

THÈSE DE L'UNIVERSITÉ DE LYON

Délivrée par

UNIVERSITÉ CLAUDE BERNARD LYON 1

DIPLÔME DE DOCTORAT

Ma spécialité

DÉCODAGE DES INTENTIONS ET DES REPRÉSENTATIONS MOTRICES CHEZ L'HOMME : ANALYSE MULTI-ÉCHELLE ET APPLICATION AUX INTERFACES CERVEAU-MACHINE

par

Etienne Combrisson

Thèse soutenue le 09/2016 devant le jury composé de :

M ^{me}	ERIKA RATÉ	Université à la Menthe	(Rapporteur)
M.	JACQUES OUILLE	Université à la Fraise	(Rapporteur)
M.	HENRI ZOTO	Laboratoire laborieux	(Rapporteur)
M.	JEAN FILE	Indienne	(Directeur)
	etc.		

*À Isabelle et Didier, mes deux parents,
qui ont tout donné pour que ceci me soit un jour possible.
Merci*

REMERCIEMENTS

Je voudrais tout d'abord exprimer mes plus profonds remerciements à...
AHÂÂÂH!

Je conclurai en remerciant de tout cœur (l'être aimé).

Montréal, le 8 juillet 2016.

TABLE DES MATIÈRES

TABLE DES MATIÈRES	vi
LISTE DES FIGURES	ix
NOTATIONS	1
I Introduction générale	3
1 PRÉSENTATION DE LA THÉMATIQUE	7
1.1 ENREGISTREMENT DE L'ACTIVITÉ NEURONALE	7
1.1.1 Enregistrement non-invasif	7
1.1.2 Enregistrement invasif	7
1.2 ÉTAT DE L'ART DES INTERFACES CERVEAU-MACHINE	7
1.2.1 ICM non-invasives	8
1.2.2 ICM invasives	8
1.3 APPRENTISSAGE MACHINE : APPLICATIONS AUX NEUROSCIENCES	8
1.4 ENCODAGE ET DÉCODAGE MOTEUR : BASES PHYSIOLOGIQUES .	8
1.5 INTENTION ET EXÉCUTION	9
1.6 DELAYED TASK : PROTOCOLE EXPÉRIMENTAL	9
2 OBJECTIFS DE LA THÈSE	11
2.1 DÉCODAGE CÉRÉBRALE À PARTIR D'ACTIVITÉ INTRACRÂNIENNE	11
2.2 EXPLORATION ET AMÉLIORATION DES FEATURES	11
2.3 COMPARATIF DES CLASSIFIEURS	11
2.4 EXPLORATION DES RÉGIONS NON-MOTRICES	12
3 MÉTHODOLOGIE	13
3.1 EXTRACTION DES FEATURES	13
3.1.1 Pré-requis	13
3.1.2 Puissance spectrale	15
3.1.3 Phase	17
3.1.4 Phase-amplitude coupling	17
3.2 APPRENTISSAGE SUPERVISÉ	22
3.2.1 Labellisation et apprentissage	23
3.2.2 <i>Training, testing</i> et validation-croisée	23
3.2.3 Classificateurs	25
3.2.4 Évaluation de la performance de décodage	27
3.2.5 Seuil de chance et évaluation statistique de la performance de décodage	27
3.2.6 Du single au multi-features	28

3.2.7	Généralisation temporelle	31
3.3	CONFIGURATION POUR DÉBUTER	33
3.4	IMPLEMENTATION DES MÉTHODES PRÉSENTÉES	33
4	DONNÉES EXPÉRIMENTALES	37
4.1	DONNÉES INTRACRÂNIENNES	37
4.1.1	Acquisition	37
4.1.2	Avantages et limitations des données intracrâniennes	37
4.1.3	Prétraitements	38
4.2	DONNÉES D'ÉTUDE	39
4.2.1	Données <i>Center-out</i>	39
4.2.2	Autres données	41
5	OUVERTURE	43
II	Étude 1 : niveau de chance et évaluation statistique des résultats de classification par apprentissage supervisé	45
5.1	PRÉSENTATION DE L'ÉTUDE	49
5.1.1	Contexte	49
5.1.2	Problématique	49
5.1.3	Résultats majeurs	49
5.2	ARTICLE	49
5.3	COMPLÉMENTS D'ÉTUDE	49
III	Étude 2 : encodage de l'intention et de l'exécution motrice	51
5.4	RÉSUMÉ DE L'ÉTUDE	55
5.5	ARTICLE	55
CONCLUSION	55	
IV	Étude 3 : décodage des directions de mouvement pendant et avant l'exécution de mouvement de membres supérieurs	57
5.6	RÉSUMÉ DE L'ÉTUDE	61
5.7	ARTICLE	61
CONCLUSION	61	
V	Étude 4 : optimisation des paramètres de la bande gamma	63
5.8	RÉSUMÉ DE L'ÉTUDE	67
5.9	ARTICLE	67
CONCLUSION	67	
VI	Étude 5 : décodage des émotions	69
5.10	RÉSUMÉ DE L'ÉTUDE	73
5.11	ARTICLE	73
CONCLUSION	73	

CONCLUSION GÉNÉRALE	75
A ANNEXES	76
A.1 COMPARATIF DE METHODES PAC (TORT ET AL., 2010)	77
A.2 PIPELINE STANDARD DE CLASSIFICATION	78
A.3 COMPARATIF DE CLASSIFIEURS (PEDREGOSA ET AL., 2011)	80
A.4 EXEMPLE DE SCHÉMA D'IMPLANTATION	82
BIBLIOGRAPHIE	83

LISTE DES FIGURES

1.1	Techniques d'enregistrement de l'activité cérébrale	7
1.2	Pipeline général d'un Interface Cerveau-Machine	8
1.3	Contrôle d'un bras robotisé	8
1.4	Comparatif	9
2.1	Mécanismes du couplage phase-amplitude	11
2.2	Localisation des aires sensorimotrices	12
3.1	"One-tailed" et "two-tailed" test	15
3.2	Exemple de représentation temps-fréquence de puissance normalisées z-score (Ossandon et al., 2011)	17
3.3	Densité de probabilité d'une distribution d'amplitudes en fonction de tranches de phases	18
3.4	(A) Exemple de cartes temps-fréquence phase locked sur le β , (B) Exemple de comodulogramme	21
3.5	Labellisation de données	23
3.6	Exemple d'une cross validation 3-folds	25
3.7	Principe du Linear Discriminant Analysis (Lotte et al., 2007)	25
3.8	Principe du Support Vector Machine (Lotte et al., 2007) . .	26
3.9	Principe du k-Nearest Neighbor (Weinberger et al., 2005) .	26
3.10	Entraînement puis test d'un classifieur linéaire	26
3.11	Calcul de l'acuité de décodage	27
3.12	Exemple d'une <i>Forward feature selection</i> appliquée sur six features	30
3.13	Exemple d'une <i>Backward feature elimination</i> appliquée sur six features	31
3.14	Exemple de décodage temporel (Waldert et al., 2008). Ici, l'auteur décode 4-directions de mouvements de la main dans le temps. A chaque instant, un classifieur est créé, entraîné puis testé à ce même instant.	32
3.15	Exemple de généralisation temporelle (King and Dehaene, 2014)	33
4.1	Détails cliniques des sujets ayant participé à la tâche <i>Center-out</i>	39
4.2	Implantation intracrâniale et couverture corticale de six sujets épileptiques ayant passés la tâche <i>Center-out</i>	40
4.3	Descriptif de la tâche <i>Center-out</i>	41
A.1	Comparatif de méthodes PAC (Tort et al., 2010)	77
A.2	Pipeline standard de classification	78

