Application of explicit precautionary principles in data-limited fisheries management

This repository (GA_MSE_PA) is a mirror of GA_MSE with the PA branch displayed as default branch.

Introduction

This repository contains the code for optimising the data-limited empirical rfb rule (ICES WKMSYCat34 catch rule 3.2.1, Fischer et al., 2020) with a genetic algorithm. The simulation is based on the Fisheries Library in R (FLR) and the Assessment for All (a4a) standard MSE framework (`FLR/mse`) developed during the Workshop on development of MSE algorithms with R/FLR/a4a (Jardim et al., 2017).

The `master` branch (GA_MSE) contains the code for the publication:

> Fischer, S. H., De Oliveira, J. A. A., Mumford, J. D., and Kell, L. T. (2021). Using a genetic algorithm to optimise a data-limited catch rule. ICES Journal of Marine Science. 78: 1311-1323. https://doi.org/10.1093/icesjms/fsab018.

This is the `PA branch` which includes the optimisation with specific risk limits for the ICES precautionary approach (PA) and contains the code for the publication:

> Fischer, S. H., De Oliveira, J. A. A., Mumford, J. D., and Kell, L. T. (2021). Application of explicit precautionary principles in data-limited fisheries management. ICES Journal of Marine Science. 12pp. https://doi.org/10.1093/icesjms/fsab169.

The `harvest_rate` branch (GA_MSE_HR) explores the use of harvest rates and contains the code for the publication:

> Fischer, S. H., De Oliveira, J. A. A., Mumford, J. D., and Kell, L. T. (2022). Exploring a relative harvest rate strategy for moderately data-limited fisheries management. ICES Journal of Marine Science. 12 pp. https://doi.org/10.1093/icesjms/fsac103.

The operating models provided as an input are those from the repository shfischer/wklifeVII as described in:

> Fischer, S. H., De Oliveira, J. A. A., and Laurence T. Kell (2020). Linking the performance of a data-limited empirical catch rule to life-history traits. ICES Journal of Marine Science, 77: 1914-1926. https://doi.org/10.1093/icesjms/fsaa054.

Repository structure

The code, input and output files from the master branch (GA_MSE) are retained:

> The root folder contains the following R scripts:
>
> - `OM.R` : This script creates the operating models (OMs),
> - `funs.R` contains functions and methods used for the creation of the operating models and for running the MSE,
> - `funs_GA.R` contains the function used in the optimisation procedure,
> - `run_ms.R` is an R script for running MSE projections and is called from a job submission script
> - `run*.pbs` are job submission scripts which are used on a high performance computing cluster and call `run_ms.R`
> - `analysis.R` is for analysing the results
>
> The following input files are provided:
>
> - `input/stocks.csv` contains the stock definitions and life-history parameters
> - `input/brps.rds` contains the FLBRP objects which are the basis for the OMs
>
> The following outputs summarising the results from running the optimisation are provided:

> - `output/pol_obj_fun_explorations_stats.csv` exploration of fitness functions for pollack
> - `output/pol_interval_MSY_stats.csv` impact of fixing the catch advice interval for pollack
> - `output/all_stocks_MSY_stats.csv` optimisation results for all 29 simulated stocks
> - `output/groups_MSY_stats.csv` optimisation results for stock groups

The following additional files specific to the `PA` branch are provided:

- `OM_sensitivity.R`, `run_ms_sensitivity.R`, and `analysis_PA_sensitivity.R` for the sensitivity analysis (for creating the operating models, running simulations and analysing the results for pollack)
- `run_PA*.pbs` are job submission scripts for the optimisation towards the precautionary approach
- `analysis_PA.R` contains the analysis of the optimisation results

Also, the following summary tables are provided:

- `pol_PA_sensitivity.csv`: summarised results from the sensitivity analysis for pollack
- `pol_PA_sensitivity_SSBs_10000.rds`, `pol_PA_sensitivity_risk_100yrs.csv`: further results from the sensitivity analysis for pollack
- `pol_PA_components_stats.csv`: exploration of including/excluding elements of the rfb rule into the optimisation for pollack
- `all_stocks_PA_multiplier_stats.csv`: optimisation towards the PA with only the multiplier of the rfb rule for all stocks
- `all_stocks_GA_optimised_stats.csv`: combined optimisation results of the rfb rule for the PA and MSY fitness functions
- `all_stocks_2over_stats.csv`: results of the 2 over 3 rule for all stocks
- `PA_summary_table_parameters.csv`: optimised rfb rule parameterisations

