



# User manual for ECA

**Version 4.0**

**Note no.**

**SAMBA/53/16**

**Authors**

**David Hirst, Sondre Aanes and Hanne Rognebakke.**

**Date**

**6. des. 2016**

### **Authors**

David Hirst, Sondre Aanes and Hanne Rognebakke.

### **Norsk Regnesentral**

Norsk Regnesentral (Norwegian Computing Center, NR) is a private, independent, non-profit foundation established in 1952. NR carries out contract research and development projects in information and communication technology and applied statistical-mathematical modelling. The clients include a broad range of industrial, commercial and public service organisations in the national as well as the international market. Our scientific and technical capabilities are further developed in co-operation with The Research Council of Norway and key customers. The results of our projects may take the form of reports, software, prototypes, and short courses. A proof of the confidence and appreciation our clients have in us is given by the fact that most of our new contracts are signed with previous customers.

**Title** **User manual for ECA**

**Authors** **David Hirst, Sondre Aanes and Hanne Rognebakke.**

Date 6. des. 2016

Year 2016

Publication number SAMBA/53/16

**Abstract**

A Bayesian hierarchical model has been developed to estimate the catch-at-age of fish, using data on age, length and age-given-length, possibly including stock separation and uncertainty in aging. The model is implemented in C with an R interface. In addition, a module doing data management to be used in the model has been developed. This note describes how to install and run the program. It can be run in Linux.

Keywords ECA

Target group ECA users

Availability open

Project number HI 15-17 220722

Research field Marine

Number of pages 54

© Copyright Norsk Regnesentral



# Table of Content

<b>1</b>	<b>Introduction.....</b>	<b>6</b>
<b>2</b>	<b>Installing ECA.....</b>	<b>6</b>
<b>3</b>	<b>Resource Files .....</b>	<b>8</b>
<b>4</b>	<b>Running the program using the GUI.....</b>	<b>11</b>
<b>5</b>	<b>Running the program using scripts.....</b>	<b>25</b>

# 1 Introduction

The ECA program consists of two main parts:

1) A data module: A data handling procedure that reads the original data, filters it and compile it to a format useable for statistical analysis using R. This also includes various quality checks, debugging of data and some descriptive analysis of data.

2) An analysis part: Implementation of the methods in Hirst et al. 2004, 2005 and 2012; a Bayesian hierarchical model has been developed to estimate the catch-at-age of fish, using data on age, length and age-given-length. The method also includes uncertainty in ageing, and stock separation coastal cod - NEA cod based on otolith type and uncertainty in classification of type. The model is implemented in C with an R interface.

A technical report describing the theory in the analysis part is available in "Catch-at-age-Version 2.0: New features and validation of program" (NR report: SAMBA/57/11) as well as in the papers by Hirst et al.

The ECA program is made as an R library and this note describes how to install and run the program. It can be run in R on Linux. There are two ways of running ECA: Either using scripts or using a graphical user interface. The script version is more flexible both in structuring the program and the setup of models, while if using the GUI a specified structure is needed and poses more restrictions on the options for fitting the model (although not more restrictions than the usual user would need). We recommend setting up ECA to use the GUI as the script version will also work with that setup.

The GUI produces a log file to which function calls generated by the GUI are written. Function calls are then read back to R and evaluated. In this way the user is always able to check the actual function calls used and to re-run parts of the analysis using a script if desired.

Prior to running ECA the original data must be available, as must a set of resource files containing information about what defines a sampling platform etc. This is described in section 3.

## 2 Installing ECA

The program uses two paths, one for the original data (***data.folder***), and one used by ECA (***ECA***). This is so that the original data can be stored centrally on a network, while the datasets generated by ECA can be stored locally. The names of these folders is optional while the name of their subfolders is required fixed. The two paths can be the same if required.

The following subdirectories are in ***data.folder***:

- ***SPD***. This contains the original IMR data in "SPD" format. The files are called fisk<YEAR>.<xxx>. This folder is only relevant if ECA is to be used for converting the

data into the STUV files which it uses. Alternatively, the STUV files may be provided directly as e.g. exported versions of IMR's SAS versions of the data.

- **STUV**. This contain all the S, T, U and V files, in .csv format, called S<YEAR>, T<YEAR>, U<YEAR> and V<YEAR>, respectively, e.g. S2011.
- **Sluttseddel**. The landings weight files are stored here with names <species>-<optional file extension>.csv, e.g. "TORSK-ark.2011.csv" or "TORSK-adj.2011.csv". The format of these files are the same as IMR "-ark" format. See the appendix for an example.

For example, if data.folder is a directory called IMRData, the structure should look like this:

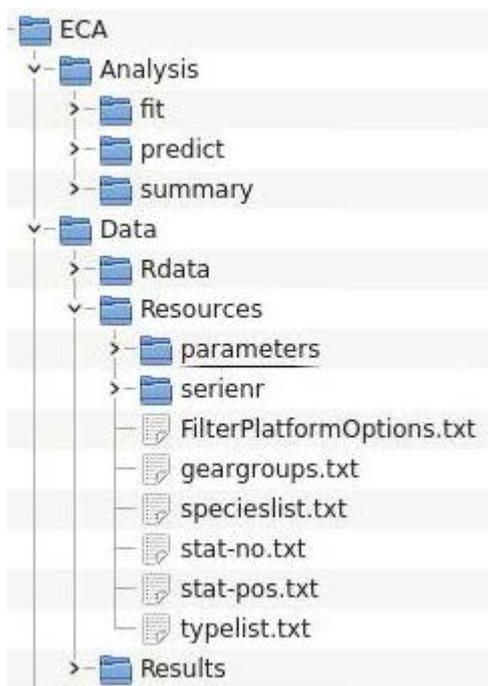


The following subdirectories are in the folder **ECA**. This directory must be created before starting to use ECA.

- **Analysis** which contains the following subdirectories
  - **Fit**. Initially empty. ECA writes results from the model fits to this folder, i.e. posterior distributions of parameters in the model.
  - **Predict**. Initially empty. Predictions from the model are written here, i.e. estimates of catch at age
  - **Summary**. Initially empty. Summary tables and plots are written here.
- **Data** which contains the following subdirectories:
  - **Rdata** which contains the following subdirectories:
    - **DataByPlatform**. Initially empty. ECA read from **data.folder/STUV** and writes R-representations of data by platform and year to this folder with names <samplingplatform><YEAR>.Rdata eg AMIGO2002.Rdata.
    - **Formatteddata**. Initially empty. ECA takes input from files in **DataByPlatform** and prepares data suitable for analysis in R by year, platform and species and saves them here.
    - **Projectdata**. Initially empty. Data prepared for analysis by the methods of Hirst et al. 2004, 2005 and 2011, are created and stored here by ECA. Takes input from files in Formatteddata
  - **Resources** must contain the files "gargroups.txt", "specieslist.txt", "statpos.txt", "typelist.txt" and "FilterPlatformOptions.txt" (See below and in the appendix for format and examples of these files) and include these subdirectories:
    - **parameters**. Must include files with names <SPECIES><optional file extension>.txt. See the appendix for format of these files.
    - **Serienr**. Must include files with names <samplingplatform>.txt. See the appendix for format of these files.

- **Results.** Initially empty. Folder where plots and reports for various exploratory data analysis provided by ECA are written to.

For example, if the ECA folder is a directory called ECA, the structure should look like this:



#### IMPORTANT:

1. Names of sampling platforms in filenames in **ECA/Data/Resources/serienr** and in the file "typelist.txt" must be identical. In addition the names of sampling platforms in "FilterPlatformOptions.txt" must be identical to these names to filter platform specific data correctly.
2. Species names occurring in S, T, U and V files in original data, total catch files, and name of species specific parameter file must be identical, as they are used to link the bits together.

For example if vanlig user is coded as "VANLIGUER" in the original data, the corresponding files must use "VANLIGUER" in its names and not for example "VANLIG UER" as the program not will see the difference and be able to extract the correct data.

## 3 Resource Files

A number of resource files must be present before running the program, and here is a rough description. It is important that the user maintains these files and makes sure they are up-to-

date at all times as they will define how data is read and filtered, and thus used for the analysis. The format in detail is left for the Appendix.

### 3.1. Species specific file

A parameter file is required for each species. This file provides necessary information both for data management as well as for setup of the model fitting. It must be called <species><optional extension>.txt and put in *ECA/Data/Resources/parameters*. The exact format of this file and an example is given in the appendix. The following elements are compulsory:

1. minage and maxage are used to exclude unreasonable data. The model will include ages from minage to maxage and any fish with observed ages outside this range will be omitted. It is usually sensible to set minage to 1. Note that fish with small lengths but no observed age will be included (if length-only data is included) even though they may be below minage.
2. "minlength" and "maxlength" are used to avoid problems caused by length-only observations being outside the range of the aged fish. This can happen for example if the tokt data are used. It is advisable to set minlength to 0.
3. "plusgroup" is only used for presentation purposes - the catch at age will be summed over all ages from plusgroup up.
4. "areagroups", "seasongroups" and "gargroups" are used for two purposes 1) to extract data from the relevant areas defining the species and 2) to define how the covariates are grouped for the analysis. This is not necessarily the same as how they are grouped for presentation of catch at age results. The model allows for different grouping of data for weight-given-length. If this is provided they should be on same form as the groups above, but with names "wglareagroups", "wglseasongroups" and "wglgargroups". If these are not provided grouping of data for weight-given-length is set equal to grouping for length-given-age.
5. Conversion factors for length and weight. These correct a fish measured as a product to whole length and round weight.
6. Parameters defining a von-Bertalanffy growth function determining how to identify outliers defining data to be removed. Should be provided. See the note debugging data for implemented method.
7. Parameters defining a log linear model for a weight-length function determining how to identify outliers to be removed. See the note debugging data for implemented method.
8. The file also includes definition of output groupings, i.e. grouping of areas, gears to summarize results over. The format is the same except that each group is named.

The following are options but must be included if the relevant fitting options are used:

1. The class error matrix. This is needed if a coastal cod analysis including uncertainty in the classification is to be run. There are 4 values, the probabilities of (1) coastal cod of the certain type being classified as coastal cod, (2) coastal cod of the uncertain type

- being classified as coastal cod, (3) skrei of the certain type being classified as skrei, (4) skrei of the uncertain type being classified as skrei.
2. The age error matrix. This is needed if uncertainty in ageing is included in the analysis. It defines the probability of observing each age given the true age. The true ages are the columns, the observed ages are the rows and the number of rows and columns must agree with the definitions of minimum and maximum ages. The element in row *i*, column *j* defines the probability of a fish of true age *j* being observed as having age *i*. The columns must sum to 1 (though the rows need not).
  3. Parameters for fixing the age-length and/or weight-length relationships. If the model is to be fitted without estimating length given age and/or weight given length the parameters must be provided. There are always three parameters for weight-given-length - the intercept, slope and residual precision for the regression of log(weight) on log(length). Length-given-age requires the same three parameters, but can also have an additional three for the Schnute-Richards model. In this case the last 3 columns are the parameters *c*, *theta* and *gamma*. If there is only 1 row, the same values will be used for all samples (ie there is no uncertainty in the parameters). To include some uncertainty in the parameters more rows (eg simulated values from some other run of the program) can be included. The rows will be resampled to get the same number as mcmc samples. Therefore the file must also include the number of samples, even if it is 1.

