

**Development of ECOTRAN models and their analysis is described in four parts:**

- 1) definition of model parameters
- 2) assembly of the ECOTRAN EnergyBudget\_matrix and BioenergeticBudget
- 3) static model scenarios and metrics
- 4) generate Monte Carlo models
- 5) footprint, reach, and web plots
- 6) sensitivity

**1) definition of model parameters**

ECOPATH models are set up within excel. For the test model, "[TESTmodel\\_01182015.xlsm](#)". A VisualBasic model provides the ECOPATH algorithms to balance the model directly, allowing us to completely bypass the ECOPATH software provided by ecopath.org. This is made available to us from Kerim Aydin (AFSC).

After all parameters are set up, there is a big red button to run ECOPATH on the MAIN tab. The second big red button is to write the entire model to a .csv file for use by ECOTRAN in matlab ("[TESTmodel\\_01182015.csv](#)").

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 ECOTRAN parameters. 1) EcotranType tab is where the type of each functional group is defined by number code, and where aggregated groups can be defined (by AggregationCode) 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. 3)

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.

OPTIONAL: After a balanced model has been generated, there is code available to generate and evaluate Monte Models.

## **2) assembly of the ECOTRAN EnergyBudget matrix and BioenergeticBudget**

The balanced ECOPATH model is exported to a .csv file directly from the excel file where the model was set up. We now switch to matlab.

To run ECOTRAN static scenarios, dynamic models, making web plots, or calculating various model metrics, a common set of steps is followed:

**STEP 1:** load EwE results from .csv file into memory & aggregate functional groups

```
[dat] = f_read_EwE_csv_01122015(readFile)
```

```
[EwEResult, PEDIGREE] = f_AggregateResults_EwE_01122015(dat)
```

Functional groups do not need to be aggregated beyond the level at which the ECOPATH model was defined, but this step is still needed in order to prepare the ECOPATH parameters for later construction of the ECOTRAN model.

**STEP 2:** load Monte Carlo models (OPTIONAL)

Monte Carlo models are generated in other code and can be loaded now from a .mat file. See “4) generate Monte Carlo models”

**STEP 3:** prepare the end-to-end ECOTRAN model

```
[ECOTRAN] = ECOTRANuncertainty_01112015(EwEResult, MonteCarloStore)
```

### **calls sub-functions:**

```
f_ECOfunction_01112015  
f_RedistributeCannibalism  
f_CalcPredationMatrix
```

The ECOTRAN structure variable has a lot of information in it, but the main outputs are the **EnergyBudget\_matrix** and the **BioenergeticBudget**.

**EnergyBudget\_matrix** --- this is 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.

- Primary producers are treated as consumers of nutrients (Consumer 1 in the example matrix below).
- Fisheries are treated as any other predator – except that the column sum of fisheries do not sum to 1. Fisheries 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\_matrix. These gains & losses are taken or added directly to production rate estimates

|                  | 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             |                  |
| emigration       |     |             |             |            | 0.1        |           |           |                  |                  |

Table 1. Format of the ECOTRAN EnergyBudget\_matrix.

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

```
row 1 = BioenergeticBudget_production
row 2 = BioenergeticBudget_feces
row 3 = BioenergeticBudget_metabolism
```

**BioenergeticBudget\_ProductionDetail** --- a more detailed breakdown of row 1 (BioenergeticBudget\_production) in the BioenergeticBudget. The sum of each column = BioenergeticBudget\_production.

```
row 1 = BioenergeticBudget_SumPredation (total consumption of each
group p going to all its consumers)
row 2 = BioenergeticBudget_eggs (total consumption of each group p
going to eggs, gametes, or live births)
row 3 = BioenergeticBudget_OtherMortality (total consumption of each
group p going to senescence "other mortality")
row 4 = BioenergeticBudget_BA (total consumption of each group p
going to biomass accumulation)
row 5 = BioenergeticBudget_EM (total consumption of each group p
going to emigration)
```

**BioenergeticBudget\_OtherMortDetail** --- fraction of "other mortality"

flowing to either pelagic or to benthic detritus pools. Each column sums to 1.

row 1 = flow to terminal pelagic detritus

row 2 = flow to terminal benthic detritus

**BioenergeticBudget\_FecesDetail** --- fraction of feces production flowing to either pelagic or to benthic detritus pools. Each column sums to 1.

row 1 = flow to terminal pelagic detritus

row 2 = flow to terminal benthic detritus

### **3) static model scenarios and metrics**

Run a static model scenario by altering a portion of the food web. Static scenarios are built and plotted in code file [ECOTRAN\\_StaticScenarios\\_TEST\\_05182014](#). Here you can...

