

# EBTtool

## Guide rapide d'utilisation

Vincent Le Bourlot      David Claessen

5 décembre 2011

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Utilisation basique</b>	<b>1</b>
2.1	Extraction de l'archive . . . . .	1
2.2	Lancement d'EBTtool . . . . .	2
2.3	Ouverture d'un projet . . . . .	3
2.4	Execution . . . . .	3
2.5	Affichage des résultats . . . . .	5
<b>3</b>	<b>Utilisation avancée</b>	<b>8</b>
3.1	Modification des paramètres . . . . .	8
3.2	Modification des conditions initiales . . . . .	10
3.3	Export de l'état final comme condition initial . . . . .	11
3.4	Bifurcation . . . . .	11
3.5	Edition du code source . . . . .	12

## 1 Introduction

Ces quelques pages présentent de façon concise le logiciel de simulation de populations physiologiquement structurées EBTtool. Ce logiciel, écrit par Andre de Roos, est disponible à l'adresse <http://staff.science.uva.nl/~aroos/>. Cette introduction a pour but de donner les clés permettant une utilisation basique du logiciel : ouvrir un modèle, éditer les paramètres, compiler et exécuter le modèle, afficher des sorties graphiques. Les exemples fournis dans cette introduction utilisent le modèle et les fichiers fournis par David Claessen dans l'archive TDebt-forstudents.zip

## 2 Utilisation basique

### 2.1 Extraction de l'archive

Lorsque vous ouvrez l'archive, huit fichiers sont extraits dans le dossier courant :

- `kmmodel.ebtp`  
Ce fichier est le fichier "projet".
- `km.c`  
Ce fichier contient le code source du modèle.
- `km.h`  
Ce fichier est le "header" du code source et contient quelques variables statiques du modèle.
- `runkm.cvf`  
Ce fichier contient les paramètres du modèle. Ce fichier constitue le fichier "run".
- `runkm.isf`  
Ce fichier contient les conditions initiales du modèle.
- `histogram.ebtsgr`  
Ce fichier est un fichier "state graph".
- `juv-adu-numbers.ebtogr`  
- `resource.ebtogr`  
Ces deux fichiers sont des fichiers "output graph".

Le rôle des différents fichiers sera expliqué au fur et à mesure de cette introduction. Pour le bon fonctionnement du modèle, il est important que les fichiers `.c` et `.h` conservent le même nom, et de même pour les fichiers `.cvf` et `.isf` (ex : `km.c` et `km.h`, `runkm.cvf` et `runkm.isf`).

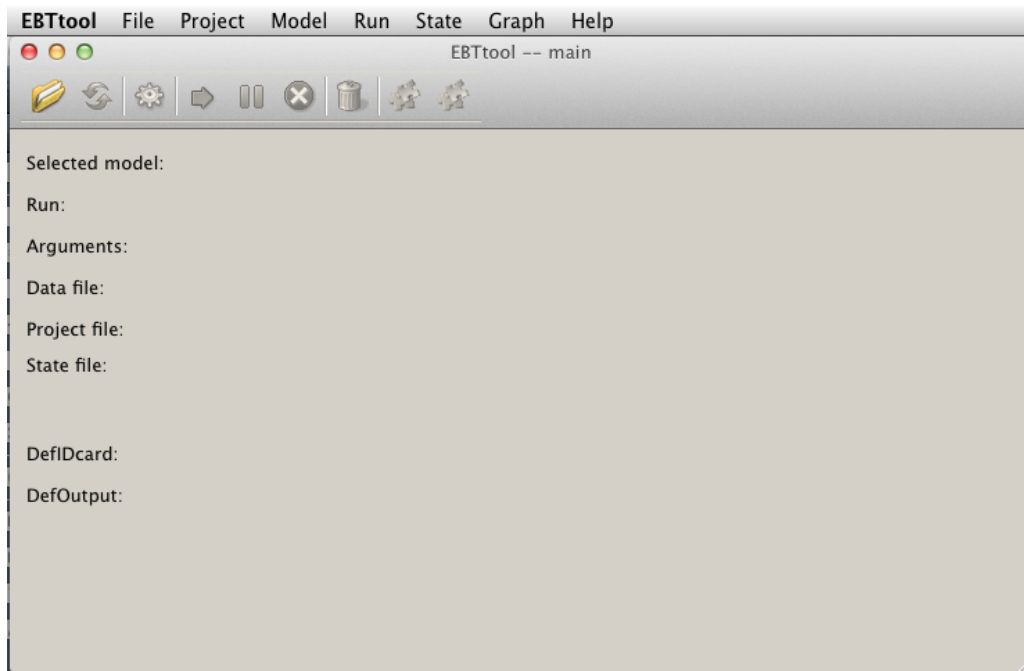


FIGURE 1 –








## 2.2 Lancement d'EBTtool

Une fois que vous avez extrait l'archive à l'endroit de votre choix, vous pouvez lancer EBTtool. Au démarrage, le programme s'ouvre sur la fenêtre principale, appelée "main" (Fig. 1).

Cette fenêtre propose plusieurs menus.

- Le menu **File** permet d'ouvrir des fichiers de données obtenus après des simulations.
- Le menu **Project** permet d'ouvrir un "projet", d'en éditer les descriptions, et de le sauvegarder.
- Le menu **Model** permet d'ouvrir un modèle, d'en éditer la source ou le header, de copier les fichiers, de créer un nouveau modèle et de compiler le modèle.
- Le menu **Run** permet d'ouvrir un "run" (fichier `.cvf` des paramètres du modèle), de voir et éditer les différents fichiers inhérents à ce run (paramètres, état initial, sorties, état final et résumé des conditions du run), de créer ou copier certains des fichiers, d'exécuter le modèle, de le stopper, et d'effacer les sorties du modèle.
- Le menu **State** permet de charger des sorties "d'état" de la simulation.
- Le menu **Graph** permet d'ouvrir et de sauvegarder les graphiques d'état (stat graph) ou de sortie (output graph) pour représenter les résultats d'une simulation.

