



HAL
open science

Mode d'emploi pour l'utilisation des modèles du projet Toolbox

Clarisse Boulenger, François Martignac, Marie Nevoux, Laurent Beaulaton

► **To cite this version:**

Clarisse Boulenger, François Martignac, Marie Nevoux, Laurent Beaulaton. Mode d'emploi pour l'utilisation des modèles du projet Toolbox. [Rapport Technique] OFB. 2018. hal-03612791

HAL Id: hal-03612791

<https://hal.science/hal-03612791>

Submitted on 18 Mar 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Avec le soutien financier de

**AGENCE FRANÇAISE
POUR LA BIODIVERSITÉ**
ÉTABLISSEMENT PUBLIC DE L'ÉTAT



Mode d'emploi pour l'utilisation des modèles du projet Toolbox

Rapport intermédiaire

Clarisse BOULENGER (INRA, Pole AFB-INRA)

François MARTIGNAC (INRA, Pole AFB-INRA)

Marie NEVOUX (INRA, Pole AFB-INRA)

Laurent BEAULATON (AFB, Pole AFB-INRA)

Janvier 2018

Document élaboré dans le cadre de :

FICHE ACTION INRA_2017_033_02

- **AUTEURS**

Clarisse BOULENGER, Ingénieure de recherche (INRA, Pôle AFB-INRA), clarisse.boulenger@inra.fr

François MARTIGNAC, Ingénieur de recherche (INRA, Pôle AFB-INRA), francois.martignac@inra.fr

Marie NEVOUX, Chargé de recherche (INRA, Pôle AFB-INRA), marie.nevoux@inra.fr

Laurent BEAULATON, Chef de pôle (AFB, Pôle AFB-INRA), laurent.beaulaton@afbiodiversite.fr

- **CORRESPONDANTS**

Agence française pour la biodiversité :

Laurent BEAULATON, Chef de pôle, laurent.beaulaton@afbiodiversite.fr

Droits d'usage : Accès restreint

Niveau géographique : National

Couverture géographique : France

Niveau de lecture : professionnels

- **MODE D'EMPLOI POUR L'UTILISATION DES MODELES DU PROJET TOOLBOX, C. BOULENGER ET AL.**

- **RESUME**

A travers la France, une cinquantaine de stations de comptage des migrateurs (STACOMI) a été mise en place pour obtenir des informations sur l'état qualitatif et quantitatif des populations de poissons migrateurs et leurs flux migratoires. Ces stations, bien que très informatives, ne permettent souvent d'obtenir qu'une observation partielle des flux de poissons (couverture du dispositif d'observation inférieure à la largeur/hauteur du cours d'eau, présence d'un bras de contournement ou d'un bief, efficacité du dispositif dépendante du débit ou de la turbidité). Les effectifs dénombrés sont donc plus ou moins différents des flux réels de poissons, et en l'absence d'une estimation des abondances, l'interprétation et la comparaison spatio-temporelle des données issues de ces stations de comptage est délicate. C'est dans ce contexte que notre travail s'inscrit. L'objectif de l'action AFB-INRA (projet Toolbox) est de proposer des outils pour estimer l'efficacité de détection des systèmes d'observation en place et l'abondance des poissons (stock en place ou de flux migrants). Pour cela, une approche combinant observations et modélisation a été développée. Dans un premier temps, une caméra acoustique ARIS est installée temporairement en complément de la station de comptage (à proximité immédiate). Après la période d'observation, les données issues des deux dispositifs de comptage (dispositif existant à la station de comptage et ARIS) seront couplées et utilisées dans un modèle hiérarchique bayésien pour estimer le flux de migration et la probabilité de détection du dispositif existant et de la caméra acoustique. Pour une utilisation facilitée des modèles mis en place pour cette approche, un mode d'emploi a été créé pour leurs potentiels utilisateurs.

- **MOTS CLES : MODE D'EMPLOI, FLUX DE MIGRATION, MODELES, TOOLBOX**

Table des matières

1. Contexte	1
2. Principe général des modèles d'estimation des flux de migrations	1
3. Le mode d'emploi pour l'utilisation des modèles	4
1 ^{ère} Etape : Téléchargement du dossier Toolbox	4
2 ^{ième} Etape : Installation des logiciels	4
3 ^{ième} Etape : Création et Mise en forme des données	5
4 ^{ième} Etape : Ouvrir le fichier R	7
5 ^{ième} Etape : Spécifier le répertoire des données	8
6 ^{ième} Etape : Téléchargement des fonctions	10
7 ^{ième} Etape : Vérification des données	11
8 ^{ième} Etape : Lancement du modèle	13
9 ^{ième} Etape : Analyse des résultats	16
10 ^{ième} Etape : Choix des résultats à utiliser (si utilisation du modèle variable) :	20

1. Contexte

A travers la France, une cinquantaine de stations de comptage des migrateurs (STACOMI) a été mise en place pour obtenir des informations sur l'état qualitatif et quantitatif des populations de poissons migrateurs et leurs flux migratoires. Ces stations, bien que très informatives, ne permettent souvent d'obtenir qu'une observation partielle des flux de poissons (couverture du dispositif d'observation inférieure à la largeur/hauteur du cours d'eau, présence d'un bras de contournement ou d'un bief, efficacité du dispositif dépendante du débit ou de la turbidité). Les effectifs dénombrés sont donc plus ou moins différents des flux réels de poissons, et en l'absence d'une estimation des abondances, l'interprétation et la comparaison spatio-temporelle des données issues de ces stations de comptage est délicate. C'est dans ce contexte que notre travail s'inscrit. L'objectif de l'action AFB-INRA (projet Toolbox) est de proposer des outils pour estimer l'efficacité de détection des systèmes d'observation en place et l'abondance des poissons (stock en place ou de flux migrants). Pour cela, une approche combinant observations et modélisation a été développée. Dans un premier temps, une caméra acoustique ARIS est installée temporairement en complément de la station de comptage (à proximité immédiate). Après la période d'observation, les données issues des deux dispositifs de comptage (dispositif existant à la station de comptage et ARIS) seront couplées et utilisées dans un modèle hiérarchique bayésien pour estimer le flux de migration et la probabilité de détection du dispositif existant et de la caméra acoustique. Pour une utilisation facilitée des modèles mis en place pour cette approche, un mode d'emploi a été créé pour leurs potentiels utilisateurs.

2. Principe général des modèles d'estimation des flux de migrations

Les modèles hiérarchiques bayésiens sont des outils très intéressants pour notre approche puisqu'ils permettent d'intégrer plusieurs sources de données mais aussi de faire des estimations lorsqu'il y a des données manquantes (ex : jour où le dispositif de comptage est en maintenance), ce qui nous permettra de faire des estimations des flux de migration pendant la période de couplage des deux dispositifs de comptage et lorsque seul le dispositif de comptage existant sera en place. De plus, cette approche de modélisation fournit un intervalle de confiance de l'abondance et des estimations de probabilités de détection donnant des informations sur la robustesse et la précision des estimations.

Compte tenu de la diversité des études de cas, nous avons besoin d'une structure générale facilement transposable. La structure générale du modèle (Fig.1) s'appuie sur les 4 différents cas qui peuvent être rencontrés quel que soit l'espèce ou le dispositif de comptage déjà en place:

Cas 1 : L'individu est vu par les deux dispositifs de comptage lors de son passage.

Cas 2 : L'individu est vu par le dispositif de comptage existant mais pas par la ARIS.

Cas 3 : L'individu n'est pas vu par le dispositif de comptage existant mais il est vu par la ARIS.

Cas 4 : L'individu n'est vu par aucun dispositif de comptage.

Pour utiliser cette structure de modèle et estimer l'abondance des poissons, deux hypothèses fortes ont été faites. La première hypothèse est que la distribution des poissons est homogène dans la section de rivière échantillonnée. La seconde hypothèse est que l'opérateur qui traite les données est efficace à cent pour cent, ce qui signifie qu'il compte tous les poissons qui passent devant le dispositif de comptage (existant ou temporaire) et identifie correctement l'espèce de chaque poisson compté.

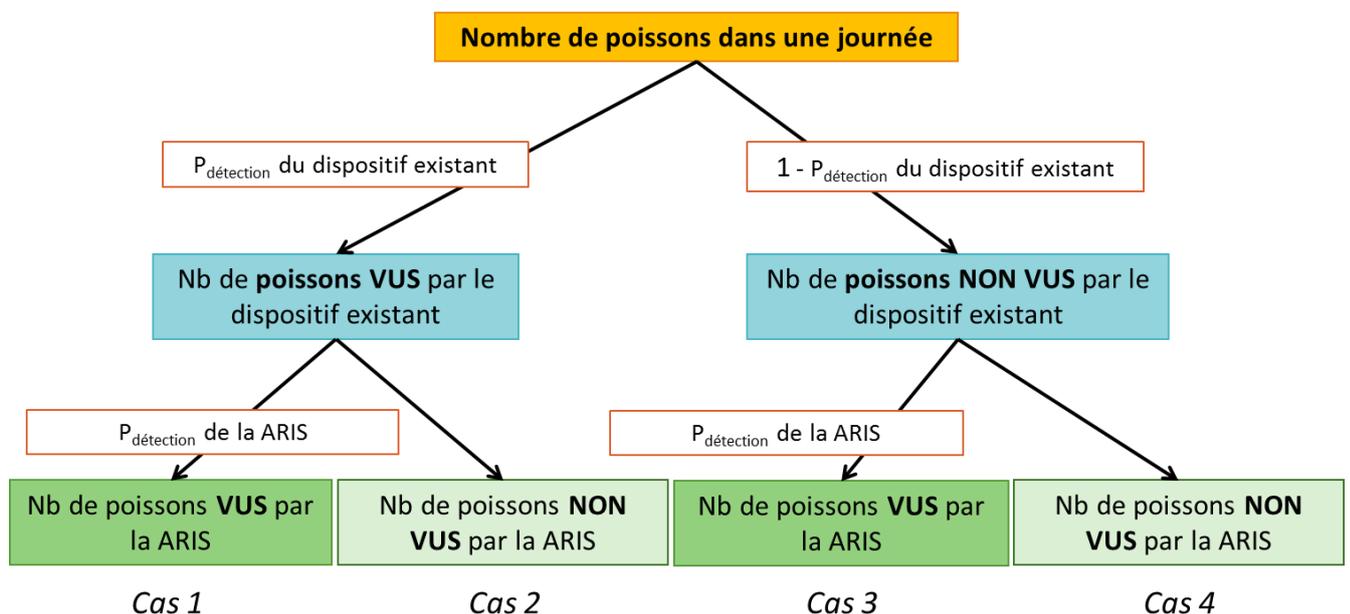


Figure 1 : Structure générale du modèle avec les 4 cas pouvant être observés dans le cadre de l'utilisation de deux dispositifs de comptage

Considérant notre approche, cette structure très généraliste a été modifiée pour s'adapter au protocole d'échantillonnage de la ARIS (Fig. 2). En effet, la ARIS enregistre les données en continu mais pour chaque espèce étudiée et rivière suivie, un filtre est mis en place pour diminuer le temps de traitements des données. Ce filtre peut prendre en compte la taille, la période de migration dans une journée, la vitesse et le sens de déplacement et d'autres paramètres qui permettent de sélectionner des séquences vidéos lors desquelles un objet ayant les caractéristiques choisies passe devant la caméra. Ces clips sont ensuite visualisés par un opérateur pour compter le nombre d'individus de l'espèce cible. Pour estimer l'efficacité du filtre, une fois par mois hors période de couplage et deux fois par mois pendant la période de couplage, un visionnage complet d'une journée (24H) est réalisé sans utiliser le filtre.

