## Document de développement de la version 1.5 de EEGpal

12.06.2025

Nouveau Layout de la fenêtre principale de EEGpal :

A screenshot of a computer

AI-generated content may be incorrect.

Chose à faire :

**1. Suppression de studyparameter dans tous les module**

~~- Il faut supprimer les fonction updateStudyParameter et applyStudyPamater ainsi que supprimer la variable studyparameter dans tout les module.   
Cela implique de supprimer définitivement les Open/Save Parameters pour chaque module.~~

- Tester un processing complet pour vérifier que cette modification ne casse rien et ne vas pas modifier les codes de MichaelDP

~~- Modifier le texte du bouton~~ *~~Retain~~* ~~par le texte~~ *~~Save in memory~~*~~. C’est plus intuitif pour l’utilisateur même si c’est un sauvegarde uniquement dans la mémoire. Dans la fenêtre principale de EEGpal, les boutons qui sauvegarder les paramètre sont forme de fichier mat seront~~ *~~Save As~~*~~. Comme ça c’est plus clair.~~

- Pour tout les module, faire appliquer tous les test d’exécutions du boutons Run aussi au bouton save. Le but est d’éviter que l’utilisateur programme des arguments faux ce qui va bloquer s’il exécute un batch. Du coup, on ne pourra plus cliquer sur save si les paramètres sont incorrect.

~~- Les session parameter ne marche pas avec les option Clean raw data. Il faudra regarder lorsque j’aurais plus le tems : redeveloppé un app au lieu du script de eeglab~~

**2. Module Bridge detection**

~~Il faut ajouter un SessionParameter au module Bridge detection pour que sont resultat soit accessible aux autres modules~~

**3. Module ICA**~~- il faut ajouter la variable sessionParameter pour sauver l’état de la fenêtre. Bien include le step1 done car on en aura besoin pour le batch.~~

~~- ajouter un switch au-dessus de la table des électrodes à ignorer qui permet de remplir automatiquement la table avec les électrodes bridge détecter par ebridge (call session parameter). Trouver une solution pour les noms des fichier input qui seront pas les même. Voici des propositions de solutions :~~

* ~~On test le nombre de fichier input de l’ICA et le nombre le ligne de la table des résultats ebridge. Si c’est identique, activé le switch et si différent le désactivé~~
* ~~On pourrait rajouter un test de plus qui est de voir si on retrouve le nom du fichier dans extension entre la table ebridge et la table en input. Mais le désavantage c’est qu’il faut pas que l’utilisateur manipule des noms après la conversion (ce qui devrait être la majorité des cas).~~

~~- Supprimer l’étape de sauver les fichier recomposer en .set/.fdt et les sauver directement en format output. Supprimer les fichier intermédiaire recomposition qui ne servent à rien.~~

~~- remplacer le bouton~~ *~~Save results~~* ~~par Retain pour sauvegarder les session parameter de l’ICA et la bouton start recomposition fera ce que le bouton save result faisait. Peut-être les mettre côte à côte~~

- ~~pour la step 1 decomposition, supprimer la box qui demande d’entrer le fichier de coordonné .locs ; Deux solutions :~~

* ~~Soit on écrit un script pour convertir n’importe quel des coordonné en fichier locs. L’aventage c’est que c’est universel. Le désavantage c’est que le display des composante risque d’être moche en fonction des coordonné~~

~~- modifier pour que la conversion en .set à lieu avant la décomposition au lieu de la faire par la starup function. Comme ça, ça évitera que le module ICA soit freeze à l’ouverture ce qui est stressant pour l’utilisateur. C’est particulièrement important si l’utilisateur devra ouvrir une première fois le module pour sauver ses session parameters~~

~~- A voir plus tard dans un second temps : Implémenter le choix automatique des mauvaises composantes en à retirer. A la place du texte sous l’inspection table, mettre un cadre avec comme titre : automatic component rejection suggestion, avec un bouton~~ *~~parameter~~* ~~et un bouton~~ *~~Compute~~*~~. Pour parameter, faire une fenêtre avec le % minimal de rejection pour les 5 category de bruit (Eye, muscle,heat, other, etc…). Une valeur de 100% indiquera que ce élément seront ignorer (comme other par exemple). Je pense qu’on peux rester sur les 24 première composante comme c’est le cas actuellement.~~

- ~~peut-être rajouter une option au-dessus de l’inspection table entre~~ *~~show me only the 24 first component~~* ~~(default),~~ *~~show me all the component~~*~~. Comme ça les puriste EEG ne m’attaquerons pas sur ce point.~~

**4. Interpolation**

- ~~Supprimer les boutons Run eBridge et Load Bridge Detection file. Les remplacer par un switch bridge channel (eBridge) qui va loader le sessionParameter de ebride. Ajoute automatiquement les channel qui sont bridge (comme le fait le code clearRawData de MichaelDP)~~