En plus de ces menus, cette fenêtre propose plusieurs raccourcis.

-  Ouvrir un fichier de données.
-  Recharger le fichier de données.
-  Compiler le modèle.
-  Exécuter le modèle.
-  Mettre l'exécution en pause.
-  Arrêter l'exécution.
-  Supprimer les sorties du modèle.

Les deux derniers raccourcis concernent des fonctions avancées du logiciel et ne seront pas utilisés.

## 2.3 Ouverture d'un projet

Un projet EBT (fichier `.eptpr`) est un fichier contenant les informations nécessaires à EBT-tool pour charger le modèle, le run sauvegardé lors de la dernière utilisation et les différents graphiques sauvegardés.

Dans le cas où il n'y a pas de projet sauvegardé, les différents éléments peuvent être chargés un à un de la façon suivante.

1. Sélectionner **Model** -> **Open**, et ouvrir le fichier `.c` contenant les sources du modèle désiré.
2. Sélectionner **Run** -> **Open**, et ouvrir le fichier `.cvf` contenant les paramètres que vous souhaitez utiliser.
3. Sélectionner **Graph** -> **Open** -> **Output**, et ouvrir les fichiers graphiques `.ebtogr` désirés, puis **Graph** -> **Open** -> **State** et ouvrir les fichiers graphiques `.ebtsgr` désirés.
4. Sélectionner **Project** -> **Save as a new project...** pour sauvegarder le projet et ne pas avoir à répéter ces étapes la prochaine fois.

Une fois le projet ouvert, la fenêtre principale affiche maintenant les noms du modèle, du run, du fichier de sortie (`.out`) et ses dimensions, et du projet (Fig. 2).

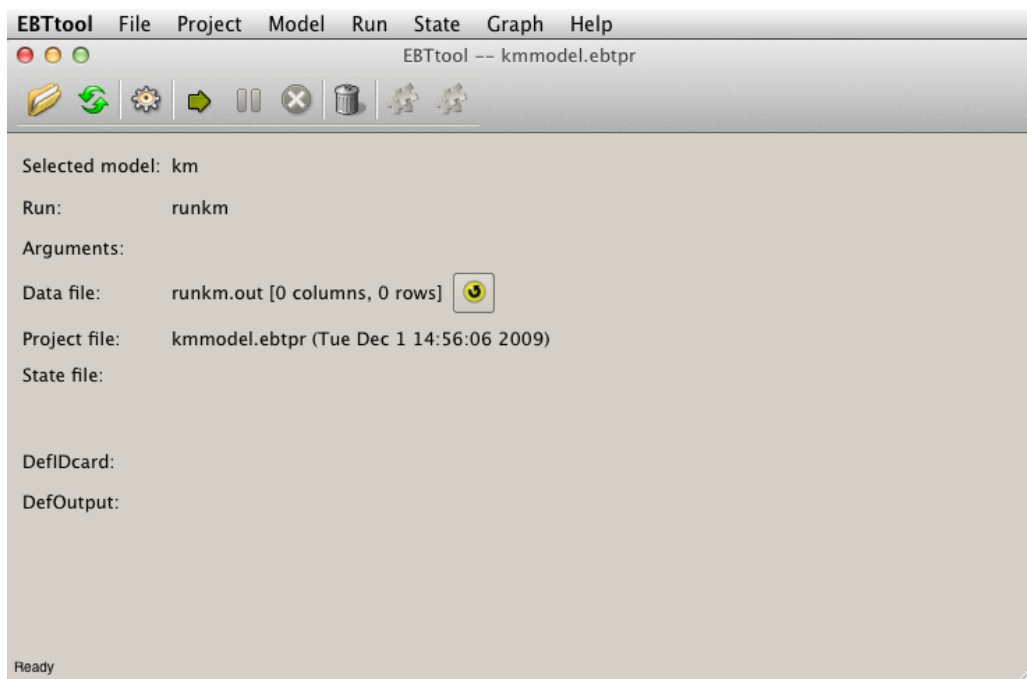


FIGURE 2 –

## 2.4 Execution

Avant d'exécuter une simulation, il est important de vérifier que le code source du modèle a été compilé pour être exécutable. Afin d'effectuer cette vérification, cliquez sur le bouton raccourcis de compilation (🔧), et dans la fenêtre qui vient de s'ouvrir (Fig. 3) cliquez sur **Compile**. Si le modèle n'est pas encore compilé, la compilation s'exécute. Une série de message s'affichent, se terminant par

```
Linking xxxxmodule.dylib.....  
Done.
```

si la compilation s'est effectuée sans erreur.