Les données de comptage sont donc décomposées ainsi :

- **Yexistant** : le nombre de poissons VUS par le dispositif existant
- **Ya1** : le nombre de poissons VUS par la ARIS (visualisation 24H) ET VUS par le dispositif existant
- **Ya2** : le nombre de poissons VUS par la ARIS (visualisation 24H) ET NON VUS par le dispositif existant
- **Yf1** : le nombre de poissons VUS par le filtre, VUS par la ARIS (visualisation 24H) ET VUS par le dispositif existant
- **Yf2** : le nombre de poissons VUS par le filtre, VUS par la ARIS (visualisation 24H) ET NON VUS par le dispositif existant

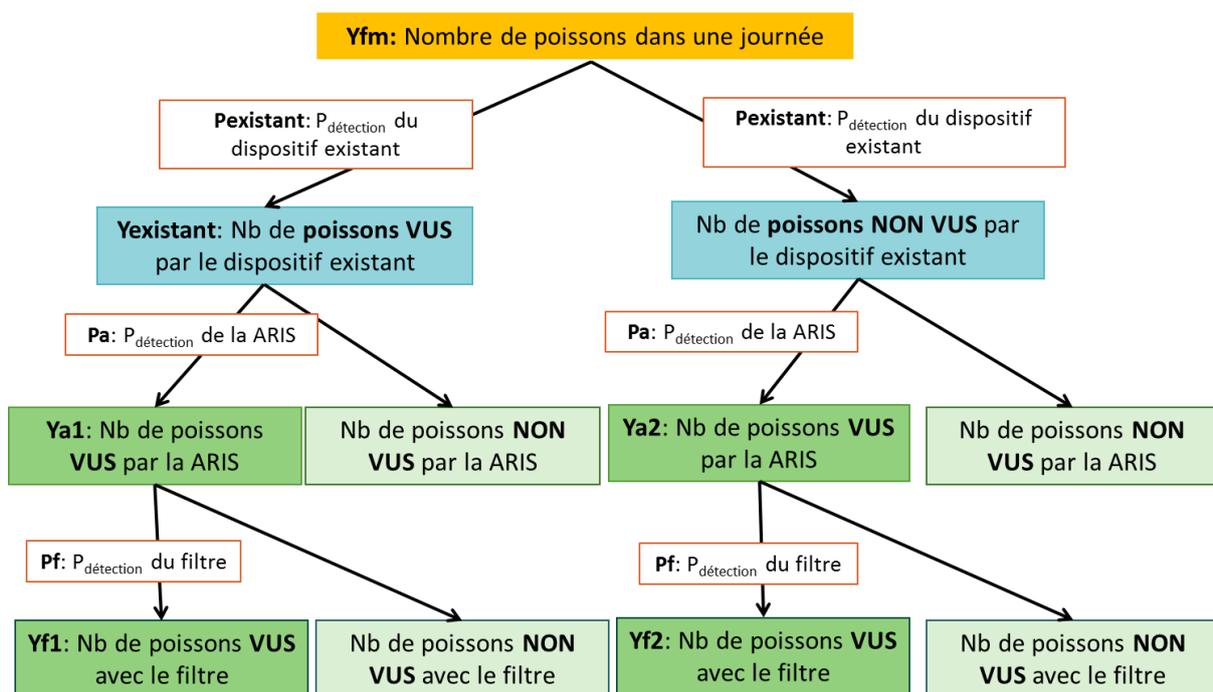


Figure 2 : Structure des modèles utilisés pour le projet Toolbox avec l'utilisation d'un filtre sur les données continues de la ARIS

Les paramètres d'intérêts (ceux que l'on cherche à estimer) sont :

- **Yfm** : le nombre de poissons passant dans une journée (le flux migratoire journalier)
- **Pexistant** : la probabilité de détection du dispositif de comptage existant
- **Pa** : la probabilité de détection de la ARIS
- **Pf** : la probabilité de détection du filtre de la ARIS
- **Ntot** : le nombre de poissons passant dans une année (le flux migratoire annuel)

Dans ce projet, deux types de modèles sont proposés. Dans le premier modèle (modèle fixe), on considère que toutes les probabilités de détection (existant, ARIS, filtre) sont fixes dans le temps : une seule estimation par probabilité de détection pour toute la période de suivi sera donc fournie. Dans le deuxième modèle (modèle variable), la probabilité de détection du

dispositif existant peut varier en fonction d'un facteur environnemental (variable avec un pas de temps journalier : turbidité, vitesse de courant, température, ...) alors que les probabilités de détection de la ARIS et du filtre restent fixes dans le temps. Le choix de tester un modèle variable reviendra à l'opérateur mais des résultats statistiques l'aideront à valider ou non l'utilisation de ce modèle par rapport à un modèle fixe.

3. Le mode d'emploi pour l'utilisation des modèles

Cette partie décrit les différentes étapes à suivre pour pouvoir utiliser les modèles et estimer les paramètres d'intérêts

1^{ère} Etape : Téléchargement du dossier Toolbox

Un dossier regroupant tous les outils nécessaires à l'utilisation des modèles a été créé. Il est téléchargeable sur le site du pole AFB-INRA Gest'Aqua ou par demande à Marie Nevoux : marie.nevoux@inra.fr . Après l'avoir téléchargé, placer le dans un répertoire facilement accessible. C'est dans ce dossier qu'il faudra placer le fichier de données et que les résultats s'enregistreront.

2^{ème} Etape : Installation des logiciels

Pour pouvoir utiliser les modèles et estimer les paramètres d'intérêts, il est nécessaire d'installer trois logiciels gratuits et open-source : **R**, **RStudio** et **JAGS**.

JAGS : permet de faire tourner les modèles hiérarchiques bayésiens

R et RStudio : permet de lancer toutes les procédures (vérification des données et lancement de JAGS), d'enregistrer et d'analyser les résultats issus des modèles.

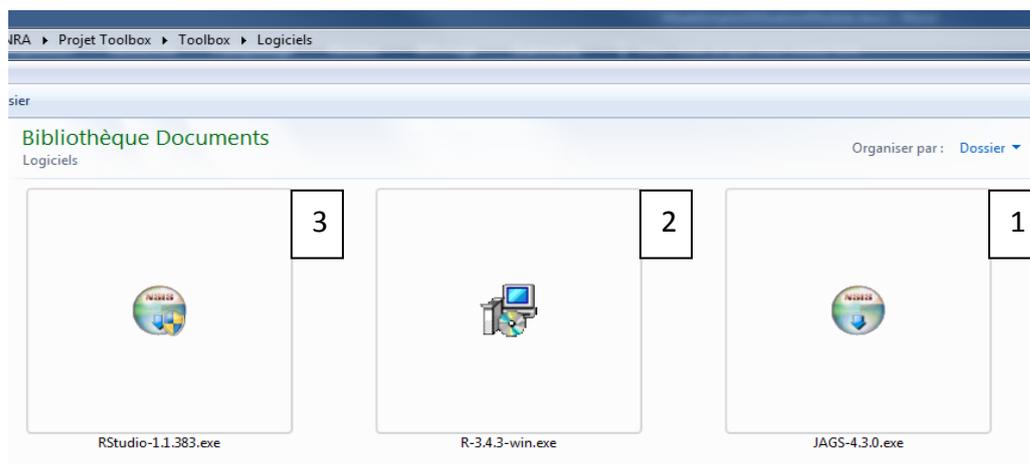


Figure 3 : Les exécutables et l'ordre d'installation des 3 logiciels à installer

Dans un premier temps installer **JAGS**, puis **R** et enfin **RStudio** grâce aux exécutables qui sont dans le sous dossier « Logiciels » du dossier « Toolbox » (Fig. 3). Pour cela il vous faut les droits administrateurs de votre ordinateur.

3^{ème} Etape : Création et Mise en forme des données

Pour pouvoir être utilisées, les données qui seront fournies aux modèles doivent avoir une mise en forme précise.

Dans le tableau de données, chaque ligne représente une journée de donnée (exceptée la première qui contient les noms de colonne). Le tableau doit être composé d'au moins 6 colonnes : « date », « Yexistant », « Ya1 », « Ya2 », « Yf1 », « Yf2 ». Dans le cas où l'on considère que la probabilité de détection du dispositif existant peut être dépendante d'une variable temporelle, une colonne « variable » doit être ajoutée au tableau (Fig. 4).

La colonne appelée « **date** » contient les dates d'échantillonnage dans l'ordre chronologique sous le format jj/mm/aaaa. Il est important que la série temporelle soit continue (Fig. 5).

	A	B	C	D	E	F	G	
1	date	Yexistant	Ya1	Ya2	Yf1	Yf2	variable	
2	01/01/2017	0	NA	NA		0	0	1,5
3	02/01/2017	0	NA	NA		0	0	1,6
4	07/01/2017	0	NA	NA		0	0	1,4
5	08/01/2017	0	NA	NA		0	0	1,6
6	09/01/2017	1	NA	NA		0	0	1,7
7	10/01/2017	0	NA	NA		0	0	1,8
8	11/01/2017	0	NA	NA		0	0	1,5

Figure 4 : Exemple de tableau de données à utiliser

Les colonnes appelées « **Yexistant** », « **Ya1** », « **Ya2** », « **Yf1** », « **Yf2** » contiennent les données de comptage décomposées (chaque groupe est défini dans la partie 2). Comme ce sont des données de comptage, ces colonnes ne peuvent contenir que des nombres entiers positifs. Il existe cependant des exceptions lorsqu'il n'y pas de données disponibles pour certaines dates pour l'un ou l'autre des dispositifs ou pour la variable environnementale. En effet, dans ces cas-là, on signifiera aux modèles qu'il n'y a pas d'information disponible en plaçant un « NA » dans la case correspondante.

	A		A
1	date	✓	date ✗
2	01/01/2017		01/01/2017
3	02/01/2017		02/01/2017
4	03/01/2017		07/01/2017
5	04/01/2017		08/01/2017
6	05/01/2017		09/01/2017
7	06/01/2017		10/01/2017
8	07/01/2017		11/01/2017

Figure 5 : Exemples de colonne « date » dans le tableau de données, à gauche format correct, à droite format non utilisable pour les modèles.