**5. Mise en place du nouveau module de Batch**

- Pour le layout graphique, voici ce que j’ai en tête :   
A screenshot of a computer screen

AI-generated content may be incorrect.

a) Liste qui est update par StartFunc en fonction de ce qui est enregistrer dans Processing/Analyse Parameter file. Seul eBridge est dispo par défaut

b) Alors on met 9 lignes car il y a 9 modules dans EEGpal (l’utilisateur ne pourra pas batcher plus de process. A louverture, seul la ligne 1 est disponible. Je met un menu déroulant qui est adapter avec la même liste que en a).   
L’utilisateur choisi un process et cela update/enable l’option sur la même ligne c).   
Lorsque l’utilisateur choisi son option, alors la ligne suivante deviens enable et ainsi de suite. On va permettre de sélectionné une seul fois chaque module. L’utilisateur devra faire un autre fichier s’il désire exécuté plusieurs fois la même step (car on fait généralement ça avec une autre sélection de fichier)

Comment géré si un utilisateur veux modifier une option précédente ? S’il modifie une option, alors ça ne change rien. S’il modifie un process alors il faut vider toutes les lignes qui sont en dessous, les désactiver et réactiver uniquement la ligne suivante.

d) comme tout les module, on a le run pour exécuter, Save pour sauvegarder les SessionParameter ou annuler

Fonctionnement :

Le module Batch va jouer l’intermédiaire entre EEGpalCS\_Main et le module.

1. Je commence par faire une copy de la variable EEGpalCS\_Main dans la start fonction du module batch :  
   app.EEGpalCS\_Main = EEGpalCS\_Main;
2. Je crais un classe spécifique pour gérer les paramètres de batch

BatchExecution = class..  
app.EEGpalCS\_Main.BatchExecution = BatchExecution

Cette class BatchExecution contiendra la variable exécution = 1 si on est en train d’exécutez un batch et = 0 si c’est pas le cas

1. - Dans chaque module, modification de la statupfunction. A la fin (après intiviatisation), mettre un test if app.EEGpalCS\_Main.BatchExecution ==1, alors appeler directement la fonction RunCallback.   
   - Dans la fonction RunCallback, il faudra conditionné l’apparition de la fenêtre de récapitulation au fait que app.EEGpalCS\_Main.BatchExecution ==0

- A la fin de la fonction Runcallback, il faudra conditionné la pop-up pour signaler la fin du processing

1. La table de sélection des fichier sera enregistrer dans la copie du app.EEGpalCS\_Main du module batch (normalement à vérifier). Si on veux updater la main windows, il faudra transmettre à la main windows dans un second temps
2. A la fin de l’execution du batch ou en cas d’echec (try catch), il faut mettre app.EEGpalCS\_Main.BatchExecution =0 ;
3. Il faudra aussi ajouter un SessionParameter au module de batch pour qu’il soit sauver dans le fichier processing/analyse file
4. Une fois l’étape exécuter, passer le ProssingDropDown en Enable = off (pour qu’il soit toujours visible mais plus modifiable
5. La stratégie la plus simple sera que la fonction qui exécute le batch soit dans le module de la fenêtre princiaple (EEGpal\_main) et que le module de Batch ne soit que une interface graphique pour progammer les succission qui sont enregistrer dans Session Parameter.

Comportement spécifique/ option pour chaque module :

**ICA Module :**

Mettre comme option :

* *Manual selection of bad compound*: à ce moment-là, l’utilisateur devra manuellement visionner toutes les composante comme normal et presser sur recompose pour continuer l’exécution du batch
* *Automatique rejection*: executé le compute automatic rejection et fait la reconstruction automatiquement

Problème résoudre : La table des bad channel enregistré dans sessionParameter va être ignorer car les noms des fichiers input ne seront pas les même que les noms enregistré dans SessionParameters.

**Interpolation module :**

Mettre comme option :

* *Manual selection*: à ce moment-là, l’utilisateur devra manuellement remplir le tableau d’interpolation
* *Automatique interpolation*: Disponible uniquement s’il a activer clearRawDatedans le Filtering. A ce moment, ça active automatiquement l’onglet et ça run l’interpolation

**Frequencying Module :**

L’option Frequenciny doit être dispo uniquement après un processing de Epoching + il doit vérifier dans les SessionParameter.Epoching que l’option concatenate all epochs in one file avait été sélectionné. Si tel est le cas, mettre comme seul option « based on epochs in one file » ce qui permettra d’informer l’utilisateur de cette façon de travailler.   
Dans les autres cas, le batch ne va pas les gérer et ça devra être faire manuellement par l’utilisateur en passant par un sélection de fichier dans la main windows.