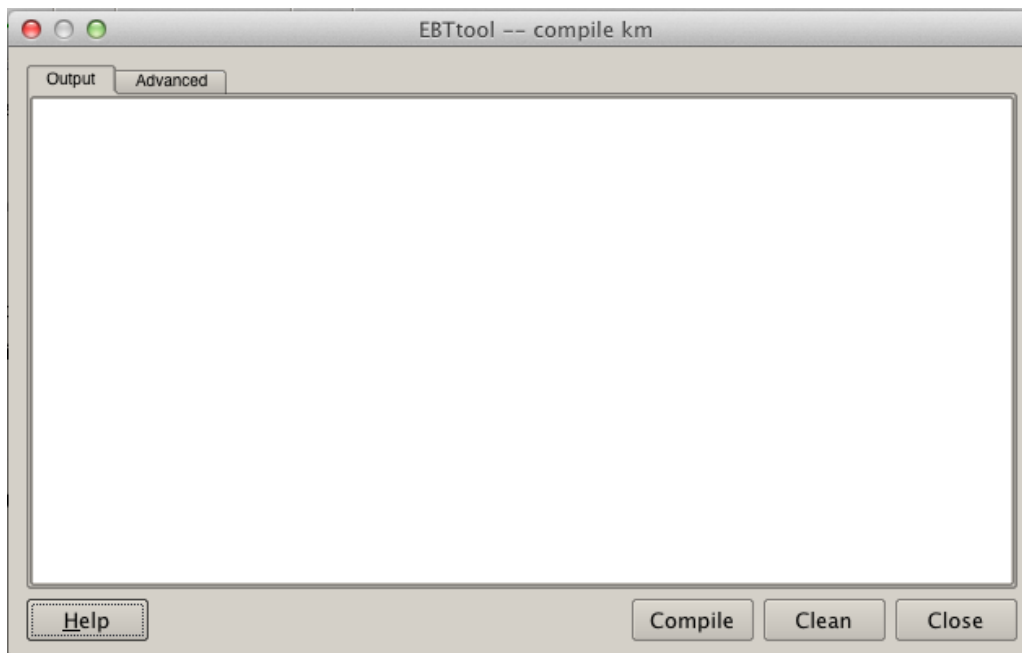


FIGURE 3 –

En cas d'erreur de compilation, un message d'erreur s'affiche vous indiquant généralement le type d'erreur et la ligne correspondant dans le code source, comme par exemple cette erreur à la ligne 214 où la variable "time" n'a pas été déclarée.

```
Building xxxxmodule.o: Compiling xxxx.c.....
```

Errors!


```
km.c: In function 'void InstantDynamics(double*, double (**)[2], double (**)[2])':  
km.c:214: error: 'time' was not declared in this scope.
```




Il faut alors aller modifier le fichier .c pour corriger cette erreur et retenter la compilation.

**Remarque:** *Le code source fourni dans l'archive compile et s'exécute sans erreur avec les paramètres et l'état initial fourni. Avant de les modifier, il est vivement conseillé d'en faire une copie.*

Si le code source est déjà compilé dans sa dernière version, le message suivant s'affiche.

```
The file xxxxmodule.dylib is up-to-date.
```

Le modèle est alors prêt à être exécuté avec les paramètres définis dans le fichier .cvf et les conditions initiales .isf. Avant l'exécution, il est conseillé de nettoyer les données de sortie afin que les graphiques n'affichent que les données du run que vous allez exécuter. Pour cela, cliquez sur , et sélectionnez "All data".

Pour exécuter le modèle, cliquez sur . La simulation démarre, seul les boutons pause () et stop () deviennent accessibles (Fig. 4), et la fenêtre du graphique d'état (histogram) montre la distribution en taille de la population au cours du temps (Fig. 5).