Exemples de cas où l'on utilisera un « NA » :

- Lorsque le dispositif de comptage existant est en panne, en maintenance ou ne fonctionne pas, on placera un « NA » dans la colonne « Yexistant » le jour correspondant (Fig. 6a)
- Lorsqu'il n'y pas de visualisation d'une journée complète des données de la ARIS, on placera un « NA » dans les colonnes « Ya1 » et « Ya2 » aux dates correspondantes (Fig. 6b)
- Lorsqu'il n'y a pas d'enregistrement avec la ARIS (hors période de couplage, en panne, ...), on placera un « NA » dans les colonnes « Ya1 », « Ya2 », « Yf1 » et « Yf2 » aux dates correspondantes (Fig. 6c)

a)	A	B	C	D	E	F	G	
1	date	Yexistant	Ya1	Ya2	Yf1	Yf2	variable	
2	01/01/2017		0 NA	NA		0	0	1,5
3	02/01/2017		0 NA	NA		0	0	1,6
4	07/01/2017		0 NA	NA		0	0	1,4
5	08/01/2017	NA		1	2	0	0	1,6
6	09/01/2017		1 NA	NA		0	0	1,7
7	10/01/2017		0 NA	NA		0	0	1,8
8	11/01/2017		0 NA	NA		0	0	1,5
b)	A	B	C	D	E	F	G	
1	date	Yexistant	Ya1	Ya2	Yf1	Yf2	variable	
2	01/01/2017		0 NA	NA		0	0	1,5
3	02/01/2017		0 NA	NA		0	0	1,6
4	07/01/2017		0 NA	NA		0	0	1,4
5	08/01/2017	NA		1	2	0	0	1,6
6	09/01/2017		1 NA	NA		0	0	1,7
7	10/01/2017		0 NA	NA		0	0	1,8
8	11/01/2017		0 NA	NA		0	0	1,5
c)	A	B	C	D	E	F	G	
1	date	Yexistant	Ya1	Ya2	Yf1	Yf2	variable	
236	27/08/2017		0 NA	NA	NA	NA		5,2
237	28/08/2017		0 NA	NA	NA	NA		5
238	29/08/2017		0 NA	NA	NA	NA		4,95
239	30/08/2017		0 NA	NA	NA	NA		3,95
240	31/08/2017		0 NA	NA	NA	NA		4,05
241	01/09/2017		1 NA	NA		0	0	4,07
242	02/09/2017		0 NA	NA		0	0	4,8
243	03/09/2017		1 NA	NA		0	0	4,8
244	04/09/2017		3 NA	NA		0	0	4,8

Figure 6 : Exemples de cas où l'on utilisera un « NA ».

La colonne appelée « **variable** » contient les valeurs de la variable environnementale que vous souhaitez tester. Cela doit être une variable quantitative ($\in (-\infty, +\infty)$) et il est possible d'utiliser des « NA » lorsqu'il n'y a pas de donnée disponible.

Le tableau de données ainsi créé doit être enregistré sous le format texte (séparateur : Tabulation) dans le dossier « Toolbox ».

Un exemple de tableau de données (sous format .txt et .xlsx) est fourni dans le sous-dossier « ExemplesDonnees » du dossier « Toolbox » (DataFunctiontest.xlsx et DataFunctiontest.txt).

4^{ème} Etape : Ouvrir le fichier R

Une fois que les données sont créées et enregistrées, RStudio va être utilisé pour lancer toutes les procédures. Pour cela, **ouvrir RStudio et ouvrir le fichier « AOuvrir.R »** qui se situe dans le dossier « Toolbox ». Pour cela, cliquez en haut à gauche sur « File » puis cliquez sur « Open File... » et sélectionner le fichier « AOuvrir.R » (Fig. 7).

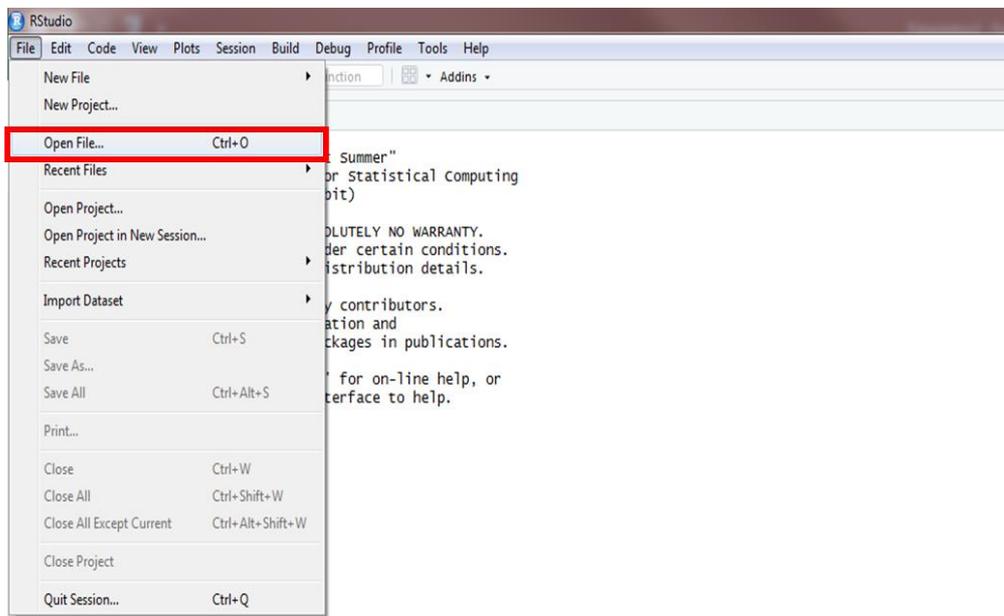


Figure 7 : Ouvrir un fichier sous RStudio

Le script R s'ouvre et va vous permettre de lancer les procédures qui suivent. Une fois que le script est ouvert, RStudio est composé de 4 blocs. Le bloc en haut à gauche contient le script avec toutes les procédures que vous allez lancer (1 dans la figure 8), le bloc en bas à gauche est la console, où vous pourrez suivre l'avancée de vos procédures (2 dans la figure 8), les deux blocs à droite de la fenêtre ne vous serviront pas (3 et 4 dans la figure 8).

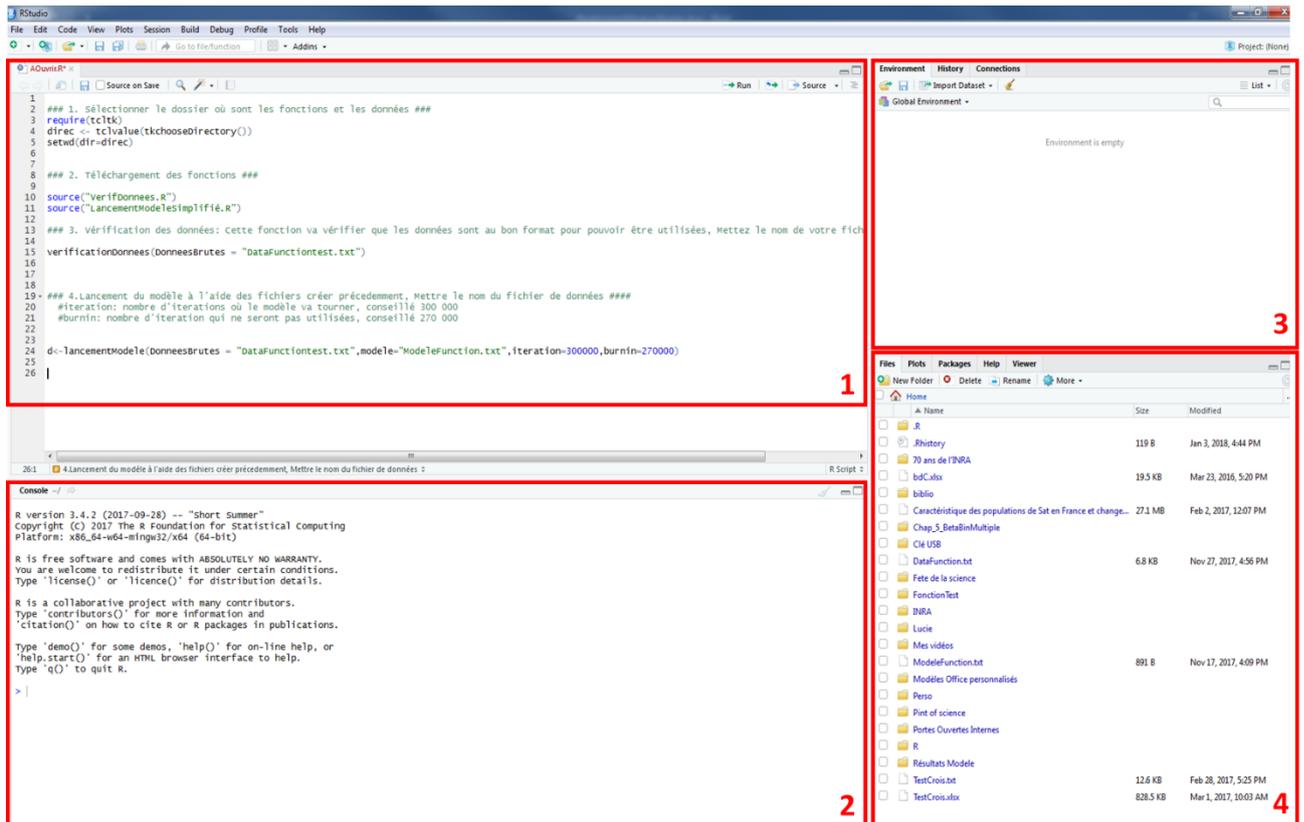


Figure 8 : Composition d'une fenêtre RStudio.

5^{ème} Etape : Spécifier le répertoire des données

Pour commencer, il faut spécifier où sont les données et les outils qui vont servir à faire tourner les modèles. Pour cela, dans le bloc 1, sélectionner à l'aide de la souris les 4 premières lignes du script et cliquez ensuite sur le bouton « Run » en haut à droite du script (Fig. 9).

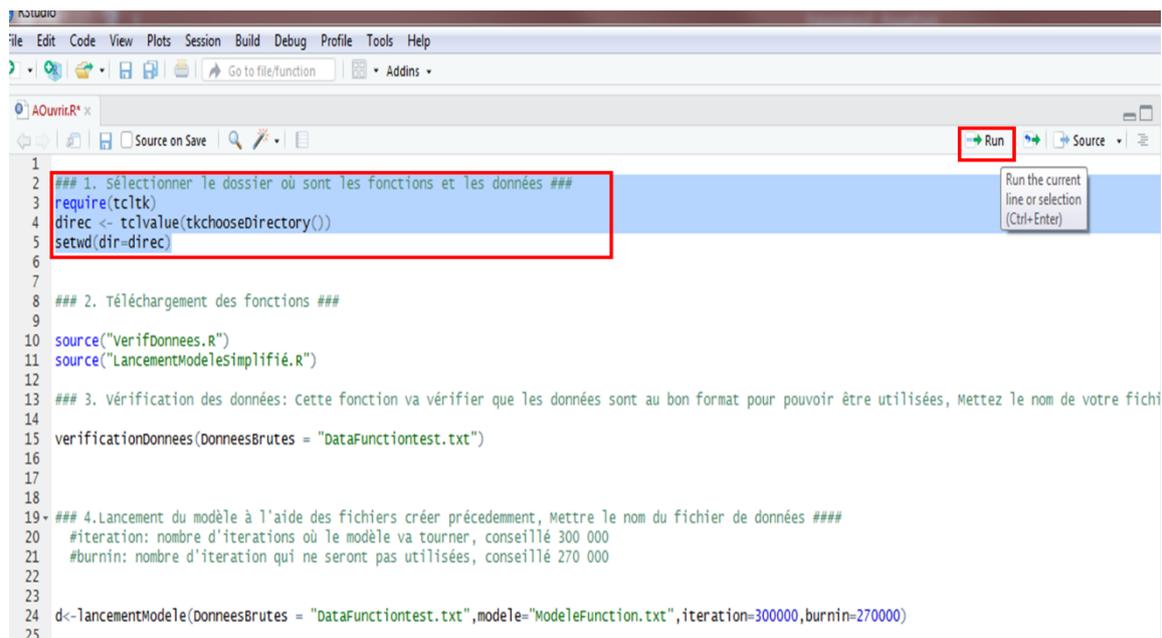


Figure 9 : Lancement des lignes du script permettant de sélectionner le répertoire du dossier « Toolbox ».

