Version: 10-16-2023

***ECOTRAN* as used in the Small Pelagic Fish review**

This is a very brief, abbreviated, practical guide to the *ECOTRAN* end-to-end modeling code used for the ICES/PICES Small Pelagic Fish Symposium. See also the comments within the code files themselves. [NOTE: some of the capabilities of ECOTRAN are not described here such as Monte Carlo analyses, time-dynamic simulations, and coupling to physical drivers]

Please also see in the references section these other publications that have used *ECOTRAN*: Steele and Ruzicka 2011; Ruzicka et al. 2012; Ruzicka et al. 2013a,b; Robinson et al. 2014; Robinson et al. 2015; Ruzicka et al. 2016a,b; de Haast et al. 2017; Chiaverano et al. 2018.

NOTE: Please note that this version of the code is provided as is and as was used for the purposes of this particular small pelagic fish study. The code has not been optimized for clarity, but was often modified “on the fly” for our own purposes.

The basic *ECOTRAN* model platform for steady-state model analyses is also included and described here (the time-dynamic functions are not included here). Please be aware that the underlying *ECOTRAN* code suite included here is frequently being revised for organizational clarity, robustness (esp. to alternate food web functional group types), functionality, and error correction. There may be errors that I have not caught or that appear in model scenario situations that I have never tried.

Please address questions to Jim Ruzicka ([james.ruzicka@noaa.gov](mailto:jim.ruzicka@oregonstate.edu)), and I will do my best to provide clarifications.

**Organization of the summary:**

1. Obtaining *Ecopath* models from the *EcoBase* archive

2. Compile *EcoBase* model metadata

3. Common model functional group names

4. Translate *Ecopath* models into VisualBasic format

5. Read mass-balanced VisualBasic models and batch-save as .mat files for *ECOTRAN* conversion

6. Compile *Ecopath* food web parameter values

7. Calculate mortality rates

8. Calculate Mixed Trophic Impact (MTI)

9. Calculate Footprint & Reach

10. Run "what if" scenario with 20% reduction of SPF biomass

11. Brief description of *ECOTRAN* models

12. Defining food web model parameters

13. Assembly of the main *ECOTRAN* variables

14. Static model sensitivity scenarios in *ECOTRAN*

15. FOOTPRINT & REACH metrics in *ECOTRAN*

16. References

**Step 1. Obtaining *Ecopath* models from the *EcoBase* archive**

*Ecopath* food web models may be queried from the *EcoBase* model repository (www.ecobase.ecopath.org). *EcoBase* is an open-access repository of published *Ecopath* model parameter sets and metadata (Colléter et al. 2015).

Models were queried individually from *EcoBase* by model catalog code number using the Matlab script **ParseEcobase\_07042023.m**.

This script saves individual models in Matlab .mat format as “**ECOBASE\_model-[EcoBase\_Number]\_[date].mat**”.

Redefine variable SaveFile\_directory as your local directory path to where you want to save the model.

This script calls function **parseXML.m**.

**2. Compile *EcoBase* model metadata**

Metadata for all models is obtained from each locally stored “**ECOBASE\_model-[EcoBase\_Number]\_[date].mat**” file using script **CompileMetadata\_09102023.m**. Metadata is stored

Define variable ReadFile\_directory to point to your local directory containing each **.mat** model file saved from Step 1, above.

Script will process all **.mat** model files stored in ReadFile\_directory. Best practice would be to have only desired .mat files stored in ReadFile\_directory.

No sub-functions are called.

Metadata is saved in variable CompiledMetadata. Copy and paste the contents of CompiledMetadata into an excel file if you are processing models in sub-batches in order to build-up a full set of metadata from all models. Manually save the total metadata set as a matlab table variable names CompiledMetadata as a **.mat** file. (See files **Compiled\_Metadata\_10042023.xlsx** and **Compiled\_Metadata\_10042023.mat**)

**3. Common model functional group names**

A common set of model functional group definitions is given in Supplemental Materials table S29 in file **SupplementalMaterial\_Tables\_S1-S29\_Ruzicka\_etal.xlsx**. This information is also stored in matlab file **EcoBase\_nametranslator\_08122023.mat**.

**4. Translate *Ecopath* models into VisualBasic format**

Use matlab script **Ecobase\_makeECOTRAN\_07032023.m** to write *Ecopath* model parameters to excel VisualBasic format (details below). This script calls excel file **FLOGTWOPI\_03252022.xlsm**.

Define local directory path variable ReadFile\_directory containing the **.mat** models saved from Step 1, above.

Define local directory path variable ExcelFile\_directory to the VisualBasic template **FLOGTWOPI\_03252022.xlsm**.

Define local directory path variable SaveFile\_directory to location for saving VisualBasic versions of each model.

Define local directory path variable NameTranslator\_directory to location of the functional group definitions file **EcoBase\_nametranslator\_08122023.mat** from Step 3, above.

Script calls files **FLOGTWOPI\_03252022.xlsm** and **EcoBase\_nametranslator\_08122023.mat**.

Process 1 model at a time. Each model is listed in commented-out code. Un-comment 1 model and the appropriate directory path at a time. The code will automatically process and save 1 model. These processed models are stored in folder

See section **12. Defining food web model parameters** for a brief explanation of the VisualBasic version of *Ecopath* mass-balancing algorithm

Check the balance of each VisualBasic model as described in **12. Defining food web model parameters**, and isolate problem models to a separate folder.

**5. Read mass-balanced VisualBasic models and batch-save as .mat files for *ECOTRAN* conversion**

Check the balance of each VisualBasic model as described in **12. Defining food web model parameters**, and isolate problem models to a separate folder.

Read and save mass-balanced VisualBasic models as a batch process with script **BatchSave\_FlogtwopiFiles\_08252023.m**.

