



Développements et mises à jour des indicateurs poissons en eaux de transition dans le SEEE et méthodes d'agrégation

C. Delpech, Hilaire Drouineau, Mario Lepage

► To cite this version:

C. Delpech, Hilaire Drouineau, Mario Lepage. Développements et mises à jour des indicateurs poissons en eaux de transition dans le SEEE et méthodes d'agrégation. [Rapport de recherche] irstea. 2011, pp.108. hal-02596680

HAL Id: hal-02596680

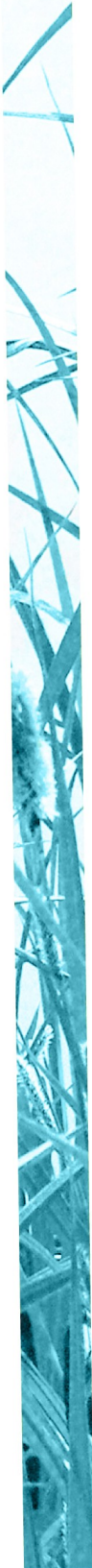
<https://hal.inrae.fr/hal-02596680>

Submitted on 15 May 2020

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.

Partenariat 2010 – Domaine 7.12 - Action 20



Développements et mises à jour des indicateurs poissons en eaux de transition dans le SEEE et méthodes d'agrégation

Rapport final

Christine Delpech (Cemagref Bordeaux)

Hilaire Drouineau (Cemagref Bordeaux)

Mario Lepage (Cemagref Bordeaux)

Janvier 2011

Contexte de programmation et de réalisation

Le Système d'Évaluation de l'État des Eaux développé par l'Onema et ses partenaires se compose de deux parties. Une partie plutôt dédiée aux scientifiques et permettant de développer les indicateurs biologiques et de tester différents scénarios à partir d'une base de données prévue à cet effet. La V1 de cet outil a été terminée en 2009. Une V2 est prévue afin d'améliorer certaines fonctionnalités et de compléter les métriques qui n'ont pas encore été spécifiées dans la V1. La deuxième partie de l'outil SEEE concerne l'évaluation et à terme le rapportage à l'union européenne. Cette partie utilisera les développements réalisés dans l'outil de simulation et sera complétée par une agrégation des différents résultats des indicateurs biologiques, physico-chimiques, hydromorphologiques et chimiques pour rapporter à l'union européenne l'état des masses d'eau.

Les auteurs

Christine Delpech*Ingénieur d'études**christine.delpech@cemagref.fr**Cemagref Bordeaux***Hilaire Drouineau***Ingénieur de recherche**hilaire.drouineau@cemagref.fr**Cemagref Bordeaux***Mario Lepage***Ingénieur de recherche**mario.lepage@cemagref.fr**Cemagref Bordeaux*

Les correspondants

Onema : Isabelle, Vial, *direction, isabelle.vial@onema.fr*

Référence du document :

Cemagref : Mario, Lepage, *Cemagref, mario.lepage@cemagref.fr*

Référence du document :

Droits d'usage :	<i>accès libre</i>
Couverture géographique :	<i>France</i>
Niveau géographique [un seul choix] :	<i>National</i>
Niveau de lecture [plusieurs choix possibles] :	<i>Professionnels, experts</i>
Nature de la ressource :	<i>Document</i>

**Développements et mises à jour des indicateurs poissons en eaux de transition
dans le SEEE et méthodes d'agrégation**

Rapport final

Christine Delpech, Hilaire Drouineau, Mario Lepage

Sommaire

Résumé.....	4
Abstract.....	5
Synthèse pour l'action opérationnelle.....	6
Corps du document.....	9
1. Contexte.....	9
2. Indicateurs développés en estuaires.....	10
3. Indicateurs développés en lagunes.....	12
4. Tests de sensibilité.....	13
4.1. Niveau d'agrégation des données.....	13
4.2. Effet de l'ordre des saisons d'échantillonnage.....	15
5. Conclusion.....	16
6. Bibliographie.....	17
ANNEXES : Fiches de spécification.....	18
<i>Métrique de densité des poissons migrateurs dans les lagunes.....</i>	<i>18</i>
<i>Métrique de densité des poissons prédateurs d'invertébrés benthiques dans les lagunes.....</i>	<i>35</i>
<i>Métrique de densité des poissons zooplanctonophages dans les lagunes</i>	<i>51</i>
<i>Indicateur multimétrique de qualité des lagunes.....</i>	<i>67</i>
ANNEXES : Algorithmes de calcul sous R.....	72
<i>ET-Poisson-DENS_CA_LAG v 0.001.....</i>	<i>72</i>
<i>ET-Poisson-DENS_IB_LAG v 0.001.....</i>	<i>83</i>
<i>ET-Poisson-DENS_Z_LAG v 0.002.....</i>	<i>94</i>
<i>INDIC_POISSON_LAG v 0.001.....</i>	<i>105</i>

**Développements et mises à jour des indicateurs poissons en eaux de transition
dans le SEEE et méthodes d'agrégation***Christine Delpech, Hilaire Drouineau, Mario Lepage***Résumé****RÉSUMÉ**

L'Onema, accompagné de plusieurs partenaires scientifiques et gestionnaires, et appuyé par de solides équipes de développeurs informaticiens, a entrepris la réalisation d'un outil national d'évaluation de la qualité des eaux encore appelé SEEE (Système d'Evaluation de l'Etat des Eaux). En 2009, seules les quatre métriques qui avaient été développées pour les estuaires étaient opérationnelles dans l'outil SEEE. Cette année, les travaux ont tout d'abord consisté à mettre à jour les spécifications et les algorithmes de ces quatre métriques estuariennes en fonction des avancées des travaux réalisés par le Cemagref. En Septembre 2010, une deuxième version de l'outil de simulation nous a été livrée et les résultats fournis ont été vérifiés. Concernant l'indicateur poisson développé pour les lagunes méditerranéennes, des spécifications ont été rédigées pour les trois métriques : densité de poissons migrateurs, densité de poissons prédateurs d'invertébrés benthiques et densité de poissons zooplanctonophages. Les algorithmes sous R ont ensuite été créés à partir de ceux déjà écrits pour les métriques estuariennes. Des mises à jour de ces indicateurs seront néanmoins nécessaires en 2011.

Afin de valider un indicateur, différents tests sont nécessaires. Ces tests concernent en particulier les méthodes d'agrégation des métriques, c'est à dire comment les différentes opérations de pêche sont agrégées (dans le temps et l'espace) pour fournir un diagnostic unique à l'échelle de la masse d'eau et du plan de gestion. Deux méthodes ont été testées : (i) en attribuant une note par zone haline en moyennant les densités, (ii) en attribuant une note par opération de pêche puis en moyennant ces notes par zone haline. Ces deux méthodes donnent des résultats relativement similaires en terme de diagnostic. En revanche, la seconde méthode semble fournir des incertitudes plus faibles, et ainsi beaucoup mieux lisser la forte variabilité liée au processus de captures. Ces résultats préliminaires seront à confirmer quand plusieurs années de données seront disponibles.

Un autre test sur l'indicateur porte sur l'importance du respect du protocole d'échantillonnage, et notamment sur l'effet que peut avoir l'inversion des saisons d'échantillonnage (échantillonner l'automne avant le printemps). Des tests ont ainsi été réalisés avec deux années d'échantillonnage successives sur l'étang de Berre, démontrant des différences significatives selon l'ordre des saisons d'échantillonnage. Cela confirme l'importance de respecter le protocole tel qu'il a été défini, en particulier si l'on veut pouvoir produire un diagnostic par année.

MOTS CLÉS (THÉMATIQUE ET GÉOGRAPHIQUE)

Système d'Evaluation de l'Etat des Eaux, R, Directive Cadre sur l'Eau, Eaux de transition, Indicateur poissons

Développements et mises à jour des indicateurs poissons en eaux de transition dans le SEEE et méthodes d'agrégation*Christine Delpech, Hilaire Drouineau, Mario Lepage***Abstract****ABSTRACT**

Onema, with several scientific partners and managers, and supported by a team of developers, has begun the implementation of a national tool for assessing the water quality also known as SEEE (Système d'Evaluation de l'Etat des Eaux). In 2009, only four metrics that had been developed for estuaries were operational in the SEEE tool. This year, updates have been made for the specifications and algorithms of the four estuarine metrics according to progress carried out by the Cemagref to improve the fish index ELFI (Estuarine and Lagoon Fish Index). In September 2010, a second version of the simulation tool has been delivered. After installation, results provided by metric calculation have been checked. Concerning the lagoon fish index, specifications were written for the three metrics: density of migrant fishes, density of benthic invertebrate predator fishes and density of zooplankton predator fishes. Algorithms on R software were then created from those already written for estuarine metrics. Updates will probably be necessary in 2011.

Several tests are required to validate an indicator. For example, it is necessary to validate the method used to aggregate the metrics, i.e. how the fishing operations are aggregated through time and space to provide a unique diagnostic at the water body and management plan level. In this context, two methods were tested: (i) giving a score by salinity zone by averaging the densities, (ii) assigning a score for each fishing operation and then averaging these notes per salinity zone. The two strategies provide similar results. However, the second method seems to provide smaller confidence interval, by better smoothing the great variability between fishing operations. Those results are preliminary, and the analysis should be carried out again when several years of data will be available.

An other test consists in assessing the sensitivity of the index to departure from the sampling protocol, especially departures from the sampling season order (autumn before spring instead of spring before autumn). An analysis has been carried out on two successive sampling years on the "Etang de Berre", and has demonstrated that season inversion may lead to significantly different diagnostic. This result points out the importance to respect the sampling protocol, especially if a yearly diagnostic is required.

ABSTRACT (THEMATIC AND GEOGRAPHICAL AREA)

SEEE, R, Water Framework Directive, Transitional waters, Fish-based index

***Développements et mises à jour des indicateurs poissons en eaux de transition
dans le SEEE et méthodes d'agrégation***

Christine Delpech, Hilaire Drouineau, Mario Lepage

Synthèse pour l'action opérationnelle

Contexte

L'Onema, accompagné de plusieurs partenaires scientifiques et gestionnaires, et appuyé par de solides équipes de développeurs informaticiens, a entrepris la réalisation d'un outil national d'évaluation de la qualité des eaux encore appelé SEEE (Système d'Evaluation de l'Etat des Eaux). Ce projet ambitieux a deux objectifs. Dans une première phase (outil de simulation), il a pour objectif de faciliter le travail des scientifiques pour le développement des différents indicateurs biologiques, physiques et chimiques prévus dans l'application de la DCE. Ce système permettra d'utiliser directement une base de données commune afin de calculer les métriques de chaque indicateur. La deuxième partie (outil d'évaluation) concerne le développement d'un outil permettant l'évaluation de l'état des eaux à plusieurs niveaux mais aussi le rapportage européen à chaque plan de gestion.

Les travaux engagés depuis 2008 sur le SEEE ont permis la réalisation d'une V1 de l'outil de simulation. Cette V1 doit cependant être complétée au cours de l'année 2010. En même temps, la mise en place de l'outil d'évaluation doit tirer profit des développements réalisés dans l'outil de simulation et passer à une phase d'agrégation des résultats des différents indicateurs biologiques, physico-chimiques, hydromorphologiques et chimiques.

Indicateurs développés en estuaires

En début d'année, l'outil SEEE a été actualisé pour permettre une première mise à disposition aux gestionnaires afin qu'ils se familiarisent avec l'outil et les fonctionnalités qu'il apporte. Les cinq indicateurs développés pour les poissons en estuaires (4 métriques et 1 indice multimétrique) ont été mis à jour avec les tableaux de seuils définis fin 2009. En Septembre 2010, la deuxième version de l'outil de simulation a été testée et les bugs rencontrés ont été corrigés. A chaque mise à jour ou correction des algorithmes R, les résultats fournis par le SEEE ont été vérifiés. Les dernières versions de ces indicateurs développés pour les estuaires, publiées dans l'outil, sont les suivantes :

✓	ET-Poisson-DENS_B_EST	v 0.012
✓	ET-Poisson-DENS_MJ_EST	v 0.009
✓	ET-Poisson-DENS_CA_EST	v 0.009
✓	ET-Poisson-DENS_DT_EST	v 0.008
✓	ET-Poisson-INDIC_POISSON_EST	v 0.012

Indicateurs développés en lagunes

Pour les lagunes, aucun indicateur n'avait encore été créé dans le SEEE. Les travaux menés pour l'action 16 ont permis de définir 3 métriques pour caractériser l'état écologique des lagunes méditerranéennes. Les spécifications décrivant le calcul de ces 3 métriques ainsi que celle décrivant le calcul de l'indice multimétrique ont été rédigées. Les algorithmes sous

R ont été écrits en se basant sur ceux déjà écrits pour les estuaires. Des adaptations spécifiques aux lagunes ont été nécessaires. A chaque étape de modification du script R, les résultats en sortie de l'outil SEEE ont été vérifiés. Néanmoins, le jeu de données disponible pour les lagunes est relativement restreint et peu d'avancées ont pu être réalisées pour l'indicateur lagune en 2010. Aucune version stabilisée de l'indicateur n'a été proposée fin 2010. De ce fait, les indicateurs développés dans l'outil SEEE pour les lagunes ne sont pas définitifs. Des modifications seront très probablement apportées en 2011 en fonction des avancés des travaux effectués sur cet indicateur.

Tests de sensibilité

Afin de valider un indicateur, différents tests sont nécessaires. Ces tests concernent en particulier les méthodes d'agrégation des métriques, c'est à dire comment les différentes opérations de pêche sont agrégées (dans le temps et l'espace) pour fournir un diagnostic unique à l'échelle de la masse d'eau et du plan de gestion. Ainsi, une méthode de test a été proposée, et mise en œuvre pour comparer sur les lagunes méditerranéennes, le système actuel :

- les métriques sont moyennées par saison et zone haline,
- puis une note est donnée par métrique et zone haline,
- la moyenne des notes constitue la valeur de l'indicateur poisson

à une stratégie alternative :

- une note est attribuée par métrique et par opération de pêche
- ces notes sont moyennées ensuite par métrique
- la moyenne des notes par métrique constitue la valeur de l'indicateur poisson.

Ces deux méthodes donnent des résultats relativement similaires en terme de diagnostic. En revanche, la seconde méthode semble fournir des incertitudes plus faibles, et ainsi beaucoup mieux lisser la forte variabilité liée au processus de captures. Pour le moment, le jeu de données disponible a permis une première analyse de l'effet de l'agrégation par saison et par zone haline. Cela dit le jeu de données lagunes est encore trop limité pour tirer trop de conclusions. La possibilité de ne plus agréger les données par cas devra malgré tout être considérée à l'avenir. Il n'a pas été possible de tester l'effet de l'agrégation de différentes années, une seule année de données étant disponible pour le moment. Ce travail sera donc à reconduire ultérieurement pour sélectionner la meilleure méthode de combinaison des 3 années de données.

Un autre test sur l'indicateur porte sur l'importance du respect du protocole d'échantillonnage, et notamment sur l'effet que peut avoir l'inversion des saisons d'échantillonnage. En effet, en 2010, les lagunes du district Rhône Méditerranée et Corse ont été échantillonnées à l'automne, et elles le seront de nouveau au printemps 2011. Les échantillonnages auront donc lieu à l'automne, puis au printemps de l'année suivante, au lieu du printemps et de l'automne de la même année tel que préconisé dans le protocole d'échantillonnage des masses d'eau de transition méditerranéennes (Lepage et al, 2008). Cela peut avoir plusieurs conséquences du point de vue des captures, notamment si les recrutements des deux années sont très différents, ou si des crises dystrophiques (malaigues) ont eu lieu au cours de l'été. Il était donc important de vérifier l'influence de cette modification du protocole sur les évaluations réalisées avec ELFI. Des tests effectués sur les échantillonnages réalisés dans l'étang de Berre en 2008 et 2009 ont ainsi montré des différences significatives dans les valeurs de l'indicateur ELFI obtenu selon les combinaisons saison/année prises en compte. Ces résultats confirment l'importance de se baser sur plusieurs années de données pour fournir un diagnostic en limitant l'influence de la variabilité naturelle (variabilité du recrutement, variabilité de l'oxygénation estivale). De plus, ces tests ont montré

que dans l'optique de fournir un diagnostic annuel, considérer le printemps et l'automne d'une même année n'est pas équivalent à considérer l'automne et le printemps de deux années consécutives (printemps 2008 + automne 2008 comparé à automne 2008 et printemps 2009 par exemple). Il est donc nécessaire de respecter le protocole d'échantillonnage préconisé pour obtenir une homogénéité de la façon de calculer ELFI.

Conclusion

La validation des métriques constituant l'indicateur poisson en estuaires n'a été achevée qu'à la fin de l'année 2010. L'exercice d'intercalibration européen se terminant en fin d'année 2011, il faut s'attendre à ce qu'il y ait quelques changements entre la version actuelle et la version stabilisée de ELFI. En 2011, les travaux relatifs à cette action consisteront tout d'abord à créer les indicateurs des métriques encore manquantes dans l'application SEEE. Pour les lagunes, les travaux ne sont pas aussi avancés en raison du faible jeu de données et l'indicateur n'est pas encore validé. Le nouveau jeu de données issus des campagnes de l'automne 2010 et du printemps 2011 permettra peut-être d'améliorer et de rendre plus robustes les métriques actuelles voire d'en créer de nouvelles malgré le fait que les deux campagnes n'aient pas été réalisées la même année. En fonction de ces avancés, des mises à jour, voire même de nouveaux indicateurs, seront apportées dans l'outil SEEE.

En 2011, nous continuerons également de suivre les modifications de l'outil de simulation V2 ainsi que dans sa phase d'évaluation, et d'apporter notre contribution au développement de cette application.

Pour en savoir plus :

Delpech, C., Drouineau, H., Lepage, M., 2011. Développements et mises à jour des indicateurs poissons en eaux de transition dans le SEEE et méthodes d'agrégation. Convention de partenariat ONEMA-Cemagref 2010 : 108p.

christine.delpech@cemagref.fr
hilaire.drouineau@cemagref.fr
mario.lepage@cemagref.fr

***Développements et mises à jour des indicateurs poissons en eaux de transition
dans le SEEE et méthodes d'agrégation***

Christine Delpech, Hilaire Drouineau, Mario Lepage

Corps du document

1. Contexte

L'Onema, accompagné de plusieurs partenaires scientifiques et gestionnaires, et appuyé par de solides équipes de développeurs informaticiens, a entrepris la réalisation d'un outil national d'évaluation de la qualité des eaux encore appelé SEEE (Système d'Evaluation de l'Etat des Eaux). Ce projet ambitieux a deux objectifs. Dans une première phase (outil de simulation), il a pour objectif de faciliter le travail des scientifiques pour le développement des différents indicateurs biologiques, physiques et chimiques prévus dans l'application de la DCE. Ce système permettra d'utiliser directement une base de données commune afin de calculer les métriques de chaque indicateur. La deuxième partie (outil d'évaluation) concerne le développement d'un outil permettant l'évaluation de l'état des eaux à plusieurs niveaux mais aussi le rapportage européen à chaque plan de gestion.

Les travaux engagés depuis 2008 sur le SEEE ont permis la réalisation d'une V1 de l'outil de simulation. Cette V1 doit cependant être complétée au cours de l'année 2010. En même temps, la mise en place de l'outil d'évaluation doit tirer profit des développements réalisés dans l'outil de simulation et passer à une phase d'agrégation des résultats des différents indicateurs biologiques, physico-chimiques, hydromorphologiques et chimiques.

L'équipe Estuaire de l'unité Ecosystèmes Estuariens et Poissons Migrateurs Amphihalins a contribué au développement de la première phase de simulation par la spécification des métriques et la validation des différentes étapes permettant le calcul de celles-ci. Aujourd'hui, la bancarisation des données « poissons dans les eaux de transition » est réalisée au Cemagref de Bordeaux et l'équipe assure la validation des données de terrain avant l'introduction des données dans la base nationale du SEEE. Cette étape d'acquisition et de validation est primordiale pour assurer la qualité des données qui seront ensuite utilisées pour l'évaluation de l'état des eaux. Ce travail fait l'objet d'une fiche détaillée à part (fiche action n° 20 bis).

En 2009, seuls les cinq indicateurs qui avaient été développés pour les estuaires étaient opérationnels dans l'outil SEEE. Les indicateurs concernant les lagunes méditerranéennes restaient à faire de même que les éventuels nouveaux indicateurs qui étaient en cours de développement. Cette année, les travaux ont tout d'abord consisté à mettre à jour les indicateurs déjà développés dans le SEEE en fonction des avancés des travaux réalisés par le Cemagref pour améliorer l'indice poisson ELFI (Estuarine and Lagoon Fish Index). Puis, une phase de vérification des résultats fournis a été entreprise lors du passage de la V1 à la V2 de l'outil de simulation. Des indicateurs ont également été créés pour les lagunes. Enfin, des tests sur la méthode d'agrégation des scores ont été effectués.

2. Indicateurs développés en estuaires

En début d'année, a eu lieu une grosse phase d'actualisation de l'outil SEEE pour permettre une première mise à disposition aux gestionnaires afin qu'ils se familiarisent avec l'outil et les fonctionnalités qu'il apporte. Pour cela, les derniers bugs signalés sous Mantis ont été traités et fermés. Les cinq indicateurs développés pour les poissons en estuaires ont été mis à jour avec les tableaux de seuils définis fin 2009 (Delpech *et al.*, 2010). Les spécifications (fiches métriques) ont également été mises à jour et incluses dans chaque indicateur. Après vérification que l'outil retourne toujours les résultats attendus suite à ces modifications, les indicateurs ont été publiés (mis à disposition sur un serveur). Il s'agit des indicateurs suivants, publiés le 26/02/2010 :

- 4 métriques de base :

ET-Poisson-DENS_B_EST	v 0.011
ET-Poisson-DENS_MJ_EST	v 0.008
ET-Poisson-DENS_CA_EST	v 0.008
ET-Poisson-DENS_DT_EST	v 0.007

- 1 indice :

ET-Poisson-INDIC_POISSON_EST	v 0.011
------------------------------	---------

Les versions antérieures des indicateurs qui sont obsolètes, ont été purgées de l'application SEEE (Tableau 1).

Tableau 1 : Liste des indicateurs obsolètes purgés de l'application SEEE.

Nom du propriétaire	Mnémonique de l'indicateur	N° de version	Type
Administrateur	ET-Poisson-DENS_B_EST	0.005	Métrique
Administrateur	ET-Poisson-DENS_B_EST	0.006	Métrique
Delpech	ET-Poisson-DENS_B_EST	0.007	Métrique
Delpech	ET-Poisson-DENS_B_EST	0.009	Métrique
Delpech	ET-Poisson-DENS_B_EST	0.010	Métrique
Administrateur	ET-Poisson-DENS_MJ_EST	0.003	Métrique
Administrateur	ET-Poisson-DENS_MJ_EST	0.004	Métrique
Delpech	ET-Poisson-DENS_MJ_EST	0.005	Métrique
Delpech	ET-Poisson-DENS_MJ_EST	0.006	Métrique
Delpech	ET-Poisson-DENS_MJ_EST	0.007	Métrique
Administrateur	ET-Poisson-DENS_CA_EST	0.003	Métrique
Administrateur	ET-Poisson-DENS_CA_EST	0.004	Métrique
Delpech	ET-Poisson-DENS_CA_EST	0.005	Métrique
Delpech	ET-Poisson-DENS_CA_EST	0.006	Métrique
Delpech	ET-Poisson-DENS_CA_EST	0.007	Métrique
Administrateur	ET-Poisson-DENS_DT_EST	0.003	Métrique
Administrateur	ET-Poisson-DENS_DT_EST	0.004	Métrique
Delpech	ET-Poisson-DENS_DT_EST	0.005	Métrique
Delpech	ET-Poisson-DENS_DT_EST	0.006	Métrique
Administrateur	TEST_COMOE:ET-Poisson-DENS_B_EST	0.006	Métrique
Administrateur	TEST_COMOE:ET-Poisson-DENS_B_EST	0.001	Métrique
Administrateur	TEST_COMOE:ET-Poisson-DENS_B_EST	0.001	Métrique
Administrateur	ET-Poisson-INDIC_POISSON_EST	0.005	Indice
Administrateur	ET-Poisson-INDIC_POISSON_EST	0.007	Indice
Delpech	ET-Poisson-INDIC_POISSON_EST	0.008	Indice
Delpech	ET-Poisson-INDIC_POISSON_EST	0.009	Indice
Delpech	ET-Poisson-INDIC_POISSON_EST	0.010	Indice

Début Septembre 2010, la version V2 de l'outil de simulation nous a été livrée (v 2.0.18). Un problème a été rencontré pour les indicateurs concernant les poissons en eaux de transition : les indicateurs ne renvoyaient aucun résultat. Ils retournaient tous un message d'erreur "aucun enregistrement valide pour ce point de controle". Avec l'indice multimétrique, l'outil retournait une erreur générale : "Paramètre de la fonction FindRootElement incorrect".

Le problème a été résolu par Logica. Il y avait en V1 un problème d'encodage de caractères sur les coordonnées (qui apparaissaient au format "46Â°15,930Â°N" au lieu de "46°15,930°N"). Pour que les métriques des estuaires fonctionnent, une correction spécifique avait été mise en place dans les métriques (aucune solution transverse n'ayant alors été trouvée). La résolution du problème d'encodage dans la bibliothèque, en V2, a eu pour

conséquence de rendre inefficace la correction spécifique ci-dessus (les coordonnées étant de nouveau correctes). La correction des métriques en question a donc consisté à supprimer la correction spécifique aux métriques des estuaires. Après mise en place des algorithmes corrigés, les métriques retournent toutes des résultats corrects. Ces nouvelles versions des indicateurs ont été publiées :

ET-Poisson-DENS_B_EST	v 0.012
ET-Poisson-DENS_MJ_EST	v 0.009
ET-Poisson-DENS_CA_EST	v 0.009
ET-Poisson-DENS_DT_EST	v 0.008
ET-Poisson-INDIC_POISSON_EST	v 0.012

3. Indicateurs développés en lagunes

En 2009, trois métriques permettant de qualifier l'état écologique des lagunes avaient été développées (Delpech *et al.*, 2010). Il s'agit de :

- la densité de poissons migrateurs,
- la densité de poissons prédateurs d'invertébrés benthiques,
- la densité de poissons zooplanctonophages.

Cette année, des travaux ont été effectués pour améliorer les modèles réalisés et de nouveaux seuils ont été calculés (cf rapport Action 16). Les indicateurs relatifs à ces 3 métriques et à l'indice multimétrique ont été créés dans l'application SEEE :

ET-Poisson-DENS_CA_LAG	v 0.001
ET-Poisson-DENS_IB_LAG	v 0.001
ET-Poisson-DENS_Z_LAG	v 0.002
INDIC_POISSON_LAG	v 0.001

Pour cela, des fiches de spécification ont été rédigées et les algorithmes de calcul en langage R ont été écrits en se basant sur ceux déjà écrits pour les estuaires. Les fiches de spécification et les algorithmes sont fournis en annexe.

Diverses adaptations ont été nécessaires par rapport aux scripts des métriques estuariennes :

- Le calcul, permettant de définir la classe de taille des estuaires, a été simplifié car les lagunes, contrairement aux estuaires, ne sont constituées que d'une seule masse d'eau. De plus, la limite entre petites et grandes lagunes est fixée à 50 km² au lieu de 100 km² pour les estuaires.

- Un tableau de paramètre a également été ajouté. Ce tableau donne pour chaque masse d'eau la valeur d'un paramètre décrivant la connexion de la lagune avec la mer ; il s'agit de la surface cumulée des graus et est appelé « section ». Ce paramètre est indispensable car des seuils différents sont appliqués selon que la section est grande ou petite. Il a donc été nécessaire d'adapter le script R pour appeler ce tableau et l'utiliser lors de la comparaison aux

seuils. L'ajout de ce tableau est l'étape qui a suscité le plus de difficultés, notamment pour faire le lien entre ce tableau et la base de données du SEEE en se servant des codes des masses d'eau.

- Le calcul des densités a été modifié et celui de la surface échantillonnée a été supprimé car dans les lagunes, l'engin de pêche est une capêchade fixe et non plus un chalut à perche.

- Les classes de salinité ont aussi été modifiées car en lagunes seules 2 zones halines sont distinguées : zone oligo-mésohaline (≤ 18) et zone polyhaline (> 18).

- La structure des tableaux des seuils a été modifiée avec 2 seuils supplémentaires. En effet, nous avons définis 2 seuils délimitant 3 classes de qualité pour la plupart des combinaisons (zone haline x saison x section x classe de taille). Cependant, dans certains cas, en raison des recouvrements des intervalles de confiance, nous avons définis 3 ou 4 seuils délimitant respectivement 4 ou 5 classes de qualité. Les scripts R ont aussi été adaptés afin de prendre en compte tous les cas de figure possibles lors de l'attribution des scores.

- Pour la métrique de densité des poissons zooplanctonophages, l'attribution des scores a été inversée par rapport aux autres métriques. En effet, lorsque la dégradation du milieu augmente, les densités des poissons zooplanctonophages augmentent, alors que pour les autres métriques les densités diminuent.

- Le calcul des densités moyennes par saison et par zone haline a été modifié car, en lagunes, il s'agit d'une moyenne des densités log-transformées et non plus d'un calcul selon la méthode delta comme dans les estuaires (voir rapport Action 16).

Les valeurs des métriques et les notes des masses d'eau obtenues avec ces nouveaux indicateurs ont été vérifiées avant de publier les indicateurs.