### 3.2. Non-species specific files:

These must be in the directory *ECA/Data/Resources*. See the appendix for details:

- 1) geargroups.txt. Defines how to group original coded gears on STUV to corresponding gear in total catch statistic
- 2) specieslist.txt. Lists species that will be included in the GUI. Must correspond to species names in STUV data.
- 3) typelist.txt. Sampling platforms that will be included in the GUI. Must correspond to serienr files.
- 4) stat-pos.txt. These define areas so that if the area is missing it can be assigned.
- 5) FilterPlatformOptions.txt. Defines how data will be filtered by various variables and year for the sampling platforms listed in typelist.txt.

### 3.3 SERIENR files

These must be in the directory *ECA/Data/Resources/serienr*. There is one file for each platform, defining the serienr ranges that correspond to the platform for each year. Called for example amigo.txt.

## 4 Running the program using the GUI

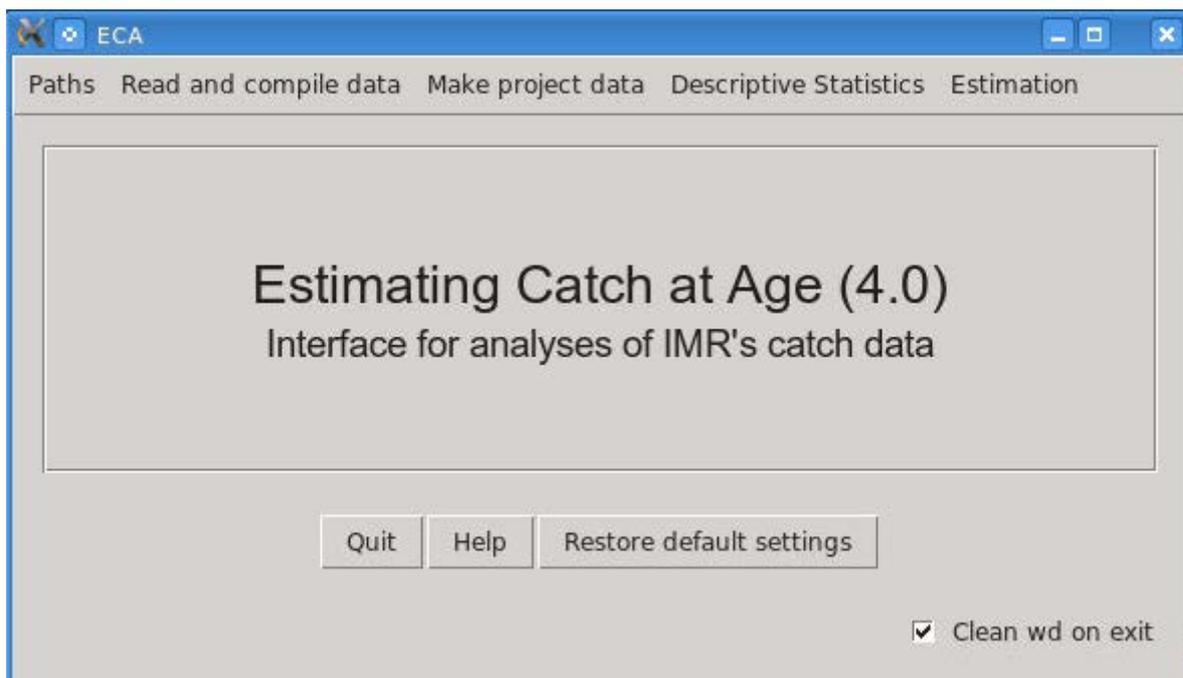
In this manual we will first go through running the program using the GUI. Then an example using scripts is provided. For those who will run the program using scripts we advise first running the gui, and then copy the LogFile as a starting point.

### 4.1. Starting the program

Use the command `library(caa,lib.loc=paste(progdir,"Rlib",sep=""))`, where "progdir" is the directory containing the R library.

Start the program with the command "ECAgui()". Since the program by default does not know the placement of the data.folder and ECA, this can either be set after starting the gui with the command "ECAgui()" and use the browse functionality, or alternatively (and perhaps easier) to supply the function with the correct paths; "ECAgui(orig.data.folder='data.folder',ECA.folder='ECA')". See '?ECAgui' for a documentation of the function and what options there are for manipulating the gui through arguments to the function.

Typing "ECAgui()", with or without arguments to the function, the following menu box will appear:



There are 4 sub-menus which are described below. There are also 3 buttons: Help gives some information that may be helpful. Most subsequent dialogs have a help file. Restore default settings is used to change later options back to default. Quit closes the dialog box. The check box Clean wd on exit removes all variables that are generated when running the gui. If the tick is removed all data generated in the session is kept.

The 4 sub-menus are as follows:

## 4.2. Paths



The default paths are subdirectories of the R working directory: “**Data**” for the input data and “**ECA**” for the ECA directory. They can be changed for the session here or input as arguments to the function ECAgui. All data created when running the program relies on the folder structure described in section 2. When the gui is closed, the paths that were set will be remembered till the next time ECAgui is run within the R session or for future R-sessions if the work space is stored when existing R.

## 4.3. Read and compile data

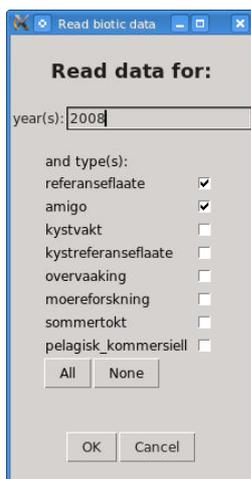
Reading and compiling data is a two- (or optionally three-) step procedure for reading the source data into R and convert it to a format useable for analysis; This has three submenus:

### 4.3.1. Read SPD data



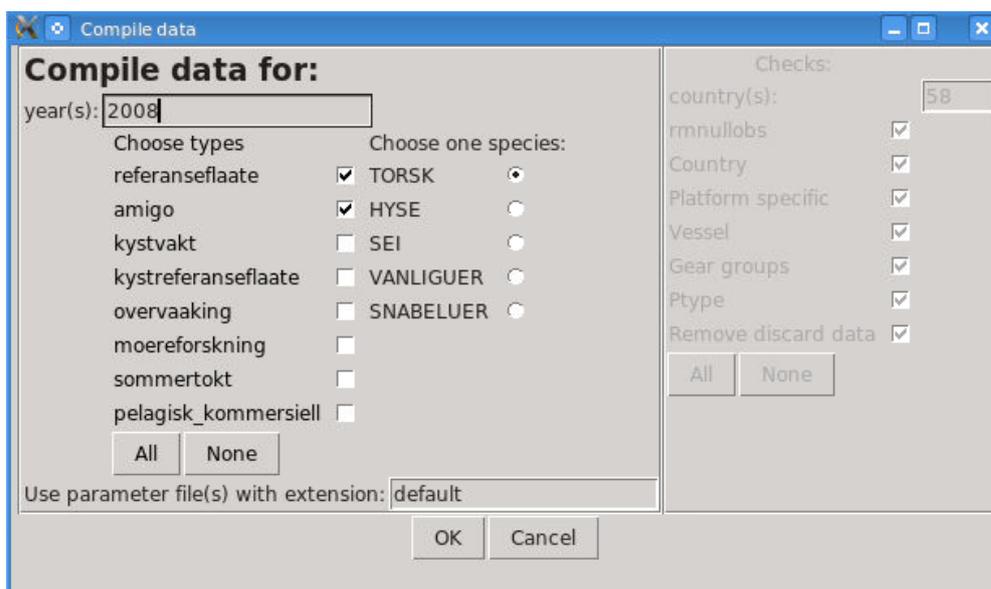
ECA uses the STUV files containing the original data. They must be in the directory **data.folder/STUV**. If they do not already exist and cannot be copied from elsewhere this menu can be used to create them from the SPD files (assumed to be in the directory **data.folder/SPD**, typically named fisk2013.316). Multiple years can be input in the format 2001:2005.

### 4.3.2. Read data



This is used to read the data from the STUV files stored in **data.folder/STUV**. It writes data in R format to **ECA/data/Rdata/DataByPlatform**. It uses the serienr files in **ECA/Resources/serienr** to produce year and type specific files. It is only necessary to read the data once for each species and year, unless the original data has been updated. Multiple years, types and species can be selected. The files will be called <name sampling platform><year>.Rdata, e.g 'amigo2008.Rdata. In the example shown the STUV data is only being read for amigo for the year 2009.

### 4.3.3. Compile data



This takes the R data created in section 4.3.2 and makes it useable for analysis by year, type and species. Only one species can be chosen but data from multiple platforms can be selected. Filtering and debugging of data are performed using the species specific parameter file defined in the bottom box in the left hand panel. The name in this case will be TORSKdefault.txt and it is assumed to be in the **ECA/Data/Resources/parameters** directory. The degree of filtering is determined by the checkboxes in the right hand panel. They are greyed out because in general they

should not be changed, though they can be if required. For a full documentation see the help file for the function "makedata", but the most important ones are:

Country: Norway is 58. Other countries can be included if there is data.

Platform specific: Applies type specific selection criteria for types in 'overvaaking', 'kystvakt', 'amigo', 'pelagisk\_kommersiell', 'leiefartøy' and 'sei\_kommersiell'.

Changing these criteria is not optional and requires changing the program code.

Vessel: Selects data only where the boat can be identified.

Gear groups: Group gears according to the resource file "gargroups.txt". Data with gears not included in this file are removed.

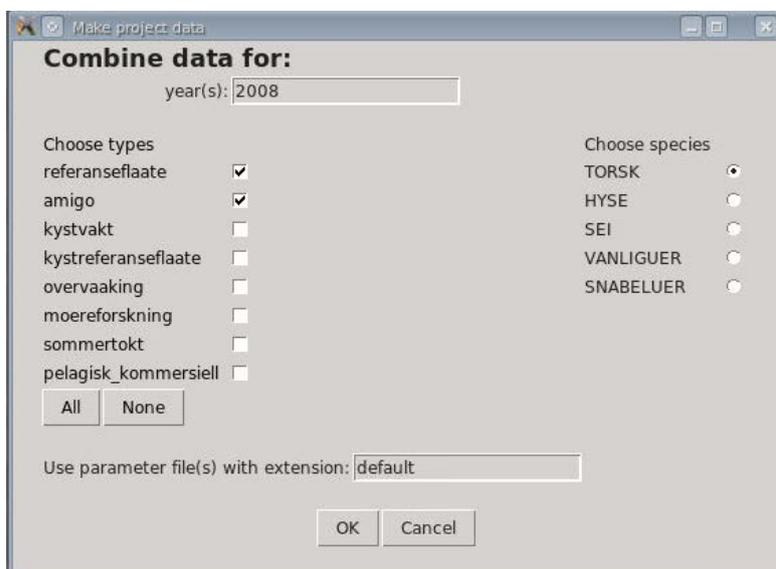
Ptype: Checks whether "PTYPE" is present, and if not data are removed.

## 4.4 Make project data

This has two sub menus

### 4.4.1. Make data

This combines the data from different platforms for a given species and one or more years. In the first menu the species, platforms and years are chosen, and the name of the parameter file is defined. By default this is SPECIESext.txt where *ext* is given in the menu, eg TORSKdefault.txt.



After clicking OK the next menu gives various options:

The project name is by default *SPECIESyear* but it can be changed to something more memorable, eg *SPECIESyearAmigo* if only Amigo data have been included.

