Getting Started with CASAL2

C. Marsh et al.

Citation: C. Marsh et al. (2019). Getting started with CASAL2. National Institute of Water &
Atmospheric Research Ltd.. 23 p.

Contents

1. Introduction

This document is an introductory help guide for CASAL2, a generalised age-structured population dynamics modelling package. CASAL2 is run from the command prompt/terminal where it reads in text files (model configuration files) which define the model. CASAL2 then prints output to the screen or a file.

This document is for users who are new to CASAL2. CASAL2 is primarily used for assessing marine fish populations, and can be used to model the population dynamics of other types of animals. CASAL2's predecessor CASAL is the primary stock assessment modelling tool used in assessing New Zealand's Tier 1 stocks, and is also the standard modelling tool used by the Convention for the Conservation of Antarctic Marine Living Resources (CCAMLR) for modelling Antarctic toothfish.

CASAL2 is a very general, highly flexible modelling framework and has no default settings. It has several run modes, settings, and user-defined population dynamics options that can be set depending on population life history characteristics and the available data.

CASAL2 is open source, and is covered under the GNU GPL 2.0 licence. See the terms and conditions in the CASAL2 Technical User Manual (Rasmussen et al., 2016), or type `casal2 -l` into the command prompt.

There is also additional information that may be useful when getting familiar with CASAL2. CASAL2 has a comprehensive User Manual (Rasmussen et al., 2016) which has further details on model components and functionality. CASAL2 also has a Contributors' Guide for users who would like to add functionality. The modular structure of the code base can make adding new processes, observation types, and likelihood types tractable.

If you have any questions, please contact the CASAL2 development team at casal2@niwa.co.nz.

The remaining content of this chapter describes operating system requirements and details about how to run, cite, and get licensing and contact info for CASAL2. If you are new to population modelling then section 2 describes the types of data needed to run a CASAL2 model. Section 2 is a good starting point to find out if you have the data for developing a model. CASAL2 can also be used as an operating model to generate simulated data.

The remaining content of this document explains how to run CASAL2, the syntax of the input configuration files that CASAL2 and a description of an example model.

1.1. Version

The results for CASAL2 models can differ between versions as issues are fixed or new features are added. The CASAL2 version number includes a date/time stamp (`yyyy-mm-dd`), which is the revision control system UTC date for the most recent modification of the source code. The User Manual will be updated for each released version of CASAL2.

1.2. Citing the CASAL2 Getting Started Guide

A reference for this document is:

C. Marsh et al. (2019). Getting started with CASAL2. National Institute of Water & Atmospheric Research Ltd.. 23 p.

1.3. Software license

1.4. System requirements

CASAL2 is available for most IBM-compatible machines running 64-bit GNU/Linux and Microsoft Windows operating systems.

Several of CASAL2's functions are highly computer intensive and a fast processor with a lot of RAM is recommended. Depending on the model specifications, some of CASAL2's functions can take a considerable amount of time (minutes to hours), and in some cases can take several days to complete a MCMC chain.

The program itself requires only a few megabytes of disk space, although output files can take up larger amounts of disk space, and some processes can use significant amounts of memory.

1.5. Necessary files

For both 64-bit Linux and Microsoft Windows, only the executable file `casal2` or `casal2.exe`, respectively, is required to run CASAL2with non-automatic differentiation minimisers. To use the automatic differentiation minimisers, the `.dll` (for Windows) or the `.so` (for Linux) must be in the same folder as the executable CASAL2 file or in your system path. There is not a version of CASAL2 for 32-bit operating systems.

CASAL2 has an **R** library for post-processing of model output for use with **R** (R Core Team, 2014). See the CASAL2User Manual Chapter 17 (Post-Processing) for more detail on the **R** package.

1.6. Getting help

CASAL2 is distributed as unsupported software. Please notify the CASAL2 development team of any bug reports, feature requests, or questions by opening an issue on the CASAL2 GitHub repository (`https://github.com/NIWAFisheriesModelling/CASAL2/issues`). See the CASAL2 User manual (Rasmussen et al., 2016) for the recommended template for reporting issues.