Une fois cette opération réalisée, vous verrez dans la console (bloc2) l'avancement des procédures (Fig. 10) et **une fenêtre « Sélectionner un dossier » doit s'ouvrir** (Fig. 11). Attention, il arrive régulièrement que cette fenêtre s'ouvre en arrière de votre fenêtre RStudio, n'hésiter donc pas à réduire votre fenêtre RStudio pour afficher la fenêtre « Sélectionner un dossier ». Dans cette fenêtre, **sélectionner le dossier Toolbox** que vous avez téléchargé et où vous avez placé les données précédemment créées (Fig. 11).

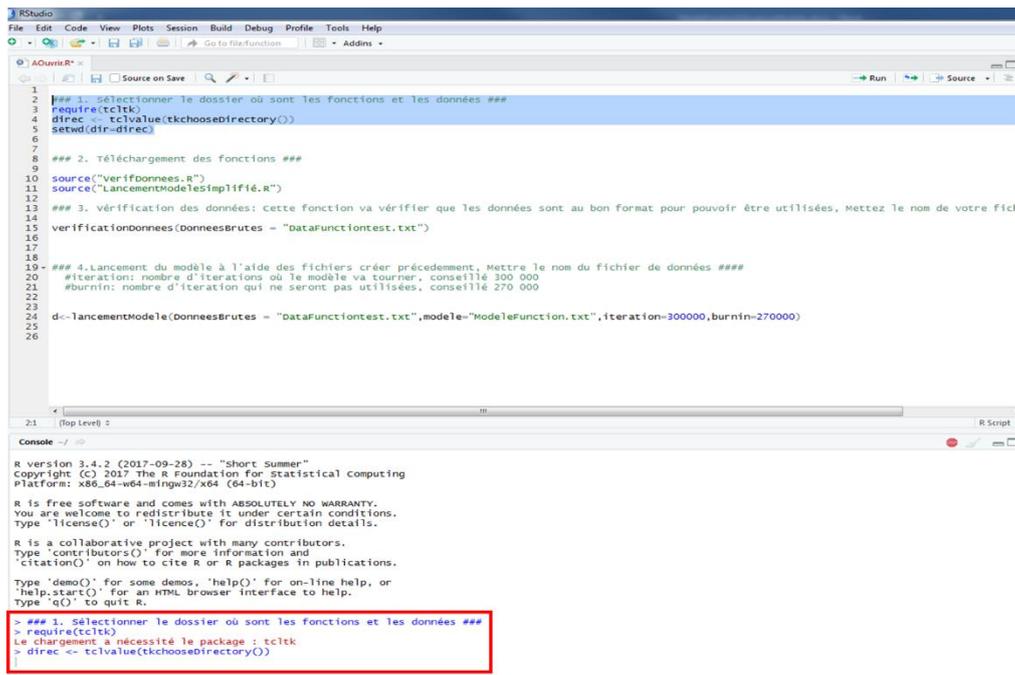


Figure 10 : Avancées des procédures dans la console

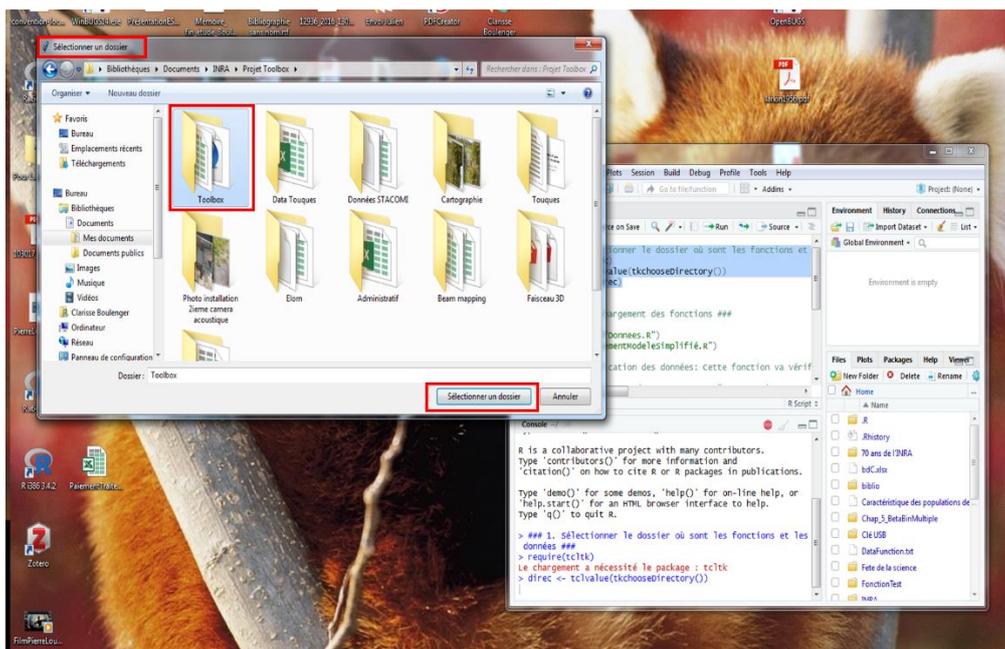


Figure 11 : Sélection du dossier Toolbox

Une fois que toutes les procédures ont été effectuées par RStudio, le signe « > » s'affiche sans ligne de code derrière signifiant que vous pouvez passer à l'étape suivante (Fig. 12).



```
Console ~/INRA/Projet Toolbox/Toolbox/

R version 3.4.2 (2017-09-28) -- "Short Summer"
Copyright (c) 2017 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

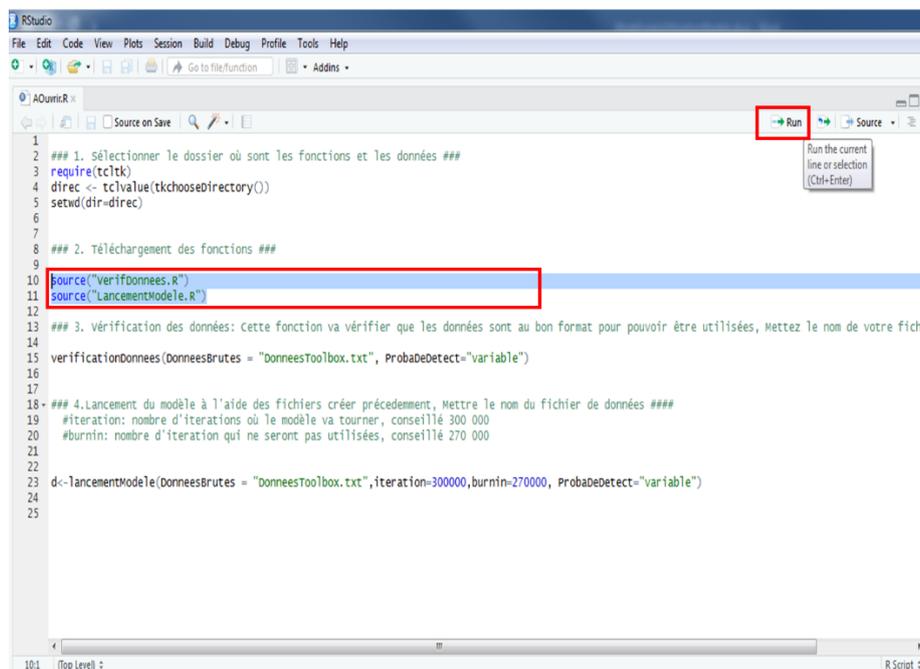
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> ### 1. Sélectionner le dossier où sont les fonctions et les données ###
> require(tcltk)
Le chargement a nécessité le package : tcltk
> direc <- tclvalue(tkchooseDirectory())
> setwd(direc)
>
```

Figure 12 : Fin des procédures lancées avec RStudio.

6^{ème} Etape : Téléchargement des fonctions

Pour pouvoir vérifier le format des données et lancer le modèle, deux fonctions R ont été créées. Afin de les utiliser, il faut les télécharger. Pour cela, **sélectionner les deux lignes situées sous « ### 2. Téléchargement des fonctions ### » dans le bloc 1 et appuyer sur le bouton « Run »** (Fig. 13). Une fois que toutes les procédures ont été effectuées par RStudio, passez à l'étape suivante.

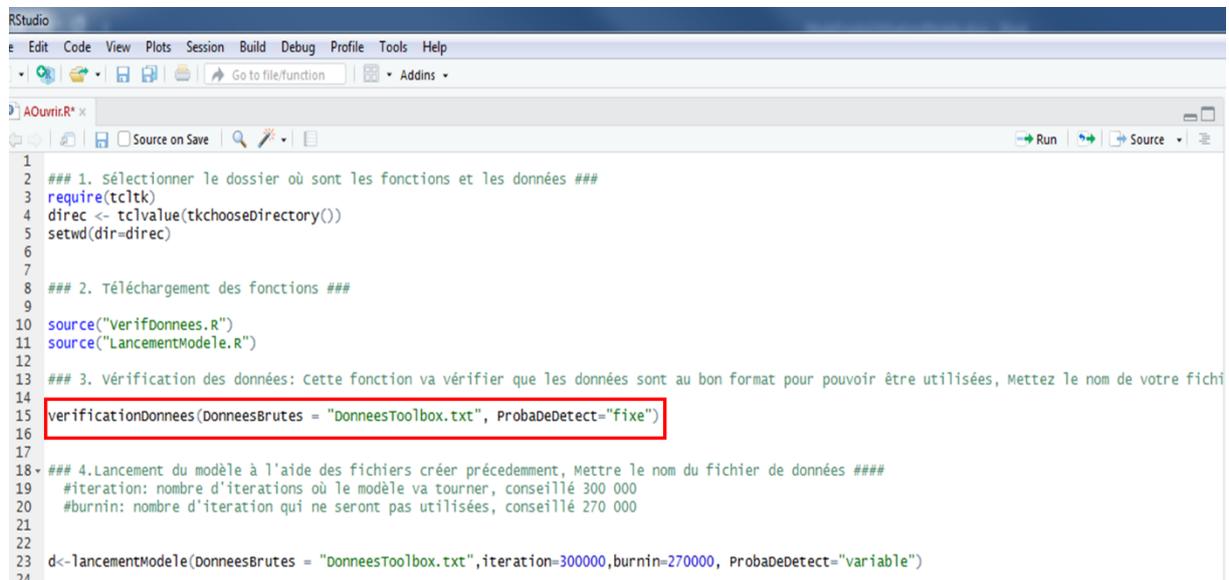


```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins
Source on Save
1
2 ### 1. Sélectionner le dossier où sont les fonctions et les données ###
3 require(tcltk)
4 direc <- tclvalue(tkchooseDirectory())
5 setwd(direc)
6
7
8 ### 2. Téléchargement des fonctions ###
9
10 source("Verifdonnees.R")
11 source("LancementModele.R")
12
13 ### 3. Vérification des données: cette fonction va vérifier que les données sont au bon format pour pouvoir être utilisées, Mettez le nom de votre fichi
14 verificationDonnees(DonneesBrutes = "DonneesToolbox.txt", ProbaDeDetect="variable")
15
16
17
18 ### 4.Lancement du modèle à l'aide des fichiers créer précédemment, Mettre le nom du fichier de données ###
19 #iteration: nombre d'iterations où le modèle va tourner, conseillé 300 000
20 #burnin: nombre d'iteration qui ne seront pas utilisées, conseillé 270 000
21
22
23 d<-lancementModele(DonneesBrutes = "DonneesToolbox.txt",iteration=300000,burnin=270000, ProbaDeDetect="variable")
24
25
```

Figure 13 : Sélection et lancement des lignes du script permettant le téléchargement des fonctions pour vérifier le format des données et lancer le modèle.