Concernant les lagunes, le jeu de données disponible est relativement restreint et peu d'avancées ont pu être réalisées pour l'indicateur lagune en 2010. Aucune version stabilisée de l'indicateur n'a été proposée fin 2010. De ce fait, les indicateurs développés dans l'outil SEEE pour les lagunes ne sont pas définitifs. Des modifications seront très probablement apportées en 2011 en fonction des avancées des travaux effectués sur cet indicateur.

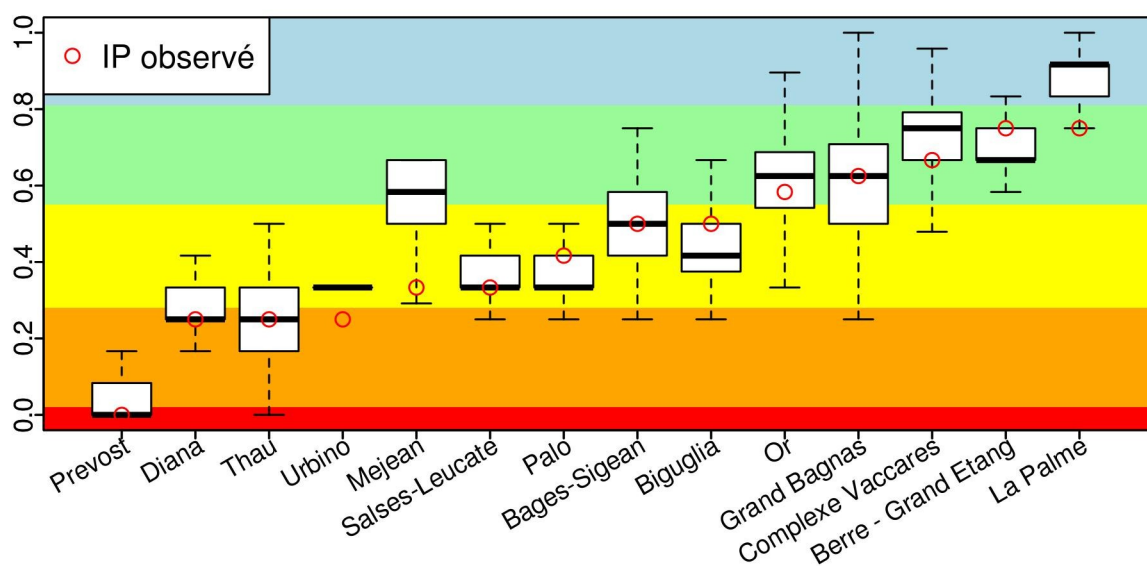
4. Tests de sensibilité

4.1. Niveau d'agrégation des données

Afin de lisser les effets de captures « exceptionnelles », et donc de proposer des diagnostics plus robustes, il a été proposé de moyenner les métriques obtenues pour chaque opération de pêche à l'échelle de la saison et de la zone haline, plutôt que de les considérer individuellement. On donne donc une note par cas (pour les lagunes, un cas désigne le croisement d'une saison, d'une zone haline, d'une classe de superficie et d'une classe de section des graux) pour chaque métrique, la moyenne de ces notes donne une note pour la métrique correspondante, la moyenne des notes pour toutes les métriques donnant l'indicateur final.

Pour voir si cette stratégie d'agrégation des observations par cas permet effectivement un lissage des données, et donc fournit un indicateur plus robuste, les notes obtenues pour les lagunes méditerranéennes ont été comparées aux notes qui seraient obtenues pour ces mêmes lagunes si une note était donnée directement à l'échelle de l'opération de pêche (en ne moyennant plus par cas). Pour quantifier l'incertitude autour de chacun des deux diagnostics, la méthode de quantification de l'incertitude proposée dans le rapport de l'action 16 de la convention ONEMA-Cemagref 2010 est appliquée (première méthode).

a)



b)

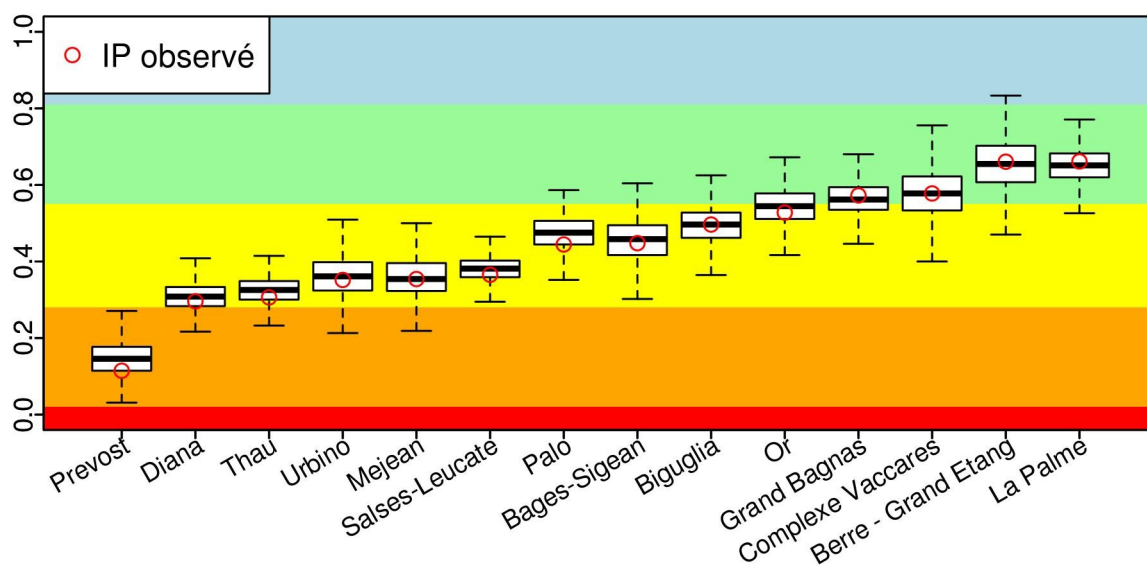


Figure 1 : Incertitudes autour du diagnostic fourni par l'indicateur ELFI quand les opérations de pêche sont agrégées par cas (figure a) ou non agrégées et directement traitées à l'échelle de l'opération de pêche (figure b).

Les diagnostics fournis par les deux méthodes sont proches du point de vue de l'ordre. En revanche, on observe que, contrairement à ce qu'on imaginait au départ, les intervalles de confiance semblent plus faibles quand les données ne sont pas agrégées par rapport à la méthode standard où l'on agrège les données par cas. Cela peut en fait s'expliquer par un effet de seuil. La présence de quelques captures exceptionnelles pour une métrique peut suffire à faire franchir d'une ou deux classes de qualité la métrique lorsqu'on agrège les données, modifiant alors de 2 ou 4 points la note pour cette métrique (le nombre d'opérations de pêche par cas étant assez limité), et donc modifiant assez fortement l'indice poisson qui n'est composé que de trois métriques pour l'instant.

Dans le cas où les observations ne sont pas agrégées par cas, à cette capture exceptionnelle correspondra peut-être une note très différente des autres, mais elle sera moyennée avec les notes obtenues par les opérations de pêche plus « normales » pour cette même métrique.

Pour le moment, le jeu de données disponible a permis une première analyse de l'effet de l'agrégation par saison et par zone haline. Cela dit le jeu de données lagunes est encore trop limité pour tirer trop de conclusions. La possibilité de ne plus agréger les données par cas devra malgré tout être considérée à l'avenir. Il n'a pas été possible de tester l'effet de l'agrégation de différentes années, une seule année de données étant disponible pour le moment. Ce travail sera donc à reconduire ultérieurement pour sélectionner la meilleure méthode de combinaisons des 3 années de données.

4.2. Effet de l'ordre des saisons d'échantillonnage

En 2010, les lagunes du district Rhône Méditerranée et Corse ont été échantillonnées à l'automne, et elles le seront de nouveau au printemps 2011. Les échantillonnages auront donc lieu à l'automne, puis au printemps de l'année suivante, au lieu du printemps et de l'automne de la même année tel que préconisé dans le protocole d'échantillonnage des masses d'eau de transition méditerranéennes (Lepage et al, 2008). Cela peut avoir plusieurs conséquences du point de vue des captures, notamment si les recrutements des deux années sont très différents, ou si des crises dystrophiques (malaigues) ont eu lieu au cours de l'été. Il était donc important de tester l'influence de cette modification du protocole sur les évaluations réalisées avec ELFI. Pour cela, nous avons utilisé des campagnes réalisées selon le protocole DCE sur l'étang de Berre au cours des printemps et automne 2008 et printemps et automne 2009 (données fournies par le GIP Posidonies pour le compte d'EDF, de l'AE-RM&C et du GIPREB). L'indicateur ELFI à trois métriques a ensuite été calculé sur toutes les combinaisons possibles de campagnes d'échantillonnage (Tableau 2).

Tableau 2 : Résultats de l'indicateur ELFI lagunes en considérant différentes campagnes du GIPREB sur l'étang de Berre.

Campagnes considérées	ELFI
Printemps 2008	0.33
Automne 2008	0.25
Printemps 2009	0.75
Automne 2009	0.75
Printemps 2008 + Automne 2008	0.28
Printemps 2008 + Printemps 2009	0.75
Printemps 2008 + Automne 2009	0.74
Automne 2008 + Printemps 2009	0.50
Printemps 2009 + Automne 2009	0.88
Automne 2008 + Automne 2009	0.58
Automne 2008 + Printemps 2009 + Automne 2009	0.67
Printemps 2008 + Automne 2008 + Printemps 2009	0.50
Printemps 2008 + Printemps 2009 + Automne 2009	0.88
Printemps 2008 + Automne 2008 + Automne 2009	0.50
Printemps 2008 + Automne 2008 + Printemps 2009 + Automne 2009	0.67

Les campagnes 2008, que ce soit au printemps ou à l'automne, donnent des résultats beaucoup plus pessimistes que les résultats 2009, avec une variabilité interannuelle assez forte. Cela s'explique principalement par une faible capture des espèces diadromes en 2008. Cela confirme l'importance de se baser sur plusieurs années de données pour fournir un diagnostic en limitant l'influence de la variabilité naturelle (variabilité du recrutement, variabilité de l'oxygénation estivale). Cela montre aussi que dans l'optique de fournir un diagnostic annuel, considérer le printemps et l'automne d'une même année n'est pas équivalent à considérer l'automne et le printemps de deux années consécutives (printemps 2008 + automne 2008 comparé à automne 2008 et printemps 2009 par exemple). Il est donc nécessaire de respecter le protocole d'échantillonnage préconisé pour obtenir une homogénéité de la façon de calculer ELFI.

5. Conclusion

La validation des métriques constituant l'indicateur poisson en estuaires n'a été achevée qu'à la fin de l'année 2010. L'exercice d'intercalibration européen se terminant en fin d'année 2011, il faut s'attendre à ce qu'il y ait quelques changements entre la version actuelle et la version stabilisée de ELFI. En 2011, les travaux relatifs à cette action consisteront tout d'abord à créer les indicateurs des métriques encore manquantes dans l'application SEEE. Pour les lagunes, comme signalés précédemment, les travaux ne sont pas aussi avancés en raison du faible jeu de données et l'indicateur n'est pas encore validé. Le nouveau jeu de données issus des campagnes de l'automne 2010 et du printemps 2011 permettra peut-être d'améliorer et de rendre plus robustes les métriques actuelles voire d'en créer de nouvelles malgré le fait que les deux campagnes n'aient pas été réalisées la même année. En fonction de ces avancées, des mises à jour, voire même de nouveaux indicateurs, seront apportées dans l'outil SEEE.

En 2011, nous continuerons également de suivre les modifications de l'outil de simulation V2 ainsi que dans sa phase d'évaluation, et d'apporter notre contribution au développement de cette application.

6. Bibliographie

Delpech, C., Drouineau, H., Lepage, M., 2010. Amélioration des performances de la méthode ELFI pour l'évaluation de la qualité des eaux de transition. Convention de partenariat ONEMA-Cemagref 2009, Cestas, 95 pp.

Lepage, M., Girardin, M., Bouju, V., 2008. Inventaire Poisson dans les eaux de transition. Mise à jour du protocole d'échantillonnage pour le District Rhône Méditerranée et Corse. Procédure EPBX_802_3, Cemagref - groupement de Bordeaux, Cestas, 30 p.

ANNEXES : Fiches de spécification

Métrie de densité des poissons migrateurs dans les lagunes

Nom de l'indicateur : Métrique de densité des poissons migrateurs dans les lagunes	Mnémonique : DENS_CA_LAG	Version : 1.0	Code SANDRE :
Profils d'évaluation applicables : <input type="checkbox"/> DCE <input type="checkbox"/> Diagnostic <input type="checkbox"/> Historique Autres ? Préciser :		Élément de qualité : Poissons	
Catégorie d'eau concernée : Eaux de transition (Une seule possible)		Thématique : Ichtyofaune	
Description succincte de l'indicateur : Cette métrique de densité des poissons migrateurs dans les lagunes permet d'évaluer la qualité des masses d'eau lagunaires du point de vue des densités de poissons appartenant à la guildes des espèces migratrices.		Propriétaire de l'indicateur pour le SEEE : Christine Delpech, Cemagref de Bordeaux	
		Coauteurs : Mario Lepage	
Références scientifiques de l'indicateur : <u>Réf principale :</u> Girardin, M., Lepage, M., Amara, R., Boët, P., Courrat, A., Delpech, C., Durozoi, B., Laffargue, P., Le Pape, O., Lobry, J., Parlier, E. and Pasquaud, S., 2009. Développement d'un indicateur poisson pour les eaux de transition. Programme Liteau II, Rapport scientifique de fin de contrat, Cemagref, Cestas. <u>Autres réf :</u> Delpech, C., Drouineau, H. and Lepage, M., 2010. Développement de l'indicateur « poisson » pour les eaux de transition, intercalibration européenne et interfaçage avec le système d'évaluation (SEEE). Convention de partenariat ONEMA-Cemagref 2009, Cestas. Delpech, C. and Lepage, M., 2009. Perfectionnements et ajustements de l'indicateur poisson pour les eaux de transition. Convention de partenariat ONEMA-Cemagref 2008, Cestas.		Références du besoin opérationnel auquel répond l'indicateur : Il s'agit des références des documents réglementaires ou non préconisant l'emploi de cet indicateur pour répondre à un besoin donné (par exemple arrêté du 25 janvier 2010 relatif à l'évaluation de l'état des eaux de surface). Un indicateur peut répondre à plusieurs besoins ; dans ce cas, indiquer le(s) besoin(s) majeurs ou le(s) plus fréquents. Dans l'outil d'évaluation, l'indicateur participe ainsi à la composition d'une ou de plusieurs stratégies d'évaluation (exemple stratégie DCE 2009). Renseigner obligatoirement si l'indicateur répond aux prescriptions pour l'évaluation DCE 2009 (mode de calcul, valeurs seuils, ..).	
Granularité temporelle de l'indicateur (en mois) : 12	Autres granularité temporelle standards (en mois) : 24, 36, 48, 60, 72	Nombre d'opérations de contrôle minimal : 1	
Mots clés utilisés dans la description : poissons, migrateurs, densité			
Abréviation	Terme	Définition/description	
CA	Migrateurs	Ensemble des espèces appartenant à la guildes des poissons migrateurs : Espèces utilisant les zones de transition comme	

				voie de migration
DONNEES DE SORTIE : PRODUITS DE L'INDICATEUR (données produites par l'indicateur: qui seront valorisées dans le SEEE). <u>Liste exhaustive</u>				
Produit Principal				
Code	Nom	Type	Forme	Description
Densité-CA	Valeur de la métrique densité de CA	M	<input type="checkbox"/> VB <input type="checkbox"/> EQR <input type="checkbox"/> CE	Moyenne des log-densités de poissons migrateurs multipliée par leur fréquence d'occurrence pour une masse d'eau, une saison et une classe de salinité définies
note-CA	Note pour la métrique densité de CA	M	<input type="checkbox"/> VB <input type="checkbox"/> EQR <input type="checkbox"/> CE	Note (normalisée entre 0 et 1) obtenue pour une masse d'eau pour la métrique de densité de poissons migrateurs
Les autres produits de l'indicateur				
<p>Ces produits supplémentaires peuvent être des Données de calcul intermédiaires (qui peuvent être des considérées comme une métrique), des Données explicatives, des Données de traçabilité, des Données d'incertitude,...</p> <p>Toutes ces données sont affichables dans le ou les tableaux des résultats de la simulation et dans le rapport d'évaluation de l'outil d'évaluation.</p> <p>Décrire la nature des données, leur format (SANDRE ou autre), les éventuelles valeurs standards et leur visibilité en fonction du type d'exécution (l'ensemble des données doivent être visible pour la <i>Simulation</i>, visible en fonction de la catégorie d'utilisateur pour l'<i>Évaluation</i>).</p> <p>Les catégories d'utilisateurs définies sont : le Grand public, les Gestionnaires, Expert (Expert institutionnel, Expert scientifique.</p>				
Code	Nom	Catégorie utilisateur	Description	
masse_eau	Nom de masse d'eau	E, Gd public, Gestionnaires	Liste des masses d'eau échantillonnées	
saison	Saison	E, Gd public, Gestionnaires	Saison à laquelle a été effectuée chaque pêche	
salinite-c	Classes de salinité	E, Gd public, Gestionnaires	Totalité des classes de salinité disponibles (calculées à partir de valeurs mesurées sur le terrain ou estimées)	
trait_id	Enregistrement	E	Identifiant unique de chaque événement de pêche / enregistrement	
superficie-c	Taille du système	E	Classes de taille des systèmes	
duree	Durée	E	Temps de pêche en minutes	
materiel_nom	Nom de l'engin	E	Nom de l'engin de pêche utilisé	
Dens-CA	Densité CA	E	Densité de poissons migrateurs capturés dans l'échantillon considéré	

Ntrait	Nombre de traits	E	Nombre de traits effectués dans la masse d'eau, saison et classe de salinité considérées
Densité-CA	Valeur de la métrique densité de CA	E, Gd public, Gestionnaires	Moyenne des log-densités de poissons migrateurs pour une masse d'eau, une saison et une classe de salinité considérées
Nscore	Nombre de scores	E	Nombre de scores calculés avec au moins 6 traits dans la masse d'eau, saison et classe de salinité considérées
som-score-CA	Somme des scores de CA	E	Somme des scores concernant la métrique de densité des poissons migrateurs dans la masse d'eau, saison et classe de salinité considérées
note-CA	Note pour la métrique densité de CA	E, Gd public, Gestionnaires	Note (normalisée entre 0 et 1) obtenue pour une masse d'eau pour la métrique de densité de poissons migrateurs

DONNEES D'ENTREE

Données externes utilisées (liste exhaustive): (données disponibles dans le SIE ou d'autres bases, données de contexte) décrire la nature des données, leur format (SANDRE ou autre), leurs éventuelles valeurs standard (ex codes des districts), ...

Ce sont les données dont l'indicateur a besoin pour pouvoir s'exécuter. Ces données peuvent être de différents types : Métrique, indice, données d'observation, données de contexte. Ces données sont codées, nommées, typées et décrites de manière unitaire afin de connaître leur forme précise et les caractéristiques liées. (ex : type de masse d'eau, température caractéristique du point de contrôle, etc.). Elles pourront aussi figurer parmi les données de sortie issues de l'exécution de l'indicateur.

Vous indiquerez le type de chaque donnée d'entrée selon la nomenclature suivante :

1. Métrique – M
2. Indice – I
3. Donnée d'observation – O
4. Donnée de contexte – C

Code	Nom	Type	Description
masse_eau	Nom de masse d'eau	C	Liste des masses d'eau échantillonnées
systeme	Système (estuaire ou lagune)	C	Liste des systèmes (certains systèmes sont composés de plusieurs masses d'eau)
ecoregion	Ecorégion	C	Ecorégion (région géographique) à laquelle appartient chaque système
superficie	Superficie	C	Superficie en km ² de chaque masse d'eau
trait_id	Enregistrement	C	Identifiant unique de chaque événement de

			pêche / enregistrement
saison	Saison	C	Saison à laquelle a été effectuée chaque pêche
engin	Engin	C	Identifiant unique de l'engin de pêche utilisé
materiel_nom	Nom de l'engin	C	Nom de l'engin de pêche utilisé
commentaire	Commentaire	C	Texte décrivant les éventuels commentaires fait lors de l'événement de pêche ou l'invalidité du trait (ex : "INVALIDE : <i>raison</i> " ; "Protocole non respecté : <i>raison</i> ")
duree	Durée	C	Temps de pêche en minutes
pos_deb_lat	Latitude de début	C	Latitude du point de début de pêche, en degrés, minutes et millièmes, (WGS84)
pos_deb_long	Longitude de début	C	Longitude du point de début de pêche, en degrés, minutes et millièmes, (WGS84)
pos_fin_lat	Latitude de fin	C	Latitude du point de fin de pêche, en degrés, minutes et millièmes, (WGS84)
pos_fin_long	Longitude de fin	C	Longitude du point de fin de pêche, en degrés, minutes et millièmes, (WGS84)
salinite	Salinité	C	Salinité de l'eau relevée lors de la pêche en psu
salinite_classe	Classe de salinité	C	Classe de salinité (1, 2 ou 3) qui a été estimée pour certaines pêches lorsque la salinité n'a pas été mesurée
nom	Nom des espèces	O	Liste des espèces capturées
nt	Abondance	M	Nombre total de poissons capturés dans chaque échantillon

Origine des données utilisées

Recenser les banques de données SIE ou autres contenant ou mettant à disposition les données d'entrée utilisées. Préciser leur gestionnaire, mettre éventuellement en annexe les structures des données si elles sont connues. Indiquer pour la phase de simulation initiale si des banques locales doivent être utilisées.

Banques de données SIE (cible)

Banque	Informations sur l'origine et la structure des données utilisées

Banque de données hors SIE (provisoire, non rattachée au SIE)

Banque	Informations sur l'origine et la structure des données utilisées
Actuellement Base Poisson DCE	Base de données du Cemagref de Bordeaux sur les captures de poisson dans les eaux de transition pour la Directive Cadre européenne sur l'Eau
En prévision Quadriga 2, Ifremer	Base de données pour la bancarisation des données des éléments de qualité biologique dans les eaux côtières et de transition

VARIABLES

Description : Entité symbolique dont la valeur est susceptible d'être modifiée pour l'exécution de l'indicateur (par ex : valeur du percentile à calculer, substance à évaluer, etc.). elle peut également représenter une modalité de calcul (percentile ou moyenne arithmétique). Sa représentation doit cependant être numérique (percentile = 0 moyenne = 1 : ceci pourrait signifier, par exemple, que l'indicateur doit fonctionner selon une moyenne et non selon un percentile)

Code	Nom	Valeur par défaut	Description
------	-----	-------------------	-------------

LE FONCTIONNEMENT DE L'INDICATEUR

Description de l'algorithme de calcul :

1- Calculs préliminaires

Nous allons tout d'abord procéder à des calculs intermédiaires nécessaires pour la suite des calculs.

Classe de salinité : notée salinite-c peut prendre la valeur 1 ou 3.

Pour la majorité des enregistrements (trait_id), les valeurs de salinité sont mesurées (salinite). La valeur intermédiaire salinite-ccal est d'abord calculée de façon à séparer les valeurs de salinité en **2 classes** selon les critères suivants :

- salinite-ccal prend la valeur "1" si salinite est \leq à 18
- salinite-ccal prend la valeur "3" si salinite est $>$ à 18
- salinite-ccal reste vide si salinite est vide.

A ces classes de salinité calculées (salinite-ccal) sont ajoutées les classes de salinité qui ont été estimées pour certains enregistrements (salinite_classe) lorsque la valeur de salinité était manquante dans la base de données. Ceci permet de regrouper dans le même champ (salinite-c) tous les renseignements disponibles concernant les classes de salinité :

si salinite-ccal est pas vide
alors salinite-c = salinite-ccal
sinon si salinite-ccal est vide ET salinite-classe est pas vide
alors si salinite-classe=3 alors salinite-c=3 sinon salinite-c=1
sinon si salinite-ccal est vide ET salinite-classe est vide
alors salinite-c reste vide

Classe de taille des systèmes lagunaires : notée superficie-c peut prendre la valeur "petit" ou "grand".

A la différence des estuaires, chaque lagune n'est constituée que d'une seule masse d'eau. La classe de taille est attribuée en fonction de la superficie (variable : superficie) de la lagune (en km²).

La variable superficie-c est créée selon les critères suivants :

- superficie-c prend la valeur "petit" lorsque superficie est \leq à 50 km²
- superficie-c prend la valeur "grand" lorsque superficie est $>$ à 50 km²

2- Sélection des enregistrements valides

La deuxième étape est de sélectionner les enregistrements (trait_id) pour lesquels la densité sera calculée. Cette sélection se fait sur les champs engin, saison, ecoregion et commentaire ainsi que sur le calcul intermédiaire salinite-c.

Un trait_id est conservé si et seulement si :

engin est égal à "CAP1"

ET
saison est égale "printemps" OU "automne"

ET
ecoregion est égale à "MED"

ET
l'expression "INVALIDE" OU l'expression "protocole non respect" ne figurent pas dans la chaîne de caractères du champ commentaire

ET
(duree >= 1200 ET duree <= 1680)

ET
salinite-c est non vide

Tous les trait_id répondant à ces critères doivent être conservés même s'ils ne contiennent aucun poisson. Ces trait_id conservés sont appelés trait_id_valide pour une meilleure compréhension des calculs qui suivent.

3- Calculs des log-densités

L'étape suivante consistera à calculer pour chaque trait_id_valide la log-densité de poissons migrateurs.

Sélection des poissons migrateurs

Pour chacun des trait_id_valide, il faut tout d'abord sélectionner uniquement les espèces (nom) dont la guildes (GuildesEcologique) est celle des espèces migratrices (appelées "DIA" dans le tableau GuildesEcologique mais qui correspond à l'abréviation CA dans cette fiche de spécification).

Calcul de la densité de CA pour chaque trait_id_valide : notée Dens-CA

Dens-CA est égale à la somme des abondances (nt) des espèces CA capturées dans le trait_id_valide.

Calcul de la log-densité de CA : notée log-Dens-CA

Toujours pour chacun des trait_id_valide, log-Dens-CA est égal au logarithme népérien de la densité de CA + 1:

$$\text{log-Dens-CA} = \ln (\text{Dens-CA} + 1)$$

4- Calcul de la métrique

Dans la section précédente, les calculs étaient effectués pour chaque trait_id_valide. Dans cette partie, nous allons procéder à un regroupement de ces valeurs en fonction de la masse d'eau, de la saison et de la classe de salinité de façon à pouvoir comparer ces valeurs aux seuils fournis.

Les calculs suivants sont effectués pour chaque combinaison masse_eau x saison x salinite-c

Par exemple : pour la masse d'eau Méjean, au printemps et dans la salinite-c 3.

Calcul du nombre total d'enregistrements effectués : noté Ntrait

Pour chaque combinaison (i) de masse_eau x saison x salinite-c,

Ntrait_(i) correspond au nombre total de trait_id_valide appartenant à la combinaison (i)

Valeur de la métrique de densité de CA : notée Densité-CA (donnée de sortie)

Pour chaque combinaison (i) de masse_eau x saison x salinite-c, Densité-CA est égale à la moyenne des log-densités de poissons migrateurs, c'est-à-dire à la somme des log-densités de CA (log-Dens-CA) appartenant à (i) divisée par le nombre total de trait_id effectués dans (i).

$$\text{moy-log-Dens-CA}_{(i)} = [\text{somme}(\text{log-Dens-CA}_{(i)})] / \text{Ntrait}_{(i)}$$

5- Comparaison aux seuils

Les valeurs de la métrique Densité-CA doivent être comparées aux seuils appropriés du tableau seuilsCALagune joint en annexe. En effet, les seuils fournis dépendent de la saison, de la classe de salinité, d'un paramètre de connectivité à la mer et de la taille de la lagune. Le paramètre de connectivité (section) est fourni dans un tableau supplémentaire (ParamLagunes), il représente la classe de taille des graus (chenaux reliant la lagune à la mer).

Rappel : pour chaque masse d'eau, la classe de taille (petit ou grand) a été calculée dans superficie-c.

Pour chaque combinaison (i) de masse_eau_(m) x saison_(s) x salinite-c_(sc) la valeur de Densité-CA doit être comparée aux seuils définis pour la classe de taille (superficie-c) et la section (section) correspondant à cette masse d'eau(m), lorsque la saison (saison) est égale à (s) et la classe de salinité (salinite-c) est égale à (sc).

Par exemple : la Densité-CA de la combinaison (Méjean x printemps x classe de salinité 3) doit être comparée aux seuils définis pour les petites lagunes ayant une petite section, au printemps et dans la classe de salinité 3 ; c'est-à-dire les seuils correspondant à :

superficie-c = petit
section = petite
saison = printemps
salinite-c = 3

Ainsi en fonction de la valeur de Densité-CA, un score (noté score-CA) sera attribué à chaque combinaison (i) de masse_eau x saison x salinite-c selon les critères suivant :

- si seuilS2 <= Densité-CA < seuilS3 alors score-CA prend la valeur 2
- si seuilS1 n'est pas null alors
 - si (Densité-CA < seuilS1) alors score-CA prend la valeur 0
 - sinon
 - si (Densité-CA < seuilS2) alors score-CA prend la valeur 1
 - sinon #(si seuilS1 est null)#
 - si(Densité-CA < seuilS2) alors score-CA prend la valeur 0

- si seuilS4 n'est pas null alors
 - si(Densité-CA >= seuilS4) alors score-CA prend la valeur 4
 - sinon
 - si(Densité-CA >= seuilS3) alors score-CA prend la valeur 3
 - sinon #(si seuilS4 est null)#
 - si(Densité-CA >= seuilS3) alors score-CA prend la valeur 4

6- Combinaison des scores

Cette combinaison se fait à l'échelle des masses d'eau. Pour chacune des masses d'eau (masse_eau), les scores (score-CA) obtenus pour les différentes saisons et classes de salinité seront additionnés puis divisés par le score maximal pouvant être atteint.