2. Data Requirements

An age-based model refers to how CASAL2 keeps track of the population, which is by keeping track of the numbers-at-age for each category in the population.

The information that is required to set up and run an age-based model in CASAL2 can be minimal, and can be expanded to include other functionality.

1. Time series of catch (currently CASAL2 assumes this is known without error)
2. Time series of relative/absolute abundance/biomass
3. Age/length composition data (if you want to estimate selectivity ogives or year class strengths)
4. Information about recruitment and the stock-recruit process
5. Biological information (e.g., growth, maturity, natural mortality, life cycle)

The minimum amount of information needed to run a CASAL2 model is recruitment and biological information. Such a model would not include any anthropogenic exploitation processes and so can be thought of as a equilibrium or steady-state model.

The biological information that is required depends on whether weight-based calculations will be performed. For an age-based model parameters for these processes need to be specified:

1. length-at-age `@age_length`
2. weight-at-length `@length_weight`
3. natural mortality-at-age `@selectivity`

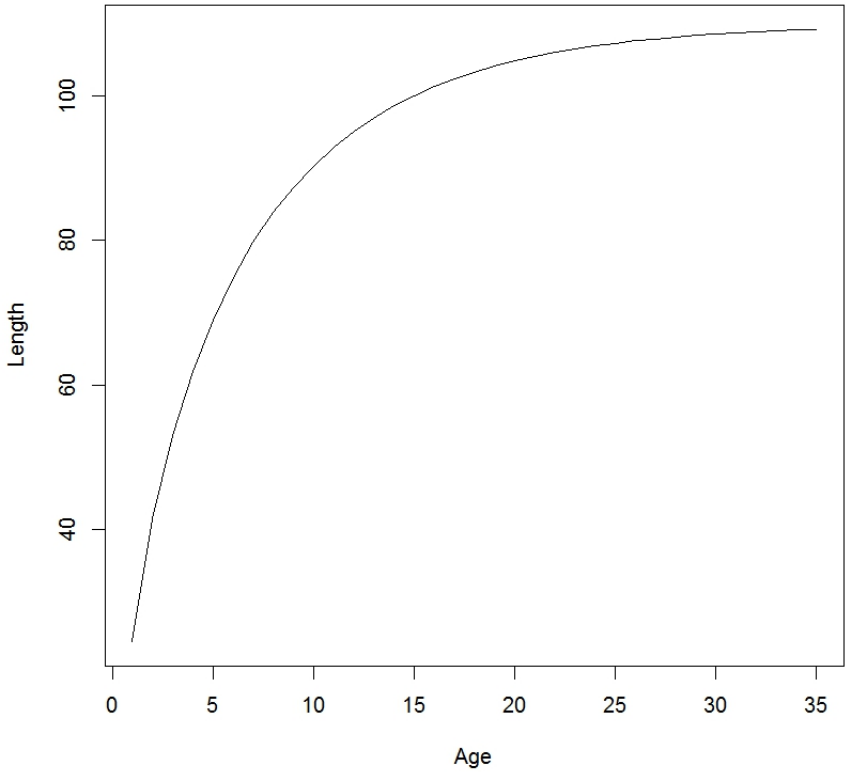

Figure 2.1: An example of a Von Bertalanfy growth curve.

3. Where to get CASAL2

See http://www.niwa.co.nz/ for information about CASAL2. The CASAL2 source code is available on GitHub at https://github.com/NIWAFisheriesModelling/CASAL2.

The CASAL2 bundle, which includes the CASAL2 executable, user manuals, example models, the **R** library, and other information, can be downloaded at

- GNU/Linux, ftp://ftp.niwa.co.nz/Casal2/linux/Casal2.tar.gz
- Microsoft Windows, ftp://ftp.niwa.co.nz/Casal2/windows/Casal2.zip

For both 64-bit Linux and Microsoft Windows, only the executable file casal2 or casal2.exe, respectively, is required to run CASAL2 with non-automatic differentiation minimisers. To use the automatic differentiation minimisers, the .so or .dll file must be in the same folder as the executable CASAL2 file.

On Linux: if the command casal2 -h does not run, then copy the file casal2_release.so to /usr/local/lib/ (or to another subdirectory in your PATH, e.g., /home/[username]/bin).