7^{ème} Etape : Vérification des données

Les données devant être dans un format spécifique pour pouvoir être utilisées, une fonction de vérification des données a été créée (la ligne sous « 3. Vérification des données », Fig. 14).

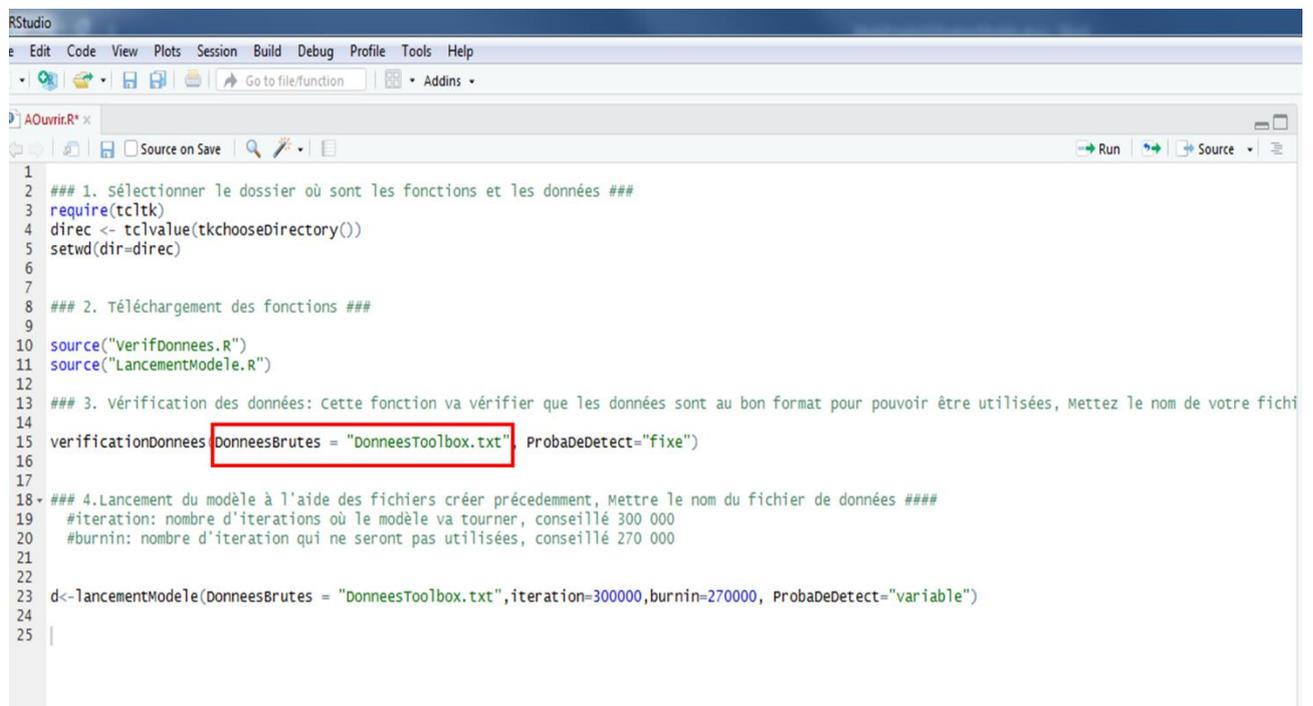


```
1
2 ### 1. Sélectionner le dossier où sont les fonctions et les données ###
3 require(tcltk)
4 direc <- tclvalue(tkchooseDirectory())
5 setwd(dir=direc)
6
7
8 ### 2. Téléchargement des fonctions ###
9
10 source("VerifDonnees.R")
11 source("LancementModele.R")
12
13 ### 3. Vérification des données: Cette fonction va vérifier que les données sont au bon format pour pouvoir être utilisées, Mettez le nom de votre fichier
14
15 verificationDonnees(DonneesBrutes = "DonneesToolbox.txt", ProbaDeDetect="fixe")
16
17
18 ### 4. Lancement du modèle à l'aide des fichiers créés précédemment, Mettre le nom du fichier de données ###
19 #iteration: nombre d'itérations où le modèle va tourner, conseillé 300 000
20 #burnin: nombre d'itérations qui ne seront pas utilisées, conseillé 270 000
21
22
23 d<-lancementModele(DonneesBrutes = "DonneesToolbox.txt",iteration=300000,burnin=270000, ProbaDeDetect="variable")
24
```

Figure 14 : Sélection et lancement de la fonction verificationDonnees.

Avant de l'utiliser, il vous faudra préciser :

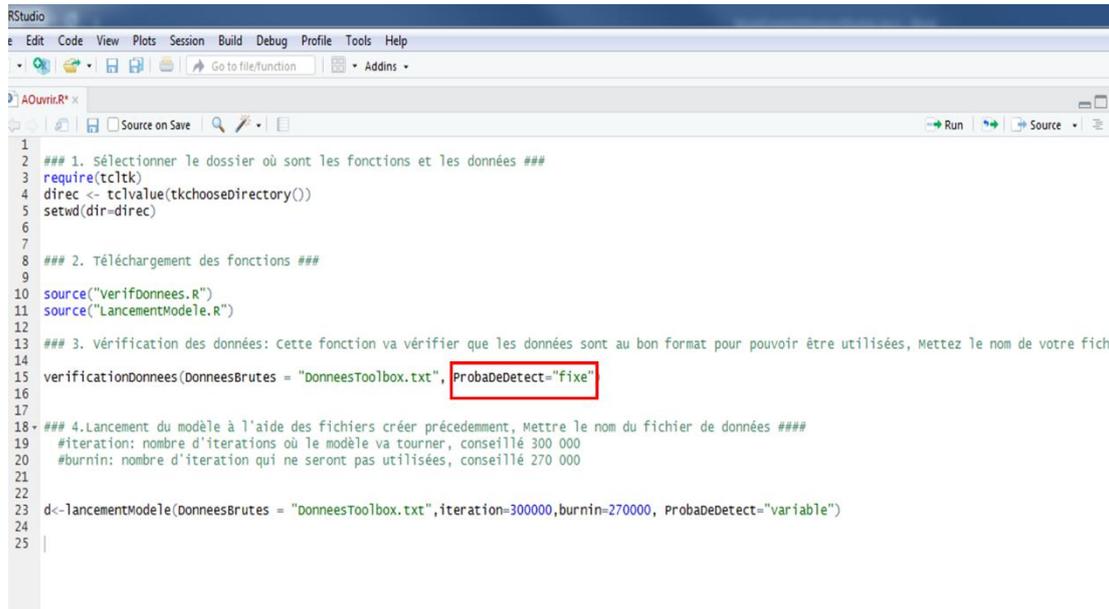
1. **Le nom du fichier contenant les données.** Pour cela après « DonneesBrutes= », préciser le nom de votre fichier comme ceci : "NomDuFichier.txt" (Fig. 15)



```
1
2 ### 1. Sélectionner le dossier où sont les fonctions et les données ###
3 require(tcltk)
4 direc <- tclvalue(tkchooseDirectory())
5 setwd(dir=direc)
6
7
8 ### 2. Téléchargement des fonctions ###
9
10 source("VerifDonnees.R")
11 source("LancementModele.R")
12
13 ### 3. Vérification des données: Cette fonction va vérifier que les données sont au bon format pour pouvoir être utilisées, Mettez le nom de votre fichier
14
15 verificationDonnees(DonneesBrutes = "DonneesToolbox.txt", ProbaDeDetect="fixe")
16
17
18 ### 4. Lancement du modèle à l'aide des fichiers créés précédemment, Mettre le nom du fichier de données ###
19 #iteration: nombre d'itérations où le modèle va tourner, conseillé 300 000
20 #burnin: nombre d'itérations qui ne seront pas utilisées, conseillé 270 000
21
22
23 d<-lancementModele(DonneesBrutes = "DonneesToolbox.txt",iteration=300000,burnin=270000, ProbaDeDetect="variable")
24
25
```

Figure 15 : Spécification du nom du fichier de données.

2. Si vous souhaitez tester le modèle fixe ou le modèle variable, après « ProbaDeDetec= », préciser "fixe" ou "variable" (Fig. 16).



```
1
2 ### 1. Sélectionner le dossier où sont les fonctions et les données ###
3 require(tcltk)
4 direc <- tclvalue(tkchooseDirectory())
5 setwd(direc)
6
7
8 ### 2. Téléchargement des fonctions ###
9
10 source("VerifDonnees.R")
11 source("LancementModele.R")
12
13 ### 3. Vérification des données: Cette fonction va vérifier que les données sont au bon format pour pouvoir être utilisées, Mettez le nom de votre fichier
14
15 verificationDonnees(DonneesBrutes = "DonneesToolbox.txt", ProbaDeDetec="fixe")
16
17
18 ### 4. Lancement du modèle à l'aide des fichiers créer précédemment, Mettre le nom du fichier de données ###
19 #iteration: nombre d'iterations où le modèle va tourner, conseillé 300 000
20 #burnin: nombre d'iteration qui ne seront pas utilisées, conseillé 270 000
21
22
23 d<-LancementModele(DonneesBrutes = "DonneesToolbox.txt", iteration=300000, burnin=270000, ProbaDeDetec="variable")
24
25
```

Figure 16 : Choix du modèle fixe ou variable.

Après ces précisions, **sélectionner à l'aide de la souris la ligne verificationDonnees du script et cliquez ensuite sur le bouton « Run »**. Les dates de début et de fin de la série temporelle vous seront précisées dans la console pour que vous puissiez les vérifier.

Dans le cas où **les données ne seraient pas au bon format**, deux types de messages écrits en rouge vont apparaitre dans la console (liste en Annexe) :

1. Les **Warning** : ils préviennent d'une erreur qui sera corrigée automatiquement lors du lancement du modèle mais qui doit être vérifiée par l'opérateur (Fig. 17). Le modèle pourra être lancé.
2. Les **Error** : ils préviennent d'une erreur qui empêchera le lancement du modèle, et qui doit être corrigée par l'opérateur (Fig. 17). Après la correction de l'erreur, il faudra relancer la fonction « verificationDonnees » du script pour vérifier qu'il n'existe pas d'autres erreurs.

```

Console ~/INRA/Projet Toolbox/Toolbox/
> verificationDonnees(DonneesBrutes = "DonneesToolbox.txt", ProbaDeDetect="fixe")
[1] "Date Debut= 2017-01-01 ,Date Fin= 2017-12-31"
[1] " Voir ligne.s= 4"
[1] "Les données sont au bon format"
warning message:
In verificationDonnees(DonneesBrutes = "DonneesToolbox.txt", ProbaDeDetect = "fixe") :
  Il existe des valeurs negatives dans la colonne yf1, elles seront remplacees par des 0 pour l'analyse. Voir ligne.s= 4

Console ~/INRA/Projet Toolbox/Toolbox/
> verificationDonnees(DonneesBrutes = "DonneesToolbox.txt", ProbaDeDetect="fixe")
[1] "Date Debut= 2017-01-01 ,Date Fin= 2017-12-31"
Error in verificationDonnees(DonneesBrutes = "DonneesToolbox.txt", ProbaDeDetect = "fixe") :
  Il manque la colonne de donnees yf2 ou le nom de votre colonne est incorrect

```

Figure 17 : Error et Warning dans la console lorsque la fonction verificationDonnees n'a pas validé le format des données

Si les données sont au bon format, cette information sera marquée dans la console. (Fig. 18).

```

Console ~/INRA/Projet Toolbox/Toolbox/
> ### 1. Sélectionner le dossier où sont les fonctions et les données ###
> require(tcltk)
> direc <- tclvalue(tkchooseDirectory())
> setwd(direc)
>
>
> ### 2. Téléchargement des fonctions ###
>
> source("VerifDonnees.R")
> source("LancementModele.R")
> verificationDonnees(DonneesBrutes = "DonneesToolbox.txt", ProbaDeDetect="fixe")
[1] "Date Debut= 2017-01-01 ,Date Fin= 2017-12-31"
[1] "Les données sont au bon format"
>

```

Figure 18 : Message de validation du format des données dans la console.