Define local directory path variable ReadFile\_directory to location of VisualBasic versions of each model from Step 4, above.

Define local directory path variable NameTranslator\_directory to location of the functional group definitions file **EcoBase\_nametranslator\_08122023.mat** from Step 3, above.

This script calls function **f\_readKpath\_08142023.m** and file **EcoBase\_nametranslator\_08122023.mat**.

**6. Compile *Ecopath* food web parameter values**

Use matlab script **CompileParameters\_10012023.m** to compile parameters and omnivory index metrics from each model that was saved in matlab format from Step 5, above.

Define local directory path variable ReadFile\_directory to location of the files that were saved in .mat matlab format from Step 5, above

Define local directory path variable SaveFile\_directory to location of the compiled parameter files.

Script calls function **f\_OmnivoryIndex\_01272023.m**.

Script batch-processes all model files stored in the folder (or regional sub-folder) from Step 5, above.

Results are stored in table format variable CompiledParameters:

CompiledParameters.TG1.label 🡨functional group name; TG1-TG23 for 23 different functional groups where “Forage Fish” are Small Pelagic Fish (SPF).

CompiledParameters.TG1.text 🡨model and group information that is in string or matrix format

ReadFile\_name

EcoBase\_ecotype

EcoBase\_OriginalGroupName

EcoBase\_GroupType

EcoBase\_SPFtype\_aggregated

EcoBase\_SPFtype\_resolved

CompiledParameters.TG1.parameters 🡨parameters

EcoBase\_ModelNumber

looky\_currentGrp

TL

OmnivoryIndex

biomass

biomass/biomass\_ALLfish

production\_rate

pb

qb

pq

ae

landings\_pooled

discards\_pooled

catch\_pooled

landings\_pooled/total landings

production\_rate / total primary production

production\_rate / total pelagic fish production

production\_rate / total fish production

EcoBase\_ecotype

**7. Calculate mortality rates**

Use matlab script **CompileMortality\_09262023.m** to calculate and compile mortality rates from different sources of mortality.

Define local directory path variable ReadFile\_directory to location of the files that were saved in **.mat** matlab format from Step 5, above

Define local directory path variable SaveFile\_directory to location of the compiled mortality rate files.

Script calls functions to convert the model into *ECOTRAN* format. See Section **13. Assembly of the main *ECOTRAN* variables**:

**f\_ECOTRANbuild\_08252023.m**

**f\_AggregateBiologicalModel\_05132022.m**

**f\_calcEE\_05122022.m**

**f\_VarianceDivision\_12132018.m**

**f\_VarianceMultiplication\_12132018.m**

**ECOTRANheart\_05132022.m**

**f\_ECOfunction\_05132022.m**

**f\_RedistributeCannibalism\_11202019.m**

**f\_calcEE\_05122022.m**

**f\_calcPredationBudget\_12102019.m**

Results are stored in table format variable CompiledMortality:

CompiledMortality.TG1.label 🡨functional group name; TG1-TG23 for 23 different functional groups where “Forage Fish” are Small Pelagic Fish (SPF). These are the groups for which mortality rates are being calculated.

CompiledMortality.TG1.text 🡨model and group information that is in string or matrix format

ReadFile\_name

group label

CompiledMortality.TG1.parameters 🡨mortality rates

EcoBase\_ModelNumber

M2\_total

F\_total

M2\_ALLzooplankton

M2\_ALLfish

M2\_PELAGICfish

M2\_DEMERSALfish M2\_seabird

M2\_mammal

M2\_cephalopod

M2\_BenthicInvertebrate

M2\_OTHERconsumers

M2\_OTHERconsumers / (M2\_total + F\_total)

**8. Calculate Mixed Trophic Impact (MTI)**

Use matlab script **CompileMTI\_10022023.m** to calculate Mixed Trophic Impact (MTI), total Mixed Trophic Impact (tMTI), and tMTI percentile rank metrics.

Define local directory path variable ReadFile\_directory to location of the files that were saved in **.mat** matlab format from Step 5, above

Define local directory path variable SaveFile\_directory to location of the compiled MTI files.

Script calls function **f\_calcMTI\_10062023.m**.

Script calls functions to convert the model into ECOTRAN format. See Section **13. Assembly of the main *ECOTRAN* variables**:

**f\_AggregateBiologicalModel\_05132022.m**

**ECOTRANheart\_07222023.m**

**f\_ECOfunction\_07212023.m**

**f\_RedistributeCannibalism\_11202019.m**

**f\_calcEE\_05122022.m**

**f\_calcPredationBudget\_12102019.m**

(Note that in this step two functions, ECOTRANheart\_07222023 and f\_ECOfunction\_07212023, are slightly modified from Step 7, above)

Results are stored in table format variable CompiledMTI:

CompiledMTI.TG1.label 🡨functional group name; TG1-TG23 for 23 different functional groups where “Forage Fish” are Small Pelagic Fish (SPF). These are the groups for which total MTI are being calculated.

CompiledMTI.TG1.text 🡨model and group information that is in string or matrix format

ReadFile\_name

EcoBase\_ModelNumber

group label

looky\_ForageFish

EcoBase\_FULLresName

EcoBase\_FULLresGroupType

MTI\_FULLres tMTI\_FULLres

DIET\_pc

A\_cp

looky\_SPFagg\_ForageFish

EcoBase\_SPFaggName

EcoBase\_SPFaggGroupType

MTI\_SPFagg

tMTI\_SPFagg

DIET\_pc\_SPFagg

A\_cp\_SPFagg

looky\_SPFres\_ForageFish

EcoBase\_SPFresName

EcoBase\_SPFresGroupType

MTI\_SPFres

tMTI\_SPFres

DIET\_pc\_SPFres

A\_cp\_SPFres

CompiledMTI.TG1.parameters 🡨 percentile\_tMTI indices

EcoBase\_ModelNumber

percentile\_tMTI

tMTI\_SPFres\_target

**9. Calculate Footprint & Reach**

Use matlab script **CompileFootprintReach\_09252023.m** to calculate Footprint and Reach metrics.