The total catch (sluttseddel) data area assumed to be called *SPECIESext.year.csv* where *ext* is given in the menu with default “-adj”, eg *TORSK-adj.2009.csv*.

There are three options for the definition of the area. “Use recorded area only” only uses data where the area is recorded, and assumes it is correct. “assign missing areas using stat.pos” will assign an area to all samples where the area is missing but latitude and longitude are present. It uses the information in the stat.pos file. It is possible that the definitions in this file will not be the same as those used to assign areas where they are present. “assign all areas using stat.pos” reassigns areas to all samples with latitude and longitude using the stat.pos file. This may change some of the assigned areas. If the names of the stat.pos file has been given in the parameter file, it will appear in the “area file” box. Otherwise it must be given here if the second or third “assign areas” options are chosen.

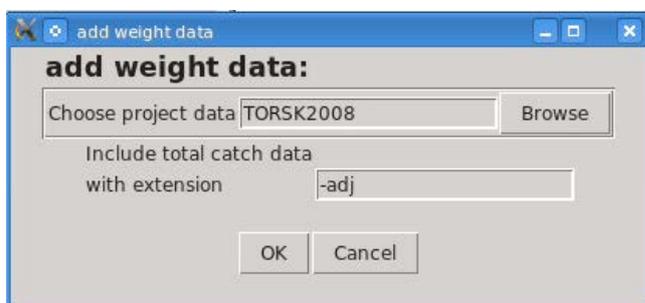
If the species is TORSK, there are three options which do not appear for other species: “remove coastal cod data by area/season” removes *all* data from those areas and seasons assumed to be coastal cod. The default is not to remove any data. “Remove types not 1,2,4 or 5” removes any fish that have been assigned other types, eg 3 or 6.

“Set type outside coastal areas to 5 (skrei)” changes all cod with an age reading from outside the coastal areas (3, 300, 301, 4, 400, 401, 5, 500, 501, 0, 6, 600, 601, 7, 700 and 701) to type 5.

The “filter options” remove data outwith the defined limits, ie below the minimum age or length, above the maximum age or length or with fewer than the specified minimum samples per unit (this is to remove hauls with very small numbers of samples, which are very likely by-catch). The default values are read from the parameter file, if they have been defined there, otherwise they are set to min age=0, max age=100, min length=0, max length=1000 and min n=0. The sixth option sets a minimum number of primary sampling units (usually boats) for each level of season and gear. Any levels of the covariates that do not have this number will be excluded. This option is included so that a data set can be created that can be analysed by both the model and design based methods. It should not normally be changed from the default of zero.

The “debug options” give the option of removing data that appear to be incorrect. They assume the parameters for plausible age-length and length-weight models are in the parameter file. If not, no debugging will be done. The range parameters give the number of standard deviations from the line outside of which data will be removed. These are allowed to be different above and below the line for length-age, but not for length-weight. If debug length-only data is checked, minimum and maximum plausible lengths will be calculated and length-only fish outside these limits removed. Note that filtering is done before debugging, which could have an effect of these plausible lengths.

#### 4.4.2. Add total weight data

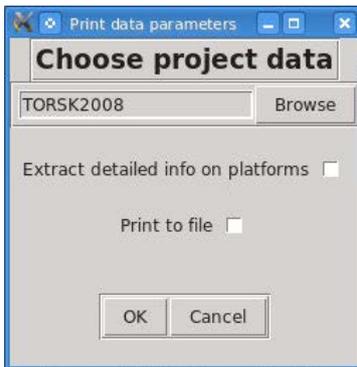


This gives the option of adding total weight data to the project data at a later date. This is because the model fitting can be done without this data. It can then be added after fitting to allow prediction.

### 4.5. Descriptive statistics

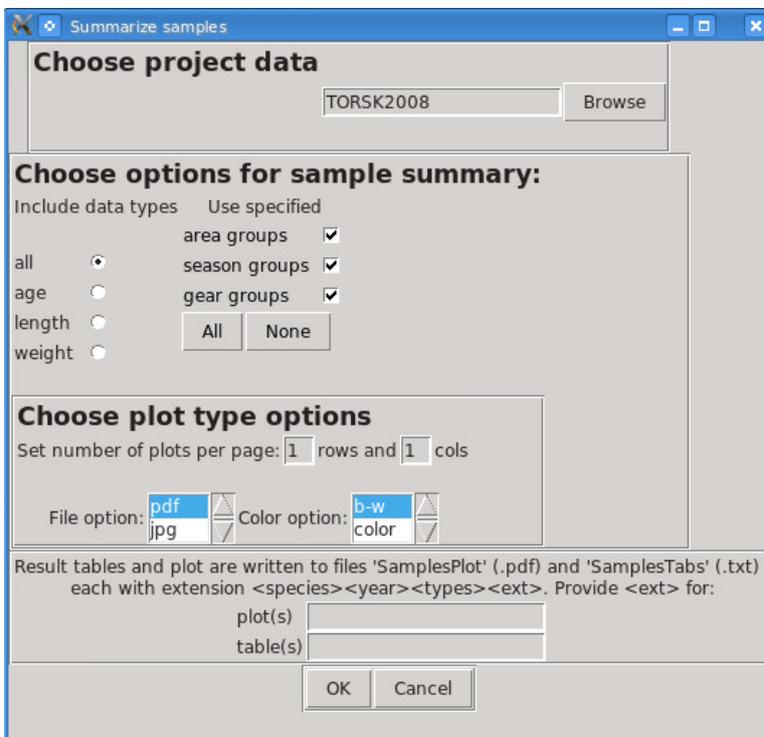
This has four sub menus

#### 4.5.1. Print data parameters



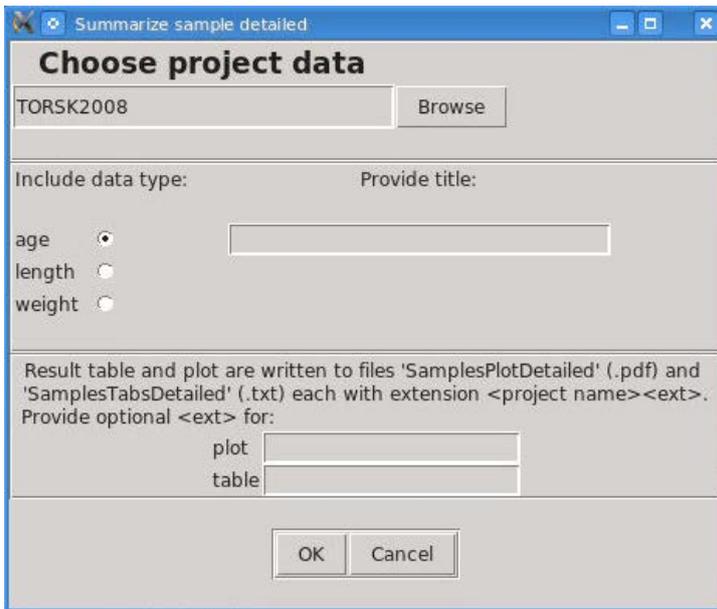
This is used to recall the options that were used to create the data set to be analyzed. If “Extract detailed info on platforms” is ticked, the function calls made to create the project data is printed. This can be printed to file if “Print to file” is ticked.

#### 4.5.2. Summarize samples



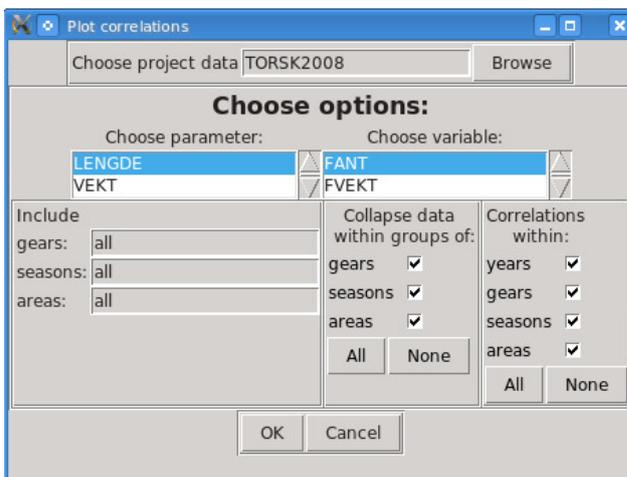
This is used to make plots and report summaries such as number of age, length or weight samples available in the project data. See documentation of R function "sample.summary" for all of the options here. It also provides an option for putting extensions for plots and folders if required to separate results.

#### 4.5.3. Summarize samples detailed



This creates a more detailed summary of the previous; plots a table by gear-, season-, and area-groups as defined in the species-specific file applied on the project data. For each entry in the table it prints the total catch, and number of samples (number of boats, number of stations/hauls, and number of individuals) of the chosen data type.

#### 4.5.4. Plot correlations



This is used for running the function "plot.correlations" and graphically explores relationships between averages of selected variables per haul versus size of cluster for optional groupings of data to indicate presence of intracluster correlation. See help file for "plot.correlations" for full documentation.