8^{ième} Etape : Lancement du modèle

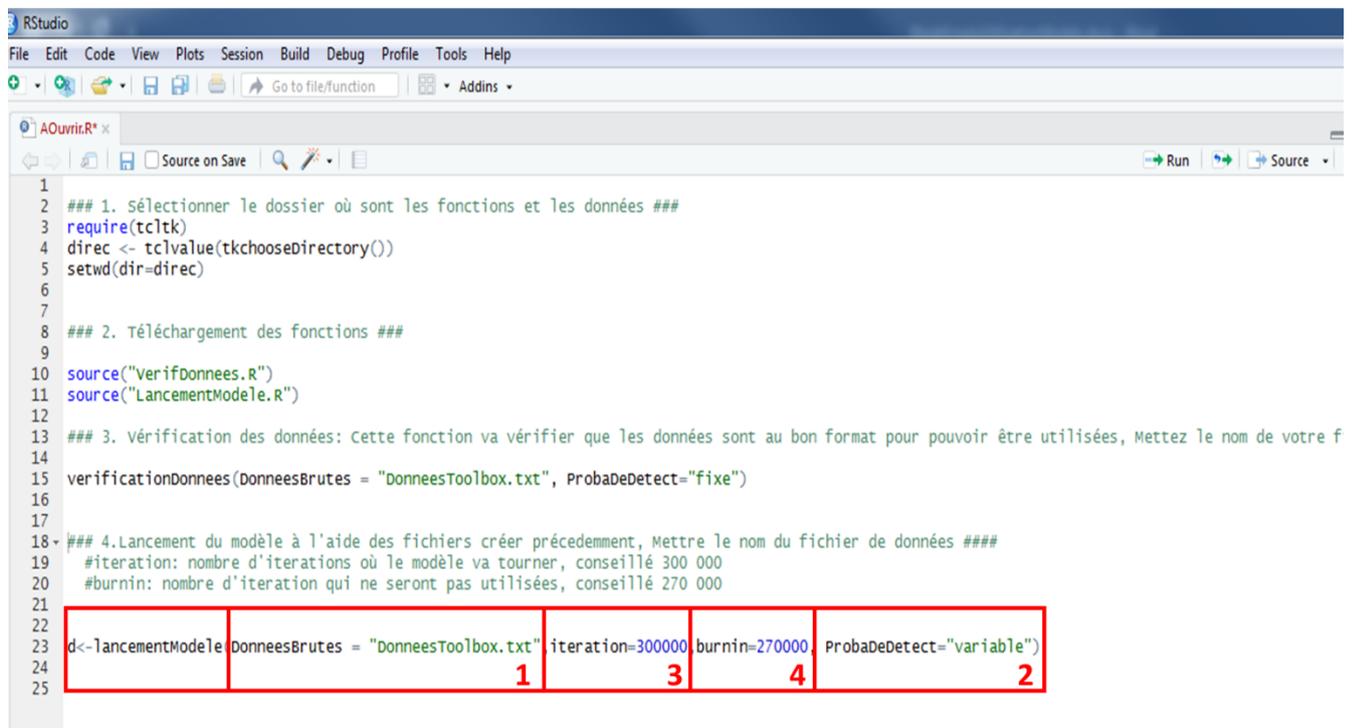
Lorsque les données ont été validées, le modèle pourra être utilisé grâce à la fonction « lancementModele » (Fig. 19). **Il est fortement conseillé d'utiliser dans un premier temps le modèle fixe et d'utiliser seulement ensuite le modèle variable si vous le souhaitez.**

Avant de l'utiliser, il vous faudra préciser :

1. **Le nom du fichier contenant les données.** Pour cela après « DonneesBrutes= », préciser le nom de votre fichier comme ceci : "NomDuFichier.txt" (Fig. 19, encart 1)
2. **Si vous souhaitez utiliser le modèle fixe ou le modèle variable**, après « ProbaDeDetec= », préciser "fixe" ou "variable" (Fig. 19, encart 2).
3. **Le nombre d'itérations** correspond au nombre de répétitions nécessaire pour que les résultats soient exploitables. En effet, pour que les résultats soient utilisables, il faut que le modèle converge, ce qui signifie que si l'on fait tourner plusieurs modèles en parallèle on obtient toujours les mêmes résultats après un certain nombre de répétitions. Dans notre approche, les simulations ont montré qu'il **fallait au moins**

300 000 itérations pour que le modèle converge. Si vous souhaitez modifier cette valeur, changer la valeur après « iteration= » (Fig. 19, encart 3).

4. **Le nombre de burnin** correspond au nombre de répétitions qui seront « jetées » (non utilisée pour les estimations). Pour interpréter les résultats du modèle, il ne faut garder que les répétitions lors desquelles le modèle a convergé. Dans notre approche, les simulations ont montré qu'il fallait « jeter » **au moins 270 000 itérations pour que le modèle converge. Si vous souhaitez modifier cette valeur, changer la valeur après « burnin= »** (Fig. 19, encart 4). **Il faudra que la différence entre le nombre d'itérations et le nombre de burnin soit de 30 000** pour pouvoir estimer correctement les paramètres d'intérêts et les intervalles de confiance autour des estimations.



```
1
2 ### 1. sélectionner le dossier où sont les fonctions et les données ###
3 require(tcltk)
4 direc <- tclvalue(tkchooseDirectory())
5 setwd(dir=direc)
6
7
8 ### 2. Téléchargement des fonctions ###
9
10 source("VerifDonnees.R")
11 source("LancementModele.R")
12
13 ### 3. vérification des données: Cette fonction va vérifier que les données sont au bon format pour pouvoir être utilisées, Mettez le nom de votre f
14
15 verificationDonnees(DonneesBrutes = "DonneesToolbox.txt", ProbaDeDetect="fixe")
16
17
18 ### 4.Lancement du modèle à l'aide des fichiers créer précédemment, Mettre le nom du fichier de données ###
19 #iteration: nombre d'itérations où le modèle va tourner, conseillé 300 000
20 #burnin: nombre d'iteration qui ne seront pas utilisées, conseillé 270 000
21
22
23 d<-lancementModele DonneesBrutes = "DonneesToolbox.txt" iteration=300000 burnin=270000, ProbaDeDetect="variable")
24
25
```

Figure 19 : Décomposition de la fonction lancementModele

Une fois ces précisions faites, **sélectionner à l'aide de la souris la ligne lancementModele du script et cliquez ensuite sur le bouton « Run »** (En dessous de «### 4. Lancement du Modèle..... »).

Dans la console, un certain nombre de procédures vont se lancer :

1. RStudio va télécharger automatiquement des fonctions qui seront nécessaires,
2. Un sous-dossier « Debug » va être créé dans le dossier « Toolbox » avec des fichiers qui pourront être envoyés au responsable du projet dans le cas où les procédures ne fonctionnent pas,
3. La fonction va paramétrer le modèle,
4. Vous verrez ensuite l'avancée des itérations par des barres d'avancement, d'abord l'*adaptive phase*, puis la *burnin phase* et enfin le *sampling* (Fig. 20),

5. La fonction va ensuite calculer les estimations pour chaque paramètre du modèle.

L'ensemble de ces procédures peut prendre entre 1h et 2h selon la puissance de votre ordinateur. Pendant ce temps, ne fermer pas RStudio ou votre ordinateur sinon il n'y aura pas d'enregistrement des résultats.

```
Console ~/INRA/Projet Toolbox/Toolbox/ ↵
> setwd(dir=dir)
> d<-lancementModele(DonneesBrutes = "DonneesToolbox.txt",iteration=300,burnin=270, ProbaDeDetect="fixe")

Processing function input.....

Done.

Compiling model graph
  Resolving undeclared variables
  Allocating nodes
Graph information:
  Observed stochastic nodes: 679
  Unobserved stochastic nodes: 1540
  Total graph size: 4919

Initializing model

Adaptive phase, 100 iterations x 3 chains
If no progress bar appears JAGS has decided not to adapt
|+++++| 100%

Burn-in phase, 270 iterations x 3 chains
|*****| 100%

Sampling from joint posterior, 30 iterations x 3 chains

calculating statistics.....
```

Figure 20 : Avancement du modèle dans la console

Si jamais le modèle n'a pas convergé, un message d'erreur va s'afficher dans la console (Fig. 21). Les résultats ne seront alors pas enregistrés, **il faudra relancer la procédure** avec un nombre d'itérations supérieur à celui utilisé (il faudra aussi augmenter le nombre de burnin pour avoir une différence égale à 30 000)

```
Console ~/INRA/Projet Toolbox/Toolbox/ ↵
Compiling model graph
  Resolving undeclared variables
  Allocating nodes
Graph information:
  Observed stochastic nodes: 679
  Unobserved stochastic nodes: 1540
  Total graph size: 4919

Initializing model

Adaptive phase, 100 iterations x 3 chains
If no progress bar appears JAGS has decided not to adapt
|+++++| 100%

Burn-in phase, 270 iterations x 3 chains
|*****| 100%

Sampling from joint posterior, 30 iterations x 3 chains

Calculating statistics.....
Warning in process.output(samples, DIC = DIC, codaonly, verbose = verbose) :
  At least one Rhat value could not be calculated.

Done.
Error in lancementModele(DonneesBrutes = "DonneesToolbox.txt", iteration = 300, :
  Problème de convergence, les résultats ne sont interprétables: relancer la fonction avec un nombre d itérations et un nombre de burnin plus importants. R
appel: valeur conseillée iteration= 300 000 et burnin=270 000
>
```

Figure 21 : Message d'erreur dans la console lorsque le modèle n'a pas convergé.

Si le modèle a convergé, un sous-dossier « Resultats Modele fixe » dans le cas de l'utilisation du modèle fixe ou « Résultats Modele variable » dans le cas de l'utilisation du modèle variable » a été créé dans le dossier « Toolbox ».

9^{ème} Etape : Analyse des résultats

Dans ces dossiers, des tableaux de résultats et des graphiques ont été enregistrés. Les tableaux sont enregistrés sous format .txt (séparateur : tabulation). Vous pouvez les ouvrir avec un bloc-notes, Excel ou encore open office. **Pour les interprétations des résultats, il est conseillé de se servir des estimations de la médiane plutôt que de celles de la moyenne** car les valeurs de la médiane sont les plus proches des modes (dans une distribution statistique, le mode est la valeur de la variable à laquelle est associée la plus grande fréquence).