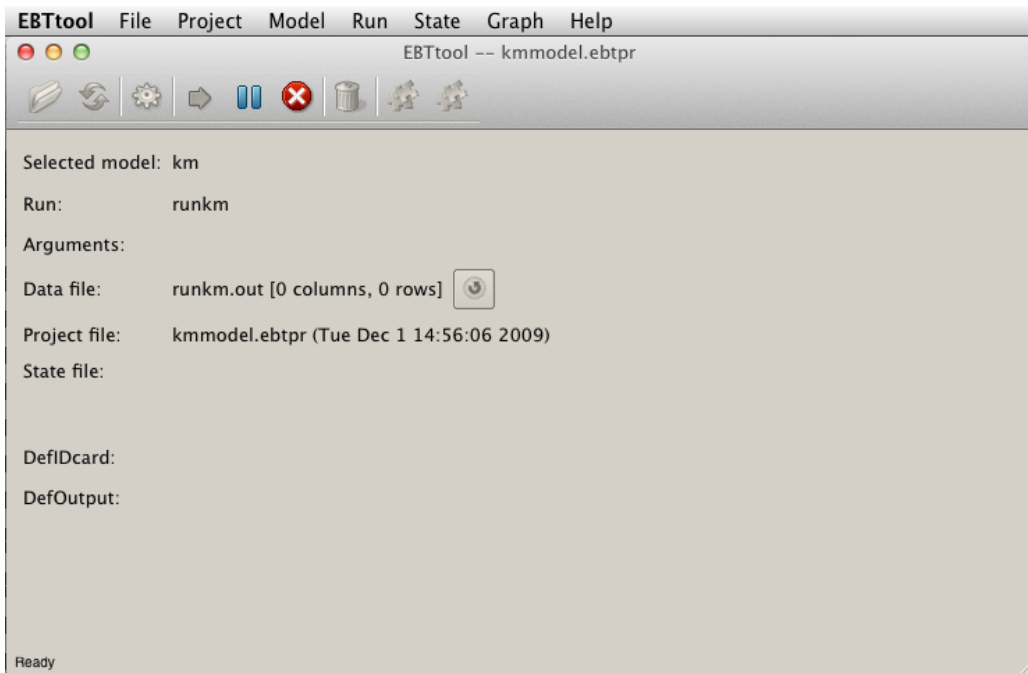


FIGURE 4 –

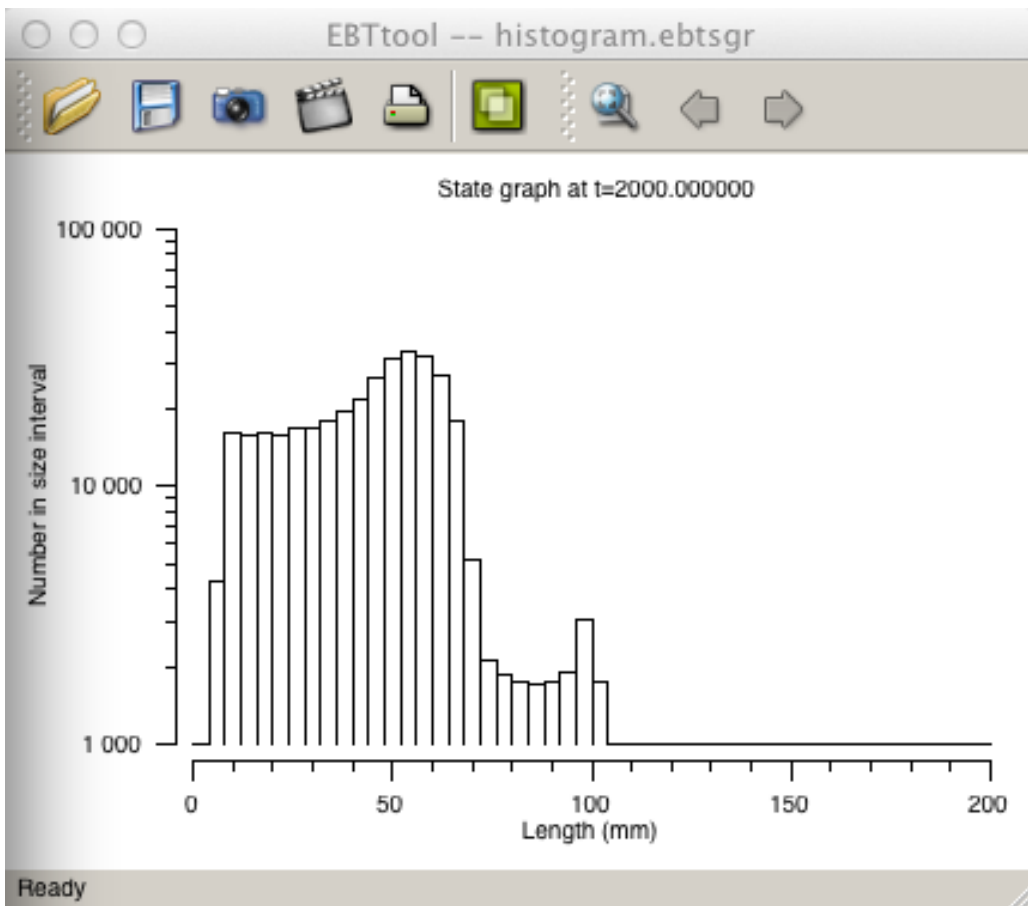


FIGURE 5 –

## 2.5 Affichage des résultats

Les différents résultats de la simulation s'affichent dans les trois fenêtres graphiques ouvertes à l'ouverture du projet.

**Etat** La fenêtre histogram montre la distribution en taille de la population au cours du temps. A la fin de la simulation, cette représentation s'arrête sur la dernière distribution calculée. Afin de revoir l'animation de l'ensemble de la simulation, il est nécessaire d'ouvrir le fichier de sortie d'état en se rendant dans le menu **State -> Open state output** et en sélectionnant le fichier **.csb** correspondant au run effectué. Vous verrez alors le nom du fichier d'état apparaître dans la fenêtre principale, accompagné d'une série de bouton (Fig. 6). Un boutons "play" permettant de rejouer l'animation, un bouton "pause", un menu déroulant permettant de sélectionner la vitesse de lecture, et un curseur permettant de se positionner à l'instant voulu de la simulation. L'animation est rejouée dans la fenêtre de graphique d'état.



FIGURE 6 –

**Sorties** Les deux autres fenêtres graphiques présentent des données de sortie du modèle. Les graphiques définis dans les fichiers **.ebtogr** montrent l'un le nombre d'adultes et de juvéniles, l'autre la quantité de ressources.

Ces fenêtres graphiques présentent plusieurs boutons (Fig. 7). Les premiers servent à ouvrir et sauvegarder le graphique dans le format EBT. L'appareil photo permet d'exporter une copie du graphique au format **.png** (pour l'inclure dans un rapport par exemple). Le clap permet d'exporter l'animation de l'état dans le cas du graphique d'état. La loupe permet de zoomer sur des portions du graphique. Enfin, le bouton vert permet d'éditer le graphique. C'est ce bouton qui permet de modifier les données affichées.

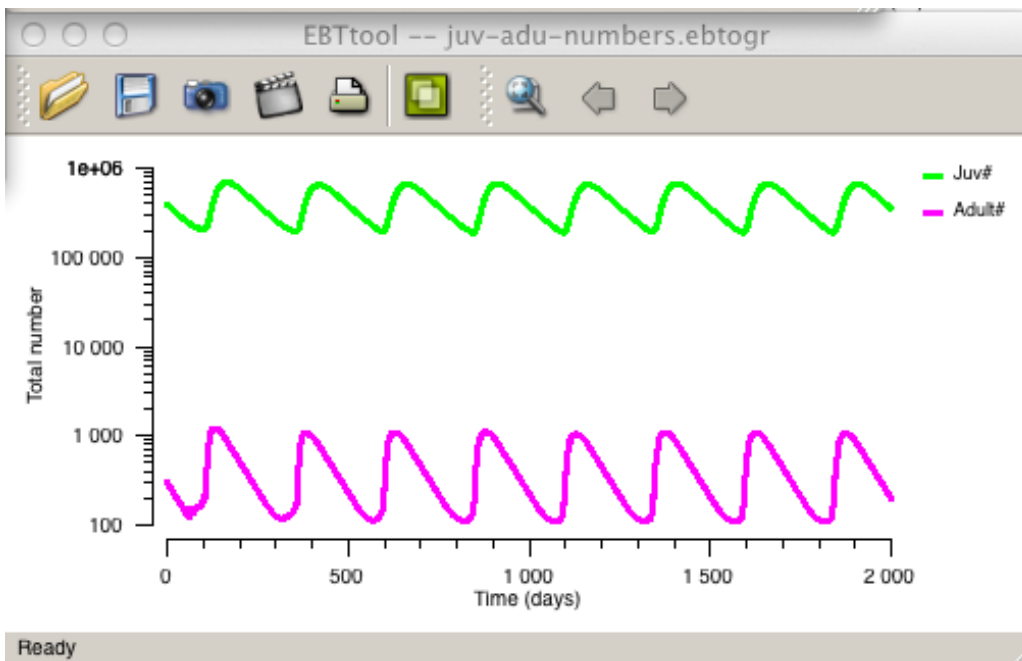


FIGURE 7 –

Si vous cliquez sur ce bouton vert, une fenêtre de configuration s'ouvre sur l'onglet "Sets" (Fig 8). Cette fenêtre permet de définir les données affichées par le graphique. La ligne "Data set" permet de choisir le set que l'on veut configurer, d'en créer un nouveau et d'effacer celui