Define local directory path variable ReadFile\_directory to location of the files that were saved in **.mat** matlab format from Step 5, above

Define local directory path variable SaveFile\_directory to location of the compiled MTI files.

Script calls functions to calculate Footprint and Reach metrics. (See Section **15. FOOTPRINT & REACH metrics in ECOTRAN**):

**f\_FootprintReach\_09252023.m**

**f\_WebProductivity\_03272019**

**f\_DietTrace\_08032020**

Script calls functions to convert the model into *ECOTRAN* format. See Section **13. Assembly of the main ECOTRAN variables**:

**f\_ECOTRANbuild\_08252023.m**,

**f\_AggregateBiologicalModel\_05132022.m**,

**f\_calcEE\_05122022.m**

**f\_VarianceDivision\_12132018.m**

**f\_VarianceMultiplication\_12132018.m**

**ECOTRANheart\_05132022.m**

**f\_ECOfunction\_05132022.m**

**f\_RedistributeCannibalism\_11202019.m**

**f\_calcEE\_05122022.m**

**f\_calcPredationBudget\_12102019.m**.

Results are stored in table format variable CompiledFootprintReach:

CompiledFootprintReach.TG1.label 🡨functional group name; TG1-TG23 for 23 different functional groups where “Forage Fish” are Small Pelagic Fish (SPF). These are the groups for which Footprint and Reach rates are being calculated.

CompiledFootprintReach.TG1.text 🡨model and group information that is in string or matrix format

ReadFile\_name

EcoBase\_OriginalGroupName

group address (ECOTRAN)

EcoBase\_GroupType

EcoBase\_ecotype

EcoBase\_SPFtype\_aggregated

EcoBase\_SPFtype\_resolved

EcoBase\_OriginalGroupName (complete list)

net\_FOOTPRINT\_array

gross\_FOOTPRINT\_array

REACH\_array

CompiledFootprintReach.TG1.parameters 🡨 Footprint and Reach metrics

model\_number

FOOTPRINT\_system

REACH\_system

net\_FOOTPRINT\_phytoplankton

gross\_FOOTPRINT\_phytoplankton

REACH\_ALLfish

REACH\_birds

REACH\_mammals

REACH\_fleets

REACH\_landedGrps

**10. Run "what if" scenario with 20% reduction of SPF biomass**

Use matlab script **CompileScenarios\_09262023.m** to calculate consequences to other consumer groups in the model following a change in the availability of a modified group.

Define local directory path variable ReadFile\_directory to location of the files that were saved in **.mat** matlab format from Step 5, above

Define local directory path variable SaveFile\_directory to location to save the compiled scenario results.

Script calls functions **f\_WebProductivity\_03272019.m**, **f\_ScenarioGenerator\_10212020.m**, and **f\_CompileScenarioResults\_03262021.m**.

Script calls functions to convert the model into ECOTRAN format. See Section **13. Assembly of the main ECOTRAN variables**:

**f\_ECOTRANbuild\_08252023.m**,

**f\_AggregateBiologicalModel\_05132022.m**,

**f\_calcEE\_05122022.m**

**f\_VarianceDivision\_12132018.m**

**f\_VarianceMultiplication\_12132018.m**

**ECOTRANheart\_05132022.m**

**f\_ECOfunction\_05132022.m**

**f\_RedistributeCannibalism\_11202019.m**

**f\_calcEE\_05122022.m**

**f\_calcPredationBudget\_12102019.m**.

Results are stored in table format variable CompiledScenario:

CompiledScenario.TG1.label 🡨functional group name; TG1-TG23 for 23 different functional groups where “Forage Fish” are Small Pelagic Fish (SPF). These are the groups for which Footprint and Reach rates are being calculated.

CompiledScenario.TG1.text 🡨model and group information that is in string or matrix format

ReadFile\_name

EcoBase\_OriginalGroupName

group address (ECOTRAN)

EcoBase\_GroupType

EcoBase\_ecotype

EcoBase\_SPFtype\_aggregated

EcoBase\_SPFtype\_resolved

EcoBase\_OriginalGroupName (complete list)

CompiledScenario.TG1.parameters 🡨 Footprint and Reach metrics

model\_number

response\_cephalopod

response\_ALLfish

response\_DemersalFish

response\_PELAGICfish\_nonSPF

response\_seabird

response\_mammal

response\_fleet

response\_landedGrps

**11: Brief description of *ECOTRAN* models**

*ECOTRAN* end-to-end (physics-to-fisheries) models usually consist of a food web model and a physical model. There may be more than one food web representing different physically connected sub-regions. The food web model includes all the trophic interactions from primary producers through top trophic level predators and fishing fleets, nutrient dynamics, and detritus dynamics. The physical model describes the physical transport of nutrients, plankton, and detritus between subregions. The physical model may also describe temperature conditions. The geometry of an *ECOTRAN* model may be variable from a single regional box, to a 1-dimensional stack of vertically connected boxes, to a 2-dimensional cross-shelf model, to a 3-dimensional model with several physically connected food webs across a range of latitudes, longitudes, and depths. Model sub-regions may also be connected by migration of living groups.