Certains tableaux et graphiques seront réalisés que vous ayez choisi de faire tourner un modèle fixe ou un modèle variable :

1. **FluxMigraJour.txt** : Dans ce fichier, vous trouverez **les estimations journalières du flux migratoire** (le nombre de poissons migrant dans une journée). Il y a une estimation par jour de la moyenne, de la médiane et des intervalles de confiance à 2.5% et à 97.5% (IC2.5 et IC97.5, respectivement). Ce sont les valeurs de la médiane et des intervalles de confiance qui serviront pour vos interprétations (Fig. 22)

	A	B	C	D	E	F	G
1	date	Moyenne	IC2.5	Mediane	IC97.5		
2	01/01/2017	0.612222222	0	0	2		
3	02/01/2017	0.612222222	0	0	3		
4	03/01/2017	1.792222222	1	1	5		
5	04/01/2017	0.462222222	0	0	2		
6	05/01/2017	0.727777777	0	0	3		
7	06/01/2017	0.537777777	0	0	2		
8	07/01/2017	0.772222222	0	1	3		
9	08/01/2017	0.747777777	0	1	3		
10	09/01/2017	1.634444444	1	1	4		
11	10/01/2017	0.741111111	0	1	3		
12	11/01/2017	0.576666666	0	0	3		
13	12/01/2017	0.568888888	0	0	3		
14	13/01/2017	1.396666666	1	1	3.524999999999998		
15	14/01/2017	0.492222222	0	0	2		
16	15/01/2017	0.572222222	0	0	3		
17	16/01/2017	0.436666666	0	0	3		
18	17/01/2017	1.676666666	1	1	6		
19	18/01/2017	0.452222222	0	0	2		
20	19/01/2017	0.988888888	0	1	4		

Figure 22 : Tableau de résultats « FluxMigraJour.txt » ouvert avec Excel

2. **VariablesInterets.txt** (Fig. 23): Dans ce fichier, vous trouverez **les estimations faites pour les paramètres d'intérêt, c'est-à-dire le flux migratoire total (Ntot), la probabilité de détection du dispositif de comptage existant fixe ou variable (Pexistant), la probabilité de détection de la ARIS (Pa) et la probabilité de détection du filtre (Pf)**. Il y a une estimation par paramètre de la moyenne, de la médiane et des intervalles de confiance à 2.5% et à 97.5% (IC2.5 et IC97.5, respectivement). Excepté quand le modèle variable a été utilisé, on a alors une estimation par jour de la probabilité de détection du dispositif de comptage existant (les estimations pour ce paramètre sont dans l'ordre chronologique, Fig. 23b) Ce sont les valeurs de la médiane et des intervalles de confiance qui serviront pour vos interprétations.

	A	B	C	D	E	F	a)	A	B	C	
1	Nom	Moyenne	IC2.5	Mediane	IC97.5			1	Nom	Moyenne	IC2.5
2	Flux Migratoire total	291.189011111111		252	288	348		2	Flux Migratoire total	281.798888888889	
3	Probabilité de détection Appareil existant	0.691403261652501	0.5638326242795	0.693657800826025	0.807523771599546			3	Probabilité de détection Aris	0.740227460607527	0.4642438749
4	Probabilité de détection Aris	0.713127714472793	0.54007329480963	0.716588668844687	0.866682791898473			4	Probabilité de détection Filtre	0.160120432036996	0.1057432831
5	Probabilité de détection Filtre	0.156532774402569	0.10111230464031	0.153781697093761	0.228012029083696			5	Probabilité de détection Appareil existant	0.26579613444157	0.0089631445
6								6	Probabilité de détection Appareil existant	0.253635298194806	0.0105058707
7								7	Probabilité de détection Appareil existant	0.384316370130993	0.0647550974
8								8	Probabilité de détection Appareil existant	0.296419276557771	0.0135634320
9								9	Probabilité de détection Appareil existant	0.277870527503587	0.0152324240
10								10	Probabilité de détection Appareil existant	0.283581396784746	0.0130306427
11								11	Probabilité de détection Appareil existant	0.259121532310381	0.0102656448
12								12	Probabilité de détection Appareil existant	0.281043642462112	0.0145289632
13								13	Probabilité de détection Appareil existant	0.39568228360748	0.0733420415
14								14	Probabilité de détection Appareil existant	0.251686284583253	0.0155476557
15								15	Probabilité de détection Appareil existant	0.292346854575347	0.0148687211
16								16	Probabilité de détection Appareil existant	0.275152089037157	0.0158979851
17								17	Probabilité de détection Appareil existant	0.422953724445005	0.0672550708
18								18	Probabilité de détection Appareil existant	0.288989825413818	0.0212457417

Figure 23 : Tableau de résultats « VariableInterets.txt » ouvert avec excel lorsque a) le modèle fixe a été utilisé, b) le modèle variable a été utilisé.

3. **Resultats.txt** (Fig. 24): Dans ce fichier, vous trouverez **les estimations faites pour les tous les paramètres du modèle**. Il y a une estimation par paramètre de la moyenne (mean), de l'écart type (sd), de la médiane (q50) et des intervalles de confiance à 2.5%, 25%, 75 % et à 97.5% (q2.5, q25, q75 et q97.5, respectivement). Il y a également une estimation du Rhat qui est un paramètre de diagnostic de convergence du modèle. En effet, si l'un des paramètres à une valeur >1.1, le modèle ne converge pas. **Ce tableau pourra servir d'outils de diagnostic si vous avez une question sur vos estimations, il suffira alors de l'envoyer aux contacts inscrits sur ce rapport.**

	A	B	C	D	E	F	G	H
1	names	mean	sd	q2.5	q25	q50	q75	q97.5
2	pa	0.713127714472793	0.0838596094333824	0.540073294809635	0.657965025463135	0.716588668844687	0.772482003042989	0.86668279189847
3	pexistant	0.691403261652501	0.0625516934086994	0.5638326242795	0.650424717041574	0.693657800826025	0.734821447825748	0.80752377159954
4	pf	0.156532774402569	0.0323163022468043	0.101112304640317	0.133701578097313	0.153781697093761	0.176361616702692	0.22801202908369
5	Yfm[1]	0.122511111111111	0.362389045313782		0	0	0	0
6	Yfm[2]	0.125255555555556	0.368738541672587		0	0	0	0
7	Yfm[3]	1.18752222222222	0.47551625245343		1	1	1	1
8	Yfm[4]	0.126544444444444	0.365694454347648		0	0	0	0
9	Yfm[5]	0.1228	0.365345314237832		0	0	0	0
10	Yfm[6]	0.130344444444444	0.371663938051345		0	0	0	0
11	Yfm[7]	0.124966666666667	0.365598953964513		0	0	0	0
12	Yfm[8]	0.119033333333333	0.367271180285084		0	0	0	0
13	Yfm[9]	1.18151111111111	0.46524845148719		1	1	1	1
14	Yfm[10]	0.121611111111111	0.360464150217658		0	0	0	0
15	Yfm[11]	0.125655555555556	0.36209410124933		0	0	0	0
16	Yfm[12]	0.124377777777778	0.359935869432716		0	0	0	0
17	Yfm[13]	1.18045555555556	0.476800082267665		1	1	1	1

Figure 24 : Tableau de résultats « Resultats.txt » ouvert avec Excel

4. **GrapheFluxMigratoire.png** (Fig. 25): Ce graphique représente **le nombre de poissons migrants par jour**. En bleu, il s'agit du nombre de poissons observés au dispositif de comptage existant (Yexistent) et en rose, il s'agit de l'échappement estimé par le modèle (Yfm-Yexistent)

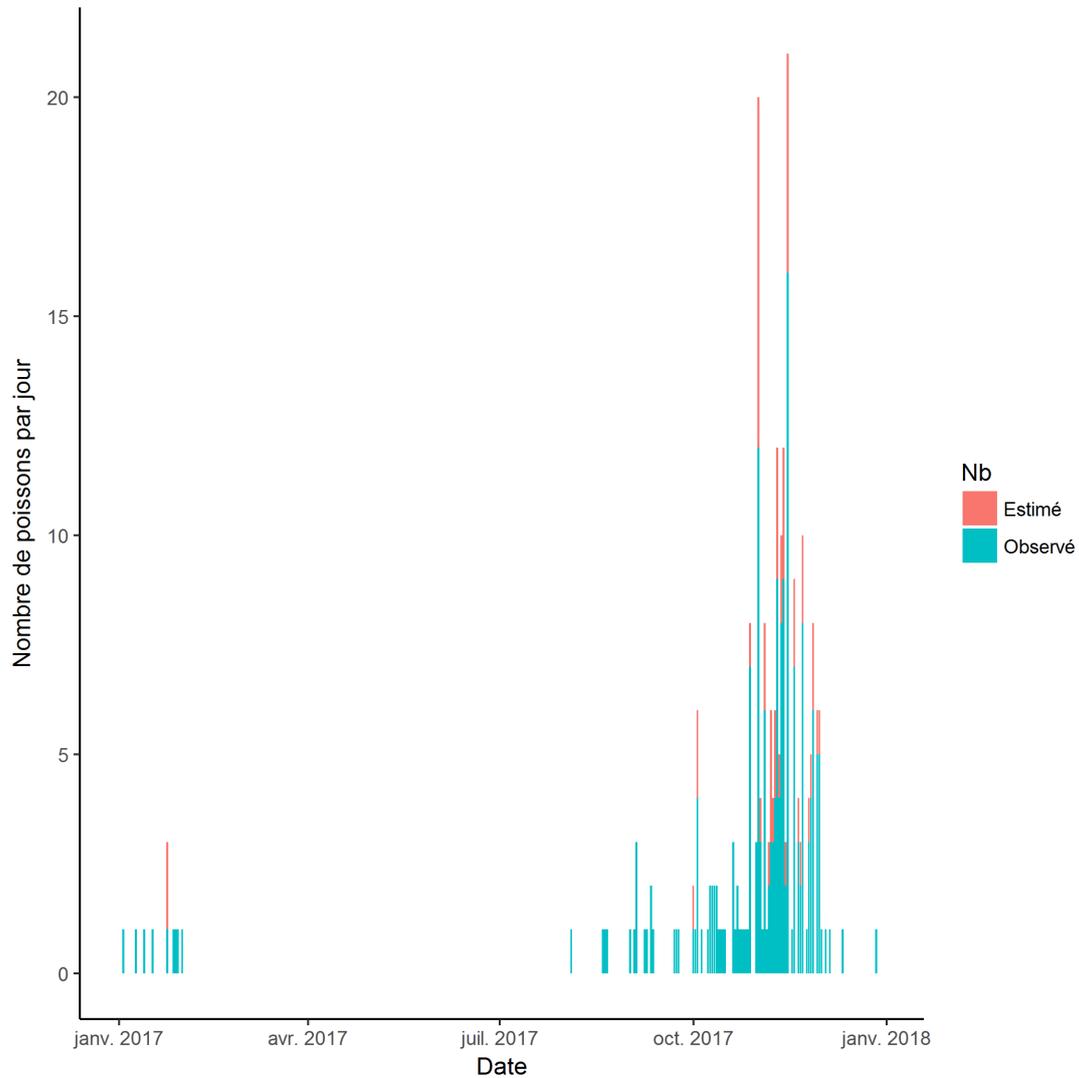


Figure 25 : Graphique issu de « GrapheFluxMigratoire.png » représentant le nombre de poissons migrants par jour. En bleu : nombre de poissons observés par le dispositif de comptage existant, en rose : l'échappement estimé par le modèle

5. **GrapheFluxMigratoireIC.png** (Fig. 26): Ce graphique représente aussi **le nombre de poissons migrant par jour** mais sous un autre format. **Les points représentent la médiane du nombre journalier de poissons** alors que **les barres représentent l'intervalle de confiance (97.5%)** autour de cette estimation.

