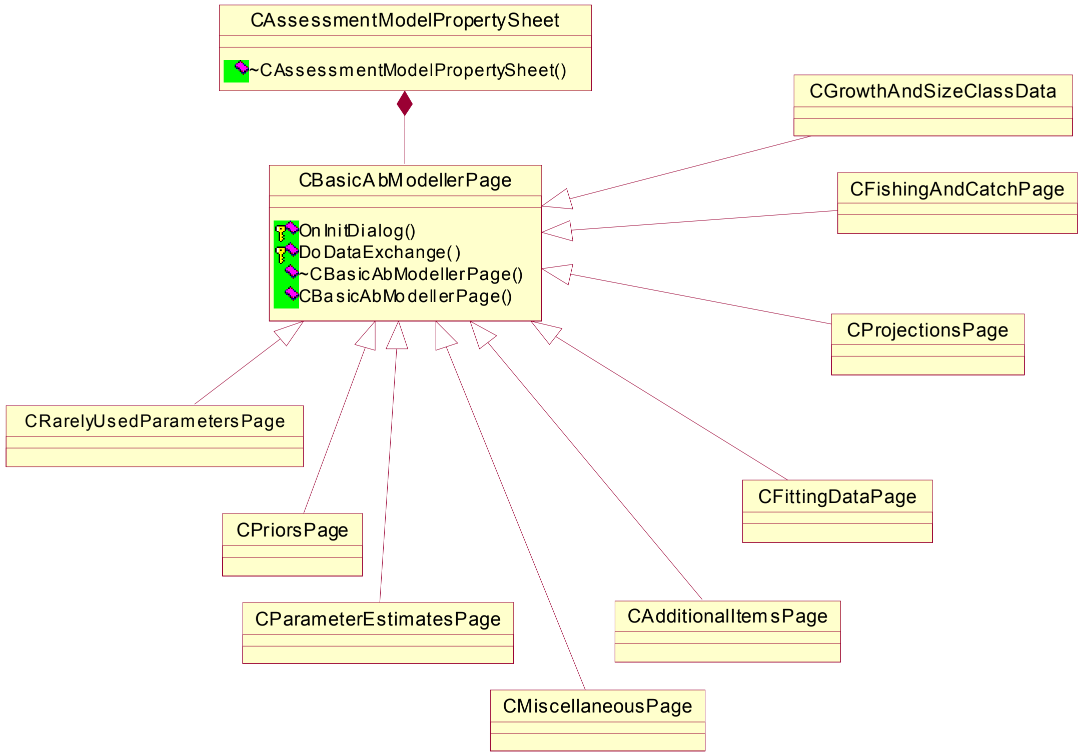


Figure 6: Modelling input data display classes

Dockable Windows
A set of open-source classes are used to implement dockable windows within the AbModeller main application window. A single instance of each child window is contained within the CMainFrame class. Each instance is created when the application starts up and is shown or hidden as required.

The CRunInfoView represents a view into the currently open results database. The database contains a record of all previously performed modelling or model runs. This view shows the main descriptive data for each run and allows an individual run to be selected for further inspection or processing. This view window is, by default, docked to the left hand side of the main application window.

The CFittingView and CParametersView represent views into the data stream generated by a modelling run. Data is passed back to AbModeller from ModelEngine via a block of shared memory, as described above, and the numbers are inserted into columns of a list box control on each view window.

CParametersSummaryView and CAdDatFileView show static views of the data that formed the input to a modelling run.

Graph Views
A set of child windows (based on the MFC class CView) are used to graph data from completed modelling runs. A Microsoft ActiveX control called MSChart provides the graphing functionality. One or more instances of this chart control are hosted in each view, depending on whether one or multiple charts are required to represent the data.

The Microsoft Visual C++ environment was used to automatically generate a large set of classes that represent the ActiveX interfaces exposed by the chart control. Instances of these classes are used to control the look of the charts.