R, R packages and version info

The MSE simulations were run on a high performance computing cluster:

```
> sessionInfo()
R version 3.6.1 (2019-07-05)
Platform: x86_64-conda_cos6-linux-gnu (64-bit)
Running under: CentOS Linux 7 (Core)

Matrix products: default
BLAS/LAPACK: /rds/general/user/shf4318/home/anaconda3/envs/R_2020/lib/R/lib/libRblas.

locale:
 [1] LC_CTYPE=en_GB.UTF-8       LC_NUMERIC=C
```

```
 [3] LC_TIME=en_GB.UTF-8        LC_COLLATE=en_GB.UTF-8
 [5] LC_MONETARY=en_GB.UTF-8    LC_MESSAGES=en_GB.UTF-8
 [7] LC_PAPER=en_GB.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_GB.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] parallel  stats     graphics  grDevices utils     datasets  methods
[8] base

other attached packages:
 [1] doMPI_0.2.2         Rmpi_0.6-9          doRNG_1.8.2
 [4] rngtools_1.5        doParallel_1.0.15   GA_3.2.1
 [7] foreach_1.4.8       mse_2.0.3           FLBRP_2.5.4
[10] data.table_1.12.2   ggplotFL_2.6.7.9001 ggplot2_3.1.1
[13] FLash_2.5.11        FLCore_2.6.14.9004  iterators_1.0.12
[16] lattice_0.20-40

loaded via a namespace (and not attached):
 [1] Rcpp_1.0.5       pillar_1.4.6     compiler_3.6.1   plyr_1.8.4
 [5] tools_3.6.1      digest_0.6.18    lifecycle_0.2.0  tibble_2.1.1
 [9] gtable_0.3.0     pkgconfig_2.0.2  rlang_0.4.5      Matrix_1.2-18
[13] cli_2.0.2        gridExtra_2.3    withr_2.3.0      dplyr_0.8.0.1
[17] stats4_3.6.1     grid_3.6.1       tidyselect_0.2.5 glue_1.3.2
[21] R6_2.4.0         fansi_0.4.1      purrr_0.3.3      magrittr_1.5
[25] scales_1.0.0     codetools_0.2-16 ellipsis_0.3.0   MASS_7.3-51.5
[29] assertthat_0.2.1 colorspace_1.4-1 lazyeval_0.2.2   munsell_0.5.0
[33] crayon_1.3.4
```

The framework is based on the Fisheries Library in R (FLR) framework. The exact versions of the packages as used here can be installed with `remotes`:

```
remotes::install_github(repo = "flr/FLCore", ref = "3d694903b9e6717b86c3e8486fc14ebf9
remotes::install_github(repo = "shfischer/FLash", ref = "d1fb86fa081aaa5b6980d74b07d9
# INSTALL_opts = "--no-multiarch" to avoid issues in Windows
remotes::install_github(repo = "flr/FLBRP", ref = "3a4d6390abc56870575fbaba3637091036
```

Furthermore, a data-limited fork of the `flr/mse` package is required:

```
remotes::install_github(repo = "shfischer/mse", ref = "mseDL2.0", INSTALL_opts = "-n
```

And a modified version of the `GA` package for genetic algorithms which also runs on HPCs and supports MPI parallelisation:

```
remotes::install_github(repo = "shfischer/GA")
```

Furthermore, some more R packages available from CRAN are required:

```
install.packages(c("foreach", "DoParallel", "doRNG", "dplyr", "tidyr", "ggplot2", "sc
```

readme.md

For using MPI parallelisation, an MPI backend such as OpenMPI and the R packages `Rmpi` and `doMPI` are required.

Releases

No releases published

Packages

No packages published

Languages

R 92.3% Shell 7.7%