4. How to run CASAL2

CASAL2 is run from a console windows on Microsoft Windows, or in a terminal window on Linux. CASAL2 is executed by typing `casal2 -[flag]`, where `flag` specifies the run type for CASAL2. When CASAL2 is run with a specific flag, CASAL2 reads in text files. These text files define the model structure and output.

For help on the flags available and their descriptions, type `casal2 --help` or `casal2 -h`, which will print a list of flags and their usage.

CASAL2 can be run in multiple modes. The modes and their corresponding flags include

- deterministic run: `-r`,
- parameter estimation: `-e`,
- parameter profiling: `-p`,
- MCMC runs: `-m`, and
- projections: `-f`.

There are two ways of printing output. The default configuration is to print all output to the screen, and the second option is to direct all output to a file. The second option is usually preferred for post-processing output, e.g., creating plots.

This invocation of CASAL2 reads in a text file that defines the model configuration (`My_model_config.csl2`) and estimates the parameters in that model, then prints the output to the file `output.txt`:

```
casal2 -e -c My_model_config.csl2 > output.txt
```

The flag `-e` specifies the estimation run mode, the flag `-c` specifies the model configuration file to read in, and `>` directs the output to the file. If the flag `-c` and a model configuration file name are not specified, then CASAL2 will (try to) read in the default model configuration file `config.csl2` in the current directory.

The main run modes for CASAL2:

- `casal2 -r` will run the model from `start_year` to `final_year` with the parameters in the configuration file, or will use parameters specified using the `-i` functionalty.
- `casal2 -e` will run the model in `-r` many times trying to solve for the global optimum.
- `casal2 -f 1` is the projection mode and will run the model from `start_year` to `projection_final_year` applying any `@project` functionality. Since much of the `@project` functionality is stochastic, many projections can be performed for a given set of parameters. For example, `casal2 -f 50` will generate 50 projection runs. If there are multiple candidate parameters (perhaps from an MCMC), `casal2 -f 50 -i mcmc_params.out` will generate 50 projections for each set, which will propagate more uncertainty into the projections.
- `casal2 -s 1` is the simulation mode, which will run from `start_year` to `final_year`. It will calculate the expected values for all `@observation` blocks which will then have random error applied to generate simulated data. Report commands can be specified that will generate observation files for re-estimation.
- `casal2 -m` is the MCMC mode and will run a Markov Chain Monte Carlo chain defined by subcommands in the `@mcmc`.

5. Syntax of a CASAL2 file

A general structure of CASAL2 files are that they are split into blocks of subcommands. A block always starts with `@` symbol. Blocks describe different aspects of the model, fundamental blocks to have in the model are `@model`, `@initialisation_phase`, `@categories`, `@time_step`, and `@process`. Within each block there will be subcommands some will be optional and important subcommands will be mandatory. An example of subcommand is shown for the `@model` block,

```
@model
type age ## is the model age or length based?
min_age 1 ## minimum age in model
max_age 17 ## maximum age in model
age_plus true ## is the last age group a plus group?
start_year 1972 ## the first year of the model
final_year 2013  ## the first year of the model
initialisation_phases phase1
## The label for the block @initialisation_phase
time_steps step1 step2
## Labels for the block @time_step
```

The subcommands are all the options that follow `@model`, then there is a space which is where the value for the subcommand goes, i.e. `min_age` specifies the minimum age in the model and we have set that equal to one but could be any integer. This brings up a useful concept to understand. Different subcommands can take different types of parameters, they can be of type int, double, string and vector. For information about which parameter type a subcommand takes, you should read the syntax section of the manual, there is a field labelled type. If you use the wrong type for a subcommand, for example `min_age 1.5`, you will get an error. A line beginning with `#` is a comment and that line is ignored by CASAL2. To comment out multiple lines the user can use the C++ syntax of * and *\, so that everything between these braces will be ignored by CASAL2. It is a useful tool for annotating models.