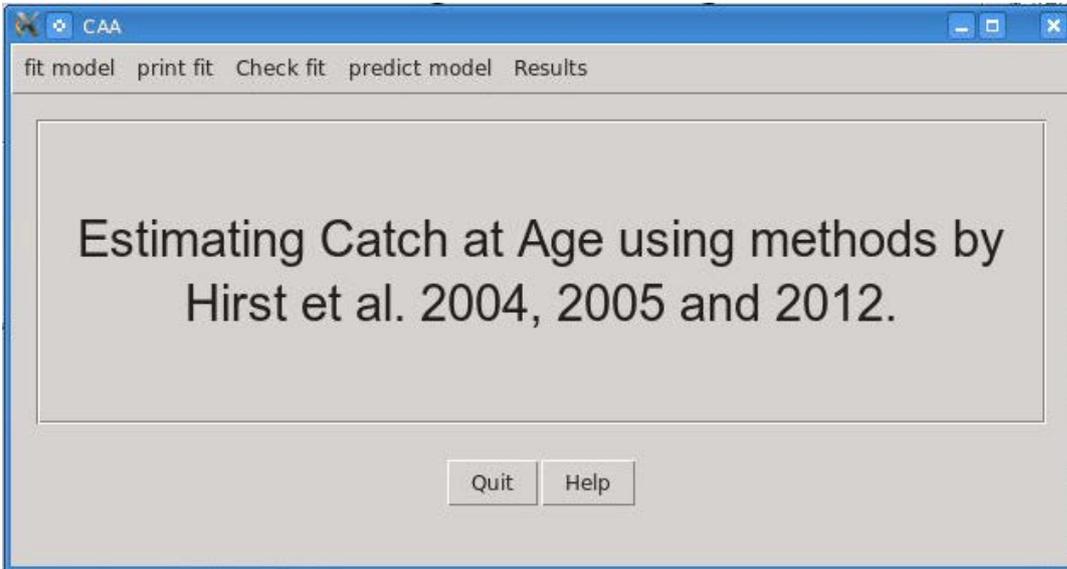
#### 4.6. Estimation

This has two submenus

#### 4.6.1. Designbased

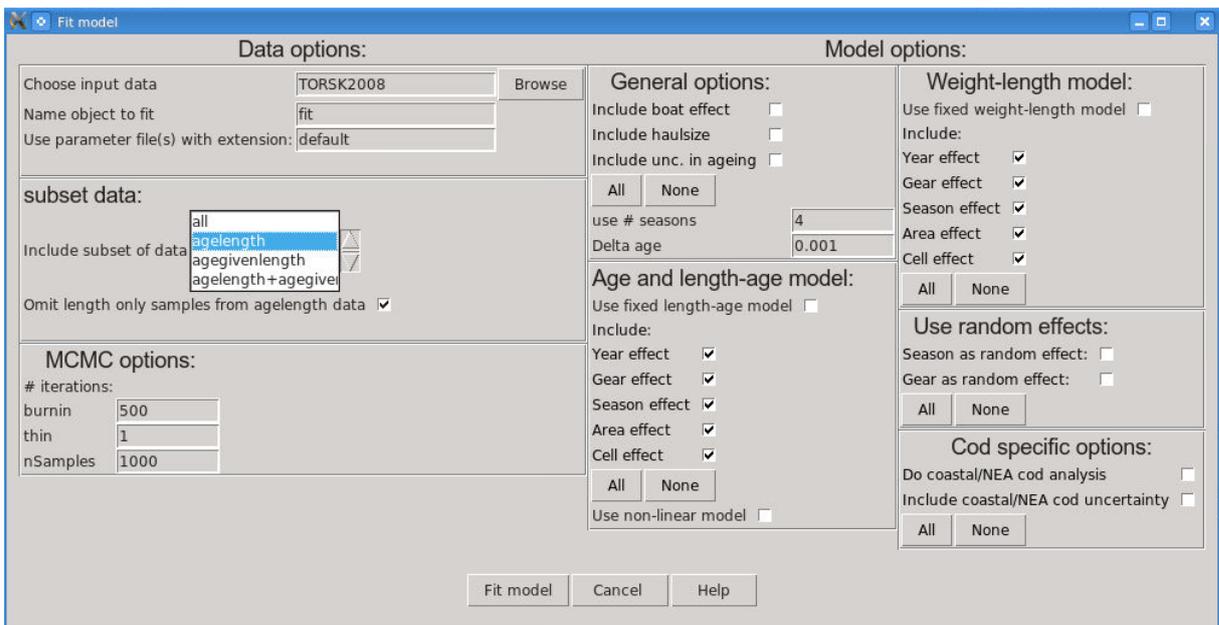
This is not functional yet, but will be developed to include methods for estimating catch at age based on methods using sampling theory and probabilistic sampling.

#### 4.6.2. CAA



This starts the interface for estimating catch at age using the methods by Hirst et al 2004, 2005 and 2012 and has 5 submenus:

##### 4.6.2.1. Fit model



This controls the function `run.model` that does the actual fitting of the model to the data. First the input data is chosen by browsing the folder **ECA/Data/Projectdata** (eg `TORSK2009`, see section 4.4). Then a name of the fit is chosen (defaults to "fit"). The rest of the menu controls some of the options that can be made when fitting the model to the data. When Fit model is pressed R calls the C program that does the MCMC sampling. The results will be saved to several files in **ECA/Analysis/fit**, all of which will have names beginning `<Projectdataname>.<fitname>` eg `TORSK2009.fit.mcmc1`. For details of these files see the appendix. The optional `fitname` allows different options for fitting the model to be tried on the same dataset.

**Subset data:** The default is "all". "agelength" means only data that can be regarded as random samples of ages and lengths are included. These can sometimes include length-only data which can be removed by ticking "omit length only samples from agelength data". "agegivenlength" means length-stratified age samples. Note that some age data must be included, so "length-only" is not an option.

**MCMC options:** The numbers of samples run and stored are controlled by "burnin", "thin" and "nSamples" parameters. "burnin" is the number of samples run and discarded before any are saved. The total number run is  $(\text{thin} * \text{nSamples}) + \text{burnin}$ , of which only `nSamples` are kept. It is advisable to do an initial testrun with a small number of samples to test that the program works with the chosen input data and setup. To get final estimates `burnin=500`, `thin=1` and `nSamples=500` is usually sufficient.

**General options:** "include boat effect" means boat is included as a random effect in all models. "include haul size" means haul size will be included as a covariate in all models. "include unc. In ageing" means the uncertainty in ageing should be accounted for using the error matrix assumed to be in the species parameter file. "use # seasons" defines how many seasons there are in a year, usually 4 or 12. This should correspond to how season is defined in the data. "Delta age" is a parameter used to improve estimation when there are ages included in the model with very few observations. The default of 0.001 should normally be used.

**Age and length-age model:** "use fixed length-age model" assumes that parameters for the model are in the parameter file. See section 3.1 and appendix. In this case there will be no covariates in the length-given-age model, but otherwise it will use the same covariates as the age model, ie those ticked in this panel. The length-given-age model will be log-linear unless the non-linear boxed is ticked, in which case the Schnute-Richards model will be used. This is the function  $\log(\text{length}) = \alpha + \beta g(\text{age})$  where

$$g(\text{age}) = -\log(\theta + \exp(-\gamma \exp(c \log(\text{age}))))$$

$\alpha$ ,  $\beta$ ,  $\theta$ ,  $\gamma$  and  $c$  are parameters to be estimated. The  $g$  function is standardised to lie between 0 and 1.

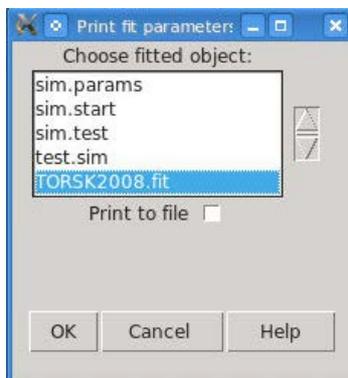
**Weight-length model:** The covariates used in the weight-given-length model can be different to those in the age and length-given-age model.

**Use random effects:** Season and/or gear can be fitted as random effects. This is recommended if some levels of the covariates have very little or no data but estimates are still required for them.

**Cod specific options:** “Do coastal/NEA cod analysis” means use the type definition in the data to estimate the two stocks. The alternative is excluding the standards coastal cod areas and seasons and only estimating skrei catch at age. “include coastal cod/NEA cod uncertainty” means include uncertainty in types 2 and 4. The parameters for this uncertainty must be included in the parameter file (see section 3.1 and appendix).

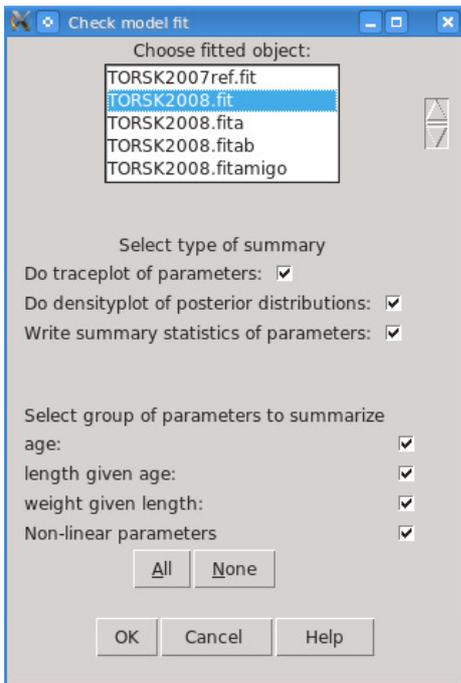
For a full documentation of all options see the help file for "run.model".

#### 4.6.2.2. Print fit



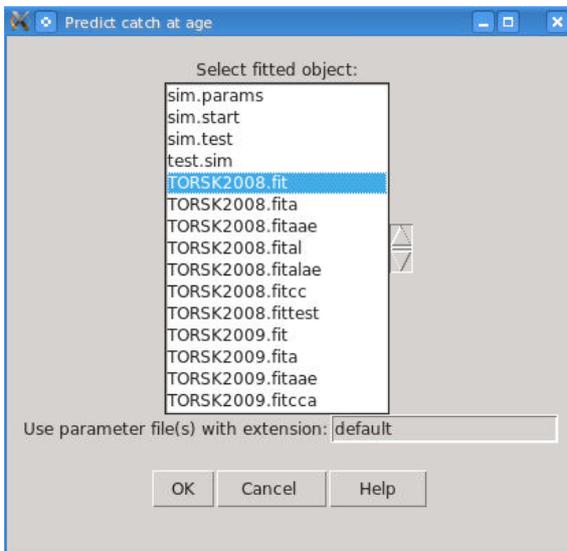
This is used to recall the options that were used to fit the model to the project data. This can be printed to file if “Print to file” is ticked.

#### 4.6.2.3. Check fit

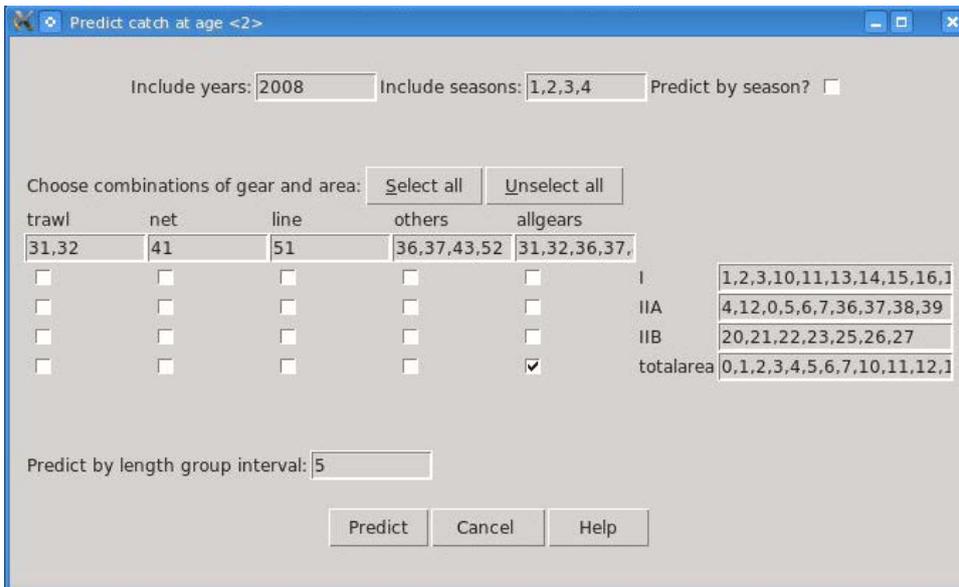


This menu let you choose one fitted object in the folder ***ECA/Analysis/fit*** and produces summaries of the fit for the chosen parts of the model.