Calcul du nombre de scores pour chaque masse d'eau : noté Nscore

Pour chacune des masses d'eau (m) recensées dans les combinaisons masse_eau x saison x salinite-c, Nscore est égal au nombre de scores (score-CA) que l'on a calculé pour la masse d'eau (m) en question.

$$Nscore_{(m)} = \text{nombre} (score-CA_{(i)})_{(m)}$$

Par exemple : si pour Méjean, on a des scores pour les 2 saisons et les 4 classes de salinité alors Nscore (Méjean) sera égal à 4 (2 saisons x 2 classes).

Calcul de la somme des scores obtenus pour chaque masse d'eau : notée som-score-CA

Toujours pour chacune des masses d'eau (m) recensées dans les combinaisons masse_eau x saison x salinite-c, som-score-CA est égale à la somme des valeurs des différents scores (score-CA) que l'on a calculé pour la masse d'eau (m) en question.

$$som-score-CA_{(m)} = \text{somme} (score-CA_{(i)})_{(m)}$$

Note obtenue pour la métrique densité de CA pour chaque masse d'eau : note-CA (donnée de sortie)

Il s'agit de donner une note globale comprise entre 0 et 1 pour la masse d'eau (m) concernant la métrique des poissons migrateurs. Cette note sera reprise pour le calcul du score final de l'indicateur (voir Fiche métrique INDIC_POISSON_LAG).

$$note-CA_{(m)} = som-score-CA_{(m)} / (4 \times Nscore_{(m)})$$

Règle d'agrégation temporelle :

Décrire ici la mécanique utilisée par l'indicateur pour effectuer l'agrégation temporelle des résultats au point de contrôle.

Il s'agit d'une moyenne des EQR (note-CA) obtenus pour chaque année donc chaque granularité temporelle de 12 mois.

Les principes de fonctionnement de l'indicateur :

Coefficients et paramètres nécessaires à l'indicateur (à joindre en annexe)

Les différents types de tableaux sont :

P : Tableau de paramètres ou de coefficients

S : Tableau de valeurs de seuils

Type	Code	Description du coefficient
P	GuildeEcologique	Tableau définissant la guilde écologique à laquelle appartient chaque taxon. DIA : Migrateurs ; ER : Espèces autochtones ; FW : Espèces dulçaquicoles ; MJ : Juvéniles marins ; MA : Espèces marines occasionnelles ; MS : Migrants marins saisonniers
S	seuilsCALagune	Tableau regroupant les paires de seuils définis pour la métrique de densité des poissons migrateurs en fonction de la taille de la lagune (superficie-c), de la section (section), de la saison (saison) et de la classe de salinité (salinite-c)
P	ParamLagunes	Tableau décrivant pour chaque masse d'eau (point de contrôle) la valeur de paramètres supplémentaires nécessaires au calcul de l'indicateur : section qui peut prendre 2 valeurs (petite ou grande) représente la classe de taille des graus reliant la lagune à la mer.

Description des limites de fonctionnement de l'indicateur

Description des conditions sur les données (reprendre les codes des tableaux précédents), modalités des cas de dégradation du fonctionnement et stratégies de gestion des cas dégradés, messages aux utilisateurs...

Identification des erreurs et description des traitements d'erreurs :

Code	Message d'erreur à afficher à l'utilisateur
EF58	Lorsque qu'aucun enregistrement n'est valide pour une masse d'eau ou qu'aucun échantillonnage n'a été effectué sur cette masse d'eau : les « Densité-CA » pour chaque combinaison saison x salinite-c et la « note-CA » ne peuvent être calculées, un message d'erreur indique " Aucun enregistrement valide pour ce point de contrôle. "
EF30	Si pour l'une des combinaisons saison x salinite-c, il n'y a pas de traits valides, un message d'erreur indique " Aucun échantillonnage pour au moins une combinaison Masse d'eau-Saison-Salinité. "

ANNEXES

Tableau : GuildeEcologique

GuildeEcologique GuildeEcologique Tableau de
paramètre

	CdSandre	Trait
2009		2009
2010		2010
2011		2011 DIA
2012		2012
2014		2014 DIA
2030		2030
2031		2031
2032		2032 DIA
2035		2035 ER
2036		2036
2037		2037
2038		2038 DIA
2039		2039
2040		2040
2041		2041 ER
2042		2042 ER
2045		2045 FW
2050		2050 FW
2053		2053 FW
2054		2054
2055		2055 DIA
2056		2056 DIA
2057		2057 DIA
2060		2060 MJ
2062		2062 MJ
2064		2064 MJ
2067		2067
2069		2069
2071		2071
2074		2074 MA
2079		2079
2080		2080
2083		2083
2084		2084 FW
2085		2085
2086		2086 FW
2088		2088 FW
2090		2090 FW
2095		2095
2096		2096 FW
2099		2099 FW
2100		2100
2102		2102 FW
2104		2104 FW
2108		2108
2110		2110 FW
2113		2113 FW
2117		2117 FW
2119		2119
2120		2120 FW
2121		2121 FW
2122		2122 FW
2124		2124
2125		2125
2129		2129 FW

2131	2131
2132	2132
2133	2133 FW
2135	2135 FW
2137	2137 FW
2142	2142 ER
2147	2147
2148	2148 MS
2151	2151 FW
2152	2152
2153	2153
2154	2154 ER
2158	2158 MJ
2160	2160 MJ
2161	2161
2162	2162 MJ
2164	2164
2165	2165 ER
2167	2167 FW
2168	2168 ER
2170	2170 ER
2172	2172 ER
2173	2173 ER
2174	2174 ER
2176	2176
2177	2177 FW
2178	2178 DIA
2179	2179
2180	2180 DIA
2181	2181 DIA
2182	2182 DIA
2183	2183 DIA
2184	2184
2185	2185 DIA
2188	2188 DIA
2191	2191 FW
2193	2193 FW
2195	2195 FW
2200	2200 ER
2201	2201
2203	2203 DIA
2204	2204
2205	2205 MJ
2207	2207
2208	2208 ER
2210	2210 MA
2211	2211 MA
2212	2212
2216	2216 FW
2219	2219
2220	2220 DIA
2221	2221 DIA
2224	2224 DIA
2231	2231 MS
2233	2233 MJ
2234	2234 MJ

2235	2235 MJ
2237	2237
2238	2238 FW
2240	2240 MJ
2241	2241 MJ
2242	2242
2243	2243 ER
2244	2244 ER
2247	2247
3218	3218
3267	3267 DIA
3347	3347
3362	3362
3369	3369
3370	3370 ER
3372	3372
3375	3375 MJ
3378	3378 MS
3384	3384 MA
3386	3386
3388	3388
3389	3389 MA
3391	3391
3393	3393
3398	3398
3400	3400
3402	3402
3404	3404
3406	3406 MA
3410	3410 MA
3415	3415 ER
3417	3417
3418	3418 ER
3421	3421
3422	3422
3424	3424
3427	3427 MJ
3428	3428 MA
3429	3429 ER
3430	3430 ER
3431	3431
3445	3445 ER
3451	3451 ER
3455	3455 ER
3456	3456 ER
3459	3459 MA
3461	3461 MA
3463	3463 MA
3466	3466 ER
3471	3471
3472	3472 MA
3473	3473 MJ
3475	3475 MA
3481	3481
3482	3482 MJ
3485	3485 MA

3490	3490	MA
3492	3492	MJ
3498	3498	
3501	3501	ER
3505	3505	MA
3506	3506	MA
3507	3507	MA
3512	3512	
3514	3514	
3518	3518	MJ
3520	3520	
3523	3523	
3531	3531	MJ
3533	3533	MA
3535	3535	MA
3537	3537	MA
3539	3539	MA
3540	3540	MA
3541	3541	MJ
3544	3544	ER
3546	3546	ER
3548	3548	MA
3549	3549	
3551	3551	
3553	3553	ER
3557	3557	
3562	3562	MJ
3563	3563	MJ
3567	3567	MA
3568	3568	
3569	3569	ER
3570	3570	ER
3571	3571	ER
3577	3577	MA
3581	3581	
3584	3584	
3585	3585	
3588	3588	
3589	3589	
3591	3591	
3592	3592	
3593	3593	
3594	3594	MA
3599	3599	
3602	3602	MA
3609	3609	
3610	3610	
3614	3614	
5070	5070	
5074	5074	
5207	5207	
5208	5208	FW
19435	19435	
19443	19443	ER
19444	19444	ER
19445	19445	

19446	19446 ER
19447	19447 ER
19448	19448 ER
19449	19449 ER
19450	19450 ER
19451	19451 MA
19452	19452 ER
19453	19453
19454	19454 ER
19455	19455 MJ
19456	19456 MA
19457	19457
19458	19458 ER
19459	19459 ER
19460	19460 MA
19461	19461 MA
19462	19462 ER
19463	19463 ER
19464	19464 ER
19465	19465 MJ
19466	19466
19467	19467
19468	19468 MJ
19469	19469 MA
19470	19470
19471	19471 ER
19472	19472 ER
19473	19473 ER
19474	19474 MA
19475	19475 MJ
19476	19476
19477	19477
19478	19478
19479	19479 MA
19480	19480 MA
19481	19481 MA
19482	19482 MJ
19483	19483 MJ
19484	19484 MJ
19485	19485 ER
19486	19486
19487	19487
19488	19488 ER
19489	19489 MA
19490	19490 ER
19491	19491 ER
19492	19492 ER
19493	19493 ER
19494	19494 ER
19511	19511 FW
19512	19512 DIA
20736	20736 MA
20738	20738 MA
20740	20740 MA
20741	20741 MJ
20742	20742 ER

20743	20743 ER
20744	20744 MA
20745	20745 MA
20746	20746 MJ

Tableau : seuilsCALagune

		Tableau regroupant les paires de seuils définis pour la métrique de densité des poissons migrateurs amphihalins en fonction de la taille de la lagune (superficie-c), la taille de la section (section), de la saison (saison) et de la classe de salinité (salinite-c)						
seuilsCALagune	seuilsCALagune	seuils	salinite					
	superficieC	section	saison	C	seuilS1	seuilS2	seuilS3	seuilS4
grandgrandeautomne1	grand	grande	automne	1		0.95	1.29	
grandgrandeautomne3	grand	grande	automne	3		0.95	1.29	
grandgrandeprintemps1	grand	grande	printemps	1		0.52	0.85	
grandgrandeprintemps3	grand	grande	printemps	3		0.52	0.85	
grandpetiteautomne1	grand	petite	automne	1	2.48	2.52	2.84	
grandpetiteautomne3	grand	petite	automne	3	2.48	2.50	2.84	
grandpetiteprintemps1	grand	petite	printemps	1	2.03	2.09	2.41	
grandpetiteprintemps3	grand	petite	printemps	3	2.03	2.06	2.41	
petitgrandeautomne1	petit	grande	automne	1		0.95	1.29	
petitgrandeautomne3	petit	grande	automne	3		0.95	1.29	
petitgrandeprintemps1	petit	grande	printemps	1		0.52	0.85	
petitgrandeprintemps3	petit	grande	printemps	3		0.52	0.85	
petitpetiteautomne1	petit	petite	automne	1	2.48	2.52	2.84	
petitpetiteautomne3	petit	petite	automne	3	2.48	2.50	2.84	
petitpetiteprintemps1	petit	petite	printemps	1	2.03	2.09	2.41	
petitpetiteprintemps3	petit	petite	printemps	3	2.03	2.06	2.41	

Tableau : ParamLagunes

ParamLagunes	ParamLagunes	Tableau de paramètre
	ParamMET	ParamSection
1	FRDT02	grande
2	FRDT03	petite
3	FRDT04	grande
4	FRDT09	petite
5	FRDT10	grande
6	FRDT11a	petite
7	FRDT11b	petite
8	FRDT11c	petite
9	FRDT14a	petite
10	FRDT15a	grande
11	FRET01	petite
12	FRET02	
13	FRET03	
14	FRET04	

Métrique de densité des poissons prédateurs d'invertébrés benthiques dans les lagunes

Nom de l'indicateur : Métrique de densité des poissons prédateurs d'invertébrés benthiques dans les lagunes	Mnémonique : DENS_IB_LAG	Version : 1.0	Code SANDRE :
Profils d'évaluation applicables : <input type="checkbox"/> DCE <input type="checkbox"/> Diagnostic <input type="checkbox"/> Historique Autres ? Préciser :			Élément de qualité : Poissons
Catégorie d'eau concernée : Eaux de transition (Une seule possible)			Thématique : Ichtyofaune
Description succincte de l'indicateur : Cette métrique de densité des poissons prédateurs d'invertébrés benthiques dans les lagunes permet d'évaluer la qualité des masses d'eau lagunaires du point de vue des densités de poissons appartenant à la guildes des espèces prédatrices d'invertébrés benthiques.			Propriétaire de l'indicateur pour le SEEE : Christine Delpech, Cemagref de Bordeaux Coauteurs : Mario Lepage
Références scientifiques de l'indicateur : <u>Réf principale :</u> Girardin, M., Lepage, M., Amara, R., Boët, P., Courrat, A., Delpech, C., Durozoi, B., Laffargue, P., Le Pape, O., Lobry, J., Parlier, E. and Pasquaud, S., 2009. Développement d'un indicateur poisson pour les eaux de transition. Programme Liteau II, Rapport scientifique de fin de contrat, Cemagref, Cestas. <u>Autres réf :</u> Delpech, C., Drouineau, H. and Lepage, M., 2010. Développement de l'indicateur « poisson » pour les eaux de transition, intercalibration européenne et interfaçage avec le système d'évaluation (SEEE). Convention de partenariat ONEMA-Cemagref 2009, Cestas. Delpech, C. and Lepage, M., 2009. Perfectionnements et ajustements de l'indicateur poisson pour les eaux de transition. Convention de partenariat ONEMA-Cemagref 2008, Cestas.		Références du besoin opérationnel auquel répond l'indicateur : Il s'agit des références des documents réglementaires ou non préconisant l'emploi de cet indicateur pour répondre à un besoin donné (par exemple arrêté du 25 janvier 2010 relatif à l'évaluation de l'état des eaux de surface). Un indicateur peut répondre à plusieurs besoins ; dans ce cas, indiquer le(s) besoin(s) majeurs ou le(s) plus fréquents. Dans l'outil d'évaluation, l'indicateur participe ainsi à la composition d'une ou de plusieurs stratégies d'évaluation (exemple stratégie DCE 2009). Renseigner obligatoirement si l'indicateur répond aux prescriptions pour l'évaluation DCE 2009 (mode de calcul, valeurs seuils, ..).	
Granularité temporelle de l'indicateur (en mois) : 12	Autres granularité temporelle standards (en mois) : 24, 36, 48, 60, 72	Nombre d'opérations de contrôle minimal : 1	
Mots clés utilisés dans la description : poissons, prédateurs d'invertébrés benthiques, densité			
Abréviation	Terme	Définition/description	
IB	Prédateurs d'invertébrés benthiques	Ensemble des espèces appartenant à la guildes des poissons prédateurs d'invertébrés benthiques : Espèces se nourrissant exclusivement ou principalement d'invertébrés benthiques.	

DONNEES DE SORTIE : PRODUITS DE L'INDICATEUR (données produites par l'indicateur: qui seront valorisées dans le SEEE). <u>Liste exhaustive</u>				
Produit Principal				
Code	Nom	Type	Forme	Description
Densité-IB	Valeur de la métrique densité de IB	M	<input type="checkbox"/> VB <input type="checkbox"/> EQR <input type="checkbox"/> CE	Moyenne des log-densités de poissons prédateurs d'invertébrés benthiques multipliée par leur fréquence d'occurrence pour une masse d'eau, une saison et une classe de salinité définies
note-IB	Note pour la métrique densité de IB	M	<input type="checkbox"/> VB <input type="checkbox"/> EQR <input type="checkbox"/> CE	Note (normalisée entre 0 et 1) obtenue pour une masse d'eau pour la métrique de densité de poissons prédateurs d'invertébrés benthiques
Les autres produits de l'indicateur				
<p>Ces produits supplémentaires peuvent être des Données de calcul intermédiaires (qui peuvent être des considérées comme une métrique), des Données explicatives, des Données de traçabilité, des Données d'incertitude,...</p> <p>Toutes ces données sont affichables dans le ou les tableaux des résultats de la simulation et dans le rapport d'évaluation de l'outil d'évaluation.</p> <p>Décrire la nature des données, leur format (SANDRE ou autre), les éventuelles valeurs standards et leur visibilité en fonction du type d'exécution (l'ensemble des données doivent être visible pour la <i>Simulation</i>, visible en fonction de la catégorie d'utilisateur pour l'<i>Évaluation</i>).</p> <p>Les catégories d'utilisateurs définies sont : le Grand public, les Gestionnaires, Expert (Expert institutionnel, Expert scientifique).</p>				
Code	Nom	Catégorie utilisateur	Description	
masse_eau	Nom de masse d'eau	E, Gd public, Gestionnaires	Liste des masses d'eau échantillonnées	
saison	Saison	E, Gd public, Gestionnaires	Saison à laquelle a été effectuée chaque pêche	
salinite-c	Classes de salinité	E, Gd public, Gestionnaires	Totalité des classes de salinité disponibles (calculées à partir de valeurs mesurées sur le terrain ou estimées)	
trait_id	Enregistrement	E	Identifiant unique de chaque événement de pêche / enregistrement	
superficie-c	Taille du système	E	Classes de taille des systèmes	
duree	Durée	E	Temps de pêche en minutes	
materiel_nom	Nom de l'engin	E	Nom de l'engin de pêche utilisé	
Dens-IB	Densité IB	E	Densité de poissons prédateurs d'invertébrés benthiques capturés dans l'échantillon	

			considéré
Ntrait	Nombre de traits	E	Nombre de traits effectués dans la masse d'eau, saison et classe de salinité considérées
Densité-IB	Valeur de la métrique densité de IB	E, Gd public, Gestionnaires	Moyenne des log-densités de poissons prédateurs d'invertébrés benthiques pour une masse d'eau, une saison et une classe de salinité considérées
Nscore	Nombre de scores	E	Nombre de scores calculés avec au moins 6 traits dans la masse d'eau, saison et classe de salinité considérées
som-score-IB	Somme des scores de IB	E	Somme des scores concernant la métrique de densité des poissons prédateurs d'invertébrés benthiques dans la masse d'eau, saison et classe de salinité considérées
note-IB	Note pour la métrique densité de IB	E, Gd public, Gestionnaires	Note (normalisée entre 0 et 1) obtenue pour une masse d'eau pour la métrique de densité de poissons prédateurs d'invertébrés benthiques

DONNEES D'ENTREE

Données externes utilisées (liste exhaustive): (données disponibles dans le SIE ou d'autres bases, données de contexte) décrire la nature des données, leur format (SANDRE ou autre), leurs éventuelles valeurs standard (ex codes des districts), ...

Ce sont les données dont l'indicateur a besoin pour pouvoir s'exécuter. Ces données peuvent être de différents types : Métrique, indice, données d'observation, données de contexte. Ces données sont codées, nommées, typées et décrites de manière unitaire afin de connaître leur forme précise et les caractéristiques liées. (ex : type de masse d'eau, température caractéristique du point de contrôle, etc.). Elles pourront aussi figurer parmi les données de sortie issues de l'exécution de l'indicateur.

Vous indiquerez le type de chaque donnée d'entrée selon la nomenclature suivante :

5. Métrique – M
6. Indice – I
7. Donnée d'observation – O
8. Donnée de contexte – C

Code	Nom	Type	Description
masse_eau	Nom de masse d'eau	C	Liste des masses d'eau échantillonnées
systeme	Système (estuaire	C	Liste des systèmes (certains systèmes sont

	ou lagune)		composés de plusieurs masses d'eau)
ecoregion	Ecorégion	C	Ecorégion (région géographique) à laquelle appartient chaque système
superficie	Superficie	C	Superficie en km ² de chaque masse d'eau
trait_id	Enregistrement	C	Identifiant unique de chaque événement de pêche / enregistrement
saison	Saison	C	Saison à laquelle a été effectuée chaque pêche
engin	Engin	C	Identifiant unique de l'engin de pêche utilisé
materiel_nom	Nom de l'engin	C	Nom de l'engin de pêche utilisé
commentaire	Commentaire	C	Texte décrivant les éventuels commentaires fait lors de l'événement de pêche ou l'invalidité du trait (ex : "INVALIDE : <i>raison</i> " ; "Protocole non respecté : <i>raison</i> ")
duree	Durée	C	Temps de pêche en minutes
pos_deb_lat	Latitude de début	C	Latitude du point de début de pêche, en degrés, minutes et millièmes, (WGS84)
pos_deb_long	Longitude de début	C	Longitude du point de début de pêche, en degrés, minutes et millièmes, (WGS84)
pos_fin_lat	Latitude de fin	C	Latitude du point de fin de pêche, en degrés, minutes et millièmes, (WGS84)
pos_fin_long	Longitude de fin	C	Longitude du point de fin de pêche, en degrés, minutes et millièmes, (WGS84)
salinite	Salinité	C	Salinité de l'eau relevée lors de la pêche en psu
salinite_classe	Classe de salinité	C	Classe de salinité (1, 2 ou 3) qui a été estimée pour certaines pêches lorsque la salinité n'a pas été mesurée
nom	Nom des espèces	O	Liste des espèces capturées
nt	Abondance	M	Nombre total de poissons capturés dans chaque échantillon

Origine des données utilisées

Recenser les banques de données SIE ou autres contenant ou mettant à disposition les données d'entrée utilisées. Préciser leur gestionnaire, mettre éventuellement en annexe les structures des données si elles sont connues. Indiquer pour la phase de simulation initiale si des banques locales doivent être utilisées.

Banques de données SIE (cible)

Banque	Informations sur l'origine et la structure des données utilisées
--------	--

--	--

Banque de données hors SIE (provisoire, non rattachée au SIE)

Banque	Informations sur l'origine et la structure des données utilisées
--------	--

Actuellement Base Poisson DCE	Base de données du Cemagref de Bordeaux sur les captures de poisson dans les eaux de transition pour la Directive Cadre européenne sur l'Eau
-------------------------------	--

En prévision Quadriges 2, Ifremer	Base de données pour la bancarisation des données des éléments de qualité biologique dans les eaux côtières et de transition
-----------------------------------	--

VARIABLES

Description : Entité symbolique dont la valeur est susceptible d'être modifiée pour l'exécution de l'indicateur (par ex : valeur du percentile à calculer, substance à évaluer, etc.). elle peut également

représenter une modalité de calcul (percentile ou moyenne arithmétique). Sa représentation doit cependant être numérique (percentile = 0 moyenne = 1 : ceci pourrait signifier, par exemple, que l'indicateur doit fonctionner selon une moyenne et non selon un percentile)

Code	Nom	Valeur par défaut	Description

LE FONCTIONNEMENT DE L'INDICATEUR

Description de l'algorithme de calcul :

1- Calculs préliminaires

Nous allons tout d'abord procéder à des calculs intermédiaires nécessaires pour la suite des calculs.

Classe de salinité : notée salinite-c peut prendre la valeur 1 ou 3.

Pour la majorité des enregistrements (trait_id), les valeurs de salinité sont mesurées (salinite). La valeur intermédiaire salinite-ccal est d'abord calculée de façon à séparer les valeurs de salinité en **2 classes** selon les critères suivants :

- salinite-ccal prend la valeur "1" si salinite est \leq à 18
- salinite-ccal prend la valeur "3" si salinite est $>$ à 18
- salinite-ccal reste vide si salinite est vide.

A ces classes de salinité calculées (salinite-ccal) sont ajoutées les classes de salinité qui ont été estimées pour certains enregistrements (salinite_classe) lorsque la valeur de salinité était manquante dans la base de données. Ceci permet de regrouper dans le même champ (salinite-c) tous les renseignements disponibles concernant les classes de salinité :

```

si salinite-ccal est pas vide
    alors salinite-c = salinite-ccal
sinon si salinite-ccal est vide ET salinite-classe est pas vide
    alors si salinite-classe=3 alors salinite-c=3 sinon salinite-c=1
sinon si salinite-ccal est vide ET salinite-classe est vide
    alors salinite-c reste vide
  
```

Classe de taille des systèmes lagunaires : notée superficie-c peut prendre la valeur "petit" ou "grand".

A la différence des estuaires, chaque lagune n'est constituée que d'une seule masse d'eau. La classe de taille est attribuée en fonction de la superficie (variable : superficie) de la lagune (en km²).

La variable superficie-c est créée selon les critères suivants :

- superficie-c prend la valeur "petit" lorsque superficie est \leq à 50 km²
- superficie-c prend la valeur "grand" lorsque superficie est $>$ à 50 km²

2- Sélection des enregistrements valides

La deuxième étape est de sélectionner les enregistrements (trait_id) pour lesquels la densité sera calculée. Cette sélection se fait sur les champs engin, saison, ecoregion et commentaire ainsi que sur le calcul intermédiaire salinite-c.

Un `trait_id` est conservé si et seulement si :

engin est égal à "CAP1"
ET
saison est égale "printemps" OU "automne"
ET
ecoregion est égale à "MED"
ET
l'expression "INVALIDE" OU l'expression "protocole non respect" ne figurent pas dans la chaîne de caractères du champ commentaire
ET
(duree >= 1200 ET duree <= 1680)
ET
salinite-c est non vide

Tous les `trait_id` répondant à ces critères doivent être conservés même s'ils ne contiennent aucun poisson. Ces `trait_id` conservés sont appelés `trait_id_valide` pour une meilleure compréhension des calculs qui suivent.

3- Calculs des log-densités

L'étape suivante consistera à calculer pour chaque `trait_id_valide` la log-densité de poissons prédateurs d'invertébrés benthiques.

Sélection des poissons prédateurs d'invertébrés benthiques

Pour chacun des `trait_id_valide`, il faut tout d'abord sélectionner uniquement les espèces (nom) dont la guild (GuildeTrophique) est celle des espèces prédatrices d'invertébrés benthiques ("IB").

Calcul de la densité de IB pour chaque `trait_id_valide` : notée Dens-IB

Dens-IB est égale à la somme des abondances (nt) des espèces IB capturées dans le `trait_id_valide`.

Calcul de la log-densité de IB : notée log-Dens-IB

Toujours pour chacun des `trait_id_valide`, log-Dens-IB est égal au logarithme népérien de la densité de IB + 1 :

$$\text{log-Dens-IB} = \ln (\text{Dens-IB} + 1)$$

4- Calcul de la métrique

Dans la section précédente, les calculs étaient effectués pour chaque `trait_id_valide`. Dans cette partie, nous allons procéder à un regroupement de ces valeurs en fonction de la masse d'eau, de la saison et de la classe de salinité de façon à pouvoir comparer ces valeurs aux seuils fournis.

Les calculs suivants sont effectués pour chaque combinaison masse_eau x saison x salinite-c

Par exemple : pour la masse d'eau Méjean, au printemps et dans la salinite-c 3.

Calcul du nombre total d'enregistrements effectués : noté Ntrait

Pour chaque combinaison (i) de masse_eau x saison x salinite-c,

Ntrait_(i) correspond au nombre total de trait_id_valide appartenant à la combinaison (i)

Valeur de la métrique de densité de IB : notée Densité-IB (donnée de sortie)

Pour chaque combinaison (i) de masse_eau x saison x salinite-c, Densité-IB est égale à la moyenne des log-densités de poissons prédateurs d'invertébrés benthiques, c'est-à-dire à la somme des log-densités de IB (log-Dens-IB) appartenant à (i) divisée par le nombre total de trait_id effectués dans (i).

$$\text{moy-log-Dens-IB}_{(i)} = [\text{somme}(\text{log-Dens-IB}_{(i)})] / \text{Ntrait}_{(i)}$$

5- Comparaison aux seuils

Les valeurs de la métrique Densité-IB doivent être comparées aux seuils appropriés du tableau seuilsIBLagune joint en annexe. En effet, les seuils fournis dépendent de la saison, de la classe de salinité, d'un paramètre de connectivité à la mer et de la taille de la lagune. Le paramètre de connectivité (section) est fourni dans un tableau supplémentaire (ParamLagunes), il représente la classe de taille des graus (chenaux reliant la lagune à la mer).

Rappel : pour chaque masse d'eau, la classe de taille (petit ou grand) a été calculée dans superficie-c.

Pour chaque combinaison (i) de masse_eau_(m) x saison_(s) x salinite-c_(sc) la valeur de Densité-IB doit être comparée aux seuils définis pour la classe de taille (superficie-c) et la section (section) correspondant à cette masse d'eau(m), lorsque la saison (saison) est égale à (s) et la classe de salinité (salinite-c) est égale à (sc).