- change biomass of one or more groups and their consumption of one or more prey types
- Change the driver production rate of one or more groups

The code starts with running STEP 1 and STEP 2 described above. Lots of options and lots of detail to be added here. But for now, see comments in the code itself...

**function:** [ECOTRAN\\_StaticScenarios\\_TEST\\_01182015](#)

**calls sub-functions:**

[f\\_read\\_EwE\\_csv\\_01122015](#)  
[f\\_AggregateResults\\_EwE\\_01122015](#)  
[ECOTRANuncertainty\\_01112015](#)  
    [f\\_ECOfunction\\_01112015](#)  
        [f\\_RedistributeCannibalism](#)  
        [f\\_CalcPredationMatrix](#)  
[f\\_E2E\\_MonteCarlo\\_09052013](#)  
    [f\\_WebProductivityWLoss](#)  
[f\\_ScenarioGenerator\\_08302013](#)  
    [f\\_WebProductivityWLoss](#)  
[f\\_CompileScenarioResults\\_08192013](#)  
[p\\_PlotScenarioResults\\_01172015](#)  
    myboxplot\_2 (and associated functions from the [Jorn\\_Diedrichsen\\_Toolbox](#);  
        found at [www.icn.ucl.ac.uk/motorcontrol/toolboxes/matlab\\_toolboxes.htm](http://www.icn.ucl.ac.uk/motorcontrol/toolboxes/matlab_toolboxes.htm))

#### **4) generate Monte Carlo models**

Code structure:

[EwE\\_MonteCarlo\\_01182015.m](#) (←main code)

Calls:

[f\\_read\\_EwE\\_csv\\_01122015](#) (←reads in EwE model .csv file)  
[f\\_AggregateResults\\_EwE\\_01122015](#) (← prepare EwE model)  
[f\\_EwEinterval\\_MonteCarlo\\_04092014](#) (←prepares sampling interval)  
[f\\_EwEnormal\\_MonteCarlo\\_04092014](#) (←generate 1 random model)  
[f\\_DietPreference\\_Readjust\\_04052014.m](#) (←adjust diets w/ prey guilds)  
[f\\_EE\\_MonteCarlo\\_04092014.m](#) (←evaluate model balance based on pre-defined EE limits of each group, NOTE: allows for “out of balance models” if wanted)

**STEP 1:** prepare EwE file as a .csv file as in “1) definition of model parameters”

**STEP 2:** in matlab code [EwE\\_MonteCarlo\\_01182015.m](#):

Define SaveFile\_directory and saveFile

Define ReadFile\_directory and readFile

**STEP 3:** pre-test difficulty of generating mass-balanced random model by adjusting the variables PEDIGREE.SamplingScale, PEDIGREE.SamplingScale\_Diet, and PEDIGREE.SamplingScale\_Fisheries. Random models are generated by sampling from within a normal distribution defined by your CV parameter pedigrees. By default, sampling is restricted to within 1 STD of the original parameter value (you can extend or shrink this by changing PEDIGREE.STD\_scaler. The problem is that for some models, millions and millions of models are tossed before 1 good model is found. So, as a compromise in order to be able to get a reasonable set of Monte Carlo models from a laptop computer, you can “steepen” the normal distribution about which random parameter values are drawn. The sampling range doesn’t change (still 1 STD be default), but the random sampling becomes more concentrated near the original parameter value as you increase the value of PEDIGREE.SamplingScale, PEDIGREE.SamplingScale\_Diet, and PEDIGREE.SamplingScale\_Fisheries. I aim to get at least 1 good model for every 50,000 random models. The justification for this compromise is that the number of models generated is so large you are still sampling over the entire parameter range even if you are focusing near the original parameter set values.

NOTES:

- If you are aggregating a model down to a smaller number of boxes, I find it best to make sure that your aggregated model is in balance before you run the Monte Carlo. Computer-aggregated models may not necessarily be in automatic balance.
- This code corrects for the potential of drawing physiology parameters that allow an implied “negative” metabolic rate by rejecting those parameter sets.