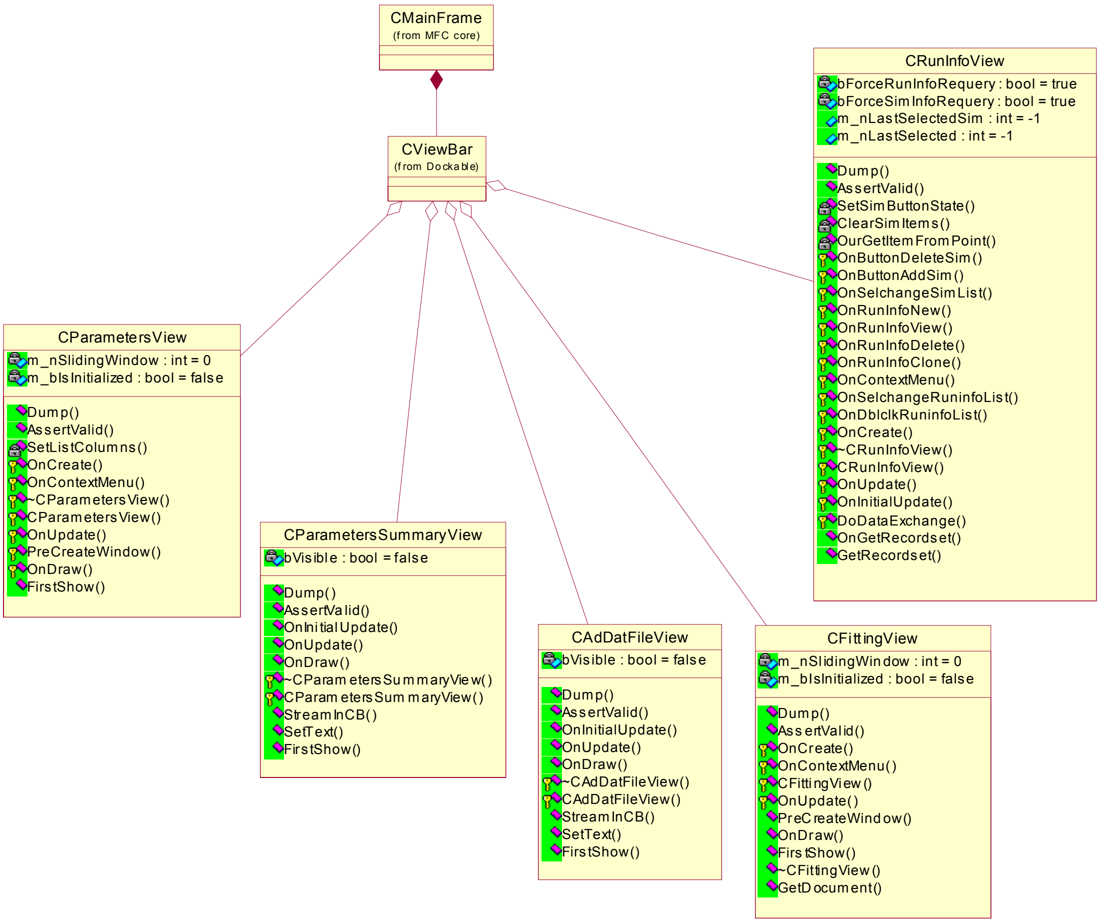


Figure 7. Dockable window classes

Each of the base classes -- CSingleChartView and CMultiChartView – is responsible for controlling the size and position of the chart control(s) on its view window. Their subclasses implement the functionality to extract the data to be charted from the database and to put that data in a format that the chart control can consume.

The subclasses are also responsible for determining axis names, X-axis labels and for ensuring that X-Y scaling factors on each chart are appropriate for the ranges of the data being graphed. In the case of views with multiple chart controls, the number of charts must be determined and a suitable label for each chart must be generated.

These data extraction and manipulation tasks are generally performed using SQL queries of the database.