#### 4.6.2.4. Predict



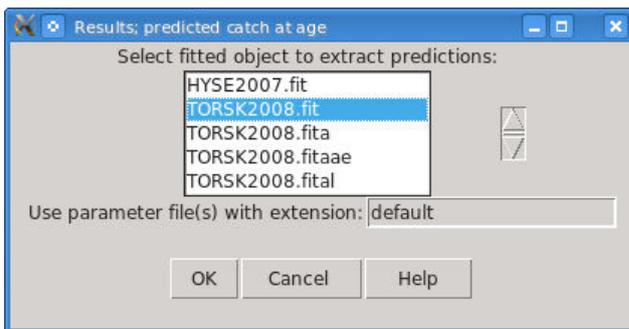
This menu let you choose one fitted objects in the folder ***ECA/Analysis/fit*** and will estimate the catch at age in numbers for the options selected in the next menu box:



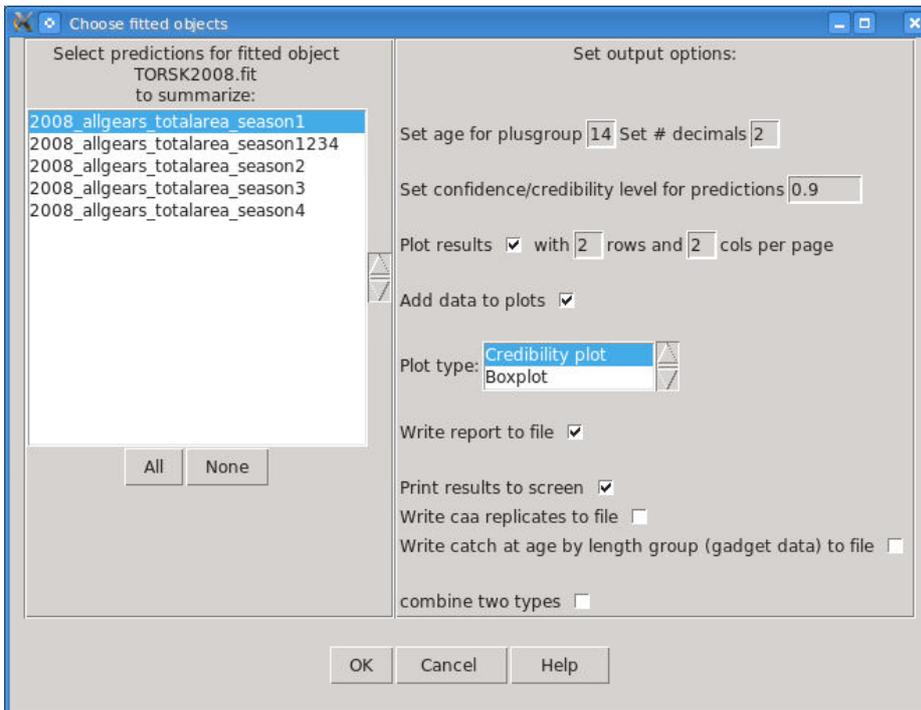
This menu controls the R function "catchatage". Catch at age will be estimated for each combination of gear and area groups selected. If "predict by season?" is ticked the estimation will be for each season separately. The estimates are written to **ECA/Analysis/fit** with the name <Projectdataname><fitname><predictname>, where <predictname> is determined by the row and column names of the matrix, e.g. "gillnet\_I\_season1234". Note that nothing is printed or plotted from this menu – the results are saved in a file for use by the next menu.

The option "Predict by length group interval" gives the length interval if catch-at-age by length is to be calculated. If it is 0 it is summed over all lengths. For a full documentation of the options see the help file for the function "catchatage".

#### 4.6.2.5. Results



This menu first let you choose one fitted object in the folder **ECA/Analysis/fit** for which any predictions have been made and will then let you choose what predictions to extract the results from:



This menu allows you to summarize the results from one or more available. Plots and tables are written to **ECA/Analysis/summary** with names similar to prediction objects, i.e. <Projectdataname><fitname><predictname>.

In the left-hand panel the results object is chosen. Be careful to choose the correct one – they appear in alphabetical order. Here the single season results were from a different run.

On the right side of the menu the plusgroup , number of decimals, and credibility coverage of the summaries are chosen. You may choose the number of plots per page, and add data to the plots. The plots that are made are catch at age in numbers, length given age, weight given length and the trace plots for the catch at age. It is possible that these will have converged even though the individual parameters have not.

There are two optional plot types: "credibility" produces a plot with credibility sets where coverage is determined by the input, while "box" produces a boxplot of length-given-age and weight-given-length.

By default plots are written to file, and a summary report is written to file and printed to screen.

"Write caa replicates to file" means the posterior samples of catch at age (summed over all lengths) are written to file for further analysis.

"Write catch at age by length group (gadget data) to file" mean the posterior samples of catch at age by length group are written to file.

“Combine two types” will combine coastal cod and skrei estimates into estimates for the combined population.

## 5 Running the program using scripts

### 5.1 The same example as above:

```
prokdir<-"/nr/user/david/testHI/Program/"
data.folder<-"/nr/project/stat/CatchAtAge/IMRdata/"
ECA.folder<-"/nr/user/david/testHI/ECA/"
#load the library
library(caa,lib.loc=paste(prokdir,"Rlib",sep=""))
#Set up needed paths and folder names
STUVpath<- paste(data.folder,"STUV/",sep="")
sluttseddelpath<- paste(data.folder,"Sluttseddel/",sep="")
DataByPlatformPath<-paste(ECA.folder,"Data/Rdata/DataByPlatform",sep="")
Srnpath<-paste(ECA.folder,"Data/Resources/serienr",sep="")
FormattedRdatapath<-paste(ECA.folder,"Data/Rdata/Formatteddata/",sep="")
stat.no.path<-paste(ECA.folder,"Data/Resources/",sep="")
stat.pos.path<-paste(ECA.folder,"Data/Resources/",sep="")
parfilepath<-paste(ECA.folder,"Data/Resources//parameters/",sep="")
gear.group.path<-paste(ECA.folder,"Data/Resources/",sep="")
FilterOptionPath<-paste(ECA.folder,"Data/Resources/",sep="")
resultpath<-paste(ECA.folder,"Data/Results/",sep="")
analysisfolder<-paste(ECA.folder,"Analysis/",sep="")
fitfolder<-"/fit/"
predictfolder<- "/predict/"
summaryfolder<-"/summary/"
read.data.main(readyears=2008,readtypes=c('referanseflaate','amigo'),indatapath= STUVpath,
outdatapath=DataByPlatformPath, sr.nr.path=Srnpath, fileext="", Rfileext='.RData')
#####
#2 Choose TORSK and compile the data: OBS; many of the arguments are given as default
values and
#may thus not be necessary to include.
makedata(infilepath= DataByPlatformPath, outdatapath= FormattedRdatapath,
readtypes=c('referanseflaate','amigo'), readyears=2008, readcountries=58,
readspecies='TORSK', rnullobs=TRUE, bycountry=TRUE, byplatformspecific=TRUE,
byvessel=TRUE, bygeargroups=TRUE, bytype=TRUE, RmDiscardData=TRUE, stat.no.path=
stat.no.path, stat.pos.path= stat.pos.path, parfile.path= parfilepath, parfileext='default',
gear.group.path= gear.group.path, FilterOptionPath = FilterOptionPath ,Rfileext='.RData')

#Put data together
caadata<-makeCAAdata(inpath= FormattedRdatapath, species='TORSK',
types=c('referanseflaate','amigo'), years=2008, plotdebug=TRUE,
plotfile='ECA/Data/Results/debugplot.pdf', debug=TRUE, debugonly=TRUE, kAl=4, kAu=4,
kW=4, minage=1, maxage=20, minlength=0, maxlength=240, minn=0, rmcc=FALSE,
rmtype=TRUE, settypebyarea=FALSE,imputemissingweight=FALSE, parameterfile=
paste0(parfilepath, 'TORSKdefault.txt'),rarea=1,stat.pos.file=NULL,nmin=0)
caadata<-add.totcatch.to.caa.data(indata=caadata, totcatchpath=sluttseddelpath, fileext='-ark')

#####
#3 make a sample summary and add catchdata to it
sampletab<-sample.summary(data=caadata, datatype='all', grouparea=TRUE,
groupseason=TRUE, groupgear=TRUE, byPlatform=TRUE)
#Plot sample summary and plot to file
pdffile<-paste0(resultpath, "SamplesPlotTORSK2008referanseflaate.amigo")
```

```

plot.sample.summary(intabs=sampletab,addcatches=T,pdffile=pdffile)
#Write to file tabfile
tabfile<-"SamplesTablesTORSK2008referanseflaate.amigo"
list2ascii(x=sampletab,path=resultpath,file=tabfile)
#####
#Do correlationplot and writetofile pdffile. Note if pdffile=NULL, it plots to screen
pdffile<-paste0(resultpath,"CorrelationPlotTORSK2008referanseflaate.amigo")
plot.correlations(indata=caadata, haulvar='FANT', readtypes='all', readgears='all',
readseasons='all', readareas='all', incyear=T, incgear=T, incseas=T, incarea=T, groupgear=T,
groupseason=T, grouparea=T, mfrow=c(2,3), pdffile= pdffile)
#####
#5 Setting up model; defining common object
pars<-read.parameter.file(paste0(parfilepath,'TORSKdefault.txt'))
ptype1.CC<-pars$classificationerror$ptype1.CC;ptype2.CC<-
pars$classificationerror$ptype2.CC;ptype4.CC<-
pars$classificationerror$ptype4.CC;ptype5.CC<-pars$classificationerror$ptype5.CC
CCError<-list(ptype1.CC=ptype1.CC,ptype1.S=1-ptype1.CC,ptype2.CC=ptype2.CC,ptype2.S=1-
ptype2.CC,ptype4.CC=ptype4.CC,ptype4.S=1-ptype4.CC,ptype5.CC=ptype5.CC,ptype5.S=1-
ptype5.CC)
Intlist<-list(year=TRUE,gear=TRUE,seas=TRUE,area=TRUE,cell=TRUE,boat=FALSE,haul=T)
Slplist<-list(year=F,gear=F,seas=F,area=F,cell=F,boat=F,haul=F)
amodlist<-list(Int=Intlist)
lgamodlist<-list(Int=Intlist,Slp=Slplist)
wglmodlist<-list(Int=Intlist,Slp=Slplist)
common<-
list(burnin=500,thin=1,nsamples=1000,burnin.split=0,thin.split=1,nsamples.split=0,split=FALSE,
usestart=F,int.len=1,cc=FALSE,minage=1,maxage=20,A2A=pars$A2A,CCError=CCError,lgarel
='log-linear',nSeason=4,lgafixed=0,iga=NULL,iga.cc= NULL, wglfixed=0,wgl= NULL,wgl.cc=
NULL,delta.age=0.001,filename='TORSK2008.fit',subset='agelength',omitlengthonly=1,agemod
el=amodlist,lgamodel=lgamodlist,wglmodel=wglmodlist,sim.ar=TRUE,rootfolder=analysisfolder,
fitfolder='/fit',print.boat=FALSE,rand.seas=FALSE,rand.gear=FALSE)
#Fit model to data
run.model(data=caadata,common=common)
#####
#Diagnostics
#Do trace plot of parameters
checkmcmc.bin(filename='TORSK2008.fit', burnin=0, plotall=TRUE, plotage=TRUE,
plotlga=TRUE, plotwgl=TRUE, plotnl=TRUE, rootfolder=analysisfolder, fitfolder=fitfolder,
outfolder=summaryfolder)
#Plot posterior densities of parameters
densities.bin(filename='TORSK2008.fit', burnin=0, plotall=TRUE, plotage=TRUE,
plotlga=TRUE, plotwgl=TRUE, plotnl=TRUE, rootfolder=analysisfolder, fitfolder=fitfolder,
outfolder=summaryfolder)
#Prints posterior mean and standard deviation of terms from specified parts of model
stats.bin(filename='TORSK2008.fit', burnin=0, printall=TRUE, printage=TRUE, printlga=TRUE,
printwgl=TRUE, printnl=TRUE, rootfolder=analysisfolder, fitfolder=fitfolder)
#####
# 6 Do prediction: i.e. estimate catch at age
#Setup common object for prediction
#Predicts within chosen combinations...
caa.weight<-caadata$totcatch[[1]]
caa.seas<-caa.weight$wseason
caa.gear<-caa.weight$wgear
caa.area<-caa.weight$warea
weight<-caa.weight$weight
caa.year<-rep(2008,length(weight))
predictname<-'2008_allgears_totalarea_season1234'
usearea<-c(0,1,2,3,4,5,6,7,10,11,12,13,14,15,16,17,18,20,21,22,23,24,25,26,27,36,37,38,39)
usegear<-c(31,32,36,37,41,43,51,52)

