Documentation Script - Contrôle qualité données POEM 2022

24 mai 2022

Sommaire

1 Objectif

Ces scripts sont une implémentation simple d'un processus de contrôle qualité semi-automatique.

Ce que font ces scripts :

- Simplifie la validatation manuelle
- applique des tests logiques sur les données avec des seuils prédéterminées pour attribuer des codes qualités

Ce que ne font pas ces scripts :

- La recalibration
- L'interpolation de données

2 Fichiers et Dossier

- `manual_validator.py` : permet d'effectuer la "validation manuelle"
- `qc_engine.py` : permet de générer un fichier avec les codes qualités en plus
- `config.py` : permet une configuration simple du script `qc_engine.py`
- `_build/` : contient les fichiers produit par le script `qc_engine.py`
- `data/` : contient les données brutes et les fichiers produits par le script `manual_validator`
- `readers/` : contient les librairies de lecture de fichiers brutes
- `tools/` : contient les librairies aditionnels d'outils

3 Préquis

Les scripts ont été écrits avec les versions de packages/librairies/logiciels suivantes. Les scripts n'ont pas été testé avec d'autres versions, cependant ils peuvent tout de même fonctionner.

- Python 3.9.7 (dans un environnement conda/anaconda, de préférence)
- Spyder 5 (disponible de base dans l'environnement conda/anaconda)
- numpy 1.20.3
- matplotlib 3.4.3
- pandas 1.3.4

Il est également nécessaire d'afficher les graphiques matplotlib en mode fenêtré. Il faut configuer celà dans Spyder.

`Tools > Preferences > IPython console > Graphics > Graphics backend > Automatic`

4 Usage / Processus

1. Validation manuelle

- Lancer le script `manual_validator.py` dans l'interface spyder
- selectionner sur quel paramètre effectuer la validation manuel dans la console (par exemple [0] puis [ENTER] pour effectuer la validation manuel de la temperature
- enregister en cliquant sur [SAVE]
- renouveler la manipulation pour chaquer paramètre

2. Contrôle Qualité Semi-Automatique

- modifier la configuration dans `config.py` si necessaire
- lancer le script `qc_engine.py` dans l'interface spyder

3. Recupérer les fichiers, ils devraient se trouver dans un sous dossier `_build/`

5 Code Qualité

Les codes qualités utilisés par le script de contrôle qualité sont référencés dans la table 1

Table 1: Valeurs des codes qualités et leur signification.

code qualité	signification
0	Contrôle qualité (QC) non effectué
1	QC effectué : bonne donnée
2	QC effectué : probablement bonne donnée
3	QC effectué : probablement mauvaise donnée
4	QC effectué : mauvaise donnée
9	Donnée manquante

6 Definitions

- **test** ou **test logique** : une comparaison logique entre une valeur calculée et une valeur seuil
- **filtre** : le resultat d'un test sous forme d'une liste de booléen

7 Processus de contrôle qualité

7.1 Validation manuelle

Du à la difficulté d'identifier programmatiquent les épisodes de fouling sur les capteurs, il est nécessaire de faire cette partie validation à la main. Cette validatio manuelle peut se faire à l'aide du script `manual_validator.py`.

7.2 Tests Logiques

Ensemble des tests logiques appliquées sur les données pour effectuer le contrôle qualité. Ils sont appliqués automatiquement le script [SCRIPT X].

*Tout les valeurs seuils définie ensuite sont définie pour la bouée POEM en prenant en compte un échantillonnage toutes les **5 minutes**. Pour une periode d'échantillonage différente les valeurs seuils du test d'accroissement et de test de pic sont suspectible de changer.*

7.2.1 Test des valeurs nulles

Test qui identifie les valeurs nulles.