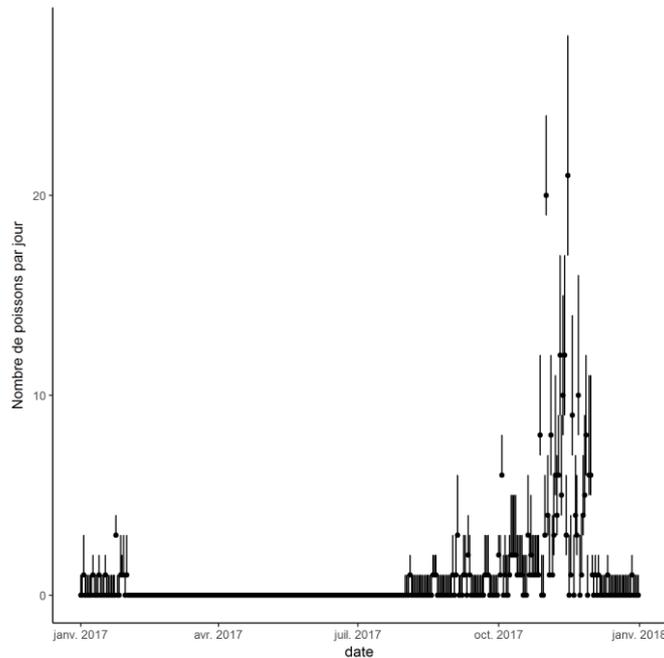


Figure 26 : Graphique issu de « GrapheFluxMigratoireIC.png » représentant le nombre de poissons migrants par jour. Un point représente une médiane journalière estimée et les barres représentent l'intervalle de confiance (97.5%) autour de cette estimation

Un graphique sera créé seulement si le modèle fixe a été utilisé :

6. **GrapheProbabilitédeDetection.png** (Fig. 27): Ce graphique représente les estimations des probabilités de détection des différents outils de comptages (**Probabilité appareil existant : $P_{existent}$, Probabilité Aris : P_a et Probabilité Filtre : P_f**). Un point représente la médiane estimée de chaque probabilité de détection et les barres représentent l'intervalle de confiance (97.5%) autour de cette estimation

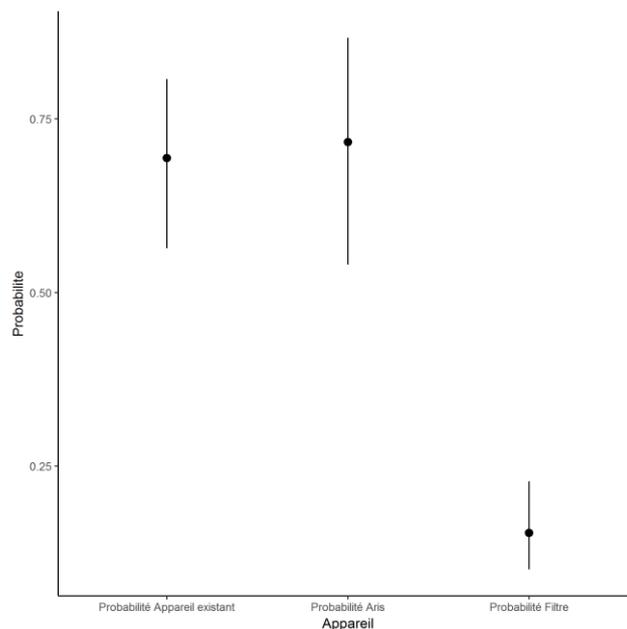


Figure 27 : Graphique issu de « GrapheProbabilitédeDetection.png » représentant les estimations des probabilités de détection. Un point représente la médiane estimée pour chaque probabilité de détection et les barres représentent l'intervalle de confiance (97.5%) autour de cette estimation

Et un graphique sera créé seulement si le modèle variable a été utilisé :

7. **GrapheProbaVariable.png** (Fig. 28). Ce graphique représente les probabilités de détection du dispositif de comptage existant en fonction de la variable testée. Un point représente la médiane estimée pour chaque probabilité de détection (une estimée par jour) et les barres représentent l'intervalle de confiance (97.5%) autour de cette estimation

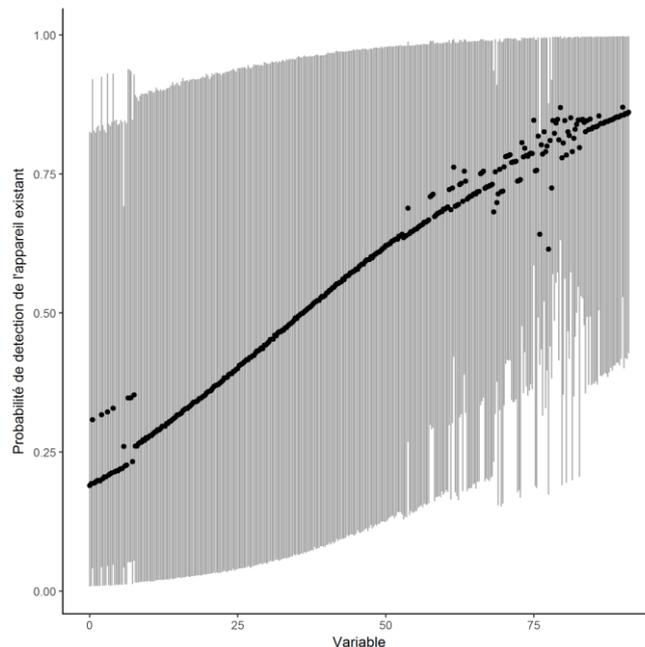


Figure 28 : Graphique issu de « GrapheProbaVariable.png » représentant les probabilités de détection du dispositif de comptage existant en fonction de la variable testée. Un point représente la médiane estimée pour chaque probabilité de détection (une estimée par jour) et les barres représentent l'intervalle de confiance (97.5%) autour de cette estimation

10^{ème} Etape : Choix des résultats à utiliser (si utilisation du modèle variable) :

Dans le cas où vous auriez choisi d'utiliser le modèle variable, il est important de savoir si:

- **Le modèle variable est plus pertinent que le modèle fixe.** Pour cela, il est important de faire tourner les deux modèles (fixe et variable) et de comparer leur DIC (Deviance information criterion) ; Le DIC est l'équivalent de l'AIC (Akaike information criterion) mais appliqué aux modèles hiérarchiques. Ce paramètre sert à faire de la sélection de modèle : plus il sera faible, meilleur le modèle sera. **On considère que pour que les modèles soient considérés comme différents en terme de performance il faut que la différence entre leur DIC soit égale ou supérieure à 4.** Pour calculer cette différence, vous trouverez dans chacun des sous-dossiers « Resultats Modele fixe » et « Resultats Modele Variable », un fichier appelé « Statistiques ». Dans ces fichiers, vous aurez une estimation du DIC. Faites la soustraction entre le DIC du modèle fixe et le DIC du modèle variable et si la différence est supérieure ou égale à 4, il est pertinent d'utiliser les résultats du modèle variable. Dans le cas contraire, il sera plus pertinent

d'utiliser les résultats du modèle fixe. **Exemple** : Dans la figure 29, on voit que la valeur du DIC du modèle fixe est égale à 650.5 et celle du modèle variable est égale à 1059.8. La différence est donc de -409.3, ce qui signifie qu'il faut utiliser les résultats du modèle fixe car le DIC du modèle fixe est très inférieur à celui du modèle variable.

a)			b)			
	A	B	A	B	C	
1	Nom	valeur	1	Nom	valeur	
2	DIC	1059.83964538446	2	DIC	650.540815165782	
3	p.value de la pente	0.0267	3			
4			4			
5			5			

Figure 29 : Résultats figurant dans les fichiers « Statistiques » a) pour le modèle variable et b) pour le modèle fixe.

- **L'effet de la variable est significativement différent de 0.** Pour cela, **une p.value a été calculée.** Si elle est inférieure ou égale à **0.05**, l'effet de la variable sur la probabilité de détection du dispositif existant est significativement différente de 0. Si elle est supérieure à 0.05, il n'y a pas d'effet significative de la variable et il est nécessaire d'utiliser les résultats du modèle fixe. **Il y a deux façons d'avoir la valeur de cette p-value** : 1) Lorsque la procédure de lancement du modèle variable est terminée (8^{ème} Etape), un message va s'afficher avec un diagnostic sur la p-value (Fig. 30), 2) dans le sous-dossier « Resultats Modele Variable », dans le fichier appelé « Statistiques », il y a une estimation de la p-value (Fig. 29a). **Exemple** : Dans notre exemple, la p-value est inférieure à 0.05 (Fig. 29a et 30) ce qui signifie qu'il y a effet significative de la variable sur la probabilité de détection. Cependant au vu de la différence de DIC, il est quand même plus pertinent d'utiliser les résultats du modèle fixe.

```

|*****| 100%

Calculating statistics.....

Done.
Saving 7 x 7 in image
Saving 7 x 7 in image
Saving 7 x 7 in image
[1] " Le DIC est égal à 1059.83964538446 vérifier qu'il est inférieur au DIC du modèle avec une probabilité de détection fixe (voir mode d'emploi)"
[1] "p.value= 0.0267 ,elle est inférieure à 0.05, il y a effet significative de la variable sur la probabilité de détection de l'appareil existant"

```

Figure 30 : Message affichant le DIC et la p-value après la fin de la procédure « Lancement du modèle » lorsque le modèle variable a été choisi.

Annexe : Liste des Error et des Warning

Warning :

- Il existe une ou des ligne.s vides dans le tableau, elle.s seront supprimees
- Il existe des valeurs negatives dans la colonne Yexistant, elles seront remplacees par des 0 pour l'analyse.
- Il existe des valeurs negatives dans la colonne Ya1, elles seront remplacees par des 0 pour l'analyse
- Il existe des valeurs negatives dans la colonne Ya2, elles seront remplacees par des 0 pour l'analyse
- Il existe des valeurs negatives dans la colonne Yf1, elles seront remplacees par des 0 pour l'analyse
- Il existe des valeurs negatives dans la colonne Yf2, elles seront remplacees par des 0 pour l'analyse

Error :

- Il manque la colonne de donnees Yexistant ou le nom de votre colonne est incorrect
- Il manque la colonne de donnees Ya1 ou le nom de votre colonne est incorrect
- Il manque la colonne de donnees Ya2 ou le nom de votre colonne est incorrect
- Il manque la colonne de donnees Yf1 ou le nom de votre colonne est incorrect
- Il manque la colonne de donnees Yf2 ou le nom de votre colonne est incorrect
- Il manque la colonne de donnees variable ou le nom de votre colonne est incorrect
- Il manque une ou des dates dans la colonne de donnees des dates
- Une date n'existant pas est presente dans le jeu de donnees.
- Il manque un ou plusieurs jour.s dans la serie temporelle, rajouter ce ou ces jour.s
- Probleme de format de date
- Problème de convergence, les résultats ne sont interprétables: relancer la fonction avec un nombre d itérations et un nombre de burnin plus important. Rappel: valeur conseillée iteration= 300 000 et burnin=270 000

Avec le soutien financier de

AGENCE FRANÇAISE
POUR LA BIODIVERSITÉ
ÉTABLISSEMENT PUBLIC DE L'ÉTAT

www.agence-francaise-biodiversite.fr



<http://www.inra.fr/>