The food web used by *ECOTRAN* is typically (but not necessarily) derived from *Ecopath* or *R-path* algorithms (Christensen & Walters 2004, Lucey et al. 2020) and represents a snapshot in time of the energy flow between all producers and consumers. Consumption matrix describes the predation pressure upon each producer *g* (rows) in terms of biomass consumed per period of time by each consumer *c* (columns):

, eq. 1

where is the diet matrix defining the fraction of each prey type ***g*** in the diet of each consumer ***c***, and is a horizontal vector defining the consumption rate of each consumer.

*ECOTRAN* models are based on the transformation into a donor-driven trophic matrix ***Acg*** that maps the fate of all production by groups ***g*** through the food web to consumers ***c*** (Steele & Ruzicka, 2011; Steele, 2009):

eq. 2

Where term is the total grazing or predation pressure upon each producer ***g*** by all consumers ***c***.

Trophic matrix ***Acg*** may expanded to include nutrient and detritus pools and account for the ultimate fate of all biomass consumed by group ***g*** among its predators, among nutrient and detritus pools via feces and metabolic nitrogenous waste excretion, or to detritus as senescence. A model expressed in this format can readily be used to quantify the consequences of changes to community composition (Chiaverano et al. 2018), changes in nutrient subsidies (Treasure et al. 2015), changes in oceanographic regime through coupling with physical models (Ruzicka et al. 2016a), changes in fishery management policy (Ruzicka et al. 2019), or physiological or diet changes to any functional group.

**12: Defining food web model parameters**

*Ecopath*-style mass-balance models are set up within Excel. Excel VisualBasic code provides the essential set of ECOPATH algorithms needed define a food web model directly without needing to use the complete *Ecopath* software package. This is made available to us from Kerim Aydin (NOAA AFSC).

Please see any of the *EcoBase* models that have been expressed in VisualBasic basic, e.g., **ECOTRAN\_Ecobase-7\_14-Apr-2022.xlsm**.

The first five tabs are where the standard *Ecopath* parameters are defined: *MAIN*, *Diets*, *Detritus*, *Fishing*, *Discards*.

The next five tabs are where various parameter uncertainty terms are defined for use in generating Monte Carlo models (if wanted, its optional).

The next two tabs are where the results from ECOPATH are returned: *MainOutputs*, *Mortalities*.

The final three tabs are for the definition of parameters specific to *ECOTRAN*. 1) *EcotranType* tab is where the type of each functional group is defined by number code, and where aggregated groups can be defined for automated aggregation. 2) *EcotranRecycling* tab is where the fates of detritus and NH4 excretion are defined. There are two detritus types: feces & non-predation mortality. Each detritus type goes to one of two terminal pools: surface or benthic. Note that there may be many detritus pools defined as functional groups in *Ecopath*, but ultimately, all detritus that is not consumed will wind up in either the terminal pelagic or the terminal benthic detritus pool. Similarly, there are two NH4 pools: pelagic NH4 and benthic NH4. This tab is also where nitrate and ammonium production is partitioned between primary producers (scroll to the left) 3) The *FunctionalResponse* tab is where functional response parameters are defined. There is room for expansion in this tab to include many optional functional response parameters; as of now, code only uses the first column.

After all food web parameters are set up in the excel workbook, there is a big red button to run *Ecopath* algorithms on the *MAIN* tab (scroll to the right to see the button) to evaluate the thermodynamic balance of the food web. A food web is in “balance” when none of the model’s living groups suffer more predation than their production rate allows. Please see the *Ecopath* manual for details on the parameters needed to build and evaluate the balance of a food web model (Christensen et al. 2005).

[NOT USED IN THE Small Pelagic Fish review🡪] A second big red button on the *MAIN* tab of the excel VisualBasic workbook writes the entire food web to a .csv file for use by *ECOTRAN* in the Matlab scripting language (e.g., **ECOTRAN\_Ecobase-7\_14-Apr-2022.csv**). Note that if Excel VisualBasic gives an error during the save process (due to outdated versions of VisualBasic), the open and formatted model parameter file can be saved as a .csv file manually.

**13: Assembly of the main *ECOTRAN* variables (EnergyBudget & ConsumptionBudget)**

The balanced *Ecopath*-style food web that was built in the excel VisualBasic .xlsm file is exported directly to a .csv file. [In the Small Pelagic Fish review, the VisualBasic files are read directly by matlab without need for conversion to .csv] We now switch to the Matlab scripting language and workspace ([www.mathworks.com](http://www.mathworks.com)).

To run *ECOTRAN* static scenarios, dynamic models, making food web plots, or calculating various model metrics, a common set of steps is followed in the Matlab scripts (see script **CompileScenarios\_09262023.m** where all these steps are combined).

**Step 1 –** Define one food web model file to read, and call function **f\_ECOTRANbuild\_08252023**. This script will load the model into memory, and automatically prepare parameter variables in *ECOTRAN* format. [It is also possible to generate a stack of Monte Carlo food webs, but this is not done in the Small Pelagic Fish review.].

>> [ECOTRAN, ECOTRAN\_MC, PhysicalLossFraction] = f\_ECOTRANbuild\_08252023(dat);

Structure variable dat names the specific EcoBase model of interest.

**calls sub-functions:**

f\_AggregateBiologicalModel\_05132022 prepare EwE model for use by ECOTRAN; also, aggregate functional groups here if wanted

ECOTRANheart\_05132022 Generate an ECOTRAN (E2E) model

f\_ECOfunction\_05132022 returns a single ECOTRAN model for 1 "type" EwE model or 1 MonteCarlo EwE model

f\_RedistributeCannibalism\_11202019 remove cannibalism terms on diagonal of matrix EwE\_diet. [NOTE: Cannibalism is directed to additional metabolism and feces production (equivalent to cannibalism fraction of diet). This is the mechanism for reduced group Transfer Efficiency that J Steele pointed out is required if cannibalism is to be removed from diet matrix.]

f\_calcEE\_05122022 calculate Ecotrophic Efficiency of each functional group to evaluate mass balance

f\_calcPredationBudget\_12102019 for each producer, g, and consumer, c: ((bg \* Qg·) / M2g)) = the fraction of total predation on each producer, g, going to each consumer, c.

f\_VarianceDivision\_12132018 calculate the variance of one term divided by another

f\_VarianceMultiplication\_12132018 calculate the variance of two products

Functional groups do not need to be aggregated beyond the level at which the ECOPATH food web was defined, but the function f\_AggregateBiologicalModel\_05132022 is still needed in order to prepare the *Ecopath* parameters for later construction of the ECOTRAN model.