Par exemple : la Densité-IB de la combinaison (Méjean x printemps x classe de salinité 3) doit être comparée aux seuils définis pour les petites lagunes ayant une petite section, au printemps et dans la classe de salinité 3 ; c'est-à-dire les seuils correspondant à :

superficie-c = petit
 section = petite
 saison = printemps
 salinite-c = 3

Ainsi en fonction de la valeur de Densité-IB, un score (noté score-IB) sera attribué à chaque combinaison (i) de masse_eau x saison x salinite-c selon les critères suivant :

- si seuilS2 <= Densité-IB < seuilS3 alors score-IB prend la valeur 2
- si seuilS1 n'est pas null alors
 - si (Densité-IB < seuilS1) alors score-IB prend la valeur 0
 - sinon
 - si (Densité-IB < seuilS2) alors score-IB prend la valeur 1
- sinon #(si seuilS1 est null)#
 - si(Densité-IB < seuilS2) alors score-IB prend la valeur 0
- si seuilS4 n'est pas null alors
 - si(Densité-IB >= seuilS4) alors score-IB prend la valeur 4
 - sinon

si(Densité-IB >= seuilS3) alors score-IB prend la valeur 3
sinon #(si seuilS4 est null)#
si(Densité-IB >= seuilS3) alors score-IB prend la valeur 4

6- Combinaison des scores

Cette combinaison se fait à l'échelle des masses d'eau. Pour chacune des masses d'eau (masse_eau), les scores (score-IB) obtenus pour les différentes saisons et classes de salinité seront additionnés puis divisés par le score maximal pouvant être atteint.

Calcul du nombre de scores pour chaque masse d'eau : noté Nscore

Pour chacune des masses d'eau (m) recensées dans les combinaisons masse_eau x saison x salinite-c, Nscore est égal au nombre de scores (score-IB) que l'on a calculé pour la masse d'eau (m) en question.

$$Nscore_{(m)} = \text{nombre} (score-IB_{(i)})_{(m)}$$

Par exemple : si pour Méjean, on a des scores pour les 2 saisons et les 4 classes de salinité alors Nscore (Méjean) sera égal à 4 (2 saisons x 2 classes).

Calcul de la somme des scores obtenus pour chaque masse d'eau : notée som-score-IB

Toujours pour chacune des masses d'eau (m) recensées dans les combinaisons masse_eau x saison x salinite-c, som-score-IB est égale à la somme des valeurs des différents scores (score-IB) que l'on a calculé pour la masse d'eau (m) en question.

$$som-score-IB_{(m)} = \text{somme} (score-IB_{(i)})_{(m)}$$

Note obtenue pour la métrique densité de IB pour chaque masse d'eau : note-IB (donnée de sortie)

Il s'agit de donner une note globale comprise entre 0 et 1 pour la masse d'eau (m) concernant la métrique des poissons prédateurs d'invertébrés benthiques. Cette note sera reprise pour le calcul du score final de l'indicateur (voir Fiche métrique INDIC_POISSON_LAG).

$$note-IB_{(m)} = som-score-IB_{(m)} / (4 \times Nscore_{(m)})$$

Règle d'agrégation temporelle :

Décrire ici la mécanique utilisée par l'indicateur pour effectuer l'agrégation temporelle des résultats au point de contrôle.

Il s'agit d'une moyenne des EQR (note-IB) obtenus pour chaque année donc chaque granularité temporelle de 12 mois.

Les principes de fonctionnement de l'indicateur :

Coefficients et paramètres nécessaires à l'indicateur (à joindre en annexe)

Les différents types de tableaux sont :

P : Tableau de paramètres ou de coefficients

S : Tableau de valeurs de seuils

Type	Code	Description du coefficient
P	GuildeTrophique	Tableau définissant la guilde trophique à laquelle appartient chaque taxon. V : Herbivores ; Z : Zooplanctonophages ; IB : Prédateurs d'invertébrés benthiques ; IS : Prédateurs d'invertébrés suprabenthiques ; F : Piscivores ; O : Omnivores
S	seuilsIBLagune	Tableau regroupant les paires de seuils définis pour la métrique de densité des poissons prédateurs d'invertébrés benthiques en fonction de la taille de la lagune (superficie-c), de la section (section), de la saison (saison) et de la classe de salinité (salinite-c)
P	ParamLagunes	Tableau décrivant pour chaque masse d'eau (point de contrôle) la valeur de paramètres supplémentaires nécessaires au calcul de l'indicateur : section qui peut prendre 2 valeurs (petite ou grande) représente la classe de taille des graus reliant la lagune à la mer.

Description des limites de fonctionnement de l'indicateur

Description des conditions sur les données (reprendre les codes des tableaux précédents), modalités des cas de dégradation du fonctionnement et stratégies de gestion des cas dégradés, messages aux utilisateurs...

Identification des erreurs et description des traitements d'erreurs :

Code	Message d'erreur à afficher à l'utilisateur
EF58	Lorsque qu'aucun enregistrement n'est valide pour une masse d'eau ou qu'aucun échantillonnage n'a été effectué sur cette masse d'eau : les « Densité-IB » pour chaque combinaison saison x salinite-c et la « note-IB » ne peuvent être calculées, un message d'erreur indique " Aucun enregistrement valide pour ce point de contrôle. "
EF30	Si pour l'une des combinaisons saison x salinite-c, il n'y a pas de traits valides, un message d'erreur indique " Aucun échantillonnage pour au moins une combinaison Masse d'eau-Saison-Salinité. "

ANNEXES

Tableau : GuildeTrophique

GuildeTrophique GuildeTrophique Tableau de
CdSandre Trait

2009	2009
2010	2010
2011	2011 parasitic
2012	2012
2014	2014 parasitic
2030	2030
2031	2031
2032	2032 IB
2035	2035 Z
2036	2036
2037	2037
2038	2038 O
2039	2039
2040	2040
2041	2041 Z
2042	2042 Z
2045	2045 IB
2050	2050 IS
2053	2053 F
2054	2054 Z
2055	2055 Z
2056	2056 Z
2057	2057 Z
2060	2060 Z
2062	2062 Z
2064	2064 Z
2067	2067
2069	2069
2071	2071
2074	2074 F
2079	2079
2080	2080
2083	2083
2084	2084
2085	2085
2086	2086 IB
2088	2088 IS
2090	2090 Z
2095	2095
2096	2096 IB
2099	2099 IB
2100	2100
2102	2102 O
2104	2104 V
2108	2108
2110	2110 O
2113	2113 IB
2117	2117 Z
2119	2119
2120	2120 O
2121	2121 O
2122	2122 IS
2124	2124
2125	2125
2129	2129 O
2131	2131

2132	2132
2133	2133 O
2135	2135 O
2137	2137 IB
2142	2142 IB
2147	2147
2148	2148 Z
2151	2151 F
2152	2152
2153	2153
2154	2154 O
2158	2158 IS
2160	2160 F
2161	2161 IS
2162	2162 IS
2164	2164
2165	2165 IB
2167	2167 IS
2168	2168
2170	2170 Z
2172	2172 IB
2173	2173
2174	2174 IB
2176	2176
2177	2177 O
2178	2178 O
2179	2179
2180	2180 O
2181	2181 O
2182	2182 O
2183	2183 O
2184	2184
2185	2185 O
2188	2188 IS
2191	2191 IB
2193	2193 IS
2195	2195 F
2200	2200 IS
2201	2201 IB
2203	2203 IB
2204	2204
2205	2205 IB
2207	2207
2208	2208 IS
2210	2210 IB
2211	2211 IB
2212	2212
2216	2216 IS
2219	2219
2220	2220 IS
2221	2221 IS
2224	2224 IS
2231	2231 IS
2233	2233 IS
2234	2234 IS
2235	2235 IS

2237	2237
2238	2238 F
2240	2240 IB
2241	2241 IB
2242	2242
2243	2243
2244	2244 Z
2247	2247
3218	3218
3267	3267 O
3347	3347
3362	3362
3369	3369
3370	3370 IB
3372	3372
3375	3375 F
3378	3378 F
3384	3384 IS
3386	3386
3388	3388
3389	3389 IS
3391	3391
3393	3393
3398	3398
3400	3400
3402	3402
3404	3404
3406	3406 IS
3410	3410 F
3415	3415 IB
3417	3417
3418	3418 IB
3421	3421
3422	3422
3424	3424
3427	3427 IS
3428	3428 Z
3429	3429 O
3430	3430 IB
3431	3431
3445	3445 Z
3451	3451 IS
3455	3455 IB
3456	3456 IB
3459	3459 IB
3461	3461 IB
3463	3463 IB
3466	3466 O
3471	3471
3472	3472 IB
3473	3473 IB
3475	3475 IS
3481	3481
3482	3482 O
3485	3485 O
3490	3490 IS

3492	3492 IS
3498	3498
3501	3501 O
3505	3505 F
3506	3506 IB
3507	3507 IB
3512	3512
3514	3514
3518	3518 IB
3520	3520
3523	3523
3531	3531 IB
3533	3533 IS
3535	3535 IB
3537	3537 IB
3539	3539 IB
3540	3540 IB
3541	3541 IB
3544	3544 IB
3546	3546 IB
3548	3548 O
3549	3549
3551	3551
3553	3553 IS
3557	3557
3562	3562 IS
3563	3563 IS
3567	3567 Z
3568	3568
3569	3569 Z
3570	3570 Z
3571	3571 F
3577	3577 F
3581	3581
3584	3584
3585	3585
3588	3588
3589	3589
3591	3591
3592	3592
3593	3593
3594	3594 IB
3599	3599
3602	3602 IB
3609	3609
3610	3610
3614	3614
5070	5070
5074	5074
5207	5207
5208	5208 O
19435	19435
19443	19443 O
19444	19444 Z
19445	19445
19446	19446 IS

19447	19447 IS
19448	19448 IS
19449	19449 IS
19450	19450 IS
19451	19451 IS
19452	19452 IS
19453	19453
19454	19454 IS
19455	19455 IS
19456	19456 IS
19457	19457
19458	19458 IB
19459	19459 Z
19460	19460 IS
19461	19461 IB
19462	19462 F
19463	19463 IB
19464	19464 IS
19465	19465 IS
19466	19466
19467	19467
19468	19468 IB
19469	19469 IB
19470	19470
19471	19471 Z
19472	19472 Z
19473	19473 V
19474	19474 IB
19475	19475 V
19476	19476
19477	19477
19478	19478
19479	19479 IB
19480	19480 IS
19481	19481 IS
19482	19482 IS
19483	19483 IS
19484	19484 IS
19485	19485 Z
19486	19486
19487	19487
19488	19488 O
19489	19489 IB
19490	19490 O
19491	19491 IS
19492	19492 IB
19493	19493 IB
19494	19494 IS
19511	19511 IB
19512	19512 parasitic
20736	20736 IS
20738	20738 IS
20740	20740 IS
20741	20741 O
20742	20742 IB
20743	20743 O

20744	20744 F
20745	20745 IS
20746	20746 IB

Tableau : seuilsIBLagune

Tableau regroupant les paires de seuils définis pour la métrique de densité des poissons prédateurs d'invertébrés benthiques en fonction de la taille de la lagune (superficie-c), la taille de la section (section), de la saison (saison) et de la classe de salinité (salinite-c)

seuilsIBLagune	seuilsIBLagune	Tableau de seuils	salinite	seuilS1	seuilS2	seuilS3	seuilS4
	superficieC	section	saison C				
grandgrandeautomne1	grand	grande	automne	1	2.83	3.95	
grandgrandeautomne3	grand	grande	automne	3	1.27	2.36	
grandgrandeprintemps1	grand	grande	printemps	1	4.27	5.39	
grandgrandeprintemps3	grand	grande	printemps	3	2.71	3.8	
grandpetiteautomne1	grand	petite	automne	1	4.17	5.3	
grandpetiteautomne3	grand	petite	automne	3	2.61	3.74	
grandpetiteprintemps1	grand	petite	printemps	1	5.61	6.74	
grandpetiteprintemps3	grand	petite	printemps	3	4.05	5.18	
petitgrandeautomne1	petit	grande	automne	1	0.77	1.9	
petitgrandeautomne3	petit	grande	automne	3	-0.79	0.34	
petitgrandeprintemps1	petit	grande	printemps	1	2.21	3.34	
petitgrandeprintemps3	petit	grande	printemps	3	0.65	1.78	
petitpetiteautomne1	petit	petite	automne	1	2.14	3.25	
petitpetiteautomne3	petit	petite	automne	3	0.58	1.68	
petitpetiteprintemps1	petit	petite	printemps	1	3.58	4.7	
petitpetiteprintemps3	petit	petite	printemps	3	2.02	3.12	

Tableau : ParamLagunes

ParamLagunes	ParamLagunes	Tableau de paramètre
	ParamMET	ParamSection
1	FRDT02	grande
2	FRDT03	petite
3	FRDT04	grande
4	FRDT09	petite
5	FRDT10	grande
6	FRDT11a	petite
7	FRDT11b	petite
8	FRDT11c	petite
9	FRDT14a	petite
10	FRDT15a	grande
11	FRET01	petite
12	FRET02	
13	FRET03	
14	FRET04	

Métrique de densité des poissons zooplanctonophages dans les lagunes

Nom de l'indicateur : Métrique de densité des poissons zooplanctonophages dans les lagunes	Mnémonique : DENS_Z_LAG	Version : 1.0	Code SANDRE :
Profils d'évaluation applicables : <input type="checkbox"/> DCE <input type="checkbox"/> Diagnostic <input type="checkbox"/> Historique Autres ? Préciser :			Élément de qualité : Poissons
Catégorie d'eau concernée : Eaux de transition (Une seule possible)			Thématique : Ichtyofaune
Description succincte de l'indicateur : Cette métrique de densité des poissons zooplanctonophages dans les lagunes permet d'évaluer la qualité des masses d'eau lagunaires du point de vue des densités de poissons appartenant à la guildes des espèces zooplanctonophages.			Propriétaire de l'indicateur pour le SEEE : Christine Delpech, Cemagref de Bordeaux Coauteurs : Mario Lepage
Références scientifiques de l'indicateur : <u>Réf principale :</u> Girardin, M., Lepage, M., Amara, R., Boët, P., Courrat, A., Delpech, C., Durozoi, B., Laffargue, P., Le Pape, O., Lobry, J., Parlier, E. and Pasquaud, S., 2009. Développement d'un indicateur poisson pour les eaux de transition. Programme Liteau II, Rapport scientifique de fin de contrat, Cemagref, Cestas. <u>Autres réf :</u> Delpech, C., Drouineau, H. and Lepage, M., 2010. Développement de l'indicateur « poisson » pour les eaux de transition, intercalibration européenne et interfaçage avec le système d'évaluation (SEEE). Convention de partenariat ONEMA-Cemagref 2009, Cestas. Delpech, C. and Lepage, M., 2009. Perfectionnements et ajustements de l'indicateur poisson pour les eaux de transition. Convention de partenariat ONEMA-Cemagref 2008, Cestas.		Références du besoin opérationnel auquel répond l'indicateur : Il s'agit des références des documents réglementaires ou non préconisant l'emploi de cet indicateur pour répondre à un besoin donné (par exemple arrêté du 25 janvier 2010 relatif à l'évaluation de l'état des eaux de surface). Un indicateur peut répondre à plusieurs besoins ; dans ce cas, indiquer le(s) besoin(s) majeurs ou le(s) plus fréquents. Dans l'outil d'évaluation, l'indicateur participe ainsi à la composition d'une ou de plusieurs stratégies d'évaluation (exemple stratégie DCE 2009). Renseigner obligatoirement si l'indicateur répond aux prescriptions pour l'évaluation DCE 2009 (mode de calcul, valeurs seuils, ...).	
Granularité temporelle de l'indicateur (en mois) : 12	Autres granularité temporelle standards (en mois) : 24, 36, 48, 60, 72	Nombre d'opérations de contrôle minimal : 1	
Mots clés utilisés dans la description : poissons, zooplanctonophages, densité			
Abréviation	Terme	Définition/description	
Z	Benthiques	Ensemble des espèces appartenant à la guildes des poissons zooplanctonophages : espèces qui vivent dans ou sur le substrat	

DONNEES DE SORTIE : PRODUITS DE L'INDICATEUR (données produites par l'indicateur: qui seront valorisées dans le SEEE). **Liste exhaustive**
Produit Principal

Code	Nom	Type	Forme	Description
Densité-Z	Valeur de la métrique densité de Z	M	<input type="checkbox"/> VB <input type="checkbox"/> EQR <input type="checkbox"/> CE	Moyenne des log-densités de poissons zooplanctonophages multipliée par leur fréquence d'occurrence pour une masse d'eau, une saison et une classe de salinité définies
note-Z	Note pour la métrique densité de Z	M	<input type="checkbox"/> VB <input type="checkbox"/> EQR <input type="checkbox"/> CE	Note (normalisée entre 0 et 1) obtenue pour une masse d'eau pour la métrique de densité de poissons zooplanctonophages

Les autres produits de l'indicateur

Ces produits supplémentaires peuvent être des Données de calcul intermédiaires (qui peuvent être des considérées comme une métrique), des Données explicatives, des Données de traçabilité, des Données d'incertitude,...

Toutes ces données sont affichables dans le ou les tableaux des résultats de la simulation et dans le rapport d'évaluation de l'outil d'évaluation.

Décrire la nature des données, leur format (SANDRE ou autre), les éventuelles valeurs standards et leur visibilité en fonction du type d'exécution (l'ensemble des données doivent être visible pour la *Simulation*, visible en fonction de la catégorie d'utilisateur pour l'*Évaluation*).

Les catégories d'utilisateurs définies sont : le Grand public, les Gestionnaires, Expert (Expert institutionnel, Expert scientifique).

Code	Nom	Catégorie utilisateur	Description
masse_eau	Nom de masse d'eau	E, Gd public, Gestionnaires	Liste des masses d'eau échantillonnées
saison	Saison	E, Gd public, Gestionnaires	Saison à laquelle a été effectuée chaque pêche
salinite-c	Classes de salinité	E, Gd public, Gestionnaires	Totalité des classes de salinité disponibles (calculées à partir de valeurs mesurées sur le terrain ou estimées)
trait_id	Enregistrement	E	Identifiant unique de chaque événement de pêche / enregistrement
superficie-c	Taille du système	E	Classes de taille des systèmes
duree	Durée	E	Temps de pêche en minutes
materiel_nom	Nom de l'engin	E	Nom de l'engin de pêche utilisé
Dens-Z	Densité Z	E	Densité de poissons zooplanctonophages capturés dans l'échantillon considéré
Ntrait	Nombre de traits	E	Nombre de traits effectués dans la masse

			d'eau, saison et classe de salinité considérées
Densité-Z	Valeur de la métrique densité de Z	E, Gd public, Gestionnaires	Moyenne des log-densités de poissons zooplanctonophages pour une masse d'eau, une saison et une classe de salinité considérées
Nscore	Nombre de scores	E	Nombre de scores calculés avec au moins 6 traits dans la masse d'eau, saison et classe de salinité considérées
som-score-Z	Somme des scores de Z	E	Somme des scores concernant la métrique de densité des poissons zooplanctonophages dans la masse d'eau, saison et classe de salinité considérées
note-Z	Note pour la métrique densité de Z	E, Gd public, Gestionnaires	Note (normalisée entre 0 et 1) obtenue pour une masse d'eau pour la métrique de densité de poissons zooplanctonophages

DONNEES D'ENTREE

Données externes utilisées (liste exhaustive): (données disponibles dans le SIE ou d'autres bases, données de contexte) décrire la nature des données, leur format (SANDRE ou autre), leurs éventuelles valeurs standard (ex codes des districts), ...

Ce sont les données dont l'indicateur a besoin pour pouvoir s'exécuter. Ces données peuvent être de différents types : Métrique, indice, données d'observation, données de contexte. Ces données sont codées, nommées, typées et décrites de manière unitaire afin de connaître leur forme précise et les caractéristiques liées. (ex : type de masse d'eau, température caractéristique du point de contrôle, etc.). Elles pourront aussi figurer parmi les données de sortie issues de l'exécution de l'indicateur.

Vous indiquerez le type de chaque donnée d'entrée selon la nomenclature suivante :

- 9. Métrique – M
- 10. Indice – I
- 11. Donnée d'observation – O
- 12. Donnée de contexte – C

Code	Nom	Type	Description
masse_eau	Nom de masse d'eau	C	Liste des masses d'eau échantillonnées
systeme	Système (estuaire ou lagune)	C	Liste des systèmes (certains systèmes sont composés de plusieurs masses d'eau)
ecoregion	Ecorégion	C	Ecorégion (région géographique) à laquelle appartient chaque système
superficie	Superficie	C	Superficie en km ² de chaque masse d'eau
trait_id	Enregistrement	C	Identifiant unique de chaque événement de pêche / enregistrement

saison	Saison	C	Saison à laquelle a été effectuée chaque pêche
engin	Engin	C	Identifiant unique de l'engin de pêche utilisé
materiel_nom	Nom de l'engin	C	Nom de l'engin de pêche utilisé
commentaire	Commentaire	C	Texte décrivant les éventuels commentaires fait lors de l'événement de pêche ou l'invalidité du trait (ex : "INVALIDE : <i>raison</i> " ; "Protocole non respecté : <i>raison</i> ")
duree	Durée	C	Temps de pêche en minutes
pos_deb_lat	Latitude de début	C	Latitude du point de début de pêche, en degrés, minutes et millièmes, (WGS84)
pos_deb_long	Longitude de début	C	Longitude du point de début de pêche, en degrés, minutes et millièmes, (WGS84)
pos_fin_lat	Latitude de fin	C	Latitude du point de fin de pêche, en degrés, minutes et millièmes, (WGS84)
pos_fin_long	Longitude de fin	C	Longitude du point de fin de pêche, en degrés, minutes et millièmes, (WGS84)
salinite	Salinité	C	Salinité de l'eau relevée lors de la pêche en psu
salinite_classe	Classe de salinité	C	Classe de salinité (1, 2 ou 3) qui a été estimée pour certaines pêches lorsque la salinité n'a pas été mesurée
nom	Nom des espèces	O	Liste des espèces capturées
nt	Abondance	M	Nombre total de poissons capturés dans chaque échantillon

Origine des données utilisées

Recenser les banques de données SIE ou autres contenant ou mettant à disposition les données d'entrée utilisées. Préciser leur gestionnaire, mettre éventuellement en annexe les structures des données si elles sont connues. Indiquer pour la phase de simulation initiale si des banques locales doivent être utilisées.

Banques de données SIE (cible)

Banque	Informations sur l'origine et la structure des données utilisées

Banque de données hors SIE (provisoire, non rattachée au SIE)

Banque	Informations sur l'origine et la structure des données utilisées
Actuellement Base Poisson DCE	Base de données du Cemagref de Bordeaux sur les captures de poisson dans les eaux de transition pour la Directive Cadre européenne sur l'Eau
En prévision Quadrigé 2, Ifremer	Base de données pour la bancarisation des données des éléments de qualité biologique dans les eaux côtières et de transition

VARIABLES

Description : Entité symbolique dont la valeur est susceptible d'être modifiée pour l'exécution de l'indicateur (par ex : valeur du percentile à calculer, substance à évaluer, etc.). elle peut également représenter une modalité de calcul (percentile ou moyenne arithmétique). Sa représentation doit cependant être numérique (percentile = 0 moyenne = 1 : ceci pourrait signifier, par exemple, que l'indicateur doit fonctionner selon une moyenne et non selon un percentile)

Code	Nom	Valeur par défaut	Description

LE FONCTIONNEMENT DE L'INDICATEUR

Description de l'algorithme de calcul :

1- Calculs préliminaires

Nous allons tout d'abord procéder à des calculs intermédiaires nécessaires pour la suite des calculs.

Classe de salinité : notée salinite-c peut prendre la valeur 1 ou 3.

Pour la majorité des enregistrements (trait_id), les valeurs de salinité sont mesurées (salinite). La valeur intermédiaire salinite-ccal est d'abord calculée de façon à séparer les valeurs de salinité en **2 classes** selon les critères suivants :

- salinite-ccal prend la valeur "1" si salinite est \leq à 18
- salinite-ccal prend la valeur "3" si salinite est $>$ à 18
- salinite-ccal reste vide si salinite est vide.

A ces classes de salinité calculées (salinite-ccal) sont ajoutées les classes de salinité qui ont été estimées pour certains enregistrements (salinite_classe) lorsque la valeur de salinité était manquante dans la base de données. Ceci permet de regrouper dans le même champ (salinite-c) tous les renseignements disponibles concernant les classes de salinité :

```
si salinite-ccal est pas vide
    alors salinite-c = salinite-ccal
sinon si salinite-ccal est vide ET salinite-classe est pas vide
    alors si salinite-classe=3 alors salinite-c=3 sinon salinite-c=1
sinon si salinite-ccal est vide ET salinite-classe est vide
    alors salinite-c reste vide
```

Classe de taille des systèmes lagunaires : notée superficie-c peut prendre la valeur "petit" ou "grand".

A la différence des estuaires, chaque lagune n'est constituée que d'une seule masse d'eau. La classe de taille est attribuée en fonction de la superficie (variable : superficie) de la lagune (en km²).

La variable superficie-c est créée selon les critères suivants :

- superficie-c prend la valeur "petit" lorsque superficie est \leq à 50 km²
- superficie-c prend la valeur "grand" lorsque superficie est $>$ à 50 km²

2- Sélection des enregistrements valides

La deuxième étape est de sélectionner les enregistrements (trait_id) pour lesquels la densité sera calculée. Cette sélection se fait sur les champs engin, saison, ecoregion, commentaire, duree ainsi que sur le calcul intermédiaire salinite-c.

Un trait_id est conservé si et seulement si :

```
engin est égal à "CAP1"
ET
saison est égale "printemps" OU "automne"
```


ET
ecoregion est égale à "MED"

ET
l'expression "INVALIDE" OU l'expression "protocole non respect" ne figurent pas dans la chaîne de caractères du champ commentaire

ET
(duree >= 1200 ET duree <= 1680)

ET
salinite-c est non vide

Tous les trait_id répondant à ces critères doivent être conservés même s'ils ne contiennent aucun poisson. Ces trait_id conservés sont appelés trait_id_valide pour une meilleure compréhension des calculs qui suivent.

3- Calculs des log-densités

L'étape suivante consistera à calculer pour chaque trait_id_valide la log-densité de poissons zooplanctonophages.

Sélection des poissons zooplanctonophages

Pour chacun des trait_id_valide, il faut tout d'abord sélectionner uniquement les espèces (nom) dont la guild (GuildeTrophique) est celle des espèces zooplanctonophages ("Z").

Calcul de la densité de Z pour chaque trait_id_valide : notée Dens-Z

Dens-Z est égale à la somme des abondances (nt) des espèces Z capturées dans le trait_id_valide.

Calcul de la log-densité de Z : notée log-Dens-Z

Toujours pour chacun des trait_id_valide, log-Dens-Z est égal au logarithme népérien de la densité de Z + 1 :

$$\text{log-Dens-Z} = \ln (\text{Dens-Z} + 1)$$

4- Calcul de la métrique

Dans la section précédente, les calculs étaient effectués pour chaque trait_id_valide. Dans cette partie, nous allons procéder à un regroupement de ces valeurs en fonction de la masse d'eau, de la saison et de la classe de salinité de façon à pouvoir comparer ces valeurs aux seuils fournis.

Les calculs suivants sont effectués pour chaque combinaison masse_eau x saison x salinite-c

Par exemple : pour la masse d'eau Méjean, au printemps et dans la salinite-c 3.

Calcul du nombre total d'enregistrements effectués : noté Ntrait

Pour chaque combinaison (i) de masse_eau x saison x salinite-c,

Ntrait_(i) correspond au nombre total de trait_id_valide appartenant à la combinaison (i)

Valeur de la métrique de densité de Z : notée Densité-Z (donnée de sortie)

Pour chaque combinaison (i) de masse_eau x saison x salinite-c, Densité-Z est égale à la moyenne des log-densités de poissons zooplanctonophages, c'est-à-dire à la somme des log-densités de Z (log-Dens-Z) appartenant à (i) divisée par le nombre total de trait_id effectués dans (i).

$$\text{moy-log-Dens-Z}_{(i)} = [\text{somme}(\text{log-Dens-Z}_{(i)})] / \text{Ntrait}_{(i)}$$

5- Comparaison aux seuils

Les valeurs de la métrique Densité-Z doivent être comparées aux seuils appropriés du tableau seuilsZLagune joint en annexe. En effet, les seuils fournis dépendent de la saison, de la classe de salinité, d'un paramètre de connectivité à la mer et de la taille de la lagune. Le paramètre de connectivité (section) est fourni dans un tableau supplémentaire (ParamLagunes), il représente la classe de taille des graus (chenaux reliant la lagune à la mer).

Rappel : pour chaque masse d'eau, la classe de taille (petit ou grand) a été calculée dans superficie-c.

Pour chaque combinaison (i) de masse_eau_(m) x saison_(s) x salinite-c_(sc) la valeur de Densité-Z doit être comparée aux seuils définis pour la classe de taille (superficie-c) et la section (section) correspondant à cette masse d'eau(m), lorsque la saison (saison) est égale à (s) et la classe de salinité (salinite-c) est égale à (sc).