7.2.2 Test des valeurs impossibles

Test qui rejette des données au delà de bornes minimals et maximals.

$$x_{min} < x_t < x_{max}$$

Avec :

- x_{min} : la valeur minimal attendu pour le paramètre x
- x_t : la valeur à l'instant t du paramètre x
- x_{max} : la valeur maximal attendu pour le paramètre x

Les valeurs minimals et maximals pour chaque paramètres sont spécifiées dans la table 2.

Table 2: Valeurs maximals et minimals attendues pour les paramètres marins de POEM.

Paramètre	Valeur minimal	Valeur maximal
Température eau (°C)	10	30
Salinité (psu)	20	45
Fluorescence (rfu)	0	15
Turbidité (ntu)	0	40
Oxygène (mg/L)	0	10

7.2.3 Test d'accroissement

Test qui rejete des données au délà d'un seuil maximal d'accroissement. Test proposé par Gronell and Wijffels (2008), Takatsuki et al. (2020) et Wong, Carval, and ADM Team (2012).

$$\frac{|x_{t+\Delta t} - x_{t-\Delta t}|}{2} < sl_0$$

Avec :

- sl_0 : la valeur seuil d'acroissement au dela de laquelle on rejette x
- $x(t)$: la valeur du paramètre x à un instant t
- Δt : l'interval de temps entre deux valeurs successives (la periode d'acquisition du paramètre x)

Les valeurs seuils pour l'accroissement sont spécifiée dans la table 3.

Table 3: Valeurs de seuils pour le test d'accroissement.

Paramètre	seuil accroissement
Température eau	1
Salinité	3
Fluorescence	4
Turbidité	6
Oxygène	0.3

7.2.4 Test de pic

Test qui identifie les pics au d'un seuil maximal. Test proposé par Takatsuki et al. (2020)] et Wong, Carval, and ADM Team (2012).

$$||x_t - \frac{x_{t+\Delta t} + x_{t-\Delta t}}{2}| - \frac{|x_{t+\Delta t} - x_{t-\Delta t}|}{2}| < sp_0$$

Avec :

- sp_0 : la valeur de seuil de pic d'un paramètres x
- $x(t)$: la valeur du paramètre x à un instant t
- Δt : l'interval de temps entre deux valeurs successfives (la periode d'aquisition du paramètre x)

Les valeurs seuils pour un pic sont spécifiée dans la table 4.

Table 4: Valeurs de seuils pour le test de pic.

Paramètre	seuil pic
Température eau	1
Salinité	3
Fluorescence	4
Turbidité	6
Oxygène	0.3

7.2.5 Test adaptatif

Test qui permet de conserver des valeurs qui ne dépasse pas 4 fois l'écart-type glissant de la composante haute fréquence et qui ne dépasserait pas 3 fois l'écart-type simple de la composante haute fréquence. Test déterminée de manière empirique.

Soit un vecteur x de valeurs de longueur $N+1$ d'un paramètre. Soit $\bar{x}_n$ la fonction moyenne glissante centrée qui renvoit la moyenne glissante centrée d'un vecteur x pour tous $n[0, N]$ sur une fenêtre w paire définie comme :

$$\bar{x}_n(x) = \frac{1}{w} \sum_{k=\frac{1}{2}(-w+1)}^{\frac{1}{2}(w-1)} x_{n+k}$$

Avec :

- w : tel que $n + \frac{1}{2}(-w+1) \geq 0$ et $n + \frac{1}{2}(w-1) \leq N$

Soit σ la fonction écart-type qui renvoit l'écart-type d'un vecteur x définie comme :

$$\sigma(x) = \sqrt{\frac{1}{N} \sum_{k=0}^{N} (x_k - \bar{x})^2}$$

Soit σ_n la fonction écart-type glissant qui renvoit l'écart-type glissant d'un vecteur x pour tous $n[0, N]$ sur une fenêtre w paire définie comme :

$$\sigma_n(x) = \sqrt{\frac{1}{w} \sum_{k=\frac{1}{2}(-w+1)}^{\frac{1}{2}(w-1)} (x_{n+k} - \bar{x})^2}$$