A.3 Comparatif de classifieurs (Pedregosa et al., 2011)	80
A.4 Exemple de schéma d'implantation	82

NOTATIONS

Général

ICM Interface Cerveau-Machine
BCI Brain Computer Interface

Enregistrements

EEG Électroencéphalographie
MEG Magnétoencéphalographie
SUA Single Unit Activity
MUA Multi Unit Activity
SEEG Stéréoélectroencéphalographie
ECoG Électrocorticographie

Features

PAC Phase Amplitude Coupling

Classifeurs

LDA Linear Discriminant Analysis
SVM Support Vector Machine
RF Random Forest
KNN k-Nearest Neighbor
NB Naive Bayes

Première partie

Introduction générale

B_LABLABLABKIBLABLOU INTRO ...

L'objectif de cette thèse a été de ...

Totalité des méthodes explorées durant ma thèse sont présentes dans une toolbox python appelée brainpipe, libre d'accès et de droit.

PRÉSENTATION DE LA THÉMATIQUE

1.1 ENREGISTREMENT DE L'ACTIVITÉ NEURONALE

- Présentation de chacun des types de données
 - Résolution et RSB
 - Avantages // inconvénients
- (Waldert et al., 2009)

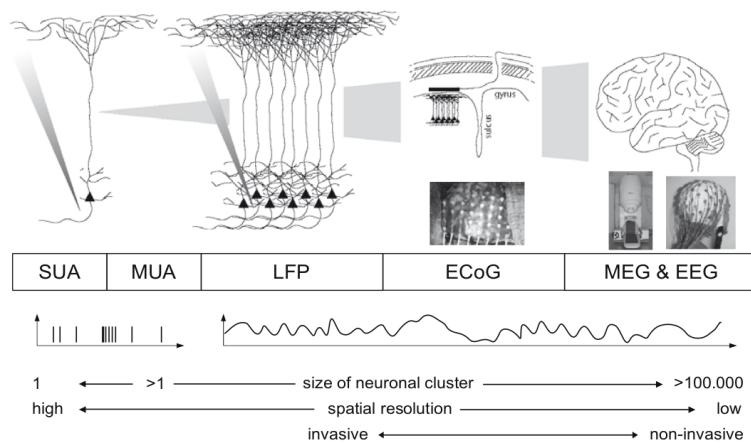


FIGURE 1.1 – Techniques d'enregistrement de l'activité cérébrale

1.1.1 Enregistrement non-invasif

Électroencéphalographie

Magnétoencéphalographie

1.1.2 Enregistrement invasif

Single Unit Activity

Multi Unit Activity

Stéréoélectroencéphalographie

Électrocorticographie

1.2 ÉTAT DE L'ART DES INTERFACES CERVEAU-MACHINE

(Bekaert et al., 2009)

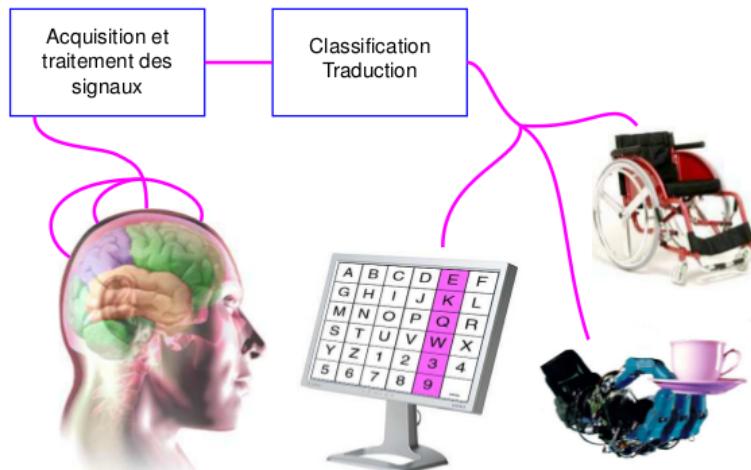


FIGURE 1.2 – Pipeline général d'un Interface Cerveau-Machine

1.2.1 ICM non-invasives

P300 speller

1.2.2 ICM invasives

(Hochberg et al., 2012)

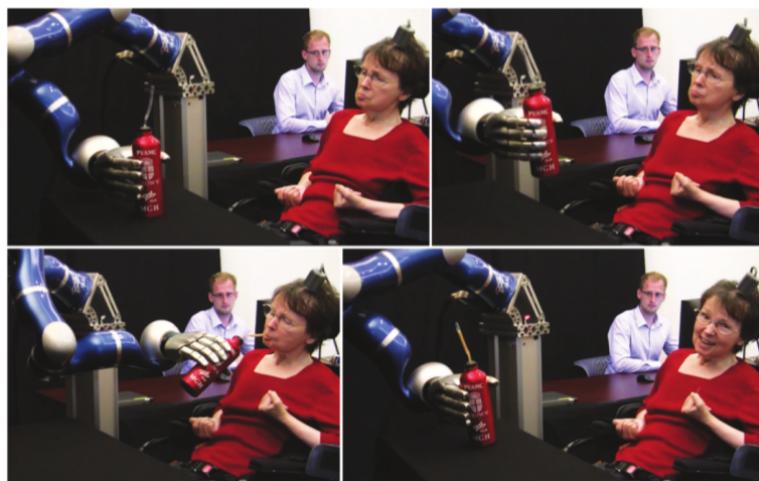


FIGURE 1.3 – Contrôle d'un bras robotisé

1.3 APPRENTISSAGE MACHINE : APPLICATIONS AUX NEUROSCIENCES

okok

1.4 ENCODAGE ET DÉCODAGE MOTEUR : BASES PHYSIOLOGIQUES

La figure 1.4 représente une trajectoire d'un processus markovien de saut, avec les notations associées. (Hanakawa et al., 2008)