6. Structure of a model

When setting up a model, it may be helpful to separate the model configuration into different files and multiple sections, e.g.,:

1. population, for the model and population characteristics
2. estimation, for the parameter and penalty characteristics
3. observation, for the fishery and survey data
4. report, for the specifications for the model output

The population file defines the categories that make up the partition, the time steps in the annual cycle and processes that occur to the partition in each time step, along with the parameters that control those processes.

The estimation file defines the estimated parameters and the prior distribution associated with parameters or processes.

The observation file defines the observations and their assumed error structure through the likelihood types which contribute to the objective function.

The report file defines the output at the end of model runs, e.g., the objective scores, residuals, derived values, and the state of the partition at a point in time.

An example `config.csl2` file may have the following commands:

```
!include "population.csl2"
!include "estimation.csl2"
!include "observation.csl2"
!include "report.csl2"
```

The `!include` function looks in the current directory for the specified filename, and reads in all of the files included in the configuration file as one model.

See the Simple example in the `Examples` subdirectory.

7. Components of a model

Components of a model that are important to know before setting up a CASAL2 model are

- how many categories are in the partition,
- what processes occur to which categories in which order,
- where observations fit in to the model, and
- what the assumed state of the partition is before the model years run.

CASAL2 runs in annual cycles, and each year is split up into time steps. Time steps are used to specify when processes such as fishing and spawning seasons occur, as well as observations such as annual surveys.

Identifying the partition is an important part of model development, and will be determined by the information available. Categories in the partition should be included if they have different biological characteristics or life history traits.

Examples why a model may have multiple categories in the partition:

- differences in growth among subgroups, e.g., sex- or area-specific growth
- spatial or temporal differences in exploitation, e.g., a fishery may target characteristics that are associated with some subgroups
- disproportionate recruitment (60% males 40% females)
- tracking maturity
- tracking tagged fish (this process is related to observations)

8. Examples

8.1. Simple Example

In this example is a model for a single stock in a single area which has one fishery associated with it. The partition is made up of a single category, with sex and maturity not defined in the partition. Processes and observations in a given year are defined to occur in the following order:

1. Recruitment
2. Fishing mortality with natural mortality
3. A survey takes place in the spawning season
4. More natural mortality
5. At the end of the year all of the fish age

This model has the structure:

```
@model
start_year 1975 # Start year
final_year 2012 # End year
min_age 1 # min age of all categories
max_age 30 # max age of all categories
age_plus true # is the last age a plus group
base_weight_units tonnes
initialisation_phases Equilibrium_state
time_steps Sep_Feb Mar_May Jun_Aug # labels for the time steps

@categories
format stock # category type
names HAL4  # category label(s)
age_lengths HAK4_AL # labels of age-length relationship for each category

@time_step Sep_Feb
processes Recruitment Instantaneous_Mortality

@time_step Mar_May
processes Instantaneous_Mortality

@time_step Jun_Aug
processes  Ageing Instantaneous_Mortality
```

The `@categories` command defines the type, label, and age-length relationship of categories that make up the partition. A category is a group of individuals that have the same attributes. Examples of such attributes include life history and growth paths. Characteristics in a partition of the population that have different attributes can be sex, maturity, area, stock, and tagging information.

An example of the `@categories` block for a simple sex-specific model with male and female in the partition:

```
@categories
format sex        # category type
names male female # category labels
age_lengths vb_male vb_female # labels for growth categories
```

The `@time_step` command describes which processes take place in each time step and in what order. Continuing on with the `@model` block example, where three time steps are defined in the annual cycle (`time_steps Sep_Feb Mar_May Jun_Aug`), and within each time step processes are specified. Each process must be defined in a `@process` block:

```
@process Recruitment
type recruitment_beverton_holt
categories HAK4
proportions 0.5 0.25 0.25
b0 44000
ycs_years 1974:2011
ycs_values 1*38
standardise_ycs_years 1975:2009
steepness 0.9
ssb SSB
age 1

@process Ageing
type ageing
categories HAK4


@process Instantaneous_Mortality
type mortality_instantaneous
m 0.2
time_step_ratio 0.42 0.25 0.33
selectivities One
categories HAK4
table catches
year FishingWest FishingEest
1975 80 111
1976 152 336
1977 74 1214
1978 28 6
1979 103 506
1980 481 269
1981 914 83
1982 393 203
1983 154 148
1984 224 120
1985 232 312
1986 282 80
1987 387 122
1988 385 189
1989 386 418
1990 309 689
end_table

table method
method    category  selectivity  u_max  time_step  penalty
FishingWest  HAK4    FSel   0.7  Sep_Feb  Catchmustbetaken
FishingEest  HAK4    FSel   0.7  Sep_Feb  Catchmustbetaken
end_table
```