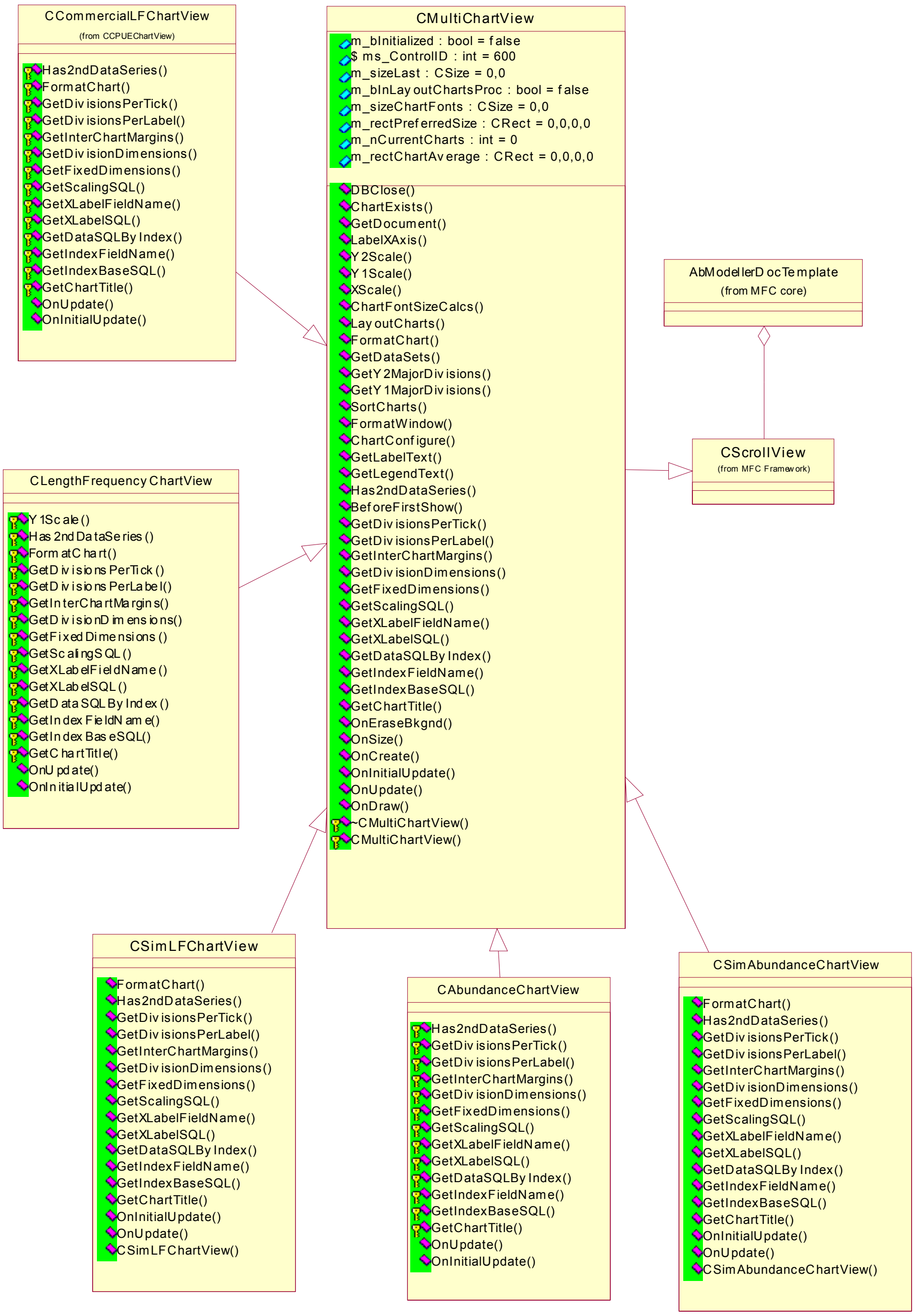


Figure 8 Chart Classes

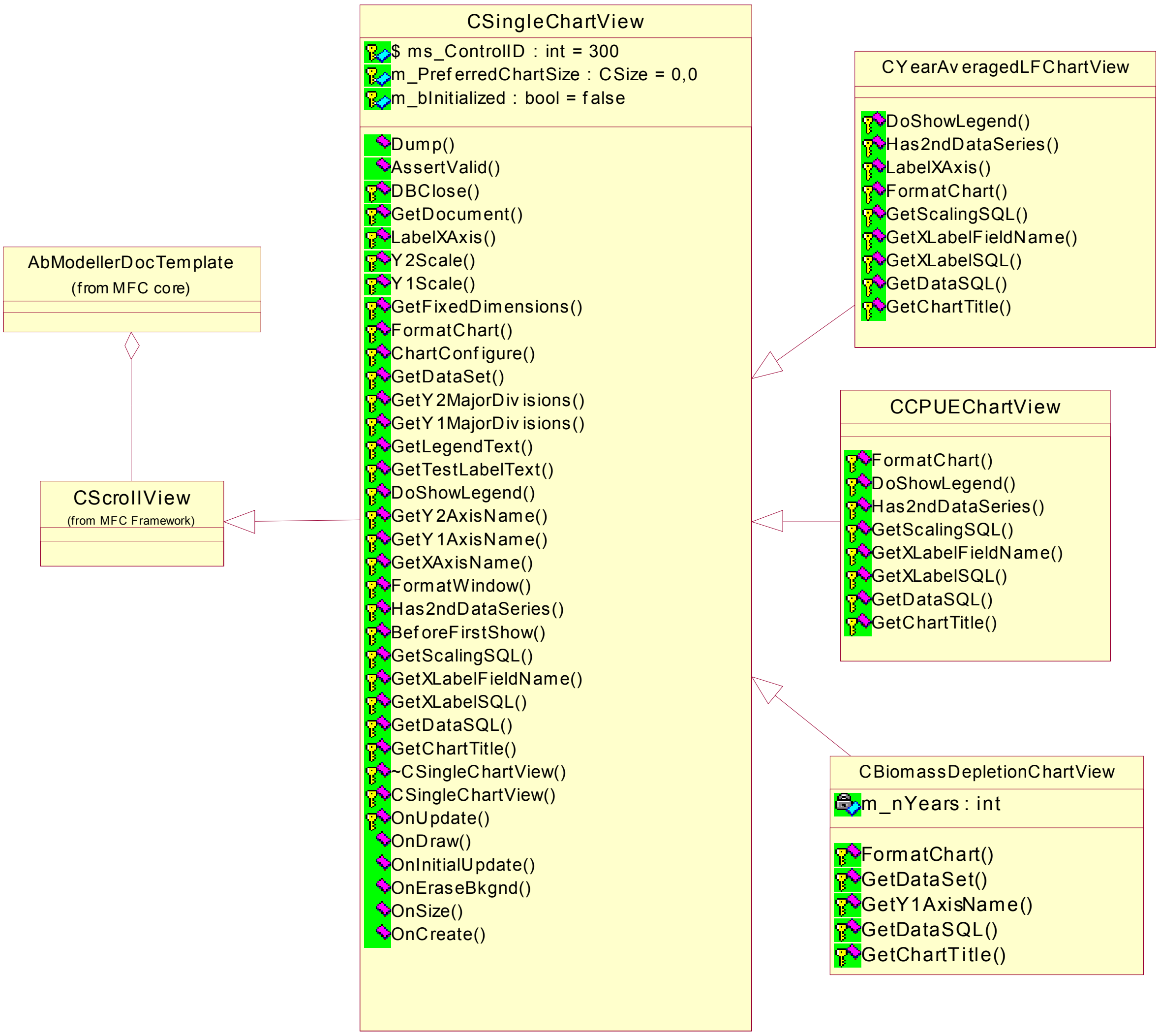


Figure 9 Chart Classes

Database Classes

AbModeller uses a mixture of two of Microsoft's Data Access Technologies – "Data Access Objects" (DAO) and "ActiveX Data Objects" (ADO). The two technologies are required because they each do slightly different things and they are not exact replacements for each other. Microsoft's chart control used in the graphing view classes consumes data from ADO classes. On the other hand DAO is the traditional solution to interfacing with Access databases and is much better supported by the Microsoft Visual C++ tools used to develop AbModeller.

Classes representing the input "modelling" database, the output "results" database and the various tables within those databases are used to read and write to the databases. Many of these classes are automatically generated by the Microsoft Visual C++ environment and are simple wrappers for the underlying table structures in the databases.

Results Analysis Database

Description

This Access database uses the outputs generated from AbModeller and stored in the Results database. It enables the statistical analysis of this data and the production of management oriented reports. ResultsAnalysys.mdb (formerly FrontEnd.mdb) comes in two versions, ResultsAnalysys2000.mdb for Access 2000 and ResultsAnalysys97.mdb for Access 97.

Tables and Modules

The available output data in the Results database is stored in the following tables:

ProjectedBiomass	ProjectedLegalBiomass	Biomass
LegalBiomass	BiomassSummary	Parameters
General	Likelihoods	Rdev