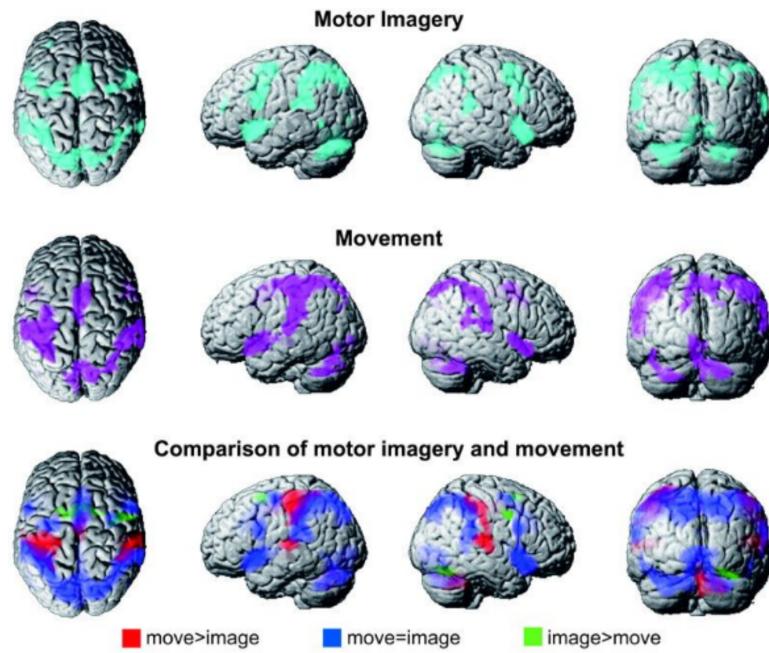


FIGURE 1.4 – Comparatif

1.5 INTENTION ET EXÉCUTION

okok

1.6 DELAYED TASK : PROTOCOLE EXPÉRIMENTAL

okok

OBJECTIFS DE LA THÈSE

2

2.1 DÉCODAGE CÉRÉBRALE À PARTIR D'ACTIVITÉ INTRACRÂ-NIENNE

- Exemple d'un schéma d'implantation + IRM
- Bipolarisation : débruitage et augmentation de la spécificité (article Karim)
- Extraction de features (ici on pourrait mentionner que le deep learning pourrait marcher sur les données brutes)

2.2 EXPLORATION ET AMÉLIORATION DES FEATURES

Rôle physiologique du phase-amplitude coupling

(Hyafil et al., 2015)

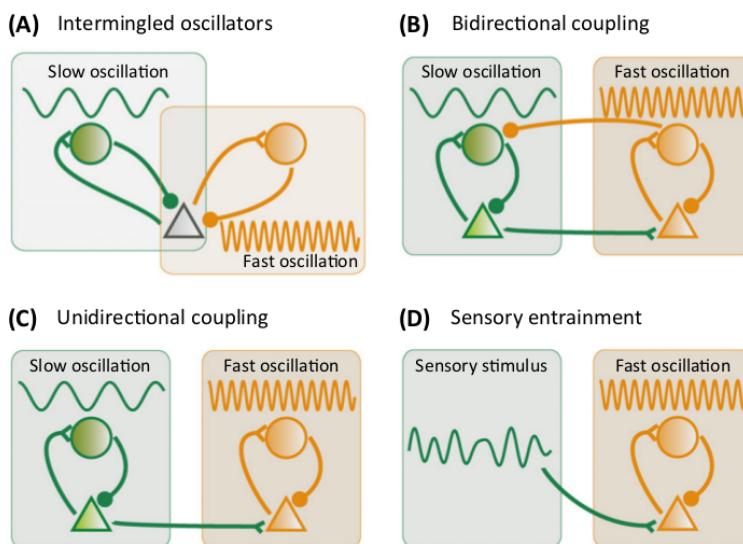


FIGURE 2.1 – Mécanismes du couplage phase-amplitude

2.3 COMPARATIF DES CLASSIFIERS

Expliquer que, chaque classifier possède une méthodologie propre permettant de répondre à des types de données différentes (en fonction des hypothèses de fonctionnement de chacun des classifiers)

2.4 EXPLORATION DES RÉGIONS NON-MOTRICES

(Van Langenhove et al., 2008)

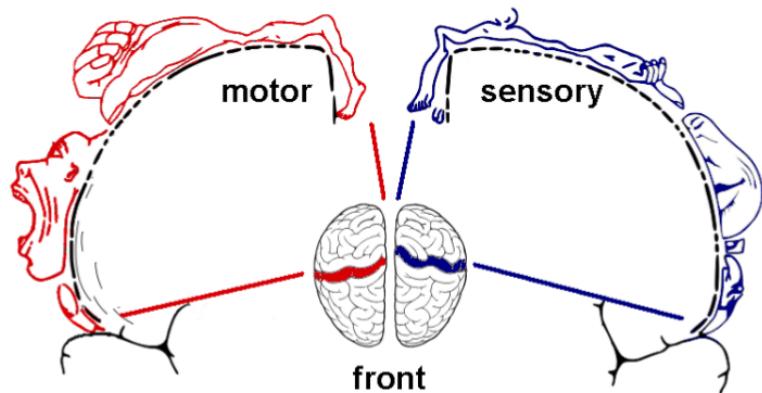


FIGURE 2.2 – Localisation des aires sensorimotrices

MÉTHODOLOGIE

3

Cette partie méthodologique sera divisée en deux grandes sous parties visant à présenter :

1. L'extraction des features : présentation des méthodes utilisées dans le cadre de l'extraction d'attributs issus de l'activité neuronale. De manière générale, nous avons étudiés des attributs spectraux comprenant :
 - Phase et puissance spectrale
 - Attributs de couplage
2. Le machine learning : présentation des principaux algorithmes testées dans le cadre du décodage de l'activité neuronale

3.1 EXTRACTION DES FEATURES

Comme nous l'avons décrit précédemment, l'objectif du décodage de l'activité neuronale est d'arriver à extraire des signaux cérébraux une information suffisamment pertinente pour pouvoir discriminer différents types de classes (exemple : mouvement vers la gauche Vs droite).

Tout les attributs testés dans le cadre de cette thèse sont des attributs spectraux, donc issus de bandes de fréquences. La plupart de ces outils partagent donc une partie méthodologique commune à savoir, le filtrage. De plus, la plupart sont extraits en utilisant la transformée d'Hilbert. Pour éviter une redondance à travers les attributs, nous allons tout d'abord introduire quelques pré-requis.

3.1.1 Pré-requis

Filtrage

L'intégralité des filtrages dans cette thèse ont été effectués avec la fonction *eegfilt* (qui a ensuite été reproduite pour le passage à python). De plus, afin d'éviter tout phénomène de déphasage, la fonction *filtfilt* a été systématiquement utilisée afin que le filtre soit appliqué dans les deux sens. Si cette dernière fonctionnalité n'est pas forcément indispensable dans le cadre d'un calcul de puissance, elle est absolument nécessaire pour un calcul de couplage phase-amplitude .

L'ordre du filtre présenté au dessus dépend de la fréquence de filtrage. Il a systématiquement été calculé en utilisant la méthode décrite par Bahramisharif et al. (2013) :

$$FiltOrder = N_{cycle} \times f_s / f_{oi} \quad (3.1)$$

où f_s est la fréquence d'échantillonnage, f_{oi} est la fréquence d'intérêt et N_{cycle} est un nombre de cycles définit par $N_{cycle} = 3$ pour les oscillations lentes et $N_{cycle} = 6$ pour les oscillations rapides.