The ECOTRANheart\_05132022 code returns the “ECOTRAN” structure variable. The ECOTRAN structure variable has a lot of information in it, but the main terms are the EnergyBudget, BioenergeticBudget, ProductionBudget, ConsumptionBudget, and five fate\_ variables.

EnergyBudget (Acg)--- this is trophic matrix, the heart of ECOTRAN. This matrix defines the fate of all energy (biomass) that enters a functional group box. Groups as producers run across columns. Groups as consumers run down the rows. Each column sums to 1. Note that if there is net immigration of a group into the model domain (emigration fraction is negative), then the entries in all other rows of that column will be greater to include distribution of immigration fate.

* Any group may be referred to in this text as either producer (g) or a consumer (c) depending upon your frame of reference within the food web.
* Primary producers are treated as consumers of nutrients (Consumer 1 in the example matrix below).
* Fleets are treated as any other predator – except that the column sum of fisheries do not sum to 1. Fleets remove production from ecosystem and only return a portion of that take as discard contribution to detritus.
* Bacteria can be included explicitly as a defined functional group, or bacteria can be defined implicitly (as in the example below) as flow from detritus to NH4 pools.
* Unconsumed pelagic detritus flows to benthic detritus. Column sum of benthic detritus does not need to sum to 1 – as this and fisheries are the ultimate loss pathways out of ecosystem. (alternatively, benthic detritus column sums to 1 but transfer efficiency is < 1)
* Advection losses and gains are not accounted for within the EnergyBudget. These gains & losses are taken or added directly to production rate estimates during model runs. [NOTE: sinking & migration gains and losses also need to be accounted for outside of the EnergyBudget]

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | NO3 | pelagic NH4 | benthic NH4 | Consumer 1 | Consumer 2 | Fishery 1 | Fishery 2 | pelagic detritus | benthic detritus |
| NO3 |  |  |  |  |  |  |  |  |  |
| pelagic NH4 |  |  |  |  | 0.05 |  |  | 0.08 |  |
| benthic NH4 |  |  |  |  | 0.2 |  |  |  | 0.25 |
| Consumer 1 | 1 | 1 | 1 |  |  |  |  |  |  |
| Consumer 2 |  |  |  | 0.42 |  |  |  |  |  |
| Fishery 1 |  |  |  | 0.3 | 0.05 |  |  |  |  |
| Fishery 2 |  |  |  | 0.05 | 0.3 |  |  |  |  |
| pelagic detritus |  |  |  | 0.2 |  |  |  |  |  |
| benthic detritus |  |  |  | 0.05 | 0.2 | 0.1 | 0.03 | 0.82 |  |

Table 1. Format of the ECOTRAN EnergyBudget (trophic matrix).

These are the array variables that describe the static, balanced state of the average biomass flows over time and bioenergetics of individual functional groups:

BioenergeticBudget --- summary of the fate of all energy (biomass) that enters a functional group box. Basically, a simplified, 3-row version of the EnergyBudget. The sum of each column = 1, except for fleets & benthic detritus):

row 1 = production

row 2 = feces

row 3 = metabolism

ProductionBudget --- a more detailed breakdown of row 1 (production) in the BioenergeticBudget. The sum of each column = production.

row 1 = predation (total consumption of each group p going to all its consumers)

row 2 = eggs (total consumption of each group p going to eggs, gametes, or live births)

row 3 = senescence (total consumption of each group p going to senescence “other mortality”)

row 4 = ba (total consumption of each group p going to biomass accumulation)

row 5 = em (total consumption of each group p going to emigration)

ConsumptionBudget --- a more concise description of the fate of all consumption given in the EnergyBudget, but also includes biomass aggregation (ba = pre-allocated biomass growth) and em (emigration). Each column sums to 1.

row 1 = feces (total consumption of each group p that is not assimilated and is defined as feces)

row 2 = metabolism (total consumption of each group p that is excreted as metabolic waste (NH4))

row 3 = eggs (total consumption of each group p going to eggs, gametes, or live births)

row 4 = predation (total consumption of each group p going to all its consumers)

row 5 = senescence (total consumption of each group p going to senescence “other mortality”)

row 6 = ba (total consumption of each group p going to biomass accumulation). Can be positive or negative.

row 7 = em (total consumption of each group p going to emigration). Can be positive or negative.

fate\_metabolism --- which nutrient pool do metabolic wastes flow to

fate\_eggs --- where does reproductive production flow to

fate\_feces --- which detritus pool does non-assimilated waste flow to

fate\_senescence --- which detritus pool do dead bodies flow to

fate\_predation --- which predator does predation pressure flow to

**14: Static model sensitivity scenarios in *ECOTRAN***

A static model scenario is run by altering a portion of the food web. A static model scenario shows the impact of changes in food web structure or changes to the strength of any pathway(s) within the food web. At present, it assumes linear predator-prey relationships. It is essentially the state of time-dynamic model after an indefinite period of time when a new equilibrium is obtained (Collie et al. 2009, Steele 2009).