The sections above define all of the processes that occur to the partitions. In the `Mortality` section natural mortality is defined with selectivity ogive `One`, and fishing mortality is associated with

selectivity ogive `FSel`:

```
@selectivity One
type constant
c 1

@selectivity FSel
type double_normal
mu 3.82
sigma_l 1.63
sigma_r 17
```

Since an age-length relationship is specified in the `@categories` block then the `@age_length` block needs to be defined. This command block is used to convert age to length which is then used to convert length to weight in an age-based model:

```
@age_length HAK4_AL # label from the @categories block
type von_bertalanffy
length_weight HAK4_LW # label for the @length_weight block
k 0.164
t0 -2.16
Linf 100.8
cv_first 0.10
cv_last  0.10

@length_weight HAK4_LW
type basic
units tonnes
a 4.79e-09
b 2.89
```

An important block to complete the population definition is the `@initialisation_phase` block. This block specifies how the partition is initialised. This block describes the state of the partition before `start_year` of the model, which is usually this is an equilibrium state:

```
@initialisation_phase phase1
type iterative ## Type of initialisation method; see manual for more information
years 100 ## How many years to run for
```

This block is an example of an iterative initialisation type. This block specifies that the annual cycle will iterate through 100 years, which may or may not result in the partition reaching an equilibrium state. **N.B.** when using this initialisation method check that the partition has reached an acceptable equilibrium state.

The next section to define is the observation section. The survey occurs in the second time step, and is for relative abundance:

```
@observation Survey ## label of observation
type biomass  ## type of observation
time_step Mar_May ## which time step the observation occurs
time_step_proportion 0.5 ## the observation occurs half way through the time step
categories HAK4
catchability q ## The label for @catchability block
```

```
selectivities One
likelihood lognormal ## likelihood to use for the objective function
process_error 0.10
years  1992  1993  1994  1995
obs  2950 3353 3303 2457
error_value  0.41 0.52 0.91 0.61

@catchability q ## label from @observation
type free
q 0.001 ## the initial value
```

A set of model configuration files is available in the `Examples/Simple` subdirectory.

9. Analyses of output

An important note about CASAL2 is that there are no default reports. That is, if a `@report` block is not specified, then CASAL2 will not report any output. So before running a CASAL2 model check that there is at least one `@report` block.

CASAL2 has an **R** library **casal2** which imports CASAL2 output files into **R** as a list. This library can be found in the directory where CASAL2 is installed.

There is also another library that helps pull out compress useful information such SSB and objective scores for model components. [Q: what is this other library?]

```
library(casal2)
output <- extract.mpd(file = "Output_file.txt", path = "Directory_of_file")
```

10. References

B Bull, R I C C Francis, A. Dunn, A McKenzie, D J Gilbert, M H Smith, R Bian, and D Fu. CASAL C++ Algorithmic Stock Assessment Laboratory): CASAL user manual v2.30-2012/03/21. Technical Report 135, National Institute of Water and Atmospheric Research Ltd (NIWA), 2012.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014. URL `http://www.R-project.org/`.

S. Rasmussen, I. Doonan, A. Dunn, C. Marsh, K. Large, and S. Mormede. Casal2 user manual. Technical Report 139, National Institute of Water and Atmospheric Research Ltd (NIWA), 2016.

11. Acknowledgements

We thank the early beta testers of CASAL2 and users of CASAL (Bull et al. 2012) for files and test cases, and their input into this document.

The development of CASAL2 was funded by the New Zealand Ministry for Primary Industries and the National Institute of Water & Atmospheric Research Ltd. (NIWA) under NIWAs Fisheries Centre Research Programme 1.

12. Index