Transformée d'Hilbert

Transformée permettant de passer un signal temporel $x(t)$ du domaine réel au domaine complexe. Le signal peut ensuite s'écrire $x_H(t) = a(t)e^{j\phi(t)}$ où $a(t)$ est l'amplitude et $\phi(t)$, la phase. Cette transformation est particulièrement exploitée car le module de $x_H(t)$ permet de récupérer l'amplitude et la phase est obtenue en prenant l'angle de $x_H(t)$.

Transformée en ondelettes

La transformée en ondelettes (Tallon-Baudry et al., 1997, Worrell et al., 2012) permet de décomposer un signal dans le domaine temps-fréquence. La décomposition en ondelettes d'une fonction f est définie par :

$$f(a, b) = \int_{-\infty}^{\infty} f(x) \bar{\psi}_{a,b} dx \quad (3.2)$$

Où ψ est appelé ondelette mère dont la définition générale est donnée par $\psi_{a,b} = \frac{1}{\sqrt{a}} \Psi(\frac{x-b}{a})$ où a est le facteur de dilatation et b le facteur de translation. Le choix de l'ondelette mère s'est porté sur l'ondelette de Morlet qui est très largement utilisée à travers la littérature et définie par :

$$w(t, f_0) = A e^{-t^2/2\sigma_t^2} e^{2i\pi f_0 t} \quad (3.3)$$

Où $\sigma_f = 1/2\pi\sigma_t$ et $A = (\sigma_t\sqrt{\pi})^{-1/2}$. L'ondelette de Morlet est caractérisée par le ratio constant $r = f_0/\sigma_f$ que nous avons fixé égale à 7 comme suggéré par Tallon-Baudry et al. (1997).

Cette décomposition peut être comparée à la transformée courte de Fourier qui décompose le signal en une somme de combinaisons linéaire de sinus et de cosinus mais part du principe qu'il existe une régularité dans le signal permettant une telle décomposition. La transformée en ondelettes résout plusieurs limitations :

- Elle permet d'obtenir l'énergie d'un signal dans le temps, ce qui permet une bien meilleure exploration des phénomènes.
- Le rapport constant r permet d'obtenir des ondelettes dont la résolution fréquentielle varie en fonction des fréquences et permet une meilleure coïncidence avec la définition des bandes physiologiques (Bertrand et al., 1994)

Tout les attributs qui vont être maintenant présentés, utilisent les méthodes décrites ci-dessus.

Évaluation statistique à base de permutations

Pour une distribution de permutations construite à partir de deux sous-ensembles A et B et comportant N observations et pour une valeur p pré-définie, on pourra conclure que :

- $A > B$ si A est parmi les $N - N \times p$ derniers échantillons ("One-tailed test upper tail")
- $A < B$ si A est parmi les $N \times p$ premiers échantillons ("One-tailed test lower tail")
- $A \neq B$ si A est soit inférieur aux $(N \times p)/2$ premiers échantillons soit supérieur aux $(N - N \times p)/2$

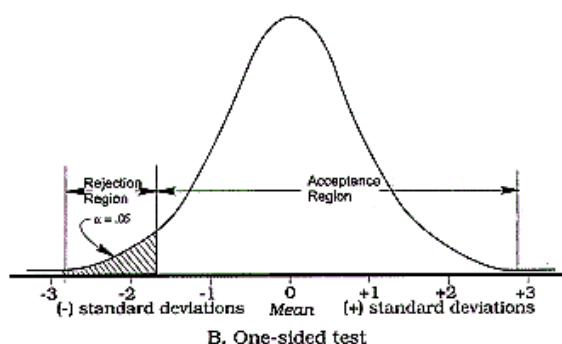
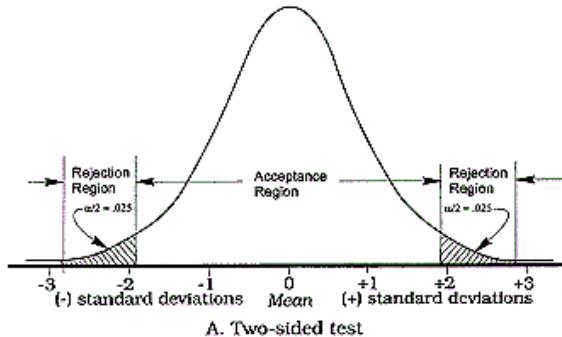


FIGURE 3.1 – "One-tailed" et "two-tailed" test

Grâce à cette méthode d'évaluation statistique, nous pourrons par exemple conclure si l'on a une augmentation, une diminution ou une différence statistique entre une valeur de puissance et la puissance contenue dans une période de baseline. Dernière précision, on comprend ainsi que pour obtenir une valeur p il faut que la taille de la distribution N soit au moins de $1/p$.

Hyperplan

Un hyperplan est un espace de co-dimension 1. Donc, dans un espace $3D$, l'hyperplan est un plan (dimension $2D + 1$). De manière générale, un espace de dimension N possède un hyperplan de dimension $N - 1$

$$\dim_{ESPACE} = \dim_{HYPERPLAN} + 1 \quad (3.4)$$

3.1.2 Puissance spectrale

Méthodes explorées

Le calcul de la puissance spectrale a été approché par deux méthodologies et qui ont été utilisés à des fins différentes :

- La transformée d'Hilbert : souvent exploité dans le cadre du décodage ainsi que pour garder une uniformité entre les attributs de phase et couplage phase-amplitude basés eux aussi sur cette transformée.
- La transformée en ondelettes : principalement utilisée pour la visualisation des cartes temps-fréquence à cause de l'adaptation des ondelettes aux bandes physiologiques.

Normalisation

On utilise la normalisation pour observer l'émergence d'un phénomène par rapport à une période définie comme baseline. A travers la littérature, quatre grands types de normalisation sont rencontrés :

1. Soustraction par la moyenne de la baseline
2. Division par la moyenne de la baseline
3. Soustraction puis division par la moyenne de la baseline
4. Z-score : soustraction de la moyenne puis division par la déviation de la baseline

La normalisation z-score est certainement la plus fréquemment rencontrée à travers la littérature. Le choix du type de normalisation dépend du type de données utilisées. Dans le cadre de nos données, β_3 était clairement la plus adaptée pour la visualisation. En revanche, dans le cadre de la classification, nous obtenions systématiquement de meilleurs résultats sans normalisation.