sélectionné. La ligne "X from column" permet de choisir la colonne du fichier .out utilisé comme axe des X. La ligne "Y from column" permet de choisir la colonne utilisée comme axe des Y. La ligne "Legend label" permet de choisir la légende qui sera affichée. Pour afficher plusieurs courbes psur un même graphique, il faut définir plusieurs sets. On peut alors choisir les mêmes axes X et Y, ou choisir d'avoir deux axes X ou Y distincts en sélectionnant Primary ou Secondary dans les menus déroulants.

Les différentes colonnes du fichier .out sont définies à la fin du fichier source .c dans la section DefineOutput et correspondent à :

1. temps
2. ressources
3. longueur de la plus vieille cohorte
4. nombre d'individus dans la population totale
5. biomasse totale
6. nombre de juvéniles
7. biomasse des juvéniles
8. nombre d'adultes
9. biomasse des adultes
10. nombre d'individus de longueur  $>50$
11. biomasse des individus de longueur  $>50$
12. nombre de nouveaux nés total
13. nombre de nouveaux nés par adultes.

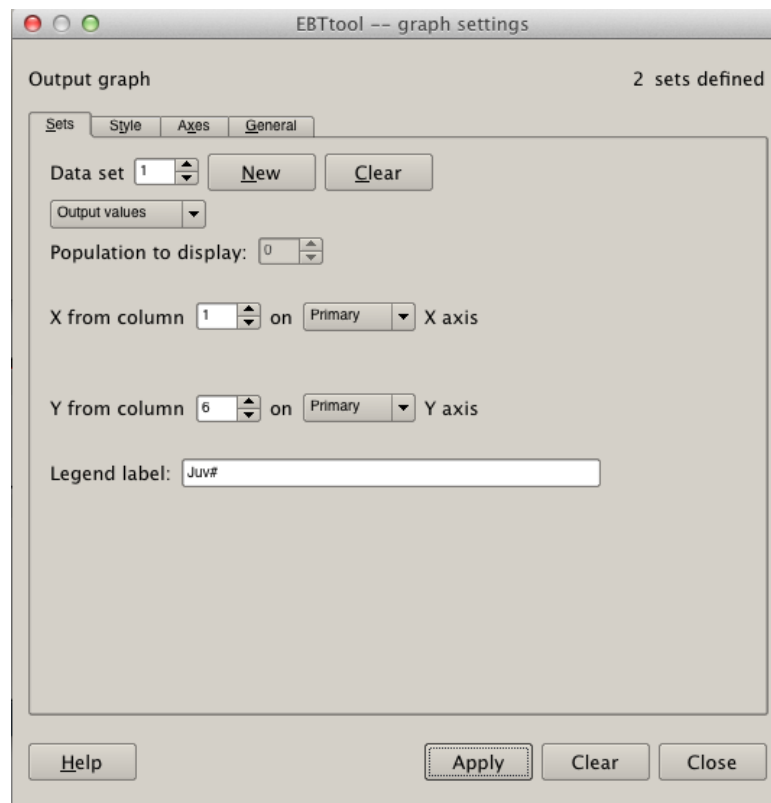


FIGURE 8 –



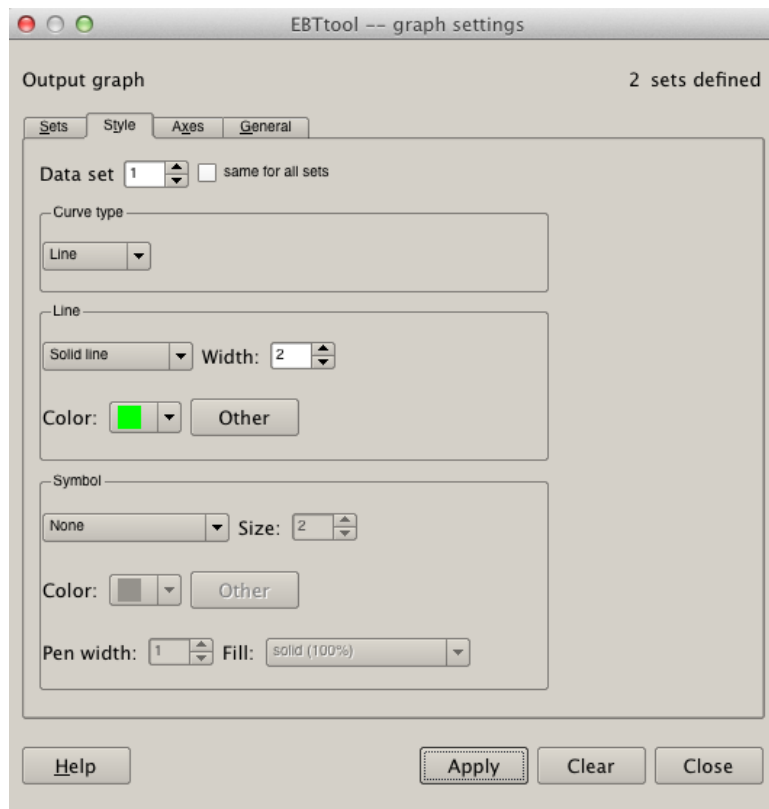



FIGURE 9 –

Dans l'onglet "Style", vous pouvez définir pour chaque set tracé, le type de représentation utilisé, le type et la couleur des lignes, la forme et la couleur des points (Fig. 9).