Static scenarios are built, analyzed, and plotted in code file CompileScenarios\_09262023.m). Here you can…

* Change biomass and consumption rate of one or more consumer groups
* Change predation pressure on one or more specific producer groups
* Change and compare ecosystem driver rates
* Change fishery group landings and/or discard rates

**script:** CompileScenarios\_09262023

**calls sub-functions:**

f\_ScenarioGenerator\_10212020 modify the EnergyBudget\_matrix according to forced scenario conditions [this file gets revised a lot, make sure this is latest version]

f\_WebProductivity\_03272019 calculate consumption rates (q) of all groups under a given driver (e.g., NO3 or primary production); also accounts for defined rates of group production export when running static scenarios. NOTE: despite the function name, consumption rates are calculated, not production.

f\_CompileScenarioResults\_03262021 a simple function to compile scenario model results relative to base model conditions (see notes in CompileScenarios\_03262023 code for column definitions)

These are the steps that take place in the code:

**Step 1** - Identify and load the *Ecopath* (EwE) mass-balanced model. Define the .csv model filename. This is the mass-balanced model constructed using K. Aydin’s VisualBasic version of the ECOPATH algorithms and exported to .csv format. Models from other sources may be used, but they need to be arranged into the same column format as produced by VisualBasic file (e.g., as produced by **ECOTRAN\_Ecobase-7\_14-Apr-2022.xlsm**). Note: the file directory path will need to be updated for the local computer.

**Step 2** – Load the current food web and build an ECOTRAN model via function **f\_ECOTRANbuild\_08252023**. (See Section **13. Assembly of the main *ECOTRAN* variables**).

**Step 3** - Initialize InputProductionVector. The model is driven by an InputProductionVector. This can defined at any trophic level in theory, but usually is a nutrient input rate or a primary production rate. By default, the InputProductionVector is defined here as the primary production rate P = b \* (p/b) for the primary producer groups. Please note that a base and a scenario vector can be defined for cases where you want to compare the effect of two alternate ecosystem drivers.

**Step 4** - Calculate productivity for each functional group within the un-altered food web. This calls function **f\_WebProductivity\_03272019**.

**Step 5** - Generate a model scenario. There are many options here. The example option is a scaling up of the abundance of one group (baleen whales) and their consumption of model resources at the expense of all other consumer groups that eat the same things (with indirect effects along all trophic chains). First, define the consumer group(s) to be modified. Second, define the amount they should be re-scaled by. Third, define “offset” consumers (those consumers that get to eat more or eat less prey because of the changes to the group with modified abundance. Fourth, define target producers. You can define whether the modified group is changing its consumption of just one, a few, or all of its prey groups. The food web trophic matrix (EnergyBudget) is re-defined (modified) by code **f\_ScenarioGenerator\_10212020**. All Monte Carlo versions of the food web are similarly modified.

NOTE: There will be a warning given for an individual Monte Carlo model in the case that one or several prey groups cannot support the changed abundance of the modified consumer, “-->WARNING in MC-996: maximum scaler exceeded for target producer(s): 14 15 85”. The modified models will still be in thermodynamic balance but the target change to the consumer biomass will not be realized (because there is not enough food available to support that amount of change). Results are expressed as the change in group production relative to the base model: (Pscenario - Pbase) / Pbase.

**Step 6** – Compile scenario results into a single array (scenario\_ResultsCompiled) via function **f\_CompileScenarioResults\_03262021**. See notes in **CompileScenarios\_09262023** code for definitions of each column of scenario\_ResultsCompiled. (These results also get saved to the table format variable CompiledScenario for the Small Pelagic Fish review.)

**15: FOOTPRINT & REACH metrics in *ECOTRAN***

The importance of each functional group ***t*** as both producer and consumer can be characterized with the two non-dimensional metrics: FOOTPRINT and REACH:

**FOOTPRINT** fraction of each PRODUCER's production required to support a particular CONSUMER group.

**REACH** fraction of particular PRODUCER group production going to support each CONSUMER group.

The Footprint of consumer *c* upon producer *g* (e.g., SPF as consumer *c* and phytoplankton as producer *g*) is calculated as the fraction of *g* production that supports production of *c*, via all direct and indirect trophic pathways:

. eq. 3

The Footprint of a specific consumer *c* on producer *g* is vector element *fc·*. Term *te·g* is a horizontal vector of transfer efficiencies for each producer *g* to the next trophic level that accounts for losses to metabolism, feces production, and senescence. Term *dg·* is the vertical food web driver vector with all elements = 0 except *dg·* = 1. To prevent double-counting the contribution of group *g* to group *c*, feedback loops and detritus recycling pathways within the trophic matrix are deactivated. Detritus uptake is set to zero (except flow between detritus pools). Elements of vector *fc·*. are recalculated for each group *c* consecutively with predation upon group *c* set to zero. Losses due to metabolic costs and feces production at each trophic step are included in the Footprint calculations by setting all transfer efficiencies *te·g* = 1 except for the terminal detritus group which is defined as 0.1. Senescence losses do not contribute to group *c* production, nor to the footprint of *c* upon producer *g*, but are directed to detritus pools in *Acg*. This calculation of the gross Footprint is analogous to the Primary Production Required (*PPR*) calculated by *Ecopath* (Christensen et al. 2005, Essington 2006) when *g* is a primary producer. As descriptive shorthand in the text, the Footprint of SPF on primary production is given as *fSPF,PrimProd*.