Évaluation statistique

La fiabilité statistique de la puissance a été évaluée en comparant chaque valeur de puissance à la puissance contenue dans une période définie comme baseline. Pour ce faire, nous avons testé deux approches :

1. Permutations : les valeurs de puissance et de baseline sont aléatoirement mélangées à travers les essais. Puis, on normalise cette puissance. En répétant cet procédure N fois, on obtient une distribution qui peut ensuite être utilisée pour en déduire la valeur p de la véritable puissance (cf : *pré-requis*)
2. "Wilcoxon signed-rank test" : ordonne les distances entre les paires de puissances (vraie valeur, baseline) (Demandt et al., 2012, Rickert, 2005, Waldert et al., 2008)

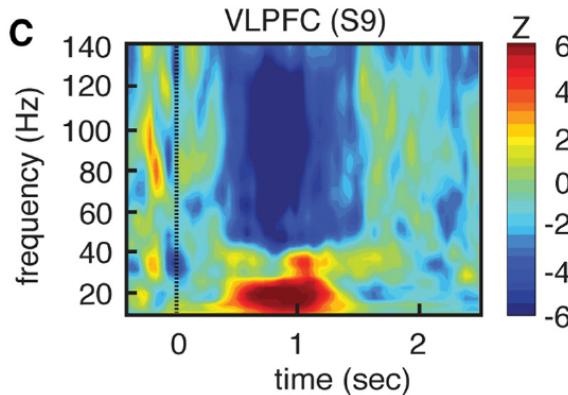


FIGURE 3.2 – Exemple de représentation temps-fréquence de puissance normalisées z-score (Ossandon et al., 2011)

3.1.3 Phase

L'extraction de la phase se fait de la même manière que pour le Couplage phase-amplitude , en prenant l'angle de la transformée d'Hilbert d'un signal filtré. La significativité peut être évaluée en utilisant le test de Rayleigh (Jervis et al., 1983, Tallon-Baudry et al., 1997). Point de vue pratique, cela correspond à la fonction *circ_rtest* de la toolbox Matlab *CircStat* (Berens and others, 2009)

3.1.4 Phase-amplitude coupling

Le calcul du Phase-amplitude coupling ne se limite pas uniquement à la méthode. En réalité, pour obtenir une estimation fiable sur des données réelles, il est indispensable de suivre les trois étapes suivantes :

1. Estimation de la véritable valeur de PAC. Il existe plusieurs méthodes.
2. Calcul de "surrogates" : on va calculer des PAC déstructurés. Idem, il existe de nombreuses méthodes
3. Correction du véritable PAC par les "surrogates". Cette correction, qui est en fait une normalisation, aura pour but de soustraire à l'estimation du PAC de l'information considérée comme bruitée.

Les sous-parties suivantes présenteront de manières succinctes les principales méthodes rencontrées dans la littérature, ainsi que différents types de corrections applicables.

Méthodologie du phase-amplitude coupling

Il existe une large variété de méthodes pour calculer le PAC, ce qui complique son exploration. Toutefois, il n'existe pas de consensus sur une méthode plus polyvalente qu'une autre, chacune possédant ses points forts et limitations. Pour aller un peu plus loin, et présenter quelques méthodes, il est nécessaire d'introduire quelques variables. Soit $x(t)$, une série temporelle de données de taille N. Pour cette série temporelle, on souhaite savoir si la phase extraite dans une bande de fréquence $f_\phi = [f_{\phi_1}, f_{\phi_2}]$ est couplée avec l'amplitude contenue dans $f_A = [f_{A_1}, f_{A_2}]$. Pour cela, on va tout

d'abord extraire $x_\phi(t)$ et $x_A(t)$ les signaux filtrés dans ces deux bandes. Enfin, la phase $\phi(t)$ est obtenue en prenant l'angle de la transformée d'Hilbert de $x_\phi(t)$ tandis que l'amplitude $a(t)$ est obtenue en prenant le module de la transformée d'Hilbert de $x_A(t)$.

1. Mean Vector Length-Modulation Index :

Cette méthode a été introduite par Canolty et al. (2006) et consiste à sommer, à travers le temps, le complexe formé de l'amplitude des hautes fréquences avec la phase des basses fréquences. L'équation est donnée par :

$$MVL = \left| \sum_{j=1}^N a(j) \times e^{j\phi(j)} \right| \quad (3.5)$$

2. Kullback-Leibler divergence :

A l'origine, la divergence de Kullback-Leibler (KLD), qui est issue de la théorie de l'information, permet de mesurer les dissimilarités entre deux distributions de probabilités. Ainsi, pour pouvoir utiliser cette mesure dans le cadre du PAC, Tort et al. (2010) propose une solution élégante qui consiste à générer une distribution de densité probabilités de l'amplitude (DPA) en fonction des valeurs de phase et d'ensuite utiliser le KLD pour comparer cette distribution à la densité de probabilité d'une distribution uniforme (DPU). Plus la DPA s'éloigne de la DPU, plus le couplage entre l'amplitude et la phase est consistant.

Pour construire la DPA, l'astuce consiste à couper le cercle trigonométrique en N tranches (dans l'article il est proposé de couper en 18 tranches de 20°). Puis, si on prend l'exemple de la tranche $[0, 20^\circ]$, on va chercher tout les instants temporels où la phase prend des valeurs comprises entre $[0, 20^\circ]$ ($t, \phi(t) \in [0, 20^\circ]$). On prend ensuite la moyenne de l'amplitude pour ces valeurs de t et on répète cette procédure pour chacune des tranches de phase. On obtient ainsi la densité d'amplitudes en fonction des valeurs de phase. Il ne reste plus qu'à normaliser cette distribution par la somme des amplitudes à travers les tranches et on récupère une distribution de densité de probabilités. La figure 3.3 (Tort et al., 2010) présente un exemple de DPA en fonction de tranches de phase.

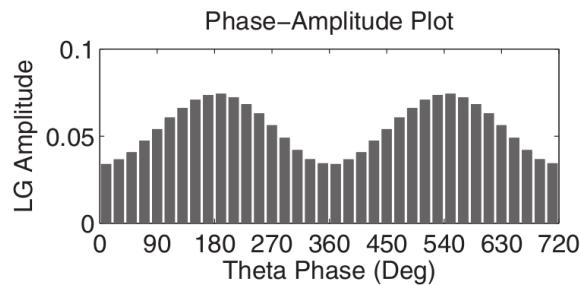


FIGURE 3.3 – Densité de probabilité d'une distribution d'amplitudes en fonction de tranches de phases

Le calcul de la divergence de Kullback-Leibler est ensuite appliqué

pour mesurer les dissimilarités entre la DPA et la DPU et c'est cette mesure qui servira d'estimation du couplage phase-amplitude :

$$D_{KL}(P, Q) = \sum_{j=1}^N P(j) \times \log \frac{P(j)}{Q(j)} \quad (3.6)$$

où $P(j)$ est la densité de probabilité de $a(t)$ en fonction de $\phi(t)$ et $Q(j)$ est la densité de probabilité d'une distribution uniforme.

3. Height Ratio

La méthode du Height Ratio (Lakatos, 2005) est extrêmement proche du Kullback-Leibler divergence . En effet, l'amplitude sera binée de la même façon en fonction des tranches de phase. La mesure du PAC est ensuite donnée par :

$$hr = (f_{max} - f_{min}) / f_{max} \quad (3.7)$$

où f_{max} et f_{min} sont respectivement le maximum et le minimum de la densité de probabilité de l'amplitude en fonction des valeurs de phase.