Par exemple : la Densité-Z de la combinaison (Méjean x printemps x classe de salinité 3) doit être comparée aux seuils définis pour les petites lagunes ayant une petite section, au printemps et dans la classe de salinité 3 ; c'est-à-dire les seuils correspondant à :

superficie-c = petit
 section = petite
 saison = printemps
 salinite-c = 3

Ainsi en fonction de la valeur de Densité-Z, un score (noté score-Z) sera attribué à chaque combinaison (i) de masse_eau x saison x salinite-c selon les critères suivant :

- si $\text{seuilS2} \leq \text{Densité-Z} < \text{seuilS3}$ alors score-Z prend la valeur 2
- si seuilS1 n'est pas null alors
 - si $(\text{Densité-Z} < \text{seuilS1})$ alors score-Z prend la valeur 4
 - sinon
 - si $(\text{Densité-Z} < \text{seuilS2})$ alors score-Z prend la valeur 3
- sinon $\#(\text{si } \text{seuilS1} \text{ est null})\#$
 - si $(\text{Densité-Z} < \text{seuilS2})$ alors score-Z prend la valeur 4
- si seuilS4 n'est pas null alors
 - si $(\text{Densité-Z} \geq \text{seuilS4})$ alors score-Z prend la valeur 0
 - sinon
 - si $(\text{Densité-Z} \geq \text{seuilS3})$ alors score-Z prend la valeur 1
- sinon $\#(\text{si } \text{seuilS4} \text{ est null})\#$

si(Densité-Z >= seuilS3) alors score-Z prend la valeur 0

NB : attention ici, l'attribution des scores est inversée par rapport aux autres métriques car la densité de poissons Z réagit positivement face à une augmentation de la pression

6- Combinaison des scores

Cette combinaison se fait à l'échelle des masses d'eau. Pour chacune des masses d'eau (masse_eau), les scores (score-Z) obtenus pour les différentes saisons et classes de salinité seront additionnés puis divisés par le score maximal pouvant être atteint.

Calcul du nombre de scores pour chaque masse d'eau : noté Nscore

Pour chacune des masses d'eau (m) recensées dans les combinaisons masse_eau x saison x salinite-c, Nscore est égal au nombre de scores (score-Z) que l'on a calculé pour la masse d'eau (m) en question.

$$Nscore_{(m)} = \text{nombre} (score-Z_{(i)})_{(m)}$$

Par exemple : si pour Méjean, on a des scores pour les 2 saisons et les 4 classes de salinité alors Nscore (Méjean) sera égal à 4 (2 saisons x 2 classes).

Calcul de la somme des scores obtenus pour chaque masse d'eau : notée som-score-Z

Toujours pour chacune des masses d'eau (m) recensées dans les combinaisons masse_eau x saison x salinite-c, som-score-Z est égale à la somme des valeurs des différents scores (score-Z) que l'on a calculé pour la masse d'eau (m) en question.

$$som-score-Z_{(m)} = \text{somme} (score-Z_{(i)})_{(m)}$$

Note obtenue pour la métrique densité de Z pour chaque masse d'eau : note-Z (donnée de sortie)

Il s'agit de donner une note globale comprise entre 0 et 1 pour la masse d'eau (m) concernant la métrique des poissons zooplanctonophages. Cette note sera reprise pour le calcul du score final de l'indicateur (voir Fiche métrique INDIC_POISSON_LAG).

$$note-Z_{(m)} = som-score-Z_{(m)} / (4 \times Nscore_{(m)})$$

Règle d'agrégation temporelle :

Décrire ici la mécanique utilisée par l'indicateur pour effectuer l'agrégation temporelle des résultats au point de contrôle.

Il s'agit d'une moyenne des EQR (note-Z) obtenus pour chaque année donc chaque granularité temporelle de 12 mois.

Les principes de fonctionnement de l'indicateur :

Coefficients et paramètres nécessaires à l'indicateur (à joindre en annexe)

Les différents types de tableaux sont :

P : Tableau de paramètres ou de coefficients

S : Tableau de valeurs de seuils

Type	Code	Description du coefficient
P	GuildeTrophique	Tableau définissant la guilde trophique à laquelle appartient chaque taxon. V : Herbivores ; Z : Zooplanctonophages ; IB : Prédateurs d'invertébrés benthiques ; IS : Prédateurs d'invertébrés suprabenthiques ; F : Piscivores ; O : Omnivores
S	seuilsZLagune	Tableau regroupant les paires de seuils définis pour la métrique de densité des poissons zooplanctonophages en fonction de la taille de la lagune (superficie-c), de la section (section), de la saison (saison) et de la classe de salinité (salinite-c)
P	ParamLagunes	Tableau décrivant pour chaque masse d'eau (point de contrôle) la valeur de paramètres supplémentaires nécessaires au calcul de l'indicateur : section qui peut prendre 2 valeurs (petite ou grande) représente la classe de taille des graus reliant la lagune à la mer.

Description des limites de fonctionnement de l'indicateur

Description des conditions sur les données (reprendre les codes des tableaux précédents), modalités des cas de dégradation du fonctionnement et stratégies de gestion des cas dégradés, messages aux utilisateurs...

Identification des erreurs et description des traitements d'erreurs :

Code	Message d'erreur à afficher à l'utilisateur
EF58	Lorsque qu'aucun enregistrement n'est valide pour une masse d'eau ou qu'aucun échantillonnage n'a été effectué sur cette masse d'eau : les « Densité-Z » pour chaque combinaison saison x salinite-c et la « note-Z » ne peuvent être calculées, un message d'erreur indique " Aucun enregistrement valide pour ce point de contrôle. "
EF30	Si pour l'une des combinaisons saison x salinite-c, il n'y a pas de traits valides, un message d'erreur indique " Aucun échantillonnage pour au moins une combinaison Masse d'eau-Saison-Salinité. "

ANNEXES

Tableau : GuildeTrophique

GuildeTrophique	GuildeTrophique	Tableau de
	CdSandre	paramètre
		Trait
2009	2009	
2010	2010	
2011	2011	parasitic

2012	2012
2014	2014 parasitic
2030	2030
2031	2031
2032	2032 IB
2035	2035 Z
2036	2036
2037	2037
2038	2038 O
2039	2039
2040	2040
2041	2041 Z
2042	2042 Z
2045	2045 IB
2050	2050 IS
2053	2053 F
2054	2054 Z
2055	2055 Z
2056	2056 Z
2057	2057 Z
2060	2060 Z
2062	2062 Z
2064	2064 Z
2067	2067
2069	2069
2071	2071
2074	2074 F
2079	2079
2080	2080
2083	2083
2084	2084
2085	2085
2086	2086 IB
2088	2088 IS
2090	2090 Z
2095	2095
2096	2096 IB
2099	2099 IB
2100	2100
2102	2102 O
2104	2104 V
2108	2108
2110	2110 O
2113	2113 IB
2117	2117 Z
2119	2119
2120	2120 O
2121	2121 O
2122	2122 IS
2124	2124
2125	2125
2129	2129 O
2131	2131
2132	2132
2133	2133 O
2135	2135 O

2137	2137 IB
2142	2142 IB
2147	2147
2148	2148 Z
2151	2151 F
2152	2152
2153	2153
2154	2154 O
2158	2158 IS
2160	2160 F
2161	2161 IS
2162	2162 IS
2164	2164
2165	2165 IB
2167	2167 IS
2168	2168
2170	2170 Z
2172	2172 IB
2173	2173
2174	2174 IB
2176	2176
2177	2177 O
2178	2178 O
2179	2179
2180	2180 O
2181	2181 O
2182	2182 O
2183	2183 O
2184	2184
2185	2185 O
2188	2188 IS
2191	2191 IB
2193	2193 IS
2195	2195 F
2200	2200 IS
2201	2201 IB
2203	2203 IB
2204	2204
2205	2205 IB
2207	2207
2208	2208 IS
2210	2210 IB
2211	2211 IB
2212	2212
2216	2216 IS
2219	2219
2220	2220 IS
2221	2221 IS
2224	2224 IS
2231	2231 IS
2233	2233 IS
2234	2234 IS
2235	2235 IS
2237	2237
2238	2238 F
2240	2240 IB

2241	2241 IB
2242	2242
2243	2243
2244	2244 Z
2247	2247
3218	3218
3267	3267 O
3347	3347
3362	3362
3369	3369
3370	3370 IB
3372	3372
3375	3375 F
3378	3378 F
3384	3384 IS
3386	3386
3388	3388
3389	3389 IS
3391	3391
3393	3393
3398	3398
3400	3400
3402	3402
3404	3404
3406	3406 IS
3410	3410 F
3415	3415 IB
3417	3417
3418	3418 IB
3421	3421
3422	3422
3424	3424
3427	3427 IS
3428	3428 Z
3429	3429 O
3430	3430 IB
3431	3431
3445	3445 Z
3451	3451 IS
3455	3455 IB
3456	3456 IB
3459	3459 IB
3461	3461 IB
3463	3463 IB
3466	3466 O
3471	3471
3472	3472 IB
3473	3473 IB
3475	3475 IS
3481	3481
3482	3482 O
3485	3485 O
3490	3490 IS
3492	3492 IS
3498	3498
3501	3501 O

3505	3505 F
3506	3506 IB
3507	3507 IB
3512	3512
3514	3514
3518	3518 IB
3520	3520
3523	3523
3531	3531 IB
3533	3533 IS
3535	3535 IB
3537	3537 IB
3539	3539 IB
3540	3540 IB
3541	3541 IB
3544	3544 IB
3546	3546 IB
3548	3548 O
3549	3549
3551	3551
3553	3553 IS
3557	3557
3562	3562 IS
3563	3563 IS
3567	3567 Z
3568	3568
3569	3569 Z
3570	3570 Z
3571	3571 F
3577	3577 F
3581	3581
3584	3584
3585	3585
3588	3588
3589	3589
3591	3591
3592	3592
3593	3593
3594	3594 IB
3599	3599
3602	3602 IB
3609	3609
3610	3610
3614	3614
5070	5070
5074	5074
5207	5207
5208	5208 O
19435	19435
19443	19443 O
19444	19444 Z
19445	19445
19446	19446 IS
19447	19447 IS
19448	19448 IS
19449	19449 IS

19450	19450	IS
19451	19451	IS
19452	19452	IS
19453	19453	
19454	19454	IS
19455	19455	IS
19456	19456	IS
19457	19457	
19458	19458	IB
19459	19459	Z
19460	19460	IS
19461	19461	IB
19462	19462	F
19463	19463	IB
19464	19464	IS
19465	19465	IS
19466	19466	
19467	19467	
19468	19468	IB
19469	19469	IB
19470	19470	
19471	19471	Z
19472	19472	Z
19473	19473	V
19474	19474	IB
19475	19475	V
19476	19476	
19477	19477	
19478	19478	
19479	19479	IB
19480	19480	IS
19481	19481	IS
19482	19482	IS
19483	19483	IS
19484	19484	IS
19485	19485	Z
19486	19486	
19487	19487	
19488	19488	O
19489	19489	IB
19490	19490	O
19491	19491	IS
19492	19492	IB
19493	19493	IB
19494	19494	IS
19511	19511	IB
19512	19512	parasitic
20736	20736	IS
20738	20738	IS
20740	20740	IS
20741	20741	O
20742	20742	IB
20743	20743	O
20744	20744	F
20745	20745	IS
20746	20746	IB

Tableau : seuilsZLagune

Tableau regroupant les paires de seuils définis pour la métrique de densité des poissons zooplanctonophages en fonction de la taille de la lagune (superficie-c), la taille de la section (section), de la saison (saison) et de la classe de salinité (salinite-c)

seuilsZLagune	seuilsZLagune	Tableau de seuils	seuils	salinite C	seuilS1	seuilS2	seuilS3	seuilS4
	superficieC	section	saison					
grandgrandeautomne1	grand	grande	automne	1	6.13	6.22	7.1	
grandgrandeautomne3	grand	grande	automne	3		3.43	4.28	
grandgrandeprintemps1	grand	grande	printemps	1	6.13	6.22	7.1	
grandgrandeprintemps3	grand	grande	printemps	3		3.43	4.28	
grandpetiteautomne1	grand	petite	automne	1		5.17	6.04	
grandpetiteautomne3	grand	petite	automne	3		2.36	3.25	
grandpetiteprintemps1	grand	petite	printemps	1		5.17	6.04	
grandpetiteprintemps3	grand	petite	printemps	3		2.36	3.25	
petitgrandeautomne1	petit	grande	automne	1	6.13	6.22	7.1	
petitgrandeautomne3	petit	grande	automne	3		3.43	4.28	
petitgrandeprintemps1	petit	grande	printemps	1	6.13	6.22	7.1	
petitgrandeprintemps3	petit	grande	printemps	3		3.43	4.28	
petitpetiteautomne1	petit	petite	automne	1		5.17	6.04	
petitpetiteautomne3	petit	petite	automne	3		2.36	3.25	
petitpetiteprintemps1	petit	petite	printemps	1		5.17	6.04	
petitpetiteprintemps3	petit	petite	printemps	3		2.36	3.25	

Tableau : ParamLagunes

ParamLagunes	ParamLagunes	Tableau de paramètre
	ParamMET	ParamSection
1	FRDT02	grande
2	FRDT03	petite
3	FRDT04	grande
4	FRDT09	petite
5	FRDT10	grande
6	FRDT11a	petite
7	FRDT11b	petite
8	FRDT11c	petite
9	FRDT14a	petite
10	FRDT15a	grande
11	FRET01	petite
12	FRET02	
13	FRET03	
14	FRET04	

Indicateur multimétrique de qualité des lagunes

Nom de l'indicateur : Indicateur multimétrique de qualité des lagunes		Mnémonique : INDIC_POISSON_LAG		Version : 1.0		Code SANDRE :	
Profils d'évaluation applicables : <input type="checkbox"/> DCE <input type="checkbox"/> Diagnostic <input type="checkbox"/> Historique Autres ? Préciser :						Élément de qualité : Poissons	
Catégorie d'eau concernée : Eaux de transition (Une seule possible)						Thématique : Ichtyofaune	
Description succincte de l'indicateur : Indicateur multimétrique de qualité des lagunes concernant la composante "poisson". Cet indicateur est la moyenne des notes obtenues pour chaque métrique.						Propriétaire de l'indicateur pour le SEEE : Christine Delpech, Cemagref de Bordeaux	
						Coauteurs : Mario Lepage	
Références scientifiques de l'indicateur : <u>Réf principale :</u> Delpech, C., Drouineau, H. and Lepage, M., 2010. Développement de l'indicateur « poisson » pour les eaux de transition, intercalibration européenne et interfaçage avec le système d'évaluation (SEEE). Convention de partenariat ONEMA-Cemagref 2009, Cestas. Delpech, C. and Lepage, M., 2009. Perfectionnements et ajustements de l'indicateur poisson pour les eaux de transition. Convention de partenariat ONEMA-Cemagref 2008, Cestas. Girardin, M., Lepage, M., Amara, R., Boët, P., Courrat, A., Delpech, C., Durozoi, B., Laffargue, P., Le Pape, O., Lobry, J., Parlier, E. and Pasquaud, S., 2009. Développement d'un indicateur poisson pour les eaux de transition. Programme Liteau II, Rapport scientifique de fin de contrat, Cemagref, Cestas.				Références du besoin opérationnel auquel répond l'indicateur : Il s'agit des références des documents réglementaires ou non préconisant l'emploi de cet indicateur pour répondre à un besoin donné (par exemple arrêté du 25 janvier 2010 relatif à l'évaluation de l'état des eaux de surface). Un indicateur peut répondre à plusieurs besoins ; dans ce cas, indiquer le(s) besoin(s) majeurs ou le(s) plus fréquents. Dans l'outil d'évaluation, l'indicateur participe ainsi à la composition d'une ou de plusieurs stratégies d'évaluation (exemple stratégie DCE 2009). Renseigner obligatoirement si l'indicateur répond aux prescriptions pour l'évaluation DCE 2009 (mode de calcul, valeurs seuils, ..).			
Granularité temporelle de l'indicateur (en mois) : 12		Autres granularité temporelle standards (en mois) : 24, 36, 48, 60, 72		Nombre d'opérations de contrôle minimal : 1			
Mots clés utilisés dans la description : indicateur multimétrique, lagunes							
Abréviation		Terme		Définition/description			
DONNEES DE SORTIE : PRODUITS DE L'INDICATEUR (données produites par l'indicateur: qui seront valorisées dans le SEEE). <u>Liste exhaustive</u>							
Produit Principal							
Code	Nom	Type	Forme	Description			

INDIC_POISSON_LAG	Indicateur poisson	I	<input type="checkbox"/> VB <input type="checkbox"/> EQR <input type="checkbox"/> CE	Note finale de l'indicateur multimétrique "poisson" (entre 0 et 1) attribuée à la masse d'eau considérée
Etat-Poisson-Lagune	Classe de qualité	I	<input type="checkbox"/> VB <input type="checkbox"/> EQR <input type="checkbox"/> CE type de classification: DCE5BVJOR	Classe de qualité renseignant sur l'état de qualité de la composante "poisson" dans la masse d'eau considérée

Les autres produits de l'indicateur

Ces produits supplémentaires peuvent être des Données de calcul intermédiaires (qui peuvent être des considérées comme une métrique), des Données explicatives, des Données de traçabilité, des Données d'incertitude,...

Toutes ces données sont affichables dans le ou les tableaux des résultats de la simulation et dans le rapport d'évaluation de l'outil d'évaluation.

Décrire la nature des données, leur format (SANDRE ou autre), les éventuelles valeurs standards et leur visibilité en fonction du type d'exécution (l'ensemble des données doivent être visible pour la *Simulation*, visible en fonction de la catégorie d'utilisateur pour l'*Évaluation*).

Les catégories d'utilisateurs définies sont : le Grand public, les Gestionnaires, Expert (Expert institutionnel, Expert scientifique).

Code	Nom	Catégorie utilisateur	Description

DONNEES D'ENTREE

Données externes utilisées (liste exhaustive): (données disponibles dans le SIE ou d'autres bases, données de contexte) décrire la nature des données, leur format (SANDRE ou autre), leurs éventuelles valeurs standard (ex codes des districts), ...

Ce sont les données dont l'indicateur a besoin pour pouvoir s'exécuter. Ces données peuvent être de différents types : Métrique, indice, données d'observation, données de contexte. Ces données sont codées, nommées, typées et décrites de manière unitaire afin de connaître leur forme précise et les caractéristiques liées. (ex : type de masse d'eau, température caractéristique du point de contrôle, etc.). Elles pourront aussi figurer parmi les données de sortie issues de l'exécution de l'indicateur.

Vous indiquerez le type de chaque donnée d'entrée selon la nomenclature suivante :

13. Métrique – M

14. Indice – I

15. Donnée d'observation – O

16. Donnée de contexte – C

Code	Nom	Type	Description
note-CA	Note de la métrique densité de CA	I	Note obtenue pour la masse d'eau (m) considérée concernant la métrique de densité des espèces migratrices
note-IB	Note de la métrique densité de IB	I	Note obtenue pour la masse d'eau (m) considérée concernant la métrique de densité des poissons prédateurs d'invertébrés benthiques
note-Z	Note de la métrique densité de Z	I	Note obtenue pour la masse d'eau (m) considérée concernant la métrique de densité des espèces zooplanctonophages

Origine des données utilisées

Recenser les banques de données SIE ou autres contenant ou mettant à disposition les données d'entrée utilisées. Préciser leur gestionnaire, mettre éventuellement en annexe les structures des données si elles sont connues. Indiquer pour la phase de simulation initiale si des banques locales doivent être utilisées.

Banques de données SIE (cible)

Banque	Informations sur l'origine et la structure des données utilisées

Banque de données hors SIE (provisoire, non rattachée au SIE)

Banque	Informations sur l'origine et la structure des données utilisées

VARIABLES

Description : Entité symbolique dont la valeur est susceptible d'être modifiée pour l'exécution de l'indicateur (par ex : valeur du percentile à calculer, substance à évaluer, etc.). elle peut également représenter une modalité de calcul (percentile ou moyenne arithmétique). Sa représentation doit cependant être numérique (percentile = 0 moyenne = 1 : ceci pourrait signifier, par exemple, que l'indicateur doit fonctionner selon une moyenne et non selon un percentile)

Code	Nom	Valeur par défaut	Description

LE FONCTIONNEMENT DE L'INDICATEUR

Les notes obtenues pour chaque métrique (note-CA, note-IB et note-Z) sont calculées individuellement selon les méthodes décrivant les métriques : DENS_CA_LAG, DENS_IB_LAG et DENS_Z_LAG

Description de l'algorithme de calcul :

L'indicateur multimétrique de qualité d'une masse d'eau lagunaire (m) est basé sur la moyenne des notes obtenues pour chacune des métriques :

$$\text{INDIC_POISSON_LAG}_{(m)} = (\text{note-CA}_{(m)} + \text{note-IB}_{(m)} + \text{note-Z}_{(m)}) / 3$$

La valeur de cet indicateur (INDIC_POISSON_LAG) pour la masse d'eau (m) est comparée aux seuils fournis dans le tableau ClassesLagune afin d'attribuer une classe de qualité à la masse d'eau (Etat-Poisson-Lagune).

Règle d'agrégation temporelle :

Décrire ici la mécanique utilisée par l'indicateur pour effectuer l'agrégation temporelle des résultats au point de contrôle.

Il s'agit d'une moyenne ou médiane des EQR (INDIC_POISSON_LAG) obtenus pour chaque année donc chaque granularité temporelle de 12 mois.

NB : L'agrégation temporelle de cet indice multimétrique prévaut sur celle des métriques de base. C'est-à-dire que si l'on demande de calculer l'indice sur 3 ans (36 mois), les EQR de chaque métrique de base doivent être

calculé pour chaque année (12 mois), puis agrégés en indice multimétrique pour chaque année et c'est l'indice multimétrique de chaque année qui sera agrégé temporellement (donc moyenne ou médiane des 3 années). Les EQR annuels des métriques de base ne seront pas agrégés sur les 3 années.

Les principes de fonctionnement de l'indicateur :

Coefficients et paramètres nécessaires à l'indicateur (à joindre en annexe)

Les différents types de tableaux sont :

P : Tableau de paramètres ou de coefficients

S : Tableau de valeurs de seuils

Type	Code	Description du coefficient
S	ClassesLagune	Ce tableau permet de distinguer 5 classes de qualité en fonction de la valeur de l'indicateur multimétrique.

Description des limites de fonctionnement de l'indicateur

Description des conditions sur les données (reprendre les codes des tableaux précédents), modalités des cas de dégradation du fonctionnement et stratégies de gestion des cas dégradés, messages aux utilisateurs...

Identification des erreurs et description des traitements d'erreurs :

Code	Message d'erreur à afficher à l'utilisateur
EF01	Lorsque la masse d'eau n'a pas été échantillonnée, l'indice ne peut pas être calculé, donc le message suivant apparaît : « Produit principal du résultat manquant »

ANNEXES

Tableau : ClassesLagune (mis en transposé pour plus de clarté)

_TableauSeuilNormalise		1
_TableauSeuilNormalise	CodeClasseEtat1	1
	InclusionBorneSup1 [
	BorneSup1	1
	BorneInf1 0.8	
	InclusionBorneInf1]	
	CodeClasseEtat2	2
	InclusionBorneSup2]	
	BorneSup2 0.8	
	BorneInf2 0.6	
	InclusionBorneInf2]	
	CodeClasseEtat3	3
	InclusionBorneSup3]	
	BorneSup3 0.6	
	BorneInf3 0.4	
	InclusionBorneInf3]	
	CodeClasseEtat4	4
	InclusionBorneSup4]	
	BorneSup4 0.4	
	BorneInf4 0.2	
	InclusionBorneInf4]	
	CodeClasseEtat5	5
	InclusionBorneSup5]	
	BorneSup5 0.2	
	BorneInf5	0
	InclusionBorneInf5]	

ANNEXES : Algorithmes de calcul sous R

ET-Poisson-DENS_CA_LAG

v 0.001

```
#####  
# Densité de poissons prédateurs d'invertébrés benthiques dans les lagunes      #  
# Author: Theunissenb  
#####  
  
#####  
# Variables globales : INTERDICTION FORMELLE DE LES MODIFIER                    #  
# fichierEntree : adresse du fichier d'entrée  
#####  
  
#Association code et label de chaque saison  
ET.SAISON.PRINTEMPS <- c("0","printemps")  
ET.SAISON.AUTOMNE   <- c("2","automne")  
#Les saisons traitées par le modèle  
DENS.CA.LAG.SAISONS <- list(ET.SAISON.PRINTEMPS,ET.SAISON.AUTOMNE)  
  
#Codes des eco-regions masse d'eau  
ET.ECOREGION.MED      <- "MED"  
#Codes des eco-regions masse d'eau traitées par le modèle  
DENS.CA.LAG.ECOREGIONS <- c(ET.ECOREGION.MED)  
  
DENS.CA.LAG.SALINITES <- c("1","3")  
  
#Declaration des tableaux associés au modèle  
DENS.CA.LAG.TAB.SEUIL.CA.LAGUNE.LABEL <- "seuilsCALagune"  
DENS.CA.LAG.TAB.SEUIL.CA.LAGUNE      <- NULL  
  
DENS.CA.LAG.TAB.GUILDE.ECOLOGIQUE.LABEL <- "GuildeEcologique"  
DENS.CA.LAG.TAB.GUILDE.ECOLOGIQUE      <- NULL  
  
DENS.CA.LAG.TAB.PARAMETRE.LAGUNES.LABEL <- "ParamLagunes"  
DENS.CA.LAG.TAB.PARAMETRE.LAGUNES      <- NULL  
  
#Declaration des noms des colonnes des tableaux manipulés  
DENS.CA.LAG.COL.SUPERFICIE      <- "superficieC"  
DENS.CA.LAG.COL.SECTION         <- "section"  
DENS.CA.LAG.COL.SAISON          <- "saison"  
DENS.CA.LAG.COL.SALINITE        <- "saliniteC"  
DENS.CA.LAG.COL.SEUILS1         <- "seuils1"  
DENS.CA.LAG.COL.SEUILS2         <- "seuils2"  
DENS.CA.LAG.COL.SEUILS3         <- "seuils3"  
DENS.CA.LAG.COL.SEUILS4         <- "seuils4"  
DENS.CA.LAG.COL.TRAIT           <- "Trait"  
DENS.CA.LAG.COL.PARAM.SECTION   <- "ParamSection"    # valeur de la section dans le  
tableau de parametre  
DENS.CA.LAG.COL.PARAM.MET       <- "ParamMET"  
  
#Trait du poisson à considérer dans le modèle  
DENS.CA.LAG.TRAIT <- "DIA"  
  
#Initialisation des erreurs  
donneesErreur <- NULL  
DENS.CA.LAG.LISTES.SYSTEMES <- NULL  
  
#Définitions de fonctions internes  
  
# FIXME Declaration temporaire d'une fonction de la bibliotheque RSEEE durant la  
Recette.  
## Récupération de la liste faunistique dans les données d'observation.  
# param operationControle le noeud de l'opération de contrôle.
```

```

# param codesUnite le(s) code(s) de l'unité bio concernant les résultats d'une
observation hydrobiologique, par défaut NULL.
# param listeDonneesContexteHydroElem le(s) libellé(s) des données de contexte
souhaitées, par défaut NULL.
# param bocal indique si une colonne contenant le bocal doit être ajoutée, par
défaut FALSE.
# param listeTypeListe liste des typeListe que la listeHydrobiologique doit
contenir, par défaut NULL.
# Si la listeHydrobiologique n'est pas d'un type compris dans typeListe alors
# l'ensemble des observationHydrobiologiques sous jacentes est ignoré.
# param force si vrai, alors les listes hydrobiologiques n'ayant pas d'observation
hydrobiologique
# apparaissent toutefois avec les données: idTaxon,resultat et codeUniteBio
valorisées à "NULL", par défaut FALSE.
# return la liste des taxons (idTaxon,resultat,codeUniteBio).
# erreur si une donnée est manquante.
findObservationsHydrobiologique <- fonction(operationControle,
      codesUnite=NULL,
      listeDonneesContexteHydroElem=NULL,
      bocal=FALSE,
      listeTypeListe=NULL,
      force=FALSE){
  # initialisation
  listeFaunistique <- NULL
  listesHydrobiologique
  findElement(operationControle,"ListesHydrobiologique")
  if("Erreur" %in% listesHydrobiologique)
    return(listesHydrobiologique)
  listesHydrobiologique
  findElement(listesHydrobiologique,"ListeHydrobiologique")
  if("Erreur" %in% listesHydrobiologique)
    return(listesHydrobiologique)

  # nombre de listesHydrobiologique
  nbListesHydrobiologique <- length(listesHydrobiologique)
  if(nbListesHydrobiologique < 1){
    erreur <- c("Erreur","EF06","Donnée d'observation manquante")
    return(erreur)
  }
  for(i in 1:nbListesHydrobiologique){
    vNumeroOrdre <- NULL
    vBocal <- NULL
    donneeContexte <- NULL
    codeTypeListe <- NULL
    idListeHydrobiologique <- NULL

    listeHydrobiologique <- listesHydrobiologique[i]

    # Recuperation du numero de passage associe a ListeHydrobiologique
    opcHydrobioElementaire
    findElement(listeHydrobiologique,"OPCHydrobioElementaire")
    if("Erreur" %in% opcHydrobioElementaire){ # Si une erreur est survenue
      vNumeroOrdre <- "NULL"
      if(bocal) # le bocal est demande et non disponible
        return(opcHydrobioElementaire) # Envoi de l'erreur
    }
    else{ # si pas d'erreur survenue
      vNumeroOrdre <- findDonneesFromElement(opcHydrobioElementaire,
"numOrdre",convertFromUTF8=FALSE)
      if("Erreur" %in% vNumeroOrdre) # si une erreur est survenue
        vNumeroOrdre <- "NULL"

      # Recherche du bocal associé a ListeHydrobiologique
      if(bocal){
        vBocal <- findDonneesFromElement(opcHydrobioElementaire,
"numRegroupement",convertFromUTF8=FALSE)
        if("Erreur" %in% vBocal) # si une erreur est survenue
          return(vBocal)
      }
    }
  }
}