Avec :

- w : tel que $n + \frac{1}{2}(-w + 1) \geq 0$ et $n + \frac{1}{2}(w - 1) \leq N$

On peut alors définir le test adaptatif proposé comme suit :

$$(|x_n - \bar{x}_n(x)| < 4\sigma_n(x_n - \bar{x}_n)) \wedge (|x_n - \bar{x}_n(x)| < 4\sigma(x_n - \bar{x}_n))$$

7.3 Validateur - Combinaison des tests logiques

Le validateur est le processus qui combine les différents tests pour attribuer les codes qualités. La combinaison des différents tests se fait avec des opérateurs logiques.

Soit :

- `f_null` : le résultat du test de valeur manquante, un vecteur qui contient `1` si la valeur est manquante et `0` sinon.
- `f_static` : le résultat du test de valeur impossible, un vecteur qui contient `1` si la valeur est inclue entre les bornes définies et `0` sinon.
- `f_sl` : le résultat du test d'accroissement, un vecteur qui contient `1` si la valeur de pente est inférieure au seuil et `0` sinon.
- `f_sp` : le résultat du test de pic, un vecteur qui contient `1` si la valeur de pic est inférieure au seuil et `0` sinon.
- `f_sk` : le résultat du test adaptatif, un vecteur qui contient `1` si la valeur est conservée par le test et `0` sinon.
- `f_manual` : le résultat de validation données manuels
- `~` : l'opérateur logique `NON`

Le validateur calcul ensuite un score pour chaque valeur qui correspond au nombre de filtres passés à la suite, sans rejet, depuis le premier filtre. Les filtres sont placés dans l'ordre suivant : `~f_null`,`f_static`,`f_sl`,`f_sp`,`f_sk`,`f_manual`. Ainsi on obtient pour chaque valeur un score entre 0 et 6. Le score 0 signifie que le premier test n'est pas passé, donc que la valeur est manquante. Le score 6 signifique que tout les tests sont passés, donc la valeur est bonne.

On peut ainsi convertir le score en code qualité en suivant la table 7.3.

Table : Table de conversion entre le score du validateur et un code qualité.

Test	Score	QC	Commentaire
	0	9	Test valeur manquante échoué : Code 9 - valeur manquante
`~f_null`	1	4	Test valeur manquante passé (et les précédents mais pas les suivants) : Code 4 - donnée mauvaise
`f_static`	2	4	Test valeur impossible passé (—) : Code 4 - donnée mauvaise
`f_sl`	3	4	Test accroissement passé (—) : Code 4 - donnée mauvaise
`f_sp`	4	3	Test pic passé (—) : Code 3 - donnée probablement mauvaise
`f_sk`	5	3	Test adaptatif passé (—) : Code 3 - donnée probablement mauvaise
f_manual	6	1	Validation manuelle passée (—) : Code 1 - donnée bonne

Bibliographie

Gronell, Ann, and Susan E. Wijffels. 2008. "A Semiautomated Approach for Quality Controlling Large Historical Ocean Temperature Archives." *Journal of Atmospheric and Oceanic Technology* 25 (6): 990–1003. https://doi.org/10.1175/JTECHO539.1.

Takatsuki, Yasushi, Yasuko Ichikawa, Taiyo Kobayashi, Keisuke Mizuno, and Kensuke Takeuchi. 2020. "Construction of the Automated Data Processing and Delayed-Mode Quality Control

System for Profiling Floats," January, 13. https://www.researchgate.net/publication/260403153_Construction_of_the_Automated_Data_Processing_and_Delayed-Mode_Quality_Control_System_for_Profiling_Floats.

Wong, Carval, and ADM Team. 2012. *Argo Quality Control Manual, Version 2.7.* Report number 341/2650. Argo Data Management Team. http://www.argodatamgt.org/content/download/341/2650/file/argo-quality-control-manual-V2.7.pdf.