Dans l'onglet "Axes", vous pouvez préciser les axes utilisés, et notamment pour chacun des axes si l'échelle est linéaire ou logarithmique et si l'échelle est automatique ou fixe (Fig. 10).

Enfin, dans l'onglet "General", vous pouvez définir un titre de figure, et si vous souhaitez que les légendes apparaissent ou non (Fig. 11).

Si vous souhaitez conserver les résultats obtenus, vous pouvez soit exporter les graphiques en cliquant sur le bouton appareil photo, soit enregistrer le fichier `.out` sous un autre nom afin de ne pas l'effacer en cliquant sur le bouton poubelle à la prochaine exécution et de pouvoir le réutiliser plus tard.

Si vous souhaitez revoir les données d'un fichier `.out` précédemment enregistré, vous pouvez l'ouvrir en cliquant sur le bouton  de la fenêtre principale, et en sélectionnant votre fichier. Les données sont alors lues et affichées dans les fenêtres graphiques.

### 3 Utilisation avancée

Dans cette section, nous présentons quelques éléments d'utilisation avancée du logiciel, et notamment, la modification des paramètres, la modification des conditions initiales, l'export de l'état final en conditions initiales, la bifurcation et l'édition du code source.

#### 3.1 Modification des paramètres

Afin de simuler différentes dynamiques, vous pouvez modifier les paramètres d'entrée du modèle. Pour cela, sélectionnez `Run -> Edit -> CVF file`. Une fenêtre d'édition s'ouvre, où vous pouvez modifier votre fichier d'entrée du modèle (Fig. 12)