4. Normalized Direct Phase-Amplitude Coupling

Le Normalized Direct Phase-Amplitude Coupling , qui n'est pas une des méthodes les plus fréquemment rencontrées, présente toutefois une avantage certain. En plus de fournir une estimation fiable du couplage phase-amplitude , Ozkurt (2012) démontre l'existence d'un seuil à partir duquel on peut considérer l'estimation du PAC comme étant statistiquement fiable. La beauté de cette méthode, c'est que ce seuil statistique, qui est une fonction de la valeur p désirée, ne dépend que de la taille de la série temporelle. Ce qui rend son utilisation particulièrement simple.

Pour estimer le PAC, une des hypothèses ayant permis d'aboutir à ce seuil statistique est de devoir normaliser l'amplitude par un z-score dénotée $\tilde{a}(t)$. L'estimation du PAC est quasiment identique au MVL puisque c'est en réalité le carré de celle-ci. Enfin, pour une valeur p désirée, l'article introduit le seuil statistique :

$$x_{lim} = N \times [erf^{-1}(1 - p)]^2 \quad (3.8)$$

où erf^{-1} est la fonction d'erreur inverse. On déduira que l'estimation PAC est significative si et seulement si cette valeur est deux fois supérieur à ce seuil.

5. Autres méthodes : Tout les algorithmes présentés ci-dessus ont été testés, implémentés et comparés. En complément, voici une liste non exhaustive d'autres méthodes existantes :

- *Phase Locking Value (PLV)* (Cohen, 2008, Penny et al., 2008) : détournement du PLV proposé par Lachaux et al. (1999) qui mesure la synchronie de phase entre deux électrodes. Cette méthode va comparer la phase des basses fréquences avec la phase de l'amplitude des hautes-fréquences.
- *Generalized Linear Model (GLM)* (Penny et al., 2008) : outil décrit comme adapté aux données courtes et bruitées.
- *Generalized Morse Wavelets (GMW)* (Nakhnikian et al., 2016) : basée sur des ondelettes, semble particulièrement utile dans le cadre de l'exploration des données.
- *Oscillatory Triggered Coupling (OTC)* (Dvorak and Fenton, 2014, Watrous et al., 2015) : issue d'une détection de maximums des hautes fréquences.

Correction du phase-amplitude coupling et évaluation statistique

Nous avons vu dans la section précédente différentes méthodes permettant de calculer un Couplage phase-amplitude . Toutefois, celui-ci peut être largement amélioré en faisant une estimation du PAC contenu dans le bruit des données. Une fois que cette estimation sera faite, on pourra retrancher ce PAC bruité à la valeur initiale. Tout comme il existe plusieurs méthodes de PAC, les équipes de recherche proposent à tour de rôle de nouvelles méthodes. Parmi elles, on peut citer :

- *Time-lag* : proposée par Canolty et al. (2006), on introduit un délai sur l'amplitude compris entre $[f_s, N - f_s]$ où f_s est la fréquence d'échantillonnage et N est le nombre de points de la série temporelle
- *Shuffling des couples [phase,amplitude]* : ici, on mélange aléatoirement les essais de phase et d'amplitude (Tort et al., 2010)
- *Swapping temporel d'amplitudes (ou de phase)* : on mélange aléatoirement les essais d'amplitude puis on recalcule le PAC avec la phase originale (Bahramisharif et al., 2013, Lachaux et al., 1999, Penny et al., 2008, Yanagisawa et al., 2012)

Ces trois méthodes produisent une distribution de *surrogates*. On pourra ensuite appliquer un z-score à la véritable estimation en utilisant la moyenne et la déviation de cette distribution. Enfin, l'évaluation statistique se fait également à partir de cette distribution (cf : *pré-requis*)

A ma connaissance, il n'existe pas de comparatif entre ces corrections et je n'ai jamais rencontré d'articles mentionnant que l'on ne puisse pas combiner les méthodes de PAC avec les différentes corrections. En revanche, ce qui est relaté c'est que le *time-lag* nécessite des données longues dû à l'introduction de ce délai temporel.

Comparatif des méthodes

Penny et al. (2008) ont comparé plusieurs méthodes dont le *MVL*, *PLV* et le *GLM* et Tort et al. (2010) ont complété cette étude avec d'autres méthodes (cf. A.1). Enfin, Canolty and Knight (2010) a fait une review qui comprend un descriptif très instructif.

Représentation du phase-amplitude coupling

Comparée à la puissance, l'exploration du PAC peut s'avérer plus complexe dû à sa dimensionnalité plus grande. Il existe donc des outils et des méthodes destinées à simplifier cette exploration et à visualiser ces résultats.

Exemple concret, si on cherche à connaître les modulations de puissance contenue dans un signal, on peut représenter une carte temps-fréquence. Pour le PAC, idéalement on voudrait visualiser les phases, les amplitudes et le temps mais ces trois dimensions empêche une représentation simple. On peut donc avoir recours à différents types de représentations complémentaires :

- Puissance phase-locked : cette représentation permet de faire émerger l'existence d'un couplage, pour une phase donnée, et d'observer sa durée. Pour cela, on aligne les phases en détectant le pic le plus proche de l'instant temporel étudié. On calcule les cartes temps-fréquence que l'on va ensuite moyennner après les avoir recalées de la même façon que les phases (c'est-à-dire avec la même latence).
- Comodulogramme : pour une tranche temporelle définie, on représente les valeurs de PAC pour différentes valeurs de phase et d'amplitude

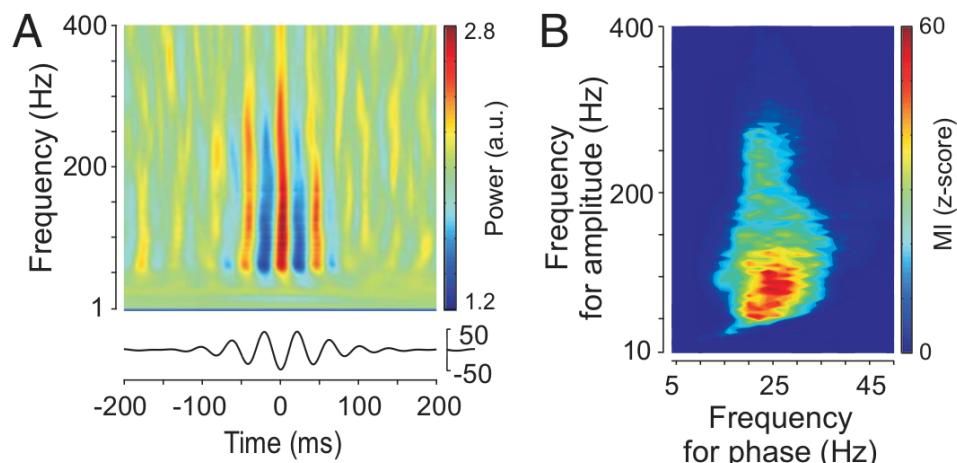


FIGURE 3.4 – (A) Exemple de cartes temps-fréquence phase locked sur le β , (B) Exemple de comodulogramme

La figure 3.4 (de Hemtinne et al., 2013) met en évidence que la représentation des cartes temps-fréquence phase-locked (**A**) est limitée d'une part, par la phase sur laquelle on choisit de recalculer et d'autre part cette méthode est également limitée par l'instant où l'on choisit de recalculer. Pour la figure (**B**), le calcul du PAC se faisant à travers la dimension temporelle, on a aucune idée de l'évolution du couplage dans le temps.