```

```

useseason<-c(1,2,3,4)
useyear<-2008
use<-caa.area%in%usearea & caa.gear%in%usegear & caa.seas%in%useseason
caa.weight.use<-weight[use]
caa.seas.use<-caa.seas[use]
caa.gear.use<-caa.gear[use]
caa.area.use<-caa.area[use]
caa.year.use<-caa.year[use]
par.haulsize<-0
common.pred<-
list(burnin=0,caa.vars=list(year=caa.year.use,seas=caa.seas.use,gear=caa.gear.use,area=caa.
area.use),weight=caa.weight.use,fit.filename='TORSK2008.fit',predict.filename=predictname,n
MC=1000,length.int=5,covtab=caadata$Covtabs,par.haulsize=par.haulsize,rootfolder=analysisf
older,fitfolder='/fit',caafolder='/predict/')
catchatage(common.pred)
#####
#7 Extract results from predictions
#Extract data that is relevant for comparison in plotting; i.e. only those that are 'agelength', i.e.
#random samples of ages and lengths and not stratified by length.
use.data<-caadata$data[caadata$data$INDSAMPLE=='agelength',]
#Extract summary from predicted object
summary.obj<-
summary.predict(fit.filename='TORSK2008.fit',pred.filename='2008_allgears_totalarea_season1
234',cred.p=0.9,rootfolder=
analysisfolder,caafolder='/predict',fitfolder='/fit',plusgroup=14,ndigs=2)
#plot it
par(mfrow=c(2,2))
plot.summary(summary.obj=summary.obj,type='cred',plot.caa=TRUE,plot.lga=TRUE,plot.wga=
TRUE,plot.cal=TRUE,plot.trace=TRUE,cc=FALSE,add.data=use.data)
#Write results as report
print.funk(summary.obj=summary.obj,report.file=paste0(analysisfolder,
summaryfolder,'/TORSK2008.fit.2008_allgears_totalarea_season1234.report.txt'),print.to.screen
=TRUE)
#####

```

## 5.2 Detailed description of commands

The file structure described in section 2 must exist and contain all the necessary files. The names of files and directories in the descriptions of the commands follow the names given in section 2.

### 5.2.1 Data reading functions

#### 5.2.1.1 read.data.main

Reads IMR's .csv versions of SPD data prepared for SAS (SPD split into tables S, T, U and V) and creates R objects for each year and specified type (i.e. sampling platform such as kystvakt). It requires the SERIENR files described in section 3.

Parameters:

readyears: Numeric vector defining what years to read.

readtypes: Character vector defining what types/sampling platforms to read. Names must match exactly with those in the SERIENR files.

indatapath: Full path to folder where original data is stored, ie **STUV**. No default

outdatapath= "ECA/Data/Rdata/DataByPlatform/": Full path to folder to write output.

sr.nr.path = "ECA/Data/Resources/serienr/": Full path to folder where SERIENR files are kept.

Fileext = ".csv": File extension to S, T, U and V files, e.g. '.csv'

Rfileext = ".RData": Optional file extension for the R data sets.

### 5.2.1.2 makedata

Takes R data created by 'read.data.main' and creates one file per species, type and year, called <species><platform><year>.RData eg TORSKamigo1995.Rdata.

Arguments:

infilepath = "ECA/Data/Rdata/DataByPlatform/": Full path to folder where input data created by read.data.main are stored.

outdatapath = "ECA/Data/Rdata/Formatteddata/": Full path to folder to store the output.

readtypes: Vector containing names of platforms, eg "amigo".

readyears: Vector of years to create data.

readcountries: Vector of nations to keep data from. See IMR's documentation of data for definition. Norway is defined by '58'. Used in conjunction with bycountry parameter.

readspecies: Vector of species to select. See IMR's documentation of data for definition.

Rmnullobs=T: Logical. Removes stations where selected species hasn't been caught. Should always be T.

bycountry=T: Logical. If T, then data are kept for countries defined in 'readcountries'

byplatformspecific=T: Logical. Applies type specific selection criteria for types in 'overvaaking', 'kystvakt', 'amigo', 'pelagisk\_kommersiell', 'leiefartøy' and 'sei\_kommersiell'.

byvessel=T: Logical. Selects data where the boat can be identified.

bygeargroups=T: Logical. Group gears according to "geargroups". Data with gears not included in this file are removed from data.

byptype=T: Logical. Perform a check whether "PTYPE" is present, and if not data are removed.

RmDiscardData=T: Logical. Remove data coded as discards.

stat.no.path ="ECA/Data/Resources/": Full path for the location of the file "stat.no.txt".

stat.pos.path ="ECA/Data/Resources/": Full path for the location of the file "stat.pos.txt".

parfile.path ="ECA/Data/Resources/": Full path for the location of the parameterfile file <species><parfileext>.txt.

parfileext="default": Extension of file name for parameterfile, allowing the user to have different versions of parameter files.

gear.group.path ="ECA/Data/Resources/": Full path for the location of the file "geargroups.txt".

FilterOptionPath ="ECA/Data/Resources/": Full path for the location of the file "FilterPlatformOptions.txt".

Rfileext=".RData": Optional file extension to the file saved to disk.

### 5.2.1.3 makeCAAdata

Makes a project object, ie puts data into the required format for fitting the catch at age model. Combines data made by 'makedata' into a list containing 1) "data"; formatted data, 2)"Covtabs"; the definition of covariate structures, and 3) "pars"; the parameterfile information for chosen species, types and years. The function removes outliers according to the parameters chosen.

Arguments:

inpath ="ECA/Data/Rdata/Formatteddata/": Folder where output from makedata is store.

species: Vector of selected species.

types: Vector of selected types/sampling platforms.

years: Vector of selected years

plotdebug = F: Logical. Plot length-age and weight-length data showing points removed by debug.

Plotfile = 'ECA/Data/Results/debugplot.pdf': Text. pdf file for debug plot.

debug = NULL: Logical. Remove outliers using length-age and weight-length distributions specified in parameter file.

debugonly = NULL: Logical. Remove outliers from length-only data using minimum and maximum ages and length-age distribution specified in parameter file.

kAl = NULL: Real. Used for debug. Number of standard deviations below which points are removed from age-length data.

kAu = NULL: Real. Used for debug. Number of standard deviations above which points are removed from age-length data.

kW = NULL: Real. Used for debug. Number of standard deviations above and below which points are removed from weight-length data.

minage = NULL: Integer. Age below which data are removed.

maxage = NULL: Integer. Age above which data are removed.

minlength = NULL: Integer. Length below which data are removed.

maxlength = NULL: Integer. Length above which data are removed.

minn = NULL: Integer. Minimum number of samples per sampling unit. Any units with fewer are deleted.

rmcc = NULL: Logical. Remove all fish assumed to be coastal cod according to area and season.

rmtree = NULL: Logical. Remove any fish with types other than 1, 2, 4 or 5 (keeps fish with missing type).

settypebyarea = NULL: Logical. If TRUE set type to 5 for all fish with ages outside the coastal areas 3, 300, 301, 4, 400, 401, 5, 500, 501, 0, 6, 600, 601, 7, 700 and 701.

imputemissingweight = NULL: Logical. Add estimated weight to fish with ages but missing weights using weight-length distribution specified in parameter file.

Parameterfile = NULL: name and path for parameter file. If NULL defaults to `paste0("ECA/Data/Resources/parameters/",species,"default.txt")`

Rarea = 1: Integer 1, 2 or 3. If 1 use areas as reported. If 2 fill in missing areas using stat.pos file. If 3 replace all areas by those found using stat.pos file.

stat.pos.file = NULL. Stat.pos file including path. If Null defaults to `paste0("ECA/Data/Resources/",pars$stat.pos.file)`

nmin = 0: Integer. Minimum number of units per level of covariates.

(note that the input parameters will default to those in the parameter file if left as NULL)

#### 5.2.1.4 **add.totcatch.to.caa.data**

Adds total catch weight data to object created by makeCAAdata. If rmcc was TRUE in this function the catch in the appropriate areas and seasons is set to zero.

Arguments:

indata: object created by makeCAAdata

totcatchpath: Directory containing total catch data, ie *Sluttseddel*.

fileext: Character. Any extension to weight file after species name, eg '-ark'.

### 5.2.2 Data summary functions

#### 5.2.2.1 **sample.summary**

Makes summary tables of samples in project data by counting PSU's at different levels and by different variables.

Arguments:

data: list with elements 'data' and 'pars'. This is the a project object created by makeCAAdata.

datatype = "all": one of "all", "length", "age", "weight". If "all" any samples in data are counted. If "length" only samples including lengths are counted, if "age" only samples including ages are counted, and if "weight" only samples including weights are counted.

grouparea =T: Logical. If T the data is grouped according to groups of areas defined by the parameterfile, otherwise the original areas are used for tabulation.

groupseason =T: If T the data is grouped according to groups of seasons defined by the parameterfile, otherwise the original seasons are used for tabulation.

groupgear =T: Logical. If T the data is grouped according to groups of gears defined by the parameterfile, otherwise the original gears are used for tabulation.

byPlatform =T: Logical. If T the summary tables are split by sampling platform.

#### 5.2.2.2 **plot.sample.summary**

Plot summary of samples by grouping variables and optionally include respective total catch for comparison

Arguments:

intabs: output from sample.summary.

mfrow =c(1,1): A vector of the form c(nr, nc). Subsequent figures will be drawn in an nr-by-nc array on the device.

mar =c(5,4,4,2): A numerical vector of the form c(bottom, left, top, right) which gives the number of lines of margin to be specified on the four sides of the plot. The default is c(5, 4, 4, 2) + 0.1.

addcatches =F: If T add total catches to the plot if provided by output from sample.summary.

pdffile=NULL: If provided plots are written to the file, otherwise plotted to the plotting device.

jpgfile=NULL: If provided plots are written to the file.

col=T: Logical. If T plot in colour.

### 5.2.2.3 plot.correlations

Summary functions for exploring relationships between hauls averages of selected variables and the size of the haul.

Arguments:

indata: Project data object.

haulvar="FANT": One of 'FANT' (number of fish in haul) or 'FVEKT' (weight of haul).

readtypes="all": Optional subset of sampling platforms in indata. Default "all".

readgears="all": Optional subset of gears in indata. Default "all".

readseasons="all": Optional subset of seasons in indata. Default "all".

readareas="all": Optional subset of areas in indata. Default "all".

incyear=T: Logical. If T include year effect, if F pool data over year before plotting.

incgear=T: Logical. If T include gear effect, if F pool data over gears before plotting.

incseas=T: Logical. If T include season effect, if F pool data over seasons before plotting.

incarea=T: Logical. If T include area effect, if F pool data over areas before plotting.

groupgear=T: Logical. If T pool data by groups of gears defined in parameterfile.

groupseason=T: Logical. If T pool data by groups of seasons defined in parameterfile.

grouparea=T: Logical. If T pool data by groups of areas defined in parameterfile.

mfrow=c(2,3): Numbers of rows and columns in plot layout. Default c(2,3).