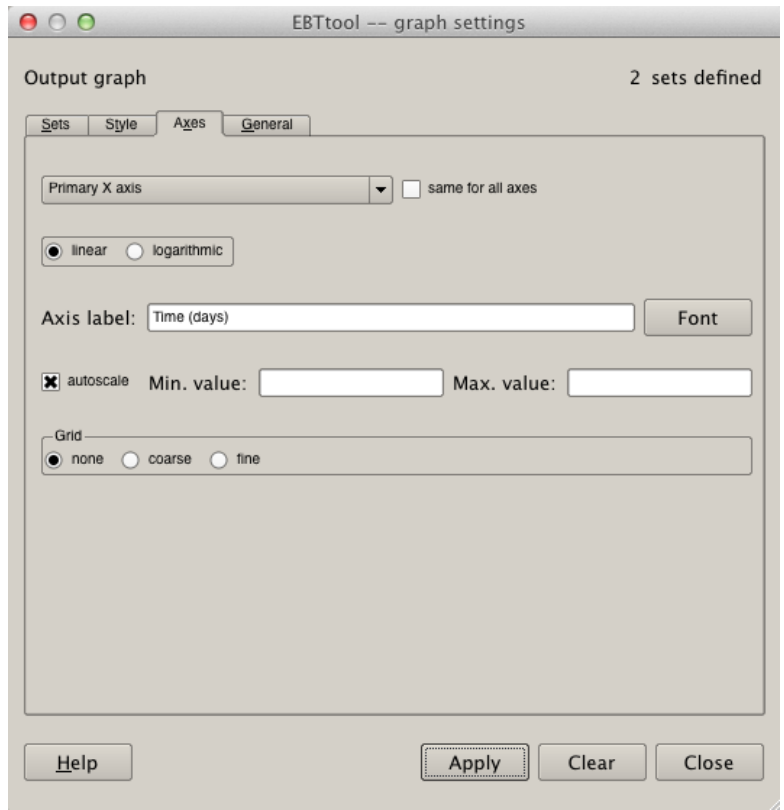


FIGURE 10 –

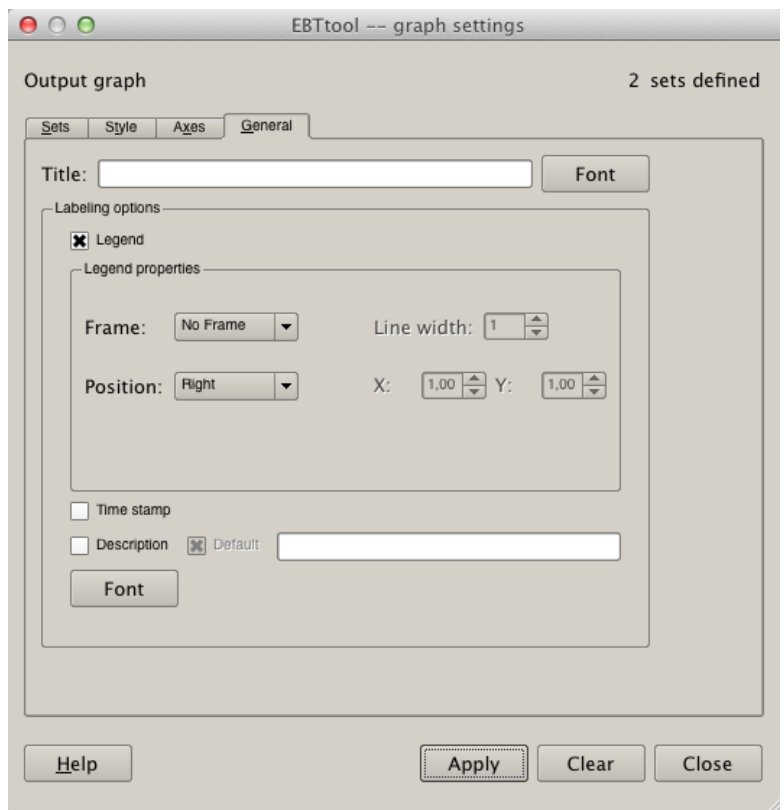


FIGURE 11 –

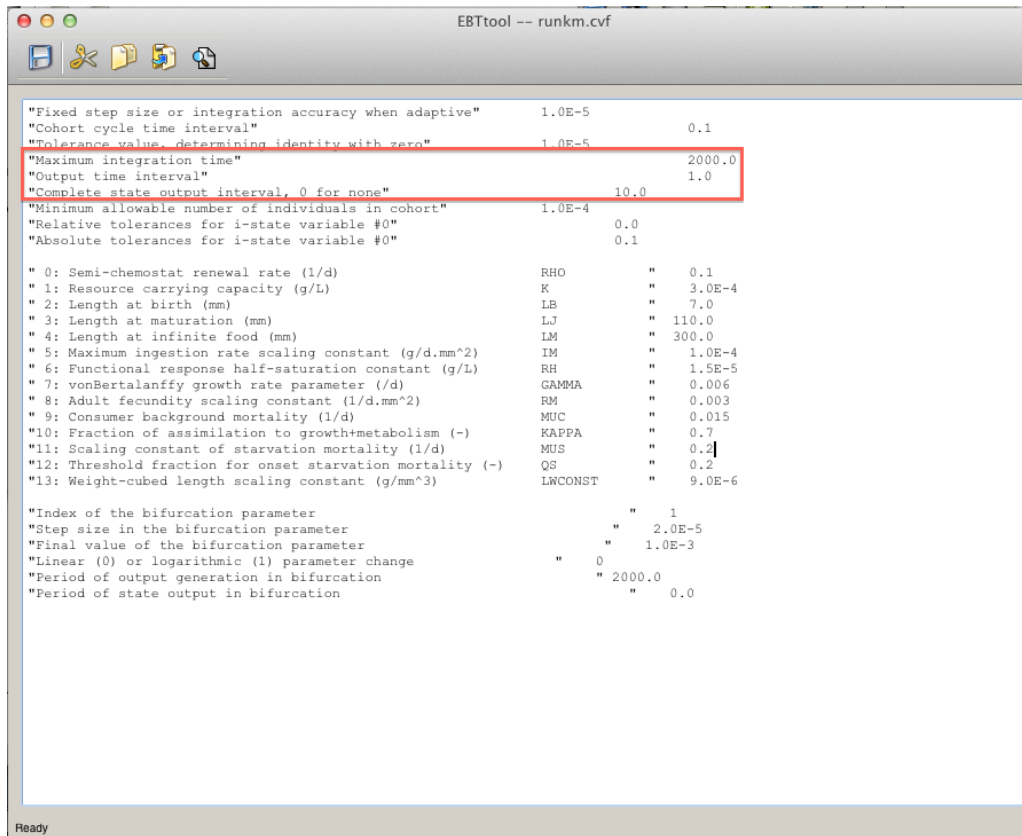


FIGURE 12 –

Le fichier de paramètre est séparé en trois blocs. Le premier bloc concernent les paramètres régulant l'intégration des équations du modèle. La plupart de ces paramètres permettent d'ajuster la précision de l'intégration, sauf exception vous n'aurez pas à les modifier. Cependant, trois de ces paramètres pourront être ajustés :

- "Maximum integration time" qui définit la durée de la simulation
- "Output time interval" qui définit l'intervalle entre deux écritures dans le fichier de sortie (.out)
- "Complete state output interval, 0 for none" qui définit l'intervalle d'écriture dans le fichier d'état (.csb). Mettre cette variable à 0 supprime l'animation durant l'exécution du modèle et ne crée aucun fichier .csb mais accélère sensiblement l'exécution du modèle.

Le second bloc contient les différents paramètres du modèle. Le troisième bloc contient les paramètres permettant contrôler les simulations de bifurcation, nous reviendront sur ce dernier plus loin.

**Remarque:** *Après édition du fichier, n'oubliez pas de le sauvegarder avant d'exécuter la simulation.*

### 3.2 Modification des conditions initiales

Les conditions initiales du run sont définies dans le fichier .isf. Pour les modifier, sélectionnez Run -> Edit -> ISF file. Une fenêtre d'édition s'ouvre, où vous pouvez modifier votre fichier de conditions initiales (Fig. 13)

Ce fichier est séparé en deux blocs. Le premier est constitué d'une ligne unique, et contient les conditions initiales des variables environnementales (ici le temps, puis la ressource).

Le second bloc contient une ligne pour chaque cohorte ajoutée au début de la simulation. Pour chaque cohorte, on définit dans l'ordre des colonnes :