ProjectedBiomassMCMC	ProjectedLegalBiomassMCMC	BiomassMCMC
LegalBiomassMCMC	BiomassSummaryMCMC	ParametersMCMC
GeneralMCMC	LikelihoodsMCMC RdevMCMC	RunInfo

The analysed data in the ResultsAnalysis database is stored in the following tables:

BiomassAnalysis	BiomassDepletionAnalysis	BiomassDepletionMCMC
GeneralParametersAnalysis	LikelihoodsAnalysis	ParametersAnalysis
RdevAnalysis		

The modules that statistically analyse the outputs are:

AnalysisModule

The reports are run from:

Modelling Report Card	Modelling Report Card (MCMC)

Finally DecimalPlaces table manages the number of decimal places displayed in any of the output tables which span more than one modelling field (eg. LikelihoodsAnalysis which holds all the components of the likelihood function). The user manual describes the Analysis process, including the linking of the two databases and the ReportAnalysis reporting capabilities.

Modifications

The results analysis database is simple enough to modify given some level of expertise in MS Access. For example, should the user require additional percentiles for any of the analysis tables, lets say 25% percentile for B_B0, the Access code change is as follows.

Add in the percentile or statistical item of interest into the table as a new field using Access table design.

then

Insert the code in the Access module that analyses the Biomass Summary table.

- rstData.Move Int(rstData.RecordCount * 0.25)
- rstAnalysis.Q25 = rstData.B_B0

Any new items can be easily incorporated into the appropriate reports. Alternatively, users may elect to cull certain items or remove sub reports entirely to tailor a report to a particular audience.