Phase-amplitude coupling : résolution temporel ?

Comment peut-on savoir si un ensemble de musiciens jouent ensemble, en rythme ? L'approche traditionnelle consiste à dire que, en fonction de la prestation du groupe, on sera en mesure de dire si ils étaient en rythme ou

non. Donc on focalise notre attention sur chaque instant du morceau et on analyse chaque note, chaque décalage. Cela signifie aussi que toute notre attention a été mobilisée par l'analyse du rythme et finalement, on passe à côté de la musique. Notre attention au détail nous a écarté du morceau global. On pourrait dire que l'on a écrasé la dimension temporelle du morceau. Une autre approche consiste à assister à toutes les répétitions du fameux groupe. Ce faisant, on est capable de dire si d'une manière générale les musiciens ont tendance à jouer ensemble. Ainsi, le jour d'une représentation, toute notre attention peut rester uniquement sur le concert. On garde donc la dimension temporelle.

C'est par ce changement de positionnement face au problème de résolution temporelle que Voytek et al. (2013) introduit le Event Related Phase-Amplitude Coupling . L'approche traditionnelle du PAC nécessitant de connaître un nombre de cycles afin d'en déduire l'existence ou non du couplage, et donc perdre la dimension temps, l'article propose de calculer le PAC à travers les essais (ou répétitions). Pour un jeu de données de M essais de longueur N , on extrait respectivement les phases et les amplitudes $\phi_M(t)$ et $a_M(t)$ puis, pour chaque point temporel, on calcule la corrélation à travers les essais (corrélation linéaire-circulaire (Berens and others, 2009) qui se fait entre l'amplitude et des sinus/cosinus de la phase). Il en résulte une valeur de corrélation pour chaque instant et donc, de couplage.

3.2 APPRENTISSAGE SUPERVISÉ

Le travail effectué durant cette thèse s'est exclusivement porté sur l'apprentissage supervisé. Celui-ci consiste à apprendre à la machine à reconnaître des événements qui ont été labellisé au préalable (cf. 3.2.1). A contrario, l'apprentissage non supervisé laisse la machine apprendre par elle-même. En pratique, l'apprentissage se fait sur des attributs. Par exemple, pour différencier des chats et des chiens, on pourra utiliser l'angle formé par le sommet des oreilles. Les attributs doivent contenir une information pertinente permettant de différencier les classes. Enfin, les algorithmes de classification vont servir de ces attributs pour définir une frontière entre les classes étudiées. A ce stade, il semble important de préciser que l'utilisation des outils d'apprentissage machine peut s'orienter (globalement) suivant deux axes :

1. Optimisation des attributs : on travail sur un raffinement des attributs afin que ceux-ci soient les plus performants possibles pour séparer les classes
2. Optimisation des paramètres de classification : on considère une base de données comme étant fixe, définitive, optimale et l'on va faire varier les différents paramètres liés à l'apprentissage machine (classificateurs, cross-validation...). C'est le cas des compétitions *BCI* où tout le monde travail sur une même base de données.

Bien sûr, ces deux axes peuvent être cumulés. Dans le cadre de cette thèse, le machine learning a été utilisé comme outil de validation d'hypothèses donc essentiellement porté sur l'optimisation des attributs. Le raffinement

des paramètres de classification a également été étudié, mais, au final, il ne constitue pas la majeure partie de l'étude.

Un schéma classique d'analyse peut-être décrit par :

1. Labellisation des données
2. Constitution de données d'entraînement (*training*) et de test (*testing*)
3. Choix d'un classifieur puis entraînement de celui-ci sur les données *training*
4. Test de ce classifieur entraîné sur les données *testing* et évaluation de la performance
5. Évaluation statistique de cette acuité de décodage

3.2.1 Labellisation et apprentissage

La labellisation c'est le fait d'associer à chaque événement l'appartenance à une classe ou à une condition. C'est par ce procédé que l'on va pouvoir apprendre ensuite au classifieur à identifier les classes. Par exemple, considérons *up* et *down* deux classes qui reflètent des mouvements de la main vers le haut ou vers le bas. On va donc construire un vecteur $y_{direction}$ qui labellise chaque essais avec direction effectuée (ce vecteur peut aussi être booléen ou contenir des entiers. L'essentiel est que à chaque classe soit attribué une valeur qui lui est propre). Ce vecteur y est appelé *vecteur label*, qui vient labelliser chaque essais d'un vecteur d'attributs x .

$$y_{direction} = \begin{pmatrix} up \\ down \\ down \\ \vdots \\ up \end{pmatrix}, y_{bool} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, x = \begin{pmatrix} x_{trial_1} \\ x_{trial_2} \\ x_{trial_3} \\ \vdots \\ x_{trial_N} \end{pmatrix} \quad (3.9)$$

D'où le nom apprentissage supervisé. Finalement, l'apprentissage machine se fera grâce à ce vecteur label y et cette matrice d'attributs x . Ce qui nous amène directement aux notions de *training set* et de *testing set*.

x	[]	[]	[]	[]	[]	[]	[]	[]	[]
y	0	0	0	1	1	1	2	2	2

FIGURE 3.5 – Labellisation de données

3.2.2 Training, testing et validation-croisée

Cette section est sans aucun doute la plus importante pour le machine learning puisque c'est elle qui assure la conformité méthodologique.

Un bon exemple pour comprendre cette partie est celui des contrôles de mathématiques. Avant l'examen, l'étudiant s'entraîne sur une série d'exercices. C'est la phase de *training*. D'ailleurs, plus il s'entraîne, plus ses chances de réussir à l'examen sont grandes. Le jour du contrôle, le professeur teste l'étudiant sur une série de nouveaux exercices en lien avec ce

qu'il a étudié. C'est le *testing*. Ici, c'est un test parfait puisque l'étudiant est naïf sur le contenu de l'examen ce qui veut dire que l'on teste ses capacités mathématiques pures. Toutefois, il peut arriver durant la scolarité que l'on soit testé sur des exercices que l'on a déjà vu dans la phase de *training*. Dans ce cas, la moyenne des notes des étudiants est généralement beaucoup plus élevée puisque l'on ne teste plus des capacités mathématiques, mais la capacité à restituer un apprentissage.

Training set, testing set et naïveté

Pour en revenir à la question du machine learning, on définit une partie des données pour entraîner la machine. Ensuite, on teste cette machine entraînée sur un nouveau jeu de données de test. Il est essentiel d'avoir une séparation stricte entre des données définies comme *training* et des données de *testing* afin d'assurer la naïveté du classifieur. Même si cela peut parître évident, nous verrons que ça n'est pas toujours aussi facile que ça.