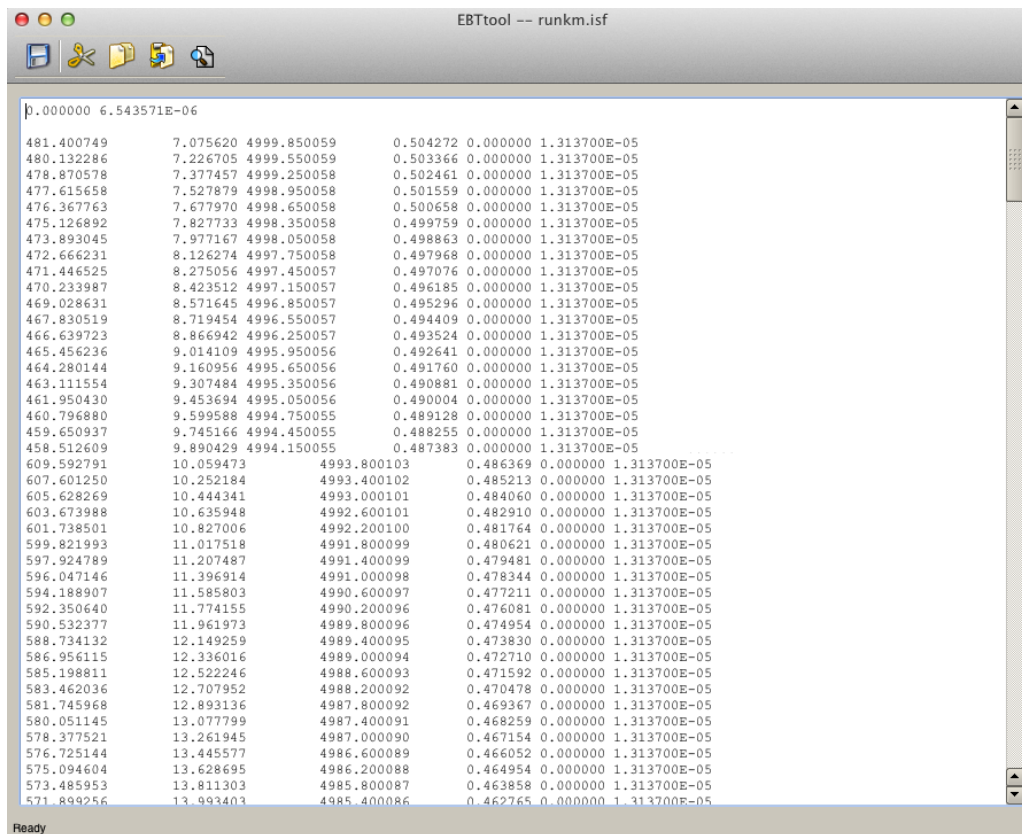


FIGURE 13 –

1. le nombre d'individus dans la cohorte
2. la taille des individus
3. ... une série de constantes dites d'identité ( $i\_const$  dans le code source) dont la valeur n'est pas très important pour les conditions initiales car sans incidence sur le fonctionnement du modèle.

**Remarque:** *Après édition du fichier, n'oubliez pas de le sauvegarder avant d'exécuter la simulation.*

### 3.3 Export de l'état final comme condition initial

EBTtool offre la possibilité d'exporter les conditions finales d'une simulation comme conditions initiales d'une nouvelle simulation. Pour réaliser cette opération, sélectionnez `Run -> Copy -> ESF to new ISF file...`. Une fenêtre s'ouvre alors où vous pouvez sélectionner l'emplacement de votre nouveau fichier `.isf`. Si vous voulez remplacer simplement vos conditions initiales, sélectionnez votre fichier `.isf` existant et confirmez le remplacement. Sinon, enregistrez votre nouveau fichier sous un autre nom (ex : `toto.isf`), et copiez votre fichier `.cvf` en lui donnant le même nom que votre nouveau fichier `.isf` (ex : `toto.cvf`). Pour lancer une simulation avec ces nouvelles conditions initiales, sélectionnez votre nouveau fichier `.cvf` dans `Run -> Open...`

### 3.4 Bifurcation

EBTtool offre la possibilité d'effectuer des pseudos bifurcations. Le principe est ici d'effectuer une série de simulations dont tous les paramètres (du second bloc dans le fichier `.cvf`) sauf

un seront fixés. Entre chacune de ces simulations, le paramètre non fixé (appelé paramètre de bifurcation) est incrémenté, l'état final de la simulation est utilisé comme nouvel état initial, et une nouvelle simulation est lancée avec la nouvelle valeur du paramètre de bifurcation.

Le paramètre de bifurcation est défini à la première ligne du troisième bloc du fichier de paramètre (Fig. 14). L'indice en question fait référence au numéro en tête de ligne dans le second bloc. La valeur initiale de la bifurcation est définie par la valeur donnée au paramètre dans le second bloc. L'incrément entre chaque simulation est défini à la deuxième ligne du troisième bloc, et la valeur finale à la troisième ligne. la ligne "Linear (0) or logarithmic (1) parameter change " permet de choisir si l'incrément est additif ou multiplicatif. La ligne "Period of output generation in bifurcation " permet de choisir le nombre de pas de temps enregistrés pour chaque simulation (*i.e.* pour chaque valeur du paramètre de bifurcation). Cette valeur doit être inférieure ou égale au temps d'intégration défini dans le premier bloc du fichier.

```

"Index of the bifurcation parameter           " 1
"Step size in the bifurcation parameter      " 2.0E-5
"Final value of the bifurcation parameter    " 1.0E-3
"Linear (0) or logarithmic (1) parameter change " 0
"Period of output generation in bifurcation  " 2000.0
"Period of state output in bifurcation      " 0.0

```

FIGURE 14 –

Afin d'indiquer à EBTtool que vous souhaitez effectuer une bifurcation, il est nécessaire de changer le paramètre BIFURCATION de 0 en 1 à la dernière ligne du header de votre modèle (fichier .h) en sélectionnant Model->Edit->Headerfile(.h) et de compiler votre modèle.

Lors d'une simulation de bifurcation, EBTtool ajoute automatiquement une colonne supplémentaire au fichier de sortie pour la valeur du paramètre de bifurcation.

### 3.5 Edition du code source

Le code source peut être édité directement dans EBTtool en sélectionnant Model -> Edit -> Program file (.c). Pour la structure du code source, se référer au poly correspondant. Editer le code source peut notamment permettre de modifier les sorties du modèle dans la section DefineOutput, ou de modifier les équations du modèle (ajout d'interactions, de prédation, modification des dépendances à la longueur, modification des réponses fonctionnelles...).

**Remarque:** *Il est vivement conseillé d'effectuer des modifications du code source sur une copie du fichier original afin de ne pas le perdre.*

*Pour rappel, les indices en C commencent **obligatoirement** à 0. Ainsi, toto[1] désignera toujours le **deuxième** élément du vecteur toto*