```

```

# recherche du typeListe de la listeHydrobiologique contenant les
observationHydrobiologiques
if(!is.null(listeTypeListe)){
  typeListe <- findElement(listeHydrobiologique,"TypeListe")
  if("Erreur" %in% typeListe) # si une erreur est survenue
    next
  codeTypeListe
}
findDonneesFromElement(typeListe,"code",convertFromUTF8=FALSE)
if(! codeTypeListe %in% listeTypeListe)
  next
}
# Recherche des donnees de contexte si demande
if(!is.null(listeDonneesContexteHydroElem)){
  opchHydrobioElementaire
}
findElement(listeHydrobiologique,"OPCHydrobioElementaire")
if("Erreur" %in% opchHydrobioElementaire) # si une erreur est survenue
  return(opchHydrobioElementaire)
  donneeContexte <- findDonneeContexte(opchHydrobioElementaire,
listeDonneesContexteHydroElem,force=TRUE,convertFromUTF8=FALSE)
  if("Erreur" %in% donneeContexte) # si une erreur est survenue
    mais cela ne devrait jamais arriver
    next
}
# Recuperation de l'id de la liste Hydrobiologique
idListeHydrobiologique
findDonneesFromElement(listeHydrobiologique,"idListeHydrobiologique",convertFromUTF
8=FALSE)
  if("Erreur" %in% idListeHydrobiologique) # si une erreur est survenue
    return(idListeHydrobiologique)

  observationsHydrobiologique
findElement(listeHydrobiologique,"ObservationsHydrobiologique")
  if("Erreur" %in% observationsHydrobiologique){
    if(force)
      listeFaunistique <- c(listeFaunistique,"NULL","NULL","NULL",
"NULL",idListeHydrobiologique,vNumeroOrdre,vBocal,codeTypeListe,donneeContexte)
    next
  }
  observationsHydrobiologique
findElement(observationsHydrobiologique, "ObservationHydrobiologique")
  if("Erreur" %in% observationsHydrobiologique){
    if(force)
      listeFaunistique <- c(listeFaunistique,"NULL","NULL","NULL",
"NULL",idListeHydrobiologique,vNumeroOrdre,vBocal,codeTypeListe,donneeContexte)
    next
  }
}
# nombre de observationsHydrobiologique
nbObservationsHydrobiologique <- length(observationsHydrobiologique)

if(nbObservationsHydrobiologique < 1){
  erreur <- c("Erreur","EF06","Donnée d'observation manquante")
  return(erreur)
}
for(j in 1:nbObservationsHydrobiologique){
  observationHydrobiologique <- observationsHydrobiologique[j]
  # on retrouve l'id du taxon
  donneesOb <- findDonneesFromElement(observationHydrobiologique,
c("taxon","resultat"),force=TRUE,convertFromUTF8=FALSE)
  if("Erreur" %in% donneesOb){
    taxon <- "NULL"
    resultat <- "NULL"
  }
  else{
    taxon <- donneesOb["taxon"]
    resultat <- donneesOb["resultat"]
  }
  uniteBio <- findElement(observationHydrobiologique,"UniteBio")
  if("Erreur" %in% uniteBio){ # si une erreur est survenue
    erreur <- c("Erreur","EF26","Information sur l'unité bio manquante")
    return(erreur)
  }
}

```

```

    }
    codeUniteBio
  }
  findDonneesFromElement(uniteBio,"code",convertFromUTF8=FALSE)
  if("Erreur" %in% codeUniteBio){ # si une erreur est survenue
    erreur <- c("Erreur","EF27","Code de l'unité bio manquant")
    return(erreur)
  }
  typeLot <- findElement(observationHydrobiologique,"TypeLot")
  if("Erreur" %in% typeLot){ # si une erreur est survenue
    codeTypeLot <- "NULL"
  }
  codeTypeLot
  findDonneesFromElement(typeLot,"code",convertFromUTF8=FALSE)
  if("Erreur" %in% codeTypeLot){ # si une erreur est survenue
    codeTypeLot <- "NULL"
  }

  if(is.null(codesUnite)){ # on prend toutes les observations
    listeFaunistique <- c(listeFaunistique,taxon,resultat,
codeUniteBio,codeTypeLot,idListeHydrobiologique,vNumeroOrdre,vBocal,codeTypeListe,d
onneeContexte)
  }
  else{
    if(codeUniteBio %in% codesUnite){ # On conserve cette
observation
      listeFaunistique
c(listeFaunistique,taxon,resultat,codeUniteBio,codeTypeLot,idListeHydrobiologique,v
NumeroOrdre,vBocal,codeTypeListe,donneeContexte)
    }
  }
}
# formatage des resultats
nomsColonnes <- c("idTaxon","resultat","codeUniteBio","codeTypeLot",
"idListeHydrobiologique","numOrdre")

if(bocal)
  nomsColonnes <- c(nomsColonnes,"bocal")
if(!is.null(listeTypeListe))
  nomsColonnes <- c(nomsColonnes,"typeListe")
if(!is.null(listeDonneesContexteHydroElem))
  nomsColonnes <- c(nomsColonnes,listDonneesContexteHydroElem)

if(length(listeFaunistique) > 0){
  listeFaunistique
matrix(listeFaunistique,ncol=length(nomsColonnes),byrow=TRUE)
  colnames(listeFaunistique) <- nomsColonnes
  # Conversion des données issues d'un flux XML (UTF-8) vers l'encoding
de R courant.
  listeFaunistique <- fromUTF8(listeFaunistique)
  return(listeFaunistique)
}
else{
  erreur <- c("Erreur","EF06","Donnée d'observation manquante")
  return(erreur)
}
} # fin fonction findObservationsHydrobiologique

#calcul de la salinite
#param :
#Salinite mesuree
#Degre de salinite estimee
#return : Salinite calculee
calculeSalinite <- function(salinite,salinite.classe) {
  salinite <- as.numeric(salinite)
  salinite.classe <- as.numeric(salinite.classe)
  salinite.ccal <- NA

  if(!is.na(salinite)){
    if(salinite <= 18) {

```

```

        salinite.ccal <- 1
      } else {
        salinite.ccal <- 3
      }
    }

    #si salinite-ccal est pas vide alors salinite-c = salinite-ccal
    if(! is.na(salinite.ccal))
      return(salinite.ccal)
    #si salinite-ccal est vide ET salinite-classe est pas vide alors
    if( is.na(salinite.ccal) && ! is.na(salinite.classe)) {
      #si salinite-classe=3 alors salinite-c=3 sinon salinite-c=1
      if(salinite.classe==3)
        return(3)
      else { return(1) } }
    else #si salinite-ccal est vide ET salinite-classe est vide alors salinite-c
    reste vide
      return(0)
  }

#Ainsi en fonction de la valeur de Densité, un score (noté score) sera
#attribué à chaque combinaison (i) de masse_eau x saison x salinite-c selon
#les critères suivant :
#- score prend la valeur 0 si Densité (i) est inférieure au seuil inférieur (inf)
#- score prend la valeur 2 si Densité (i) est supérieure ou égale au seuil
inférieur (inf) et inférieure au seuil supérieur (sup)
#- score prend la valeur 4 si Densité (i) est supérieure ou égale au seuil
supérieur (sup)
#param:
#superficieCMasseDEau      :
#SectionMasseDEau         :
#saison                    : le nom de la saison
#salinite.c                : la classe de la salinite
#densite                   :
calculScore                                                         <-
function(superficieCMasseDEau,SectionMasseDEau,lbSaison,salinite.c,Densite){
  tab <- DENS.CA.LAG.TAB.SEUIL.CA.LAGUNE

  #Selection des lignes du tableau verifiant les criteres donnés en argument
  tab <- subset(tab,
                tab[,DENS.CA.LAG.COL.SUPERFICIE] == superficieCMasseDEau &
                tab[,DENS.CA.LAG.COL.SECTION]   == SectionMasseDEau &
                tab[,DENS.CA.LAG.COL.SAISON]     == lbSaison &
                tab[,DENS.CA.LAG.COL.SALINITE]   == salinite.c)

  if(nrow(tab) == 0){
    return(NULL)
  }

  S1 <- as.numeric(tab[1,DENS.CA.LAG.COL.SEUILS1])
  S2 <- as.numeric(tab[1,DENS.CA.LAG.COL.SEUILS2])
  S3 <- as.numeric(tab[1,DENS.CA.LAG.COL.SEUILS3])
  S4 <- as.numeric(tab[1,DENS.CA.LAG.COL.SEUILS4])

  if((Densite >= S2) && (Densite < S3)) { return(2) }

  if(!is.na(S1)) {
    if(Densite < S1) { return(0) }
    else {
      if(Densite < S2) { return(1) } } }
  else {
    if(Densite < S2) { return(0) } }

  if(!is.na(S4)) {
    if(Densite >= S4) { return(4) }
    else {
      if(Densite >= S3) { return(3) } } }
  else {
    if(Densite >= S3) { return(4) } }

```

```

}

#Exécution de l'algo
#run : fonction appelée par SEEE pour effectuer l'exécution d'un algorithme
run <- function() {
  #Initialisation de la liste des erreurs générales
  erreursGenerales <- NULL
  #Récupération des données dans le fichier XML d'entrée
  #Attention : ces variables ne doivent pas être modifiées
  #le nom du fichier d'entrée est passé en variable globale (fichierEntree)
  #Récupération de l'élément racine du fichier d'entrée
  rootElement <- findRootElement(fichierEntree)
  #Si la récupération s'est mal passée (l'élément racine n'existe pas dans le
XML)
  if("Erreur" %in% rootElement) {
    erreursGenerales <- c(erreursGenerales, rootElement)
    rootElement <- NULL
    utilisateur <- NULL
  }else {
    #Récupération de l'utilisateur qui a lancé l'algorithme
    utilisateur <- findUtilisateur(rootElement)
    #Si l'utilisateur a été mal spécifié dans le fichier d'entrée
    if("Erreur" %in% utilisateur) {
      erreursGenerales <- c(erreursGenerales, utilisateur)
      utilisateur <- NULL
    }

    #Récupération de l'adresse du fichier de sortie
    FichierSortie <- findFichierSortie(rootElement)
    #Si l'adresse du fichier de sortie a été mal spécifiée
    if("Erreur" %in% FichierSortie) {
      erreursGenerales <- c(erreursGenerales, FichierSortie)
      FichierSortie <- NULL
    }

    #Récupération de l'adresse du fichier des données d'observation
    fichierDonneesObservation <-
findFichierDonneesObservation(rootElement)
    #Si l'adresse du fichier des données d'observation a été mal spécifiée
    if("Erreur" %in% fichierDonneesObservation) {
      erreursGenerales <- c(erreursGenerales,
fichierDonneesObservation)
      fichierDonneesObservation <- NULL
    }

    #récupération des adresses des fichiers contenant les données de
référence (si nécessaire)
    fichiersDonneesReference <- findListeTabReference(rootElement)
    #si l'adresse du fichier des données d'observation a été mal spécifiée
    if("Erreur" %in% fichiersDonneesReference) {
      erreursGenerales <- c(erreursGenerales, fichiersDonneesReference)
      fichiersDonneesReference <- NULL
    }else{
      referentiel_taxon <- findTabRefTaxon(fichiersDonneesReference["TabRefTaxon"],
c("cdTaxonSandre", "embranchementTaxon"))
      if("Erreur" %in% referentiel_taxon){
        erreursGenerales <- c(erreursGenerales, referentiel_taxon)
        referentiel_taxon <- NULL
      }
    }

    #recuperation des donnees des tableaux
    DENS.CA.LAG.TAB.SEUIL.CA.LAGUNE <-
findTableau(rootElement, DENS.CA.LAG.TAB.SEUIL.CA.LAGUNE.LABEL)
    if("Erreur" %in% DENS.CA.LAG.TAB.SEUIL.CA.LAGUNE) {
      erreursGenerales <- c(erreursGenerales,
DENS.CA.LAG.TAB.SEUIL.CA.LAGUNE)
      DENS.CA.LAG.TAB.SEUIL.CA.LAGUNE <- NULL
    }
  }
}

```

```

    DENS.CA.LAG.TAB.GUILDE.ECOLOGIQUE <- NULL
  }
  findTableau(rootElement,DENS.CA.LAG.TAB.GUILDE.ECOLOGIQUE.LABEL)
  if("Erreur" %in% DENS.CA.LAG.TAB.GUILDE.ECOLOGIQUE) {
    erreursGenerales <- c(erreursGenerales,
DENS.CA.LAG.TAB.GUILDE.ECOLOGIQUE)
    DENS.CA.LAG.TAB.GUILDE.ECOLOGIQUE <- NULL
  }
  DENS.CA.LAG.TAB.PARAMETRE.LAGUNES <- NULL
  findTableau(rootElement,DENS.CA.LAG.TAB.PARAMETRE.LAGUNES.LABEL)
  if("Erreur" %in% DENS.CA.LAG.TAB.PARAMETRE.LAGUNES) {
    erreursGenerales <- c(erreursGenerales,
DENS.CA.LAG.TAB.PARAMETRE.LAGUNES)
    DENS.CA.LAG.TAB.PARAMETRE.LAGUNES <- NULL
  }

  #libération de la mémoire
  rm(rootElement)
}
#Initialisation de l'arbre Xml de sortie
SortieXmlTree <- InitialiserXmlTreeSortie(utilisateur,FichierSortie)

#On boucle sur les points de contrôle
funPointDeControle = function(pointDeControle) {
  #####Préparation des resultats au format XML
  ### Xml de sortie
  #Initialisation des variables pour les noeuds permettant de faire le
fichier xml de sortie
  produitPrincipalNode <- NULL
  produitsSecondairesNode <- NULL
  DonneesIntermediairesNode <- NULL
  DonneesPreCalculeesNode <- NULL
  DonneesExplicativesNode <- NULL
  DonneesDeContexteNode <- NULL
  DonneesDeTracabiliteNode <- NULL
  DonneesIncertitudeNode <- NULL

  #Initialisation des produits secondaires
  produitsSecondairesNode <- InitNodeProduitsSecondaires()

  #Initialisation des erreurs
  donneesGenerales <- NULL
  donneesErreur <- NULL

  #Declaration de la liste des enregistrements valides
  listeTraitIdValide <- list()
  localisation <- NULL
  #récupération de l'id du point de controle

  #récupération des données générales de l'opération de contrôle
  donneesGenerales <- findDonneesGenerales(pointDeControle, NULL)
  #Si les données générales n'ont pas été correctement saisies
  if("Erreur" %in% donneesGenerales) {
    donneesErreur <- donneesGenerales
    SortieXmlTree <- AjouterNodeResultats(SortieXmlTree,
NULL,NULL,NULL,donneesErreur,NULL,NULL,NULL,NULL,NULL)
    donneesGenerales <- NULL
  }
  else {
    #Recuperation de la localisation (systeme-estuaire) du point de controle

    #Recuperation du code Ecoregion et de la superficie totale de la masse
d'eau au niveau du point de controle
    EcoRegionMasseDEau <- findDonneeContexte(pointDeControle, "CdEcoregion")
    if("Erreur" %in% EcoRegionMasseDEau) {
      donneesErreur <- EcoRegionMasseDEau
      EcoRegionMasseDEau <- NULL
    } else {
      #Initialisation des données

```

```

    superficieCMasseDEau <- NULL
    superficie <- findDonneeContexte(pointDeControle, "SuperficieMasseDEauET")
    if("Erreur" %in% superficie) {
      donneesErreur <- superficie
      superficie <- NA
    }
    #Affectation de la superficie de la masse d'eau
    if(!is.na((superficie))) {
      if(as.numeric(superficie) <= 50) {
        superficieCMasseDEau <- "petit"
      } else {
        superficieCMasseDEau <- "grand"
      }
    }
    #Recuperation de la section de la masse d'eau au niveau du point de controle
    tabConn <- DENS.CA.LAG.TAB.PARAMETRE.LAGUNES
    SectionMasseDEau<-NULL
    #Selection de la ligne pour le point de controle analysé
    CdPointDeControle <- findDonneesFromElement(pointDeControle, "CdPointDeControle")
    tabConn <- subset(tabConn, tabConn[,DENS.CA.LAG.COL.PARAM.MET] ==CdPointDeControle)
    if (nrow(tabConn)!=1) {
      donneesErreur <- c(donneesErreur, SectionMasseDEau)
    } else {
      SectionMasseDEau <- tabConn[1,DENS.CA.LAG.COL.PARAM.SECTION]

    #On récupère les opérations de controle liées au point de controle
    operationsDeControle <- findOperationsControle(pointDeControle)
    if("Erreur" %in% operationsDeControle) {
      donneesErreur <- c(donneesErreur, operationsDeControle)
      operationsDeControle <- NULL
      donneesGenerales <- NULL
    } else {

    #Pour chaque opération de contrôle : récupération des données puis exécution
    de l'algorithme
      for(operationDeControle in operationsDeControle) {

        #Recuperation de la saison au niveau de l'operation de controle
        saison <- findDonneeContexte(operationDeControle,"Saison")
        if("Erreur" %in% saison) {
          donneesErreur <- c(donneesErreur,saison)
          saison <- NULL
        } else {

        #Récupération des données de l'opération de contrôle
        listeCtxHydroElem <-
        c("CdEnginPêche","Salinite","ClasseDeSalinite","DureePêche" ,"Commentaires")
        listeTraitDePêche <- findListeTraitDePêche(operationDeControle,listeCtxHydroElem,
        "13")

        #si une erreur est survenue
        if("Erreur" %in% listeTraitDePêche) {
          donneesErreur <- c(donneesErreur,listeTraitDePêche)
          listeTraitDePêche <- NULL
        } else {

    #Exécution de l'algo
    #Parcours des trait de pêche et calculs intermediaires
    for(trait_id in listeTraitDePêche) {

      #Récupération du code engin pêche

      CdEnginPêche <- trait_id[1,"CdEnginPêche"]
      DureePêche <- trait_id[1,"DureePêche"]
      Commentaires <- trait_id[1,"Commentaires"]

      #Declarations des variables pour les calculs intermediaires
      salinite.c <- 0

      #Calcul de la salinite
      salinite.c<- calculeSalinite(trait_id[1,"Salinite"],trait_id[1,"ClasseDeSalinite"])

      #Calcul de la duree

```


Janvier 2011 – Page 80/108

```

#4- Dans la section précédente, les calculs étaient effectués pour
#chaque trait_id_valide. Dans cette partie, nous allons procéder
#à un regroupement de ces valeurs en fonction de la masse d'eau,
#de la saison et de la classe de salinité de façon à pouvoir comparer
#ces valeurs aux seuils fournis. Les calculs suivants sont effectués
#pour chaque combinaison masse_eau x saison x salinite-c
#Par exemple : pour la masse d'eau Loire, au printemps et dans la salinite-c 1.

# FIXME /\ La masse d'eau est assimilée au point de contrôle.
#On travail donc sur la combinaison PointDeContrôle x saison x salinite-c

listeScore <- NULL

#Parcours des saisons
for(saison in DENS.CA.LAG.SAISONS) {
  cdSaison <- saison[1] # Code de la saison
  lbSaison <- saison[2] # Label de la saison

  #Parcours des salinités
  for (salinite.c in DENS.CA.LAG.SALINITES) {

    Densite <- 0
    selection1 <- subset(listeTraitIdValide,
      listeTraitIdValide[, "saison"] == cdSaison &
      listeTraitIdValide[, "salinite"] == salinite.c )
    NTrait <- nrow(selection1)

    if(NTrait != 0)
      sommeLogDens <- sum(as.numeric(selection1[, "logDens"]))

    if(NTrait != 0)
      Densite <- sommeLogDens / NTrait

    score
calculScore(superficieCMasseDEau, SectionMasseDEau, lbSaison, salinite.c, Densite) <-

#Attention, en estuaires seuls les scores dont le calcul est basé sur un nombre
#d'enregistrements (NTrait) supérieur ou égal à 6 sont comptabilisés.
# Pas pour les lagunes
    if(NTrait != 0){
      listeScore <- c(listeScore, score)
    }

    code <- paste("dens_ca_lag", cdSaison, salinite.c, sep="")
    libelle <- paste("Densité CA: ", lbSaison, "-Salinité ", salinite.c, sep="")
    produitsSecondairesNode <- AjouterNodeProduitSecondaire(produitsSecondairesNode, code,
    libelle, Densite, NULL, NULL)
    if("Erreur" %in% produitsSecondairesNode) {
      donneesErreur <- c(donneesErreur, produitsSecondairesNode)
      produitsSecondairesNode <- NULL
    }
  }
} # fin de boucle sur les saisons

Nscore <- length(listeScore)
if(Nscore > 0 ){
  som.score <- sum(listeScore)
  note <- som.score / (4 * Nscore)
}else
  note <- 0

produitPrincipalNode <- CreerNodeProduitPrincipal("Note-CA",
NULL, note, NULL, "dens_ca_lag")
if("Erreur" %in% produitPrincipalNode) {
  donneesErreur <- c(donneesErreur, produitPrincipalNode)
  produitPrincipalNode <- NULL
}
}

```

```
SortieXmlTree <- AjouterNodeResultats(SortieXmlTree,
                                     donneesGenerales,
                                     produitPrincipalNode,
                                     produitsSecondairesNode,
                                     donneesErreur,
                                     DonneesIntermediairesNode,
                                     DonneesPreCalculeesNode,
                                     DonneesExplicativesNode,
                                     DonneesDeContexteNode,
                                     DonneesDeTracabiliteNode,
                                     DonneesIncertitudeNode)
}#fin de fonction point de controle

#Si absence d'erreur de lecture et présence de point(s) de contrôle
#on lance l'algo, sinon on écrit seulement les erreurs générales
if((length(erreursGenerales) == 0) ) {
  #Lancement du parsing + listage des differents systemes et calcul de
leur superficie (estuaires)
  # doPointControlesSAX(fichierDonneesObservation,funSysteme)
  #Lancement du parsing + traitement a la volée des points de controles
  doPointControlesSAX(fichierDonneesObservation,funPointDeControle)
}else
  SortieXmlTree <- AjouterNodeErreursGenerales(SortieXmlTree,
erreursGenerales)

Metrique <- "Metrique"
Indice <- "Indice"
#Il faut définir le type du modèle : soit mettre Metrique ou Indicateur.
type <- Metrique

#Ecriture du fichier de sortie
EcrireFichierXml(type, SortieXmlTree, FichierSortie)
}
```

ET-Poisson-DENS_IB_LAG**v 0.001**

```
#####
# Densité de poissons prédateurs d'invertébrés benthiques dans les lagunes
# Author: Theunissenb
#####

#####
# Variables globales : INTERDICTION FORMELLE DE LES MODIFIER
# fichierEntree : adresse du fichier d'entrée
#####

#Association code et label de chaque saison
ET.SAISON.PRINTEMPS <- c("0","printemps")
ET.SAISON.AUTOMNE   <- c("2","automne")
#Les saisons traitées par le modèle
DENS.IB.LAG.SAISONS <- list(ET.SAISON.PRINTEMPS,ET.SAISON.AUTOMNE)

#Codes des eco-regions masse d'eau
ET.ECOREGION.MED      <- "MED"
#Codes des eco-regions masse d'eau traitées par le modèle
DENS.IB.LAG.ECOREGIONS <- c(ET.ECOREGION.MED)

DENS.IB.LAG.SALINITES <- c("1","3")

#Declaration des tableaux associés au modèle
DENS.IB.LAG.TAB.SEUIL.IB.LAGUNE.LABEL      <- "seuilsIBLagune"
DENS.IB.LAG.TAB.SEUIL.IB.LAGUNE           <- NULL

DENS.IB.LAG.TAB.GUILDE.TROPHIQUE.LABEL     <- "GildeTrophique"
DENS.IB.LAG.TAB.GUILDE.TROPHIQUE          <- NULL

DENS.IB.LAG.TAB.PARAMETRE.LAGUNES.LABEL    <- "ParamLagunes"
DENS.IB.LAG.TAB.PARAMETRE.LAGUNES         <- NULL

#Declaration des noms des colonnes des tableaux manipulés
DENS.IB.LAG.COL.SUPERFICIE                  <- "superficieC"
DENS.IB.LAG.COL.SECTION                     <- "section"
DENS.IB.LAG.COL.SAISON                      <- "saison"
DENS.IB.LAG.COL.SALINITE                    <- "saliniteC"
DENS.IB.LAG.COL.SEUILS1                     <- "seuils1"
DENS.IB.LAG.COL.SEUILS2                     <- "seuils2"
DENS.IB.LAG.COL.SEUILS3                     <- "seuils3"
DENS.IB.LAG.COL.SEUILS4                     <- "seuils4"
DENS.IB.LAG.COL.TRAIT                       <- "Trait"
DENS.IB.LAG.COL.PARAM.SECTION               <- "ParamSection" # valeur de la section dans le
tableau de parametre
DENS.IB.LAG.COL.PARAM.MET                   <- "ParamMET"

#Trait du poisson à considérer dans le modèle
DENS.IB.LAG.TRAIT <- "IB"

#Initialisation des erreurs
donneesErreur <- NULL
DENS.IB.LAG.LISTES.SYSTEMES <- NULL

#Définitions de fonctions internes

# FIXME Declaration temporaire d'une fonction de la bibliotheque RSEEE durant la
Recette.
## Récupération de la liste faunistique dans les données d'observation.
# param operationControle le noeud de l'opération de contrôle.
# param codesUnite le(s) code(s) de l'unité bio concernant les résultats d'une
observation hydrobiologique, par défaut NULL.
```

```
# param listeDonneesContexteHydroElem le(s) libellé(s) des données de contexte
souhaitées, par défaut NULL.
# param bocal indique si une colonne contenant le bocal doit être ajoutée, par
défaut FALSE.
# param listeTypeListe liste des typeListe que la listeHydrobiologique doit
contenir, par défaut NULL.
# Si la listeHydrobiologique n'est pas d'un type compris dans typeListe alors
# l'ensemble des observationHydrobiologiques sous jacentes est ignoré.
# param force si vrai, alors les listes hydrobiologiques n'ayant pas d'observation
hydrobiologique
# apparaissent toutefois avec les données: idTaxon,resultat et codeUniteBio
valorisées à "NULL", par défaut FALSE.
# return la liste des taxons (idTaxon,resultat,codeUniteBio).
# erreur si une donnée est manquante.
findObservationsHydrobiologique <- fonction(operationControle,
      codesUnite=NULL,
      listeDonneesContexteHydroElem=NULL,
      bocal=FALSE,
      listeTypeListe=NULL,
      force=FALSE){
  # initialisation
  listeFaunistique <- NULL
  listesHydrobiologique
  findElement(operationControle,"ListesHydrobiologique")
  if("Erreur" %in% listesHydrobiologique)
    return(listesHydrobiologique)
  listesHydrobiologique
  findElement(listesHydrobiologique,"ListeHydrobiologique")
  if("Erreur" %in% listesHydrobiologique)
    return(listesHydrobiologique)

  # nombre de listesHydrobiologique
  nbListesHydrobiologique <- length(listesHydrobiologique)
  if(nbListesHydrobiologique < 1){
    erreur <- c("Erreur","EF06","Donnée d'observation manquante")
    return(erreur)
  }
  for(i in 1:nbListesHydrobiologique){
    vNumeroOrdre      <- NULL
    vBocal             <- NULL
    donneeContexte    <- NULL
    codeTypeListe     <- NULL
    idListeHydrobiologique <- NULL

    listeHydrobiologique <- listesHydrobiologique[i]

    # Recuperation du numero de passage associe a ListeHydrobiologique
    opcHydrobioElementaire
    findElement(listeHydrobiologique,"OPCHydrobioElementaire")
    if("Erreur" %in% opcHydrobioElementaire){ # Si une erreur est survenue
      vNumeroOrdre <- "NULL"
      if(bocal) # le bocal est demande et non disponible
        return(opcHydrobioElementaire) # Envoi de l'erreur
    }
    else{ # si pas d'erreur survenue
      vNumeroOrdre
      findDonneesFromElement(opcHydrobioElementaire,"numOrdre",convertFromUTF8=FALSE)
      if("Erreur" %in% vNumeroOrdre) # si une erreur est survenue
        vNumeroOrdre <- "NULL"

      # Recherche du bocal associé a ListeHydrobiologique
      if(bocal){
        vBocal <- findDonneesFromElement(opcHydrobioElementaire,
"numRegroupement",convertFromUTF8=FALSE)
        if("Erreur" %in% vBocal) # si une erreur est survenue
          return(vBocal)
      }
    }
  }
  # recherche du typeListe de la listeHydrobiologique contenant les
observationHydrobiologiques
```

```

    if(!is.null(listeTypeListe)){
      typeListe <- findElement(listeHydrobiologique,"TypeListe")
      if("Erreur" %in% typeListe) # si une erreur est survenue
        next
    }
    codeTypeListe <- findDonneesFromElement(typeListe,"code",convertFromUTF8=FALSE)
    if(! codeTypeListe %in% listeTypeListe)
      next
  }
  # Recherche des donnees de contexte si demande
  if(!is.null(listeDonneesContexteHydroElem)){
    opcHydrobioElementaire
    findElement(listeHydrobiologique,"OPCHydrobioElementaire")
    if("Erreur" %in% opcHydrobioElementaire) # si une erreur est survenue
      return(opcHydrobioElementaire)
    donneeContexte <- findDonneeContexte(opcHydrobioElementaire,
    listeDonneesContexteHydroElem,force=TRUE,convertFromUTF8=FALSE)
    if("Erreur" %in% donneeContexte) # si une erreur est survenue
    mais cela ne devrait jamais arriver
      next
  }
  # Recuperation de l'id de la liste Hydrobiologique
  idListeHydrobiologique <- findDonneesFromElement(listeHydrobiologique,
  "idListeHydrobiologique",convertFromUTF8=FALSE)
  if("Erreur" %in% idListeHydrobiologique) # si une erreur est survenue
    return(idListeHydrobiologique)

  observationsHydrobiologique
  findElement(listeHydrobiologique,"ObservationsHydrobiologique")
  if("Erreur" %in% observationsHydrobiologique){
    if(force)
      listeFaunistique <- c(listeFaunistique,"NULL","NULL","NULL",
  "NULL",idListeHydrobiologique,vNumeroOrdre,vBocal,codeTypeListe,donneeContexte)
      next
    }
    observationsHydrobiologique
    findElement(observationsHydrobiologique, "ObservationHydrobiologique")
    if("Erreur" %in% observationsHydrobiologique){
      if(force)
        listeFaunistique <- c(listeFaunistique,"NULL","NULL","NULL",
  "NULL",idListeHydrobiologique,vNumeroOrdre,vBocal,codeTypeListe,donneeContexte)
        next
      }
      # nombre de observationsHydrobiologique
      nbObservationsHydrobiologique <- length(observationsHydrobiologique)

      if(nbObservationsHydrobiologique < 1){
        erreur <- c("Erreur","EF06","Donnée d'observation manquante")
        return(erreur)
      }
      for(j in 1:nbObservationsHydrobiologique){
        observationHydrobiologique <- observationsHydrobiologique[j]
        # on retrouve l'id du taxon
        donneesOb <- findDonneesFromElement(observationHydrobiologique,
  c("taxon","resultat"),force=TRUE,convertFromUTF8=FALSE)
        if("Erreur" %in% donneesOb){
          taxon <- "NULL"
          resultat <- "NULL"
        }
        else{
          taxon <- donneesOb["taxon"]
          resultat <- donneesOb["resultat"]
        }
        uniteBio <- findElement(observationHydrobiologique,"UniteBio")
        if("Erreur" %in% uniteBio){ # si une erreur est survenue
          erreur <- c("Erreur","EF26","Information sur l'unité bio manquante")
          return(erreur)
        }
        codeUniteBio
        findDonneesFromElement(uniteBio,"code",convertFromUTF8=FALSE)
        if("Erreur" %in% codeUniteBio){ # si une erreur est survenue