The Reach of SPF group *g* is the fraction of any consumer group *c* production that is supported by the SPF group. From Ruzicka et al. (2012), the Reach of *g* is calculated by iteratively multiplying the contribution of *g* to the diets of each consumer through diet matrix . Matrix *Tgc* represents the fraction of biomass passing through each trophic linkage in the food web that originated with *g*. *Tgc* is estimated through iteration as

. eq. 4

Vertical Reach vector *θc·* is the fractional contribution of *g* to the diet of each consumer. *θc·* is initialized as *θc·* = *Dg·'*, the vertical transpose of row *g* in the diet matrix. In each iteration of eq. 4, *Tgc* represents the contribution of *g* to the diet of each consumer through direct and indirect trophic pathways up to length *l* = iteration count. The total contribution of *g* to each consumer is recalculated after each iteration by summing *Tgc* down all rows,

. eq. 5

The Reach of *g* (e.g., SPF) to any specific consumer *c* is then element *c* in the final Reach vector *θc*. Before calculation of the Reach, the diet composition of each consumer (*D*.*c*) was renormalized to sum to 1 after setting all cannibalism elements *Dcc* = 0, and the contribution of *g* to itself was set to *θg·* = 1 in each iteration. We performed a maximum of *l* = 1000 iterations with progression halted once no element of *θ* differed by more than 0.0001 from the previous iteration. As descriptive shorthand in the text, the Reach of an SPF group to a specific consumer group is given as *θSPF,consumer*.

See also references:

Ruzicka et al. ICES Journal of Marine Science (2019), doi:10.1093/icesjms/fsz244

Ruzicka et al. Progress in Oceanography 102 (2012) 19–41

**script:** CompileFootprintReach\_09252023.m

**calls sub-functions:**

f\_ECOTRANbuild\_08252023 Generate an *ECOTRAN* (E2E) model (See Section **13. Assembly of the main *ECOTRAN* variables**)

f\_FootprintReach\_09252023 calculate FOOTPRINT & REACH metrics and network trace variables

f\_WebProductivity\_03272019 calculate consumption rate (q) for each group given a driver group input production rate (e.g., primary productivity). NOTE: despite the function name, group consumption is calculated, not production.

f\_DietTrace\_08032020 use DIET matrix to calculate the REACH metrics of each functional group, and calculate the FOOTPRINT & REACH traces for each trophic linkage for one (1) specific functional group of interest = TraceGroup.

These are the steps that take place in the code:

**Step 1** - Identify and load the *Ecopath* (EwE) mass-balanced model. Define the .csv model filename. This is the mass-balanced model constructed using K. Aydin’s VisualBasic version of the ECOPATH algorithms and exported to .csv format. Models from other sources may be used, but they need to be arranged into the same column format as produced by VisualBasic file (e.g., as produced by **ECOTRAN\_Ecobase-7\_14-Apr-2022.xlsm**). Note: the file directory path will need to be updated for the local computer.

**Step 2** – Load the current food web and build an ECOTRAN model via function **f\_ECOTRANbuild\_08252023**. (See Section **13. Assembly of the main *ECOTRAN* variables**).

**Step 3** - Calculate FOOTPRINT & REACH metrics. First, identify one consumer group for which you want to calculate footprint and reach network traces and ecosystem-level metrics. Second, call function **f\_FootprintReach\_09252023** and sub-function **f\_DietTrace\_08032020** to calculate the metrics. The code blocks nutrient recycling in ALL cases. Several variables are returned, with values for each Monte Carlo model:

**DIET\_MC** - diet matrix of each consumer. One DIET matrix for each Monte Carlo model.

- (3D matrix: num\_grps (producers + 1 import) X num\_grps (consumers + 1 import) X num\_MC)

- NOTE: this diet matrix may differ slightly from a defined diet in EwE if the fates of the terminal detritus pools (tab “EcotranRecycling”) differ from those defined as EwE parameters (tab “Detritus”).

**net\_FOOTPRINT\_array\_MC** - net "FOOTPRINT" of each TraceGroup = consumer (rows) upon each producer (columns)

- does NOT include metabolic, feces, and other metabolism losses at each trophic step (not the same as Primary Production Required)

- the fraction of each producer group's production flowing to each consumer = TraceGroup

- (3D matrix: num\_grps (consumer) X num\_grps (producer) X num\_MC) (num\_MC = number of Monte Carlo models)

- NOTE: use for web plotting footprint box colors relative to TraceGroup

- NOTE: does NOT include import diet. There is no footprint calculated for producer = import because the value of import production can be arbitrary.

**gross\_FOOTPRINT\_array\_MC** - gross "FOOTPRINT" of each TraceGroup = consumer (rows) upon each producer (columns)

- DOES include metabolic, feces, and other metabolism losses at each trophic step (if you look just at footprint on primary producers, this should be the same as Primary Production Required)

- the fraction of each producer group's production flowing to each consumer = TraceGroup

- (3D matrix: num\_grps (consumer) X num\_grps (producer) X num\_MC) (num\_MC = number of Monte Carlo models)

- NOTE: use for web plotting footprint box colors relative to TraceGroup

- NOTE: does NOT include import diet. There is no footprint calculated for producer = import because the value of import production can be arbitrary.

**REACH\_array\_MC** - "REACH" of TraceGroup = producer (rows) to each consumer (columns)

- the fraction of each consumer group's production ultimately originating from producer = TraceGroup

- (3D matrix: num\_grps (producers + 1 import) X num\_grps (consumers + 1 import) X num\_MC)

- NOTE: use to plot REACH box colors in food web diagram

- NOTE: DOES include import diet

**FOOTPRINT\_trace\_MC** - Fraction of each trophic link ultimately contributing to production of CONSUMER = TraceGroup

- the relative contribution of each linkage to production of CONSUMER = TraceGroup

- (3D matrix: num\_grps+1 (producers, +import) X num\_grps+1 (consumers, +import) X num\_MC)

- NOTE: use to plot FOOTPRINT arrow colors in food web diagram

- NOTE: DOES include import diet

**REACH\_trace\_MC** - Fraction of each trophic link ultimately originating from PRODUCER = TraceGroup

- It is the fraction of energy within each linkage ultimately originating from PRODUCER = TraceGroup.