Pdffile=NULL: If provided, plots are written to the file, otherwise plotted to plotting device.

### **5.3 Fitting model functions**

#### **5.3.1 run.model**

Runs the C code to fit the parameters.

Arguments:

data: object created by makeCAAdata function.

common: A list with elements:

burnin: Integer. Number of samples at start of run to be discarded.

thin: Thinning rate (eg 10 keeps every 10th sample). Does not apply to burnin.

nsamples: Number of samples to keep.

usestart: Logical. If T program will be restarted using values from the end of a previous run as starting values. Assumes that a run with exactly the same data and filename exists.

int.len: Length interval in cm used to record data.

cc: Logical. If T run coastal cod analysis.

minage: Minimum age to be used in analysis. Any lower values in the data will be omitted.

maxage: Maximum age to be used in analysis. Any higher values in the data will be omitted.

A2A: Matrix defining age errors. Number of rows and columns must be equal to  $\text{maxage} - \text{minage} + 1$ . Columns give probability of observed age given true age, and must sum to 1. If NULL no age errors assumed.

CCerror: Coastal cod class uncertainty. If NULL no class uncertainty. Otherwise a list with 8 elements each of length 1:

ptype1.CC: Probability of a certain coastal cod being classified as type 1.

ptype1.S: Probability of a certain Atlantic cod being classified as type 1.

ptype2.CC: Probability of an uncertain coastal cod being classified as type 2.

ptype2.S: Probability of an uncertain Atlantic cod being classified as type 2.

ptype4.CC: Probability of an uncertain coastal cod being classified as type 4.

ptype4.S: Probability of an uncertain Atlantic cod being classified as type 4.

ptype5.CC: Probability of a certain coastal cod being classified as type 5

ptype5.S: Probability of an uncertain Atlantic cod being classified as type 5.

lgarel: Which model to use for length-given-age: "log-linear" or "non-linear".

nSeason: Integer. Number of seasons in a year (usually 4 or 12).

lgafixed: Logical. T if fixed parameters to be used for length-given-age model (must be given in next argument).

lga: Matrix containing fixed parameters for length-given-age model. 3 columns if log-linear model. 4 if non-linear. Each column contains a constant or random samples from a distribution for the samples in order Constant, intercept, tau.obs, gamma. Columns must be of length  $\text{burnin} + \text{thin} * \text{nsamples}$ .

lga.cc: As lga for coastal cod model.

wglfixed: Logical. T if fixed parameters to be used for wgl.

wgl: Matrix containing fixed parameters for weight-given-length model. 3 columns each containing a constant or random samples from a distribution for the samples in order Constant, intercept, tau.obs. Columns must be of length  $\text{burnin} + \text{thin} * \text{nsamples}$ .

wgl.cc: As wgl for coastal cod model.

delta.age: A small value (eg 0.001) added to each age class in each haul in order to stabilise estimates for very rare age classes.

filename: Character. Name of analysis. Will be used as part of name for output file, and as part of any catch-at-age estimation.

subset: Character. The type of samples to be used. Must exist in data\$INDSAMPLE or no data will be selected. The default is NULL which selects all data. An alternative is "agelength".

omitlengthonly: Logical. Only applies if subset="agelength". Omits any length only samples which have been classified as agelength.

lgamodel: List defining the covariates to be used in the lga model. Has submodels for intercept, slope and the haulsize regression. Each of these must have logical elements for year, seas, gear, area, cell, boat and haul. An example could be:

```
lgamodel=list(Int=list(year=T,gear=T,seas=T,area=T,cell=T,boat=T,haul=T),  
Slp=list(year=F,gear=F,seas=F,area=F,cell=F,boat=F,haul=F),Hsz=NULL)
```

This will fit all covariates for the intercept, only a constant for the slope and not fit the haulsize regression. The haulsize regression is the only part of the model that can be omitted entirely, Int and Slope must have at least a constant fitted.

wglmodel: A list similar to that for lgamodel. Different covariates can be included if required.

agemodel: A list similar to lgamodel except that there is no Slp term.

sim.ar: Logical. If F the area effect is fitted without any spatial component, ie as a standard random effect.

rootfolder: The folder which contains subfolders for the results, any catch-at-age estimates and any plots. If NULL defaults to **ECA/Analysis/**.

fitfolder: Name of the subfolder of rootfolder which will contain the binary file of results from the fit. The folder must exist (it cannot be created by the program). If NULL defaults to **fit**.

print.boat: Logical. If TRUE print boat effects to binary file.

rand.seas: Logical. If TRUE fit season as a random effect.

rand.gear: Logical. If TRUE fit gear as a random effect.

### 5.3.2 Summary and plotting functions:

The mcmc trace plots from a fit can be produced using the function checkmcmc.bin. This takes the following arguments:

filename: Name of fit, ie fit.commands\$filename.

burnin=0: Number of samples to omit from beginning of MCMC samples in plots.  
Default 0

plotall=T: Logical. If T (default) the next 4 parameters are set to T. Set to F if any of the next 4 parameters are T.

plotage=F: Logical. If T plot the age model parameters.

plotlga=F: Logical. If T plot the length-given-age model parameters.

plotwgl=F: Logical. If T plot the weight-given-length model parameters.

plotnl=F: Logical. If T plot the non-linear parameters in the length-given-age model.

rootfolder="ECA/Analysis/": The pdf file will be created in the outfolder subfolder of this folder.

fitfolder="fit": Subdirectory of rootfolder in which the fitted samples are found.

outfolder="summary": Subdirectory of rootfolder into which the plots will be written.

If the filename is eg "TORSK2008" the plots will be written to a pdf file called TORSK2008.checkmcmc.pdf in outfolder.

Posterior densities are plotted with the function densities.bin which takes the same arguments as checkmcmc.bin.

Posterior means and standard deviations are printed to the screen with the function stats.bin. It takes the same arguments as checkmcmc.bin except that the logical arguments "plotall" etc are called "printall" etc and there is no outfolder parameter.

### 5.3.3 Prediction

#### 5.3.3.1 Run prediction

The catch-at-age estimation program is run with the command

catchatage(caa.commands). caa.commands is a list with the following elements:

burnin: Number of mcmc samples to be omitted from start of fitted parameter distributions. Used if the samples have not converged until midway through the chains.

fit.filename: The name of the fit which is to be used, ie fit.commands\$filename

predict.filename: Name of the catch-at-age results, eg ".all.cells.". This will be added to fit.filename to give a combined name for the analysis, eg "run1.all.cells".

rootfolder: Character. The folder within which fitfolder and caafolder are found. If NULL or missing defaults to **ECA/Analysis**

fitfolder: Subfolder of rootfolder in which the fitted mcmc sample and parameter files are found. If NULL or missing defaults to **fit**

caafolder: Subfolder of rootfolder in which the fitted mcmc catch-at-age results will be written. If NULL or missing defaults to **predict**

caa.vars: list with elements year, seas, gear and area, each containing the values of the covariates (on the original scale) over which the catch at age is to be summed. Each element must be the same length, For example if catch at age is required for gear 41, for the whole of 2008, summed over areas 0,3,4 and 5, this can be used:  
caa.vars=list(year=rep(2008,16),seas=rep(1:4,times=4), gear=rep(41,16),  
area=rep(c(0,3,4,5),each=4)).

It is assumed that every level of every covariate given in caa.vars existed in the fit defined by fit.filename. If not cells with missing covariates are omitted and a warning printed.

weight: vector of total catch weights for each cell defined by caa.vars. Must be the same length as each element of caa.vars.

nMC: Total number of units to be simulated. These will be divided amongst the cells defined by caa.vars in proportion to the total catch weight for the cells, with the exception that at least one unit is simulated in each cell.

length.int: Length interval if catch-at-age by length is to be calculated. If 0 catch at age is summed over all lengths.

covtab: Covtabs sublist of list produced by makeCAAdata function. Contains information on true and index values of all covariates.

par.haulsize: Values for haulsize for individual hauls. Possibly simulated.

### 5.3.3.2 Create object containing results

The object containing the results is created with the function summary.predict. It takes the following arguments:

fit.filename: The name of the fit which is to be used, ie caa.commands\$fit.filename eg 'TORSK2008.fit'.

pred.filename= Name of the catch-at-age results ie caa.commands\$predict.filename eg '2008\_allgears\_I\_season1234'.

cred.p: width of credible intervals, eg=0.9.

rootfolder = "ECA/Analysis": Character. The folder within which fitfolder and caafolder are found.

fitfolder = "/fit/": Subfolder of rootfolder in which the fitted mcmc sample and parameter files are found.

caafolder = "/predict/": Subfolder of rootfolder in which the fitted mcmc catch-at-age results will be written.

Plusgroup: age above which results are combined.

Ndigs: number of digits for results

### 5.3.3 Plot predictions

Catch-at-age estimates are plotted with the function plot.summary.

It takes the following arguments:

summary.obj: object created by summary.predict

type: type of plots, either='cred' for credible intervals, or 'box' for boxplots.

plot.caa=T: logical, plot catch at age predictions

plot.lga=T: logical, plot length at age predictions

plot.wga=T,: logical, plot weight at age predictions

plot.cal=T: Logical. Plot catch at length predictions

plot.trace=T: logical, plot catch at age trace plots

cc=F: Logical. Must be T if this is a coastal cod analysis.

add.data: Length given age observations to add to lga plot. Can be given as caadata\$data[caadata\$data\$INDSAMPLE=='agelength',]

### 5.3.4 Print predictions

Catch-at-age estimates are printed with the function print.funk.

It takes the following arguments:

summary.obj: object created by summary.predict

report.file: File to write report to. If NULL don't write to file.

print.to.screen: logical. Print results to screen.

printl=T: Logical. If T print catch-at –length.

## 6 Parameter files

### 6.1. Reading a parameter file

The function that reads a parameter file into R is `read.parameter.file`. It searches for key words in each line and assumes subsequent lines contain the relevant pieces of information. That means that the format of each element of the file must be exactly correct, although the order in which the elements occur can be changed. Also any line containing the symbol “#” is ignored, even if it contains a keyword. There are 6 different formats as follows (see example file below for examples of all formats):

- 1) Number. Eg “minage”. The keyword is followed on the next line by a single number.
- 2) Matrix. Eg “Age error matrix”. The keyword is followed by each line of the matrix, with the elements separated by commas. The matrix must be symmetric, since the number of lines to be read is determined by the length of the first line.
- 3) List. Eg “areagroups”. The keyword is followed by “ngroups”, then by the length of the list, then by “groups”, then by one line for each element of the list (ie each area group). The elements of each line must be separated by spaces.
- 4) Output. Eg “gears”. The keyword is followed by “ngroups”, then by the number of groups, then by two lines for each group. The first gives the names of the group, eg “trawl”, the next the definition of the group, eg 31 32, since gears 31 and 32 make up the trawl group. The elements of these lines must be separated by spaces.
- 5) Table. Eg “length correction”. The keyword is followed by “nfac”, then by the number of rows of the table to be read, then by the column names (separated by spaces), then by the values for the table, one row per table row, separated by spaces.
- 6) Paramfix. Eg “lgafix”. The keyword is followed by the number of samples of each parameter, then by this number of rows, each of which has either 3 (log linear length-given-age or weight given-length parameters) or 4 (non-linear length-given-age) elements, separated by commas.

The keywords of the parameters that fall into each of these classes are as follows:

Number: minage, maxage, minlength, maxlength, plusgroup

Matrix: Age error matrix

List: areagroups, geargroups, seasongroups, wglareagroup, wglgeargroup, wglseasongroup

Output: gears, areas

Table: length correction, weight correction, classification probabilities, length age parameters, weight length parameters

Paramfix: lgafix, lgafix.cc, wglfix

The example parameter file is divided into 4 sections, ie those parameters used in (1) reading the original data, (2) creating the project data, (3) fitting the model, and (4) predicting catch at age. This is only to make the file easier to read, as the parameters can appear in any order. However, only those parameters needed at each stage need be present at that stage, and the file can be edited before each stage and the values changed.

## 6.2 Example of a parameter file

```
### stage 1 parameters ###
```

```
### used in reading STUV files ###
```

```
#Konverteringsfaktor lengde for produkttype definert ved LMAAL på STUV slik at korrigert lengde
```

```
#l=a*I'+b hvor 'l' er lengden registrert på skjema. SJEKK OFFISIELLE KONVERTERINGSFAKTORER
```

```
length correction
```

```
nfac
```

```
1
```

```
LMAAL a b
```

```
K 1.289 1.414
```

```
#Konverteringsfaktor vekt for produkttype definert ved VEKTVOL på STUV slik at korrigert vekt
```

```
#v=a*v' hvor v' er vekten registrert på skjema
```

weight correction

nfac

3

VEKTVOL a

1 1

3 1.5

4 1.18

#####  
#####

### stage 2 parameters ###

### used in creating project data ###

minn

20

minlength

0

maxlength

240

minage

1

maxage

20

areagroups

ngroups

9

groups

3

2 10 11 13 14 15 16 17 18 24 1

12 37 36 39 38

4

5

6

7

0

20 21 22 23 25 26 27

geargroups

ngroups

5

groups

41

51

52

32 36 37 43

31

seasongroups

ngroups

4

groups

1

2

3

4

wglareagroup

ngroups

9

groups

3

2 10 11 13 14 15 16 17 18 24 1

12 37 36 39 38

4

5

6

7

0

20 21 22 23 25 26 27

wglgaregroup

ngroups

5

groups

41

51

52

32 36 37 43

31

wglseasongroup

ngroups

4

groups

1

2

3

4

#Parametre basert på vonBertalanffy vekstmodell  $L = L_{inf}(1 - \exp(-K \cdot \text{age})) \cdot \exp(\epsilon)$ ;  $\epsilon \sim N(0, \sigma^2)$  som brukes til å fjerne uteliggende datapunkt

#sigma angir hvor stort avvik fra gjennomsnitt som aksepteres

length age parameters

nfac

1

Linf K sigma kl ku

232.98028344 0.05284384 0.16180306 4 4

#Parametre basert på modellen  $\log(W) = \alpha + \beta \cdot \log(L)$  som brukes til å fjerne uteliggende datapunkt

#sigma angir hvor stort avvik fra gjennomsnitt som aksepteres

weight length parameters

nfac

1

logalfa beta sigma k

-5.0061 3.0716 0.1454 4

#####





ngroups

4

trawl

31 32

net

41

line

51

others

36 37 43 52

areas

ngroups

3

I

1 2 3 10 11 13 14 15 16 17 18 24

IIA

4 12 0 5 6 7 36 37 38 39

IIB

20 21 22 23 25 26 27

Serienr file

An example is provided:

#Begin serienr file

AAR FRA TIL KOMMENTAR

1981 96000 97999  
1982 96000 97999  
1983 96000 97999  
1984 96000 97999  
1985 96000 97999  
1986 96000 97999  
1987 96000 97999  
1988 96000 97999  
1989 96000 97999  
1990 96000 97999  
1991 96000 97999  
1992 96000 97999  
1993 96000 97999  
1994 96000 97999  
1995 96000 97999  
1996 96000 97999  
1997 96000 97999  
1998 96000 97999  
1999 96000 97999  
2000 96000 97999  
2001 96000 96999  
2002 96000 96999  
2003 96000 96999  
2004 96000 96999

2005 96000 96999

2006 96000 96999

2007 96000 96999

2008 96000 96999

2009 96000 96999

2010 1 1

2011 97301 97999

#End serienr file

Specieslist file

#Begin specieslist file

TORSK

HYSE

SEI

VANLIGUER

SNABELUER

#End specieslist file

Typelist file

#Begin typelist file

referanseflaate

amigo

kystvakt

kystreferanseflaate

#End typelist file

Geargroup file

#Begin geargroup file

41

4000 4110 4111 4112 4113 4114 4115 4116 4117 4118 4120 4121 4123 4125 4126

4130 4131 4132 4134 4135 4140 4141 4142 4143 4144 4146 4148 4149 4150 4151

4152 4160 4166

51

5100 5101 5110 5111 5113 5120 5140

52

5200 5210 5211 5212 5220 5230 5231

37

3700 3710 3711 3712 3713 3720 3730 3731

36

3600

32

3202 3203 3204 3206 3210 3211 3212 3214 3230 3231 3232 3233 3234 3235 3236  
3237 3238 3239 3240 3241 3242 3243 3244 3245 3246 3247 3248 3249 3250 3251  
3252 3253 3254 3255 3256 3258 3259 3260 3261 3262 3263 3264 3265 3266 3267  
3268 3269 3270 3271 3272 3273 3275 3276 3279 3280 3281 3282 3283 3288 3290

31

3100 3101 3103 3110 3111 3112 3113 3114 3116 3118 3120 3121 3122 3123 3124  
3125 3126 3127 3130 3131 3132 3133 3134 3135 3136 3137 3139 3141 3142 3143  
3149 3151 3174 3175 3176 3177 3185

43

4300 3400 3401 3410 3415 3422 3441 3442 3500 3501 3510 3511 3512 3513 3514  
3515 3516 3517 3518 3519 3520 3521 3522 3523 3524 3530 3531 3532 3533 3534  
3540 3541 3542 3543 3546 3547 3548 3552 3580 3582 3590 4311 4312 4314 5300  
5305 5310 5320 5330 5331 5332 5333 5334 5335 5337 5339 5341 5342 5343 5345  
5346 5347 5348 5349 5351 5352

#End geargroup file

#Begin FilterPlatformOptions file

#Merk

#Alle tabeller innen linjer markert med "-" MÅ være tab separert

#blank verdi for variabel skrives i denne sammenheng NA

#variabel det ikke skal kontrolleres for angis NULL

#Dersom en variabel har flere verdier kan dette angis på 2 alternative måter;

#1) skriv lovlige verdier som vektor i R-syntax

#2) repeter linjer, men skriv hver variabel

-----

Proevebaat

AAR	GRUPPE	STTYPE	SKIPKODE
-----	--------	--------	----------

1980:1994	c(NA,20,26)	NA	21
-----------	-------------	----	----

-----

Kystvakten

AAR	GRUPPE	STTYPE	SKIPKODE
-----	--------	--------	----------

1980:2008	c(51,53:90)	NA	1
-----------	-------------	----	---

2009	c(51,53:92)	NA	1
------	-------------	----	---

2010:2013	c(51,53:95)	NA	1
-----------	-------------	----	---

-----

Kontrollverket

AAR	GRUPPE	STTYPE	SKIPKODE
-----	--------	--------	----------

1980:1998	c(NA,20,26,41)	NA	NULL
-----------	----------------	----	------

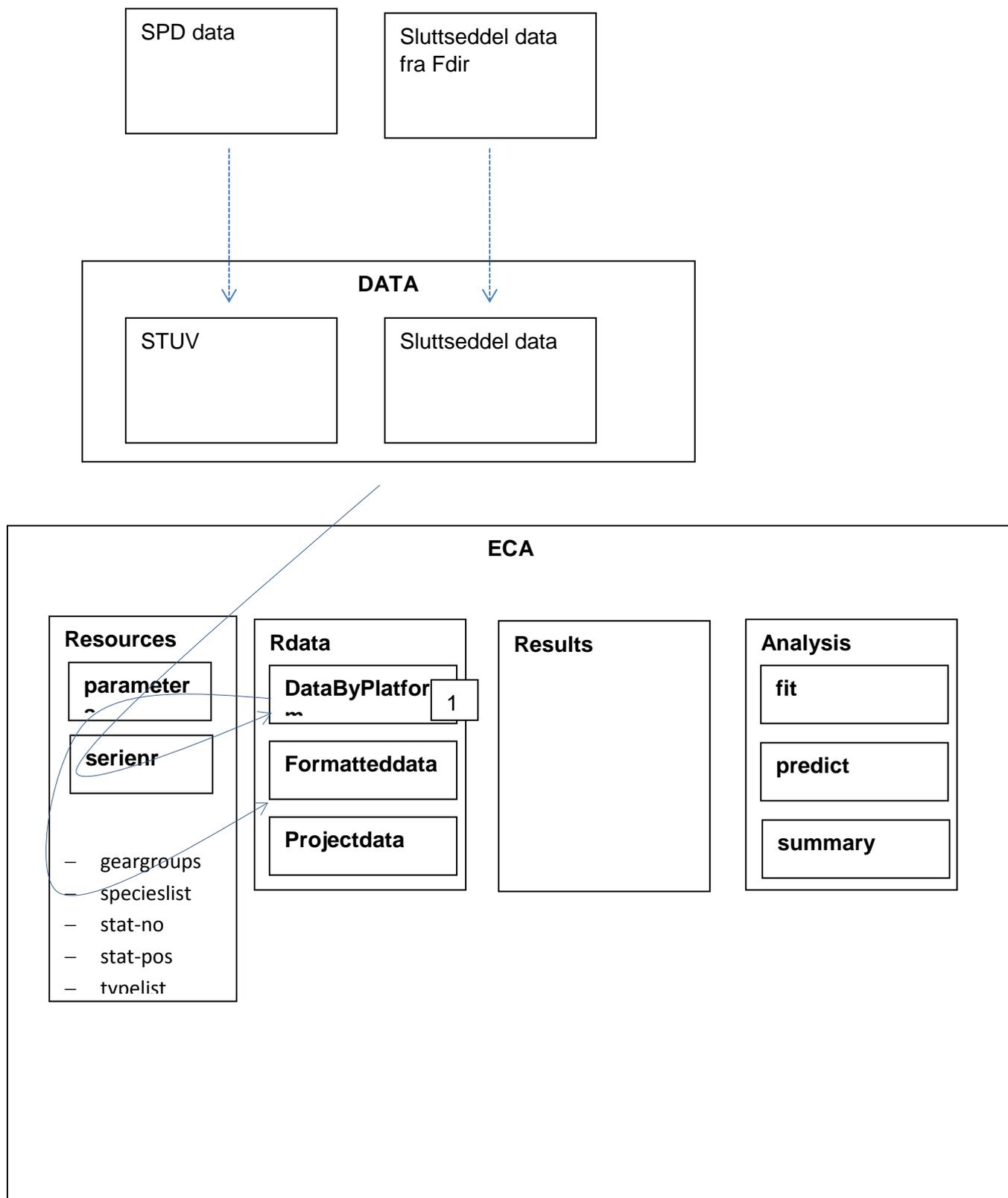
1999:2001	41	NA	NULL
-----------	----	----	------

2002:2013	NULL	NA	1
-----------	------	----	---

-----

#End FilterPlatformOptions file

## 7 Data model of ECA



- Schematic description of data flow before analyzing data