```

```

    erreur <- c("Erreur", "EF27", "Code de l'unité bio manquant")
    return(erreur)
  }
  typeLot <- findElement(observationHydrobiologique, "TypeLot")
  if("Erreur" %in% typeLot){ # si une erreur est survenue
    codeTypeLot <- "NULL"
  }
  codeTypeLot
  findDonneesFromElement(typeLot, "code", convertFromUTF8=FALSE)
  if("Erreur" %in% codeTypeLot){ # si une erreur est survenue
    codeTypeLot <- "NULL"
  }

  if(is.null(codesUnite)){ # on prend toutes les observations
listeFaunistique <- c(listeFaunistique, taxon, resultat, codeUniteBio, codeTypeLot,
idListeHydrobiologique, vNumeroOrdre, vBocal, codeTypeListe, donneeContexte)
  }
  else{
    if(codeUniteBio %in% codesUnite){ # On conserve cette observation
listeFaunistique <- c(listeFaunistique, taxon, resultat, codeUniteBio, codeTypeLot,
idListeHydrobiologique, vNumeroOrdre, vBocal, codeTypeListe, donneeContexte)
    }
  }
}
# formatage des resultats
nomsColonnes
c("idTaxon", "resultat", "codeUniteBio", "codeTypeLot", "idListeHydrobiologique", "numOr
dre")

if(bocal)
  nomsColonnes <- c(nomsColonnes, "bocal")
if(!is.null(listeTypeListe))
  nomsColonnes <- c(nomsColonnes, "typeListe")
if(!is.null(listeDonneesContexteHydroElem))
  nomsColonnes <- c(nomsColonnes, listeDonneesContexteHydroElem)

if(length(listeFaunistique) > 0){
  listeFaunistique
matrix(listeFaunistique, ncol=length(nomsColonnes), byrow=TRUE)
  colnames(listeFaunistique) <- nomsColonnes
  # Conversion des données issues d'un flux XML (UTF-8) vers l'encoding
de R courant.
  listeFaunistique <- fromUTF8(listeFaunistique)
  return(listeFaunistique)
}
else{
  erreur <- c("Erreur", "EF06", "Donnée d'observation manquante")
  return(erreur)
}
} # fin fonction findObservationsHydrobiologique

#calcul de la salinite
#param :
#Salinite mesuree
#Degre de salinite estimee
#return : Salinite calculee
calculeSalinite <- function(salinite, salinite.classe) {
  salinite <- as.numeric(salinite)
  salinite.classe <- as.numeric(salinite.classe)
  salinite.ccal <- NA

  if(!is.na(salinite)){
    if(salinite <= 18) {
      salinite.ccal <- 1
    } else {
      salinite.ccal <- 3
    }
  }
}

```

```
#si salinite-ccal est pas vide alors salinite-c = salinite-ccal
if(! is.na(salinite.ccal))
  return(salinite.ccal)
#si salinite-ccal est vide ET salinite-classe est pas vide alors
if( is.na(salinite.ccal) && ! is.na(salinite.classe)) {
  #si salinite-classe=3 alors salinite-c=3 sinon salinite-c=1
  if(salinite.classe==3)
    return(3)
  else { return(1)} }
else #si salinite-ccal est vide ET salinite-classe est vide alors salinite-c
reste vide
  return(0)
}
```

```
#Ainsi en fonction de la valeur de Densité, un score (noté score) sera
#attribué à chaque combinaison (i) de masse_eau x saison x salinite-c selon
#les critères suivant :
#- score prend la valeur 0 si Densité (i) est inférieure au seuil inférieur (inf)
#- score prend la valeur 2 si Densité (i) est supérieure ou égale au seuil
inférieur (inf) et inférieure au seuil supérieur (sup)
#- score prend la valeur 4 si Densité (i) est supérieure ou égale au seuil
supérieur (sup)
#param:
#superficieCMasseDEau      :
#SectionMasseDEau         :
#saison                    : le nom de la saison
#salinite.c                : la classe de la salinite
#densite                   :
calculScore                                                         <-
function(superficieCMasseDEau,SectionMasseDEau,lbSaison,salinite.c,Densite){
  tab <- DENS.IB.LAG.TAB.SEUIL.IB.LAGUNE

  #Selection des lignes du tableau verifiant les criteres donnés en argument
  tab <- subset(tab,
    tab[,DENS.IB.LAG.COL.SUPERFICIE] == superficieCMasseDEau &
    tab[,DENS.IB.LAG.COL.SECTION]   == SectionMasseDEau &
    tab[,DENS.IB.LAG.COL.SAISON]    == lbSaison &
    tab[,DENS.IB.LAG.COL.SALINITE]  == salinite.c)

  if(nrow(tab) == 0){
    return(NULL)
  }

  S1 <- as.numeric(tab[1,DENS.IB.LAG.COL.SEUILS1])
  S2 <- as.numeric(tab[1,DENS.IB.LAG.COL.SEUILS2])
  S3 <- as.numeric(tab[1,DENS.IB.LAG.COL.SEUILS3])
  S4 <- as.numeric(tab[1,DENS.IB.LAG.COL.SEUILS4])

  if((Densite >= S2) && (Densite < S3)) { return(2) }

  if(!is.na(S1)) {
    if(Densite < S1) { return(0) }
    else {
      if(Densite < S2) { return(1) } } }
  else {
    if(Densite < S2) { return(0) } }

  if(!is.na(S4)) {
    if(Densite >= S4) { return(4) }
    else {
      if(Densite >= S3) { return(3) } } }
  else {
    if(Densite >= S3) { return(4) } }

  }

#Exécution de l'algo
#run : fonction appelée par SEEE pour effectuer l'exécution d'un algorithme
run <- function() {
```



```

#Initialisation de la liste des erreurs générales
erreursGenerales <- NULL
#Récupération des données dans le fichier XML d'entrée
#Attention : ces variables ne doivent pas être modifiées
#le nom du fichier d'entrée est passé en variable globale (fichierEntree)
#Récupération de l'élément racine du fichier d'entrée
rootElement <- findRootElement(fichierEntree)
#Si la récupération s'est mal passée (l'élément racine n'existe pas dans le
XML)
if("Erreur" %in% rootElement) {
  erreursGenerales <- c(erreursGenerales, rootElement)
  rootElement <- NULL
  utilisateur <- NULL
}else {
  #Récupération de l'utilisateur qui a lancé l'algorithme
  utilisateur <- findUtilisateur(rootElement)
  #Si l'utilisateur a été mal spécifié dans le fichier d'entrée
  if("Erreur" %in% utilisateur) {
    erreursGenerales <- c(erreursGenerales, utilisateur)
    utilisateur <- NULL
  }

  #Récupération de l'adresse du fichier de sortie
  FichierSortie <- findFichierSortie(rootElement)
  #Si l'adresse du fichier de sortie a été mal spécifiée
  if("Erreur" %in% FichierSortie) {
    erreursGenerales <- c(erreursGenerales, FichierSortie)
    FichierSortie <- NULL
  }

  #Récupération de l'adresse du fichier des données d'observation
  fichierDonneesObservation <-
findFichierDonneesObservation(rootElement)
  #Si l'adresse du fichier des données d'observation a été mal spécifiée
  if("Erreur" %in% fichierDonneesObservation) {
    erreursGenerales <- c(erreursGenerales, fichierDonneesObservation)
    fichierDonneesObservation <- NULL
  }

  #récupération des adresses des fichiers contenant les données de
référence (si nécessaire)
  fichiersDonneesReference <- findListeTabReference(rootElement)
  #si l'adresse du fichier des données d'observation a été mal spécifiée
  if("Erreur" %in% fichiersDonneesReference) {
    erreursGenerales <- c(erreursGenerales, fichiersDonneesReference)
    fichiersDonneesReference <- NULL
  }else{
    referentiel_taxon <-
findTabRefTaxon(fichiersDonneesReference["TabRefTaxon"],
c("cdTaxonSandre","embranchementTaxon"))
    if("Erreur" %in% referentiel_taxon){
      erreursGenerales <- c(erreursGenerales, referentiel_taxon)
      referentiel_taxon <- NULL
    }
  }

  #recuperation des donnees des tableaux
  DENS.IB.LAG.TAB.SEUIL.IB.LAGUNE <-
findTableau(rootElement,DENS.IB.LAG.TAB.SEUIL.IB.LAGUNE.LABEL)
  if("Erreur" %in% DENS.IB.LAG.TAB.SEUIL.IB.LAGUNE) {
    erreursGenerales <- c(erreursGenerales, DENS.IB.LAG.TAB.SEUIL.IB.LAGUNE)
    DENS.IB.LAG.TAB.SEUIL.IB.LAGUNE <- NULL
  }
  DENS.IB.LAG.TAB.GUILDE.TROPHIQUE <-
findTableau(rootElement,DENS.IB.LAG.TAB.GUILDE.TROPHIQUE.LABEL)
  if("Erreur" %in% DENS.IB.LAG.TAB.GUILDE.TROPHIQUE) {
    erreursGenerales <- c(erreursGenerales, DENS.IB.LAG.TAB.GUILDE.TROPHIQUE)
    DENS.IB.LAG.TAB.GUILDE.TROPHIQUE <- NULL
  }
}

```

```

DENS.IB.LAG.TAB.PARAMETRE.LAGUNES
findTableau(rootElement,DENS.IB.LAG.TAB.PARAMETRE.LAGUNES.LABEL)
  if("Erreur" %in% DENS.IB.LAG.TAB.PARAMETRE.LAGUNES) {
    erreursGenerales <- c(erreursGenerales, DENS.IB.LAG.TAB.PARAMETRE.LAGUNES)
    DENS.IB.LAG.TAB.PARAMETRE.LAGUNES <<- NULL
  }

  #libération de la mémoire
  rm(rootElement)

}
#Initialisation de l'arbre Xml de sortie
SortieXmlTree <- InitialiserXmlTreeSortie(utilisateur,FichierSortie)

#On boucle sur les points de contrôle
funPointDeControle = function(pointDeControle) {
  #####Préparation des resultats au format XML
  ### Xml de sortie
  #Initialisation des variables pour les noeuds permettant de faire le
fichier xml de sortie
  produitPrincipalNode      <- NULL
  produitsSecondairesNode   <- NULL
  donneesIntermediairesNode <- NULL
  donneesPreCalculeesNode   <- NULL
  donneesExplicativesNode    <- NULL
  donneesDeContexteNode     <- NULL
  donneesDeTracabiliteNode  <- NULL
  donneesIncertitudeNode    <- NULL

  #Initialisation des produits secondaires
  produitsSecondairesNode <- InitNodeProduitsSecondaires()

  #Initialisation des erreurs
  donneesGenerales <- NULL
  donneesErreur    <- NULL

  #Declaration de la liste des enregistrements valides
  listeTraitIdValide <- list()
  localisation      <- NULL
  #récupération de l'id du point de controle

  #récupération des données générales de l'opération de contrôle
  donneesGenerales <- findDonneesGenerales(pointDeControle, NULL)
  #Si les données générales n'ont pas été correctement saisies
  if("Erreur" %in% donneesGenerales) {
    donneesErreur <- donneesGenerales
    SortieXmlTree <<- AjouterNodeResultats(SortieXmlTree,
NULL,NULL,NULL,donneesErreur,NULL,NULL,NULL,NULL,NULL,NULL)
    donneesGenerales <- NULL
  }
  else {
    #Recuperation de la localisation (systeme-estuaires) du point de controle

    #Recuperation du code Ecoregion et de la superficie totale de la masse
d'eau au niveau du point de controle
    EcoRegionMasseDEau <- findDonneeContexte(pointDeControle, "CdEcoregion")
    if("Erreur" %in% EcoRegionMasseDEau) {
      donneesErreur <- EcoRegionMasseDEau
      EcoRegionMasseDEau <- NULL
    } else {
      #Initialisation des données
      superficieCMasseDEau <- NULL
      superficie <- findDonneeContexte(pointDeControle, "SuperficieMasseDEauET")
      if("Erreur" %in% superficie) {
        donneesErreur <- superficie
        superficie <- NA
      }
    }
    #Affectation de la superficie de la masse d'eau
    if(!is.na((superficie))) {

```

```

if(as.numeric(superficie) <= 50) {
    superficieCMasseDEau <- "petit"
} else {
    superficieCMasseDEau <- "grand"
}

#Recuperation de la section de la masse d'eau au niveau du point de controle
tabConn <- DENS.IB.LAG.TAB.PARAMETRE.LAGUNES
SectionMasseDEau<-NULL
#Selection de la ligne pour le point de controle analysé
CdPointDeControle <- findDonneesFromElement(pointDeControle, "cdPointDeControle")
tabConn <- subset(tabConn, tabConn[,DENS.IB.LAG.COL.PARAM.MET] ==CdPointDeControle)
if (nrow(tabConn)!=1) {
    donneesErreur <- c(donneesErreur, SectionMasseDEau)
} else {
    SectionMasseDEau <- tabConn[1,DENS.IB.LAG.COL.PARAM.SECTION]

    #On récupère les opérations de controle liées au point de controle
    operationsDeControle <- findOperationsControle(pointDeControle)
    if("Erreur" %in% operationsDeControle) {
        donneesErreur <- c(donneesErreur, operationsDeControle)
        operationsDeControle <- NULL
        donneesGenerales <- NULL
    } else {

    #Pour chaque opération de contrôle : récupération des données puis
exécution de l'algorithme
    for(operationDeControle in operationsDeControle) {

        #Recuperation de la saison au niveau de l'operation de controle
        saison <- findDonneeContexte(operationDeControle,"Saison")
        if("Erreur" %in% saison) {
            donneesErreur <- c(donneesErreur,saison)
            saison <- NULL
        } else {

        #Récupération des données de l'opération de contrôle
        listeCtxHydroElem
c("CdEnginPêche","Salinite","ClasseDeSalinite","DureePêche","Commentaires") <-
        listeTraitDePêche <- findListeTraitDePêche(operationDeControle,
listeCtxHydroElem, "13")
        #si une erreur est survenue
        if("Erreur" %in% listeTraitDePêche) {
            donneesErreur <- c(donneesErreur,listeTraitDePêche)
            listeTraitDePêche <- NULL
        } else {

        #Exécution de l'algo
        #Parcours des trait de pêche et calculs intermediaires
        for(trait_id in listeTraitDePêche) {

            #Récupération du code engin pêche
            CdEnginPêche <- trait_id[1,"CdEnginPêche"]
            DureePêche <- trait_id[1,"DureePêche"]
            Commentaires <- trait_id[1,"Commentaires"]

            #Declarations des variables pour les calculs intermediaires
            salinite.c <- 0

            #Calcul de la salinite
            salinite.c<- calculeSalinite(trait_id[1,"Salinite"],trait_id[1,"ClasseDeSalinite"])

            #Calcul de la duree
            DureePêche <- as.numeric(DureePêche)
            if(is.na(DureePêche)) {
                DureePêche <- 0
            }

            #2- La deuxième étape est de sélectionner les enregistrements (trait_id) pour
            # lesquels la densité sera calculée. Cette sélection se fait sur les champs
            # engin, saison, ecoregion, commentaire et duree

            if((CdEnginPêche == "CAP1")

```

Janvier 2011 – Page 91/108

```

listeScore <- NULL

#Parcours des saisons
for(saison in DENS.IB.LAG.SAISONS) {
  cdSaison <- saison[1] # Code de la saison
  lbSaison <- saison[2] # Label de la saison

  #Parcours des salinités
  for (salinite.c in DENS.IB.LAG.SALINITES) {

    Densite <- 0

    selection1 <- subset(listeTraitIdValide,
      listeTraitIdValide[, "saison"] == cdSaison &
      listeTraitIdValide[, "salinite"] == salinite.c )
    NTrait <- nrow(selection1)

    if(NTrait != 0)
      sommeLogDens <- sum(as.numeric(selection1[, "logDens"]))

    if(NTrait != 0)
      Densite <- sommeLogDens / NTrait

    score <- calculScore(superficieCMasseDEau, SectionMasseDEau, lbSaison, salinite.c, Densite)

    #Attention, en estuaires seuls les scores dont le calcul est basé sur un nombre
    #d'enregistrements (NTrait) supérieur ou égal à 6 sont comptabilisés.
    # Pas pour les lagunes
    if(NTrait != 0){
      listeScore <- c(listeScore, score)
    }

    code <- paste("dens_ib_lag", cdSaison, salinite.c, sep="")
    libelle <- paste("Densité IB: ", lbSaison, "-Salinité ", salinite.c, sep="")
    produitsSecondairesNode <- AjouterNodeProduitSecondaire(produitsSecondairesNode,
      code, libelle, Densite, NULL, NULL)
    if("Erreur" %in% produitsSecondairesNode) {
      donneesErreur <- c(donneesErreur, produitsSecondairesNode)
      produitsSecondairesNode <- NULL
    }
  }
}

Nscore <- length(listeScore)
if(Nscore > 0 ){
  som.score <- sum(listeScore)
  note <- som.score / (4 * Nscore)
}else
  note <- 0

produitPrincipalNode <- CreerNodeProduitPrincipal("Note-IB",
  NULL, note, NULL, "dens_ib_lag")
if("Erreur" %in% produitPrincipalNode) {
  donneesErreur <- c(donneesErreur, produitPrincipalNode)
  produitPrincipalNode <- NULL
}

SortieXmlTree <- AjouterNodeResultats(SortieXmlTree,
  donneesGenerales,
  produitPrincipalNode,
  produitsSecondairesNode,
  donneesErreur,
  DonneesIntermediairesNode,
  DonneesPreCalculeesNode,
  DonneesExplicativesNode,

```

```
DonneesDeContexteNode,  
DonneesDeTracabiliteNode,  
DonneesIncertitudeNode)  
}#fin de fonction point de controle  
  
#Si absence d'erreur de lecture et présence de point(s) de contrôle  
#on lance l'algo, sinon on écrit seulement les erreurs générales  
if((length(erreursGenerales) == 0) ) {  
  #Lancement du parsing + listage des differents systemes et calcul de  
leur superficie (estuaires)  
  # doPointControleSAX(fichierDonneesObservation,funSysteme)  
  #Lancement du parsing + traitement a la volée des points de controles  
  doPointControleSAX(fichierDonneesObservation,funPointDeControle)  
}else  
SortieXmlTree <- AjouterNodeErreursGenerales(SortieXmlTree, erreursGenerales)  
  
Metrique <- "Metrique"  
Indice <- "Indice"  
#Il faut définir le type du modèle : soit mettre Metrique ou Indicateur.  
type <- Metrique  
  
#Ecriture du fichier de sortie  
EcrireFichierXml(type, SortieXmlTree, FichierSortie)  
}
```

ET-Poisson-DENS_Z_LAG

v 0.002

```
#####
# Densité de poissons zooplanctonophages dans les lagunes
# Author: Theunissenb
#####

#####
# Variables globales : INTERDICTION FORMELLE DE LES MODIFIER #
# fichierEntree : adresse du fichier d'entrée
#####

#Association code et label de chaque saison
ET.SAISON.PRINTEMPS <- c("0","printemps")
ET.SAISON.AUTOMNE   <- c("2","automne")
#Les saisons traitées par le modèle
DENS.Z.LAG.SAISONS  <- list(ET.SAISON.PRINTEMPS,ET.SAISON.AUTOMNE)

#Codes des eco-regions masse d'eau
ET.ECOREGION.MED    <- "MED"
#Codes des eco-regions masse d'eau traitées par le modèle
DENS.Z.LAG.ECOREGIONS <- c(ET.ECOREGION.MED)

DENS.Z.LAG.SALINITES <- c("1","3")

#Declaration des tableaux associés au modèle
DENS.Z.LAG.TAB.SEUIL.Z.LAGUNE.LABEL <- "seuilsZLagune"
DENS.Z.LAG.TAB.SEUIL.Z.LAGUNE      <- NULL

DENS.Z.LAG.TAB.GUILDE.TROPHIQUE.LABEL <- "GildeTrophique"
DENS.Z.LAG.TAB.GUILDE.TROPHIQUE      <- NULL

DENS.Z.LAG.TAB.PARAMETRE.LAGUNES.LABEL <- "ParamLagunes"
DENS.Z.LAG.TAB.PARAMETRE.LAGUNES      <- NULL

#Declaration des noms des colonnes des tableaux manipulés
DENS.Z.LAG.COL.SUPERFICIE <- "superficieC"
DENS.Z.LAG.COL.SECTION    <- "section"
DENS.Z.LAG.COL.SAISON     <- "saison"
DENS.Z.LAG.COL.SALINITE   <- "saliniteC"
DENS.Z.LAG.COL.SEUILS1    <- "seuils1"
DENS.Z.LAG.COL.SEUILS2    <- "seuils2"
DENS.Z.LAG.COL.SEUILS3    <- "seuils3"
DENS.Z.LAG.COL.SEUILS4    <- "seuils4"
DENS.Z.LAG.COL.TRAIT      <- "Trait"
DENS.Z.LAG.COL.PARAM.SECTION <- "ParamSection" # valeur de la section dans le
tableau de parametre
DENS.Z.LAG.COL.PARAM.MET   <- "ParamMET"

#Trait du poisson à considérer dans le modèle
DENS.Z.LAG.TRAIT <- "Z"

#Initialisation des erreurs
donneesErreur <- NULL
DENS.Z.LAG.LISTES.SYSTEMES <- NULL

#Définitions de fonctions internes

# FIXME Declaration temporaire d'une fonction de la bibliotheque RSEEE durant la
Recette.
## Récupération de la liste faunistique dans les données d'observation.
# param operationContrôle le noeud de l'opération de contrôle.
# param codesUnite le(s) code(s) de l'unité bio concernant les résultats d'une
observation hydrobiologique, par défaut NULL.
```

```
# param listeDonneesContexteHydroElem le(s) libellé(s) des données de contexte
souhaitées, par défaut NULL.
# param bocal indique si une colonne contenant le bocal doit être ajoutée, par
défaut FALSE.
# param listeTypeListe liste des typeListe que la listeHydrobiologique doit
contenir, par défaut NULL.
# Si la listeHydrobiologique n'est pas d'un type compris dans typeListe alors
# l'ensemble des observationHydrobiologiques sous jacentes est ignoré.
# param force si vrai, alors les listes hydrobiologiques n'ayant pas d'observation
hydrobiologique
# apparaissent toutefois avec les données: idTaxon,resultat et codeUniteBio
valorisées à "NULL", par défaut FALSE.
# return la liste des taxons (idTaxon,resultat,codeUniteBio).
# erreur si une donnée est manquante.
findObservationsHydrobiologique <- fonction(operationControle,
      codesUnite=NULL,
      listeDonneesContexteHydroElem=NULL,
      bocal=FALSE,
      listeTypeListe=NULL,
      force=FALSE){
  # initialisation
  listeFaunistique <- NULL
  listesHydrobiologique
  findElement(operationControle,"ListesHydrobiologique")
  if("Erreur" %in% listesHydrobiologique)
    return(listesHydrobiologique)
  listesHydrobiologique
  findElement(listesHydrobiologique,"ListeHydrobiologique")
  if("Erreur" %in% listesHydrobiologique)
    return(listesHydrobiologique)

  # nombre de listesHydrobiologique
  nbListesHydrobiologique <- length(listesHydrobiologique)
  if(nbListesHydrobiologique < 1){
    erreur <- c("Erreur","EF06","Donnée d'observation manquante")
    return(erreur)
  }
  for(i in 1:nbListesHydrobiologique){
    vNumeroOrdre      <- NULL
    vBocal             <- NULL
    donneeContexte    <- NULL
    codeTypeListe      <- NULL
    idListeHydrobiologique <- NULL

    listeHydrobiologique <- listesHydrobiologique[i]

    # Recuperation du numero de passage associe a ListeHydrobiologique
    opcHydrobioElementaire
    findElement(listeHydrobiologique,"OPCHydrobioElementaire")
    if("Erreur" %in% opcHydrobioElementaire){ # Si une erreur est survenue
      vNumeroOrdre <- "NULL"
      if(bocal) # le bocal est demande et non disponible
        return(opcHydrobioElementaire) # Envoi de l'erreur
    }
    else{ # si pas d'erreur survenue
      vNumeroOrdre
      findDonneesFromElement(opcHydrobioElementaire,"numOrdre",convertFromUTF8=FALSE)
      if("Erreur" %in% vNumeroOrdre) # si une erreur est survenue
        vNumeroOrdre <- "NULL"

      # Recherche du bocal associé a ListeHydrobiologique
      if(bocal){
        vBocal <- findDonneesFromElement(opcHydrobioElementaire,
"numRegroupement",convertFromUTF8=FALSE)
        if("Erreur" %in% vBocal) # si une erreur est survenue
          return(vBocal)
      }
    }
  }
  # recherche du typeListe de la listeHydrobiologique contenant les
observationHydrobiologiques
```



```

    if(!is.null(listeTypeListe)){
      typeListe <- findElement(listeHydrobiologique,"TypeListe")
      if("Erreur" %in% typeListe) # si une erreur est survenue
        next
    }
    codeTypeListe <- findDonneesFromElement(typeListe,"code",convertFromUTF8=FALSE)
    if(! codeTypeListe %in% listeTypeListe)
      next
  }
  # Recherche des donnees de contexte si demande
  if(!is.null(listeDonneesContexteHydroElem)){
    opchHydrobioElementaire <-
findElement(listeHydrobiologique,"OPCHydrobioElementaire")
    if("Erreur" %in% opchHydrobioElementaire) # si une erreur est survenue
      return(opchHydrobioElementaire)
    donneeContexte <- findDonneeContexte(opchHydrobioElementaire,
listeDonneesContexteHydroElem,force=TRUE,convertFromUTF8=FALSE)
    if("Erreur" %in% donneeContexte) # si une erreur est survenue
      return(donneeContexte)
    # si une erreur est survenue
    mais cela ne devrait jamais arriver
    next
  }
  # Recuperation de l'id de la liste Hydrobiologique
  idListeHydrobiologique <- findDonneesFromElement(listeHydrobiologique,
"idListeHydrobiologique",convertFromUTF8=FALSE)
  if("Erreur" %in% idListeHydrobiologique) # si une erreur est survenue
    return(idListeHydrobiologique)

  observationsHydrobiologique <-
findElement(listeHydrobiologique,"ObservationsHydrobiologique")
  if("Erreur" %in% observationsHydrobiologique){
    if(force)
      listeFaunistique <- c(listeFaunistique,"NULL","NULL","NULL",
"NULL",idListeHydrobiologique,vNumeroOrdre,vBocal,codeTypeListe,donneeContexte)
    next
  }
  observationsHydrobiologique <-
findElement(observationsHydrobiologique, "ObservationHydrobiologique")
  if("Erreur" %in% observationsHydrobiologique){
    if(force)
      listeFaunistique <- c(listeFaunistique,"NULL","NULL","NULL",
"NULL",idListeHydrobiologique,vNumeroOrdre,vBocal,codeTypeListe,donneeContexte)
    next
  }
  # nombre de observationsHydrobiologique
  nbObservationsHydrobiologique <- length(observationsHydrobiologique)

  if(nbObservationsHydrobiologique < 1){
    erreur <- c("Erreur","EF06","Donnée d'observation manquante")
    return(erreur)
  }
  for(j in 1:nbObservationsHydrobiologique){
    observationHydrobiologique <- observationsHydrobiologique[j]
    # on retrouve l'id du taxon
    donneesOb <- findDonneesFromElement(observationHydrobiologique,
c("taxon","resultat"),force=TRUE,convertFromUTF8=FALSE)
    if("Erreur" %in% donneesOb){
      taxon <- "NULL"
      resultat <- "NULL"
    }
    else{
      taxon <- donneesOb["taxon"]
      resultat <- donneesOb["resultat"]
    }
    uniteBio <- findElement(observationHydrobiologique,"UniteBio")
    if("Erreur" %in% uniteBio){ # si une erreur est survenue
      erreur <- c("Erreur","EF26","Information sur l'unité bio manquante")
      return(erreur)
    }
  }
  codeUniteBio <- findDonneesFromElement(uniteBio,"code",convertFromUTF8=FALSE)
  if("Erreur" %in% codeUniteBio){ # si une erreur est survenue
    erreur <- c("Erreur","EF27","Code de l'unité bio manquant")
  }