- (3D matrix: num\_grps+1 (producers, +import) X num\_grps+1 (consumers, +import) X num\_MC)

- NOTE: use to plot REACH arrow colors in food web diagram

- NOTE: DOES include import diet

**FOOTPRINT\_ system \_MC** - SYSTEM-LEVEL footprint = ratio of total TraceGroup footprint on all PRODUCERS over total production of all CONSUMER groups

- total consumer production of ecosystem excludes microzooplankton

- (vertical vector: num\_MC X 1)

**REACH\_system\_MC** - SYSTEM-LEVEL reach = ratio of total TraceGroup production going to all consumers over total production of all CONSUMER groups

- total consumer production of ecosystem excludes microzooplankton

- (vertical vector: num\_MC X 1)

And, the means & standard deviations of each variable across all Monte Carlo models are given in variables (where row or layer 1 = mean; row or layer 2 = standard deviation):

DIET\_recalculated

FOOTPRINT\_array (you decide in the code whether to use net or gross)

REACH\_array

FOOTPRINT\_trace

REACH\_trace

FOOTPRINT\_system

REACH\_system

**16: references**

Chiaverano LM, Robinson, KL, Tam, J, Ruzicka JJ, Quiñones, J, Aleksa KT, Hernandez FJ, Brodeur RD, Leaf R, Uye S, Decker MB, Acha M, Mianzan HW, Graham WM (2018) Evaluating the role of large jellyfish and forage fishes as energy pathways, and their interplay with fisheries, in the Northern Humboldt Current System. Progress in Oceanography 164:28-36.

Christensen V, Walters CJ (2004) Ecopath with Ecosim: methods, capabilities and limitations. Ecological Modelling 172:109-139

Christensen V, Walters CJ, Pauly D (2005). Ecopath with Ecosim: a User’s Guide. Fisheries Centre University of British Columbia

Colléter M, Valls A, Guitton J, Gascuel D, Pauly D, Christensen V (2015) Global overview of the applications of the Ecopath with Ecosim modeling approach using the EcoBase models repository. Ecological Modelling 302:42–53

Collie JS, Gifford DJ, Steele JH (2009) End-to-end foodweb control of ﬁsh production on Georges Bank. ICES Journal of Marine Science. 66:2223-2232.

de Haast, J.A., Treasure, A.M. , Ruzicka, J. J., Moloney, C.L. (2017) A donor-driven approach to modelling anchovy-sardine dominance shifts in the southern Benguela ecosystem. Journal of Marine Systems. <http://dx.doi.org/10.1016/j.jmarsys.2017.09.001>

Lucey SM, Gaichas SK, Aydin KY (2020) Conducting reproducible ecosystem modeling using the open source mass balance model Rpath. Ecological Modelling 427:109057

Robinson, K.L., Ruzicka, J.J., Decker, M.B., Brodeur, R.D., Hernandez, F.J., Quiñones, J., Acha, E.M., Uye. S.-I., Mianzan, H., Graham, W.M. (2014) Jellyfish, forage fish, and the world’s major fisheries. Oceanography 27:78-89.

Robinson, K.L., Ruzicka, J.J., Hernandez, F.J., Graham, W.M., Decker, M.B., Brodeur, R.D., Sutor, M. (2015) Evaluating energy flows through jellyfish and gulf menhaden (Brevoortia patronus) and the effects of fishing on the northern Gulf of Mexico ecosystem. ICES Journal of Marine Science. DOI: 10.1093/icesjms/fsv088.

Ruzicka, J. J., K. H. Brink, K.H., Gifford, D.J., Bahr, F. (2016a) A physically coupled end-to-end model platform for coastal ecosystems: Simulating the effects of climate change and changing upwelling characteristics on the Northern California Current ecosystem. Ecological Modelling 331:86-99.

Ruzicka, J.J., Brodeur, R.D., Emmett, R.L., Steele, J.H., Zamon, J.H., Morgan, C.A., Thomas, A.C., Wainwright, T.C. (2012) Interannual variability in the Northern California Current food web structure: changes in energy flow pathways and an End-to-End model to investigate system response to alternate forcing scenarios. Progress in Oceanography 102:19-41.

Ruzicka, J.J., E.A. Daly, Brodeur, R.D. (2016b) Evidence that summer jellyfish blooms impact Pacific Northwest salmon production. Ecosphere 7(4):e01324. 10.1002/ecs2.1324.

Ruzicka, J.J., Steele, J.H., Ballerini, T., Gaichas, S.K., Ainley, D.G. (2013a) Dividing up the pie: Whales, fish, and humans as competitors. Progress in Oceanography 116:207-219.

Ruzicka, J.J., Steele, J.H., Gaichas, S.K., Ballerini, T., Gifford, D., Brodeur, R.D., Hofmann, E.E. (2013b) Analysis of energy flow in US GLOBEC ecosystems using end-to-end models. Oceanography 26:24-39.

Ruzicka JJ, Cieciel K, Decker MB, Brodeur R (2019) Examining the ecological role of jellyfish in the Eastern Bering Sea. ICES Journal of Marine Science doi:10.1093/icesjms/fsz244.

Steele, J.H. (2009) Assessment of some linear food web methods. Journal of Marine Systems 76:186–194.

Steele, J.H., Ruzicka, J.J. (2011) Constructing end-to-end models using ECOPATH data. Journal of Marine Systems 87:227-238.

Treasure, A., Ruzicka, J., Moloney, C., Gurney, L., Ansorge, I. (2015) Land-sea interactions and consequences for sub-Antarctic marine food webs. Ecosystems. 18:752-768.