Se pose maintenant la question de comment l'on choisit de couper les données en *training* et *testing*. Une méthode serait de prendre une partie des données de manière aléatoire, de la définir comme *training* et sur tester sur les données restantes. Toutefois, ce choix ne représenterai qu'une partie des données. Une méthode plus exhaustive et plus rigoureuse consiste à utiliser une validation-croisée (ou cross-validation).

Validation-croisée

La validation-croisée (CV) est une procédure permettant de séparer les données en *training* et *testing*. Pour comprendre comment cela fonctionne, prenons un ensemble composé de N échantillons. Il existe plusieurs de CV mais de manière générale, toutes dérivent du même principe qui est la cross-validation k-Fold (Efron and Tibshirani, 1994, ?). On coupe les N échantillons en k paquets de tailles égales (ou proches). Ensuite, le classifieur est entraîné sur $k - 1$ paquets puis on le test sur le paquet restant. Cette procédure est ensuite appliquée k fois afin que chaque paquet passe au *testing*. On dira que la cross-validation est *stratified* si la proportion de classes représentées au sein de chaque dossier est approximativement uniforme à travers les folds. on pourra aussi rencontrer le terme *shuffle* si il y a un mélange supplémentaire. Tout cela nous emmène à des CV k-fold, k-fold stratified, k-fold shuffle ou encore k-fold stratified shuffle.

Concernant le nombre de folds, on rencontre en générale 3 valeurs à travers la littérature : 3-folds, 5-folds ou 10-folds (Latinne et al., 2001, Yanagisawa et al., 2009, Besserve et al., 2007, Waldert et al., 2008). Un cas particulier, mais si le nombre de folds $k = N$, ça revient à entraîner la machine sur $N - 1$ échantillons tester sur celui qui a été isolé et on répète cette procédure N fois. C'est ce que l'on appelle le *Leave-One-Out*. Toutefois cette dernière possède une grande variance et peut conduire à des estimations non fiables (Efron and Tibshirani, 1994, ?).

Un autre cas particulier, est celui du *Leave-p-Subject-Out* (Vidaurre et al., 2009, Lajnef et al., 2015) qui consiste à entraîner sur p sujets et tester sur les sujets restants. Cette procédure est particulièrement exigeante puis-

qu'elle nécessite d'avoir une certaine reproductibilité entre les sujets. Cette validation-croisée est fréquente avec des données EEG mais impossible à mettre en œuvre pour la sEEG à cause de l'implantation unique de chaque sujet.

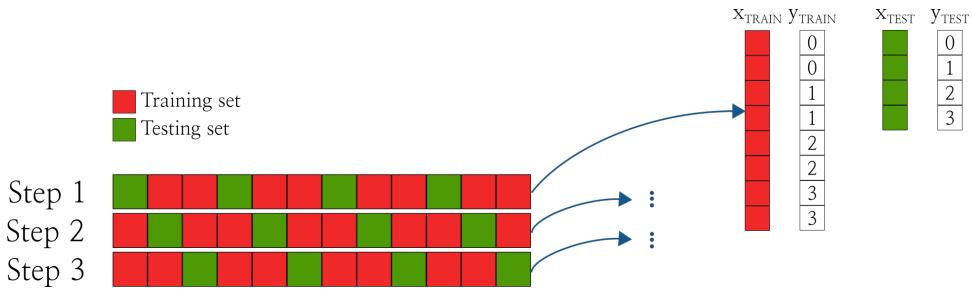


FIGURE 3.6 – Exemple d'une cross validation 3-folds

3.2.3 Classifieurs

1. Linear Discriminant Analysis (LDA)

Le LDA (Fisher, 1936) est un classifieur linéaire. Pour un problème à deux classes, le LDA tente de trouver un hyperplan qui va maximiser la distance entre les classes tout en minimisant la variance inter-classes. Ce classifieur fait l'hypothèse que les données sont normalement distribuées avec la même co-variance. Un problème multi-classes pouvant être transformée en multiple bi-classes, le LDA tente de trouver un hyperplan séparant la classe du reste (*One-vs-All*)

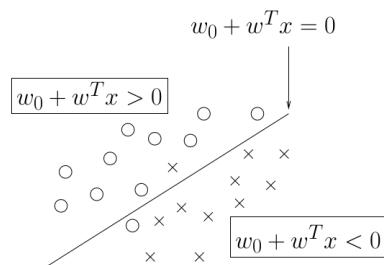


FIGURE 3.7 – Principe du Linear Discriminant Analysis (Lotte et al., 2007)

2. Support Vector Machine (SVM)

Le SVM (Boser et al., 1992, Cortes and Vapnik, 1995, Vladimir and Vapnik, 1995) utilise également un hyperplan pour séparer deux classes. Toutefois, cet hyperplan optimal est trouvé en maximisant les marges (ou distance) entre ce plan et les attributs les plus proches. Le SVM possède une particularité, il utilise un noyau qui peut permettre de résoudre les problèmes linéaire (*linear SVM*) mais également les problèmes non-linéaire en projetant les données dans un espace de dimension supérieure (*kernel trick*). Un noyau que l'on retrouve assez régulièrement est le *Radial Basis Function (RBF)* (?). Les problèmes multi-classes peuvent également être traités en *One-vs-All*

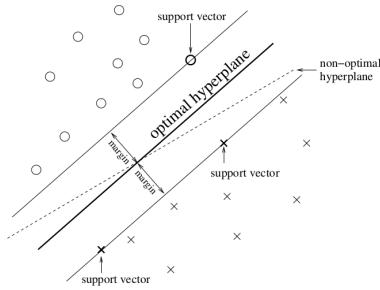


FIGURE 3.8 – Principe du Support Vector Machine (Lotte et al., 2007)

3. k-Nearest Neighbor (KNN)

Pour un nouveau point de testing, le KNN (Fix and Hodges Jr, 1951) mesure la distance avec les k plus proches voisins et déduit la classe de ce point en fonction des classes de ces k -voisins (l'attribution de la classe se fait donc par vote)

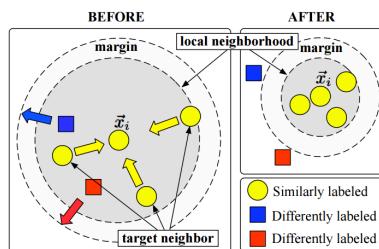


FIGURE 3.9 – Principe du k-Nearest Neighbor (Weinberger et al., 2005)

4. Naive Bayes (NB)

Le NB (Fukunaga, 1990) est un classifieur probabiliste. Une des hypothèses du NB est que les données dans les classes doivent être normalement distribuées et indépendante.

La figure A.3 en annexe, issue de l'excellentissime librairie python scikit-learn dédiée au machine learning, illustre le comportement de chaque classifieur face à trois types de données. D'autres informations détaillées à propos des classificateurs peuvent être trouvées dans Lotte et al. (2007), Wieland and Pittore (2014), Wu et al. (2008)