```

```

    return(erreur)
  }
  typeLot <- findElement(observationHydrobiologique,"TypeLot")
  if("Erreur" %in% typeLot){ # si une erreur est survenue
    codeTypeLot <- "NULL"
  }
  codeTypeLot <- findDonneesFromElement(typeLot,"code",convertFromUTF8=FALSE)
  if("Erreur" %in% codeTypeLot){ # si une erreur est survenue
    codeTypeLot <- "NULL"
  }

  if(is.null(codesUnite)){ # on prend toutes les observations
    listeFaunistique <- c(listeFaunistique,taxon,resultat,codeUniteBio,
codeTypeLot,idListeHydrobiologique,vNumeroOrdre,vBocal,codeTypeListe,donneeContexte
)
  }
  else{
    if(codeUniteBio %in% codesUnite){ # On conserve cette observation
      listeFaunistique <- c(listeFaunistique,taxon,resultat,codeUniteBio,
codeTypeLot,idListeHydrobiologique,vNumeroOrdre,vBocal,codeTypeListe,donneeContexte
)
    }
  }
}
# formatage des resultats
nomsColonnes <- c("idTaxon","resultat","codeUniteBio","codeTypeLot","idListeHydrobiologique","numOrdre")
if(bocal)
  nomsColonnes <- c(nomsColonnes,"bocal")
if(!is.null(listeTypeListe))
  nomsColonnes <- c(nomsColonnes,"typeListe")
if(!is.null(listeDonneesContexteHydroElem))
  nomsColonnes <- c(nomsColonnes,listeDonneesContexteHydroElem)

if(length(listeFaunistique) > 0){
  listeFaunistique <- matrix(listeFaunistique,ncol=length(nomsColonnes),byrow=TRUE)
  colnames(listeFaunistique) <- nomsColonnes
  # Conversion des données issues d'un flux XML (UTF-8) vers l'encoding
  de R courant.
  listeFaunistique <- fromUTF8(listeFaunistique)
  return(listeFaunistique)
}
else{
  erreur <- c("Erreur","EF06","Donnée d'observation manquante")
  return(erreur)
}
} # fin fonction findObservationsHydrobiologique

#calcul de la salinite
#param :
#Salinite mesuree
#Degre de salinite estimee
#return : Salinite calculee
calculeSalinite <- function(salinite,salinite.classe) {
  salinite <- as.numeric(salinite)
  salinite.classe <- as.numeric(salinite.classe)
  salinite.ccal <- NA

  if(!is.na(salinite)){
    if(salinite <= 18) {
      salinite.ccal <- 1
    } else {
      salinite.ccal <- 3
    }
  }
}

```

```

#si salinite-ccal est pas vide alors salinite-c = salinite-ccal
if(! is.na(salinite.ccal))
  return(salinite.ccal)
#si salinite-ccal est vide ET salinite-classe est pas vide alors
if( is.na(salinite.ccal) && ! is.na(salinite.classe)) {
  #si salinite-classe=3 alors salinite-c=3 sinon salinite-c=1
  if(salinite.classe==3)
    return(3)
  else { return(1)} }
else #si salinite-ccal est vide ET salinite-classe est vide alors salinite-c
reste vide
  return(0)
}

```

```

#Ainsi en fonction de la valeur de Densité, un score (noté score) sera
#attribué à chaque combinaison (i) de masse_eau x saison x salinite-c selon
#les critères suivant :
#- score prend la valeur 0 si Densité (i) est inférieure au seuil inférieur (inf)
#- score prend la valeur 2 si Densité (i) est supérieure ou égale au seuil
inférieur (inf) et inférieure au seuil supérieur (sup)
#- score prend la valeur 4 si Densité (i) est supérieure ou égale au seuil
supérieur (sup)
#param:
#superficieCMasseDEau      :
#SectionMasseDEau         :
#saison                    : le nom de la saison
#salinite.c                : la classe de la salinite
#densite                   :
calculScore                                                         <-
function(superficieCMasseDEau,SectionMasseDEau,lbSaison,salinite.c,Densite){
  tab <- DENS.Z.LAG.TAB.SEUIL.Z.LAGUNE

  #Selection des lignes du tableau verifiant les criteres donnés en argument
  tab <- subset(tab,
    tab[,DENS.Z.LAG.COL.SUPERFICIE] == superficieCMasseDEau &
    tab[,DENS.Z.LAG.COL.SECTION] == SectionMasseDEau &
    tab[,DENS.Z.LAG.COL.SAISON] == lbSaison &
    tab[,DENS.Z.LAG.COL.SALINITE] == salinite.c)

  if(nrow(tab) == 0){
    return(NULL)
  }

  S1 <- as.numeric(tab[1,DENS.Z.LAG.COL.SEUILS1])
  S2 <- as.numeric(tab[1,DENS.Z.LAG.COL.SEUILS2])
  S3 <- as.numeric(tab[1,DENS.Z.LAG.COL.SEUILS3])
  S4 <- as.numeric(tab[1,DENS.Z.LAG.COL.SEUILS4])

  if((Densite >= S2) && (Densite < S3)) { return(2) }

  if(!is.na(S1)) {
    if(Densite < S1) { return(4) }
    else {
      if(Densite < S2) { return(3) } } }
  else {
    if(Densite < S2) { return(4) } }

  if(!is.na(S4)) {
    if(Densite >= S4) { return(0) }
    else {
      if(Densite >= S3) { return(1) } } }
  else {
    if(Densite >= S3) { return(0) } }

}

#Exécution de l'algo
#run : fonction appelée par SEEE pour effectuer l'exécution d'un algorithme
run <- function() {

```

```

#Initialisation de la liste des erreurs générales
erreursGenerales <- NULL
#Récupération des données dans le fichier XML d'entrée
#Attention : ces variables ne doivent pas être modifiées
#le nom du fichier d'entrée est passé en variable globale (fichierEntree)
#Récupération de l'élément racine du fichier d'entrée
rootElement <- findRootElement(fichierEntree)
#Si la récupération s'est mal passée (l'élément racine n'existe pas dans le XML)
if("Erreur" %in% rootElement) {
  erreursGenerales <- c(erreursGenerales, rootElement)
  rootElement <- NULL
  utilisateur <- NULL
}else {
  #Récupération de l'utilisateur qui a lancé l'algorithme
  utilisateur <- findUtilisateur(rootElement)
  #Si l'utilisateur a été mal spécifié dans le fichier d'entrée
  if("Erreur" %in% utilisateur) {
    erreursGenerales <- c(erreursGenerales, utilisateur)
    utilisateur <- NULL
  }

  #Récupération de l'adresse du fichier de sortie
  FichierSortie <- findFichierSortie(rootElement)
  #Si l'adresse du fichier de sortie a été mal spécifiée
  if("Erreur" %in% FichierSortie) {
    erreursGenerales <- c(erreursGenerales, FichierSortie)
    FichierSortie <- NULL
  }

  #Récupération de l'adresse du fichier des données d'observation
  fichierDonneesObservation <-
findFichierDonneesObservation(rootElement)
  #Si l'adresse du fichier des données d'observation a été mal spécifiée
  if("Erreur" %in% fichierDonneesObservation) {
    erreursGenerales <- c(erreursGenerales, fichierDonneesObservation)
    fichierDonneesObservation <- NULL
  }

  #récupération des adresses des fichiers contenant les données de
référence (si nécessaire)
  fichiersDonneesReference <- findListeTabReference(rootElement)
  #si l'adresse du fichier des données d'observation a été mal spécifiée
  if("Erreur" %in% fichiersDonneesReference) {
    erreursGenerales <- c(erreursGenerales, fichiersDonneesReference)
    fichiersDonneesReference <- NULL
  }else{
    referentiel_taxon <-
findTabRefTaxon(fichiersDonneesReference["TabRefTaxon"],
c("cdTaxonSandre", "embranchementTaxon"))
    if("Erreur" %in% referentiel_taxon){
      erreursGenerales <- c(erreursGenerales, referentiel_taxon)
      referentiel_taxon <- NULL
    }
  }

  #recuperation des donnees des tableaux
  DENS.Z.LAG.TAB.SEUIL.Z.LAGUNE <-
findTableau(rootElement, DENS.Z.LAG.TAB.SEUIL.Z.LAGUNE.LABEL)
  if("Erreur" %in% DENS.Z.LAG.TAB.SEUIL.Z.LAGUNE) {
    erreursGenerales <- c(erreursGenerales,
DENS.Z.LAG.TAB.SEUIL.Z.LAGUNE)
    DENS.Z.LAG.TAB.SEUIL.Z.LAGUNE <- NULL
  }
  DENS.Z.LAG.TAB.GUILDE.TROPHIQUE <-
findTableau(rootElement, DENS.Z.LAG.TAB.GUILDE.TROPHIQUE.LABEL)
  if("Erreur" %in% DENS.Z.LAG.TAB.GUILDE.TROPHIQUE) {
    erreursGenerales <- c(erreursGenerales,
DENS.Z.LAG.TAB.GUILDE.TROPHIQUE)
    DENS.Z.LAG.TAB.GUILDE.TROPHIQUE <- NULL
  }
}

```

```

DENS.Z.LAG.TAB.PARAMETRE.LAGUNES <- NULL
findTableau(rootElement,DENS.Z.LAG.TAB.PARAMETRE.LAGUNES.LABEL)
  if("Erreur" %in% DENS.Z.LAG.TAB.PARAMETRE.LAGUNES) {
    erreursGenerales <- c(erreursGenerales,
DENS.Z.LAG.TAB.PARAMETRE.LAGUNES)
    DENS.Z.LAG.TAB.PARAMETRE.LAGUNES <- NULL
  }

#libération de la mémoire
rm(rootElement)

}
#Initialisation de l'arbre Xml de sortie
SortieXmlTree <- InitialiserXmlTreeSortie(utilisateur,FichierSortie)

#On boucle sur les points de contrôle
funPointDeControle = function(pointDeControle) {
  #####Préparation des resultats au format XML
  ### Xml de sortie
  #Initialisation des variables pour les noeuds permettant de faire le
fichier xml de sortie
  produitPrincipalNode <- NULL
  produitsSecondairesNode <- NULL
  DonneesIntermediairesNode <- NULL
  DonneesPreCalculeesNode <- NULL
  DonneesExplicativesNode <- NULL
  DonneesDeContexteNode <- NULL
  DonneesDeTracabiliteNode <- NULL
  DonneesIncertitudeNode <- NULL

  #Initialisation des produits secondaires
  produitsSecondairesNode <- InitNodeProduitsSecondaires()

  #Initialisation des erreurs
  donneesGenerales <- NULL
  donneesErreur <- NULL

  #Declaration de la liste des enregistrements valides
  listeTraitIdValide <- list()
  localisation <- NULL
  #récupération de l'id du point de controle

  #récupération des données générales de l'opération de contrôle
  donneesGenerales <- findDonneesGenerales(pointDeControle, NULL)
  #Si les données générales n'ont pas été correctement saisies
  if("Erreur" %in% donneesGenerales) {
    donneesErreur <- donneesGenerales
    SortieXmlTree <- AjouterNodeResultats(SortieXmlTree,
NULL,NULL,NULL,donneesErreur,NULL,NULL,NULL,NULL,NULL,NULL)
    donneesGenerales <- NULL
  }
  else {
    #Recuperation de la localisation (systeme-estuaire) du point de
controle

    #Recuperation du code EcoRegion et de la superficie totale de la
masse d'eau au niveau du point de controle
    EcoRegionMasseDEau <- findDonneeContexte(pointDeControle, "CdEcoregion")
    if("Erreur" %in% EcoRegionMasseDEau) {
      donneesErreur <- EcoRegionMasseDEau
      EcoRegionMasseDEau <- NULL
    } else {
      #Initialisation des données
      superficieCMasseDEau <- NULL
superficie <- findDonneeContexte(pointDeControle, "SuperficieMasseDEauET")
      if("Erreur" %in% superficie) {
        donneesErreur <- superficie
        superficie <- NA
      }
    }
  }
}

```

```

#Affectation de la superficie de la masse d'eau
    if(!is.na((superficie))) {
    if(as.numeric(superficie) <= 50) {
        superficieCMasseDEau <- "petit"
    } else {
        superficieCMasseDEau <- "grand"
    }
}

#Recuperation de la section de la masse d'eau au niveau du point de controle
    tabConn <- DENS.Z.LAG.TAB.PARAMETRE.LAGUNES
    SectionMasseDEau<-NULL
    #Selection de la ligne pour le point de controle analysé
    CdPointDeControle <- findDonneesFromElement(pointDeControle,
"cdPointDeControle")
    tabConn <- subset(tabConn, tabConn[,DENS.Z.LAG.COL.PARAM.MET] ==
CdPointDeControle )
    if (nrow(tabConn)!=1) {
        donneesErreur <- c(donneesErreur, SectionMasseDEau)
    } else {
        SectionMasseDEau <- tabConn[1,DENS.Z.LAG.COL.PARAM.SECTION]

#On récupère les opérations de controle liées au point de controle
    operationsDeControle <- findOperationsControle(pointDeControle)
    if("Erreur" %in% operationsDeControle) {
        donneesErreur <- c(donneesErreur, operationsDeControle)
        operationsDeControle <- NULL
        donneesGenerales <- NULL
    } else {

#Pour chaque opération de contrôle : récupération des données puis exécution
de l'algorithme
        for(operationDeControle in operationsDeControle) {

#Recuperation de la saison au niveau de l'operation de controle
            saison <- findDonneeContexte(operationDeControle,"Saison")
            if("Erreur" %in% saison) {
                donneesErreur <- c(donneesErreur,saison)
                saison <- NULL
            } else {

#Récupération des données de l'opération de contrôle
                listeCtxHydroElem <- c("CdEnginPêche","Salinite","ClasseDeSalinite","DureePêche","Commentaires")
                listeTraitDePêche <- findListeTraitDePêche(operationDeControle,
listeCtxHydroElem, "13")
                #si une erreur est survenue
                if("Erreur" %in% listeTraitDePêche) {
                    donneesErreur <- c(donneesErreur,listeTraitDePêche)
                    listeTraitDePêche <- NULL
                } else {

#Exécution de l'algo
#Parcours des trait de pêche et calculs intermediaires
for(trait_id in listeTraitDePêche) {

#Récuperation du code engin pêche
                CdEnginPêche <- trait_id[1,"CdEnginPêche"]
                DureePêche <- trait_id[1,"DureePêche"]
                Commentaires <- trait_id[1,"Commentaires"]

#Declarations des variables pour les calculs intermediaires
                salinite.c <- 0

#Calcul de la salinite
salinite.c<- calculeSalinite(trait_id[1,"Salinite"],trait_id[1,"ClasseDeSalinite"])

#Calcul de la duree
                DureePêche <- as.numeric(DureePêche)
                if(is.na(DureePêche)) {
                    DureePêche <- 0
                }

#2- La deuxième étape est de sélectionner les enregistrements (trait_id) pour

```

```
# lesquels la densité sera calculée. Cette sélection se fait sur les champs
# engin, saison, ecoregion, commentaire et duree

if((CdEnginPêche == "CAP1")
  && (saison %in% unlist(DENS.Z.LAG.SAISONS))
  && (EcoRegionMasseDEau %in% DENS.Z.LAG.ECOREGIONS)
  && !grepl("INVALIDE", Commentaires, ignore.case = TRUE)
  && !grepl("protocole non respect", Commentaires, ignore.case = TRUE)
  && ((DureePêche >= 1200) && (DureePêche <= 1680))
  && (salinite.c != 0)
) {

  Ab      <- 0

#Parcours des observations du trait_id_valide
for(k in 1:nrow(trait_id)){
  idTaxon  <- trait_id[k,"idTaxon"]
  resultat <- trait_id[k,"resultat"]

#3- L'étape suivante consistera à calculer pour chaque trait_id_valide la
log-densité de poissons appartenant à la guilde.

  if(idTaxon %in% rownames(referentiel_taxon)) {
    cdTaxonSandre <- referentiel_taxon[idTaxon,"cdTaxonSandre"]
    if(cdTaxonSandre %in% rownames(DENS.Z.LAG.TAB.GUILDE.TROPHIQUE)) {

      if(DENS.Z.LAG.TAB.GUILDE.TROPHIQUE[cdTaxonSandre,DENS.Z.LAG.COL.TRAIT] ==
DENS.Z.LAG.TRAIT && resultat != "NULL") {

        Ab <- Ab + as.integer(resultat)

      }
    }
  }

log.Dens      <- log(Ab+1 , base = exp(1))
listeTraitIdValide <- c(listeTraitIdValide,saison,salinite.c,log.Dens)
}
}

#Transformation de la liste des TraitIdValide en matrice
listeTraitIdValide <- matrix(listeTraitIdValide,ncol=3,byrow=TRUE)
colnames(listeTraitIdValide) <- c("saison","salinite","logDens")
#Nombre de trait valide, que ce soit Pres=1 ou Pres=0
nbTraitValide <- nrow(listeTraitIdValide)

#Dans certains cas, il se peut qu'il n'y ait pas eu d'échantillonnages
#et donc cette valeur de Densité ne pourra pas être calculée.
if(nbTraitValide == 0){
  donneesErreur <- c(donneesErreur, c("Erreur", "EF58", "Aucun
enregistrement valide pour ce point de contrôle.))
}else{

#4- Dans la section précédente, les calculs étaient effectués pour
#chaque trait_id_valide. Dans cette partie, nous allons procéder
#à un regroupement de ces valeurs en fonction de la masse d'eau,
#de la saison et de la classe de salinité de façon à pouvoir comparer
#ces valeurs aux seuils fournis. Les calculs suivants sont effectués
#pour chaque combinaison masse_eau x saison x salinite-c
#Par exemple : pour la masse d'eau Loire, au printemps et dans la salinite-c 1.
```

```
# FIXME /\ La masse d'eau est assimilée au point de contrôle.
#On travail donc sur la combinaison PointDeControle x saison x salinite-c

listeScore <- NULL

#Parcours des saisons
for(saison in DENS.Z.LAG.SAISONS) {
  cdSaison <- saison[1] # Code de la saison
  lbSaison <- saison[2] # Label de la saison

  #Parcours des salinités
  for (salinite.c in DENS.Z.LAG.SALINITES) {

    Densite <- 0

    selection1 <- subset(listeTraitIdValide,
      listeTraitIdValide[, "saison"] == cdSaison &
      listeTraitIdValide[, "salinite"] == salinite.c )
    NTrait <- nrow(selection1)

    if(NTrait != 0)
      sommeLogDens <- sum(as.numeric(selection1[, "logDens"]))

    if(NTrait != 0)
      Densite <- sommeLogDens / NTrait

    score <- calculScore(superficieCMasseDEau, SectionMasseDEau, lbSaison, salinite.c, Densite)

    #Attention, en estuaires seuls les scores dont le calcul est basé sur un nombre
    #d'enregistrements (NTrait) supérieur ou égal à 6 sont comptabilisés.
    # Pas pour les lagunes
    if(NTrait != 0){
      listeScore <- c(listeScore, score)
    }

    code <- paste("dens_z_lag", cdSaison, salinite.c, sep="")
    libelle <- paste("Densité Z: ", lbSaison, "-Salinité ", salinite.c, sep="")
    produitsSecondairesNode <- AjouterNodeProduitSecondaire(produitsSecondairesNode, code, libelle, Densite, NULL,
      NULL)

    if("Erreur" %in% produitsSecondairesNode) {
      donneesErreur <- c(donneesErreur, produitsSecondairesNode)
      produitsSecondairesNode <- NULL
    }
  } # fin de boucle sur les saisons

  Nscore <- length(listeScore)
  if(Nscore > 0 ){
    som.score <- sum(listeScore)
    note <- som.score / (4 * Nscore)
  }else
    note <- 0

  produitPrincipalNode <- CreerNodeProduitPrincipal("Note-Z",
    NULL, note, NULL, "dens_z_lag")
  if("Erreur" %in% produitPrincipalNode) {
    donneesErreur <- c(donneesErreur, produitPrincipalNode)
    produitPrincipalNode <- NULL
  }
}

SortieXmlTree <- AjouterNodeResultats(SortieXmlTree,
  donneesGenerales,
```



```
        produitPrincipalNode,  
        produitsSecondairesNode,  
        donneesErreur,  
        DonneesIntermediairesNode,  
        DonneesPreCalculeesNode,  
        DonneesExplicativesNode,  
        DonneesDeContexteNode,  
        DonneesDeTracabiliteNode,  
        DonneesIncertitudeNode)  
    }#fin de fonction point de controle  
  
    #Si absence d'erreur de lecture et présence de point(s) de contrôle  
    #on lance l'algo, sinon on écrit seulement les erreurs générales  
    if((length(erreursGenerales) == 0) ) {  
        #Lancement du parsing + listage des differents systemes et calcul de  
leur superficie (estuaires)  
        # doPointControleSAX(fichierDonneesObservation,funSysteme)  
        #Lancement du parsing + traitement a la volée des points de controles  
        doPointControleSAX(fichierDonneesObservation,funPointDeControle)  
    }else  
    SortieXmlTree <- AjouterNodeErreursGenerales(SortieXmlTree, erreursGenerales)  
  
    Metrique <- "Metrrique"  
    Indice <- "Indice"  
    #Il faut définir le type du modèle : soit mettre Metrique ou Indicateur.  
    type <- Metrique  
  
    #Ecriture du fichier de sortie  
    EcrireFichierXml(type, SortieXmlTree, FichierSortie)  
}
```

INDIC_POISSON_LAG**v 0.001**

```
#####
# Calcul du modèle INDIC_POISSON_LAG
# Description succincte : Indicateur multimétrique de qualité des lagunes
# concernant la composante "poisson"
# Author: Logica
#####

#####
# Variables globales : INTERDICTION FORMELLE DE LES MODIFIER
# fichierEntree : adresse du fichier d'entrée
#####

IPE.TYPE.CLASSE.ETAT <- "DCE5BVJOR"

#exécution de l'algo
run <- function() {

  #Initialisation de la liste des erreurs
  erreursGenerales <- NULL

  #Recuperation des donnees dans le fichier XML d'entree
  #Attention : ces variables ne doivent pas être modifiées
  #le nom du fichier d'entrée en passé en variable globale (FichierEntree)
  #recuperation de l'element racine du fichier d'entrée
  rootElement <- findRootElement(fichierEntree)
#si la récupération s'est mal passée (l'élément racine n'existe pas dans le XML)
  if("Erreur" %in% rootElement) {
    erreursGenerales <- c(erreursGenerales, rootElement)
    rootElement <- NULL
    utilisateur <- NULL
  }else {
    #recuperation de l'utilisateur qui a lance l'algo
    utilisateur <- findUtilisateur(rootElement)
    #si l'utilisateur a été mal rentré dans le fichier d'entrée
    if("Erreur" %in% utilisateur) {
      erreursGenerales <- c(erreursGenerales, utilisateur)
      utilisateur <- NULL
    }

    #récupération de l'adresse du fichier de sortie
    FichierSortie <- findFichierSortie(rootElement)
    #si l'adresse du fichier de sortie a été mal spécifiée
    if("Erreur" %in% FichierSortie) {
      erreursGenerales <- c(erreursGenerales, FichierSortie)
      FichierSortie <- NULL
    }

    #récupération des adresses des fichiers de sortie des métriques
    précédentes (si nécessaire)
    fichiersSortieIndicateurs <- findListeIndicateurs(rootElement)
    #si l'adresse du fichier des données d'observation a été mal spécifiée
    if("Erreur" %in% fichiersSortieIndicateurs) {
      erreursGenerales <- c(erreursGenerales, fichiersSortieIndicateurs)
      fichiersSortieIndicateurs <- NULL
    }

    # récupère la liste des mnémoniques
    listeMnemonic <- findListeMnemonic(rootElement)
    if("Erreur" %in% listeMnemonic){
      erreursGenerales <- c(erreursGenerales,listeMnemonic)
      listeMnemonic <- NULL
    }

    #libération de la mémoire
  }
```

```

    rm(rootElement)
  }

  #Initialisation de l'arbre Xml de sortie
  SortieXmlTree <- InitialiserXmlTreeSortie(utilisateur)

  # position type de classe d'etat
  positionnerTypeClasseEtat(IPE.TYPE.CLASSE.ETAT)
  # parametrage calcul classe etat
  paramettrerOrigineCEEQR()

  ## Traitement de l'indice.
  # param resultats un résultat.
  funIndice <- fonction(resultats){

    ## Effectue les traitements d'un résultat.
    # param resultat une ligne de résultat.
    funTraitementResultat <- fonction(resultat){
      # Initialisation des erreurs
      donneesErreur <- NULL

      # preparation des resultats au format XML
      # XML de sortie
      # Initialisation des variables pour les noeuds permettant de
      faire les XML de sortie
      produitPrincipalNode          <- NULL
      produitsSecondairesNode      <- NULL
      donneesIntermediairesNode    <- NULL
      donneesPreCalculeesNode      <- NULL
      donneesExplicativesNode      <- NULL
      donneesContexteNode          <- NULL
      donneesTracabiliteNode       <- NULL
      donneesIncertitudeNode       <- NULL

      # Variables auxiliaires
      eqrsProduitPrincipal         <- NULL
      eqrAgregeProduitPrincipal    <- NULL
      classeEtatProduitPrincipal   <- NULL

      # donnees generales
      donneesGenerales <- findDonneesGeneralesResultat(resultat)
      if("Erreur" %in% donneesGenerales){
        donneesErreur <- c(donneesErreur,donneesGenerales)
        donneesGenerales <- NULL
      }

      for(mnemonique in listeMnemonique){
        # produit principal
        produitPrincipal <- findProduitPrincipalResultat(resultat,mnemonique)
        if("Erreur" %in% produitPrincipal){
          donneesErreur <- c(donneesErreur,produitPrincipal)
          produitPrincipal <- NULL
        }
        else{
          stringEQR <- produitPrincipal["EQR"]
          eqr <- as.numeric(stringEQR)
          if(is.na(eqr)){
            erreur <- c("Erreur","ERR-T","EQR produit principal non numérique")
            donneesErreur <- c(donneesErreur,erreur)
          }
          else
            eqrsProduitPrincipal <- c(eqrsProduitPrincipal,eqr)
        }
      }

      if(!is.null(eqrsProduitPrincipal))
        eqrAgregeProduitPrincipal <- mean(eqrsProduitPrincipal)

      if(!is.null(eqrAgregeProduitPrincipal)){
        classeEtatProduitPrincipal <- findCodeClasseEtat(eqrAgregeProduitPrincipal)
      }
    }
  }

```

```

        if("Erreur" %in% classeEtatProduitPrincipal){
donneesErreur <- c(donneesErreur,classeEtatProduitPrincipal)
classeEtatProduitPrincipal <- NULL
        }
    }

    # Produit principal
    produitPrincipalNode <- CreerNodeProduitPrincipal("Indicateur poisson",
NULL,eqrAgregeProduitPrincipal,classeEtatProduitPrincipal,"INDIC_POISSON_LAG")
    if("Erreur" %in% produitPrincipalNode){
        donneesErreur <- c(donneesErreur,produitPrincipalNode)
        produitPrincipalNode <- NULL
    }

    # Ajout des nouveaux resultats a l'arbre XML de sortie
    SortieXmlTree <- AjouterNodeResultats(SortieXmlTree,
        donneesGenerales,
        produitPrincipalNode,
        produitsSecondairesNode,
        donneesErreur,
        donneesIntermediairesNode,
        donneesPreCalculeesNode,
        donneesExplicativesNode,
        donneesContexteNode,
        donneesTracabiliteNode,
        donneesIncertitudeNode)
    } # fin fonction funTraitementResultat

    doTraitementResultat(resultats,funTraitementResultat)
} # fin fonction funIndice

if(length(erreursGenerales) == 0){
    # tableau intermediaire

doCalculIndiceSurTableauIntermediaire(fichiersSortieIndicateurs,funIndice)
}
else{
SortieXmlTree <- AjouterNodeErreursGenerales(SortieXmlTree, erreursGenerales)
}

Metrique <- "Metrique"
Indice <- "Indice"
#il faut définir le type du modèle : soit mettre Metrique ou Indice
type <- Indice
#Ecriture du fichier de sortie
EcrireFichierXml(type, SortieXmlTree, FichierSortie)
}

```

Onema
Hall C – Le Nadar
5 square Félix Nadar
94300 Vincennes
01 45 14 36 00
www.onema.fr

Cemagref
Parc de Tourvoie
BP 44,
92163 Antony cedex
01 40 96 61 21
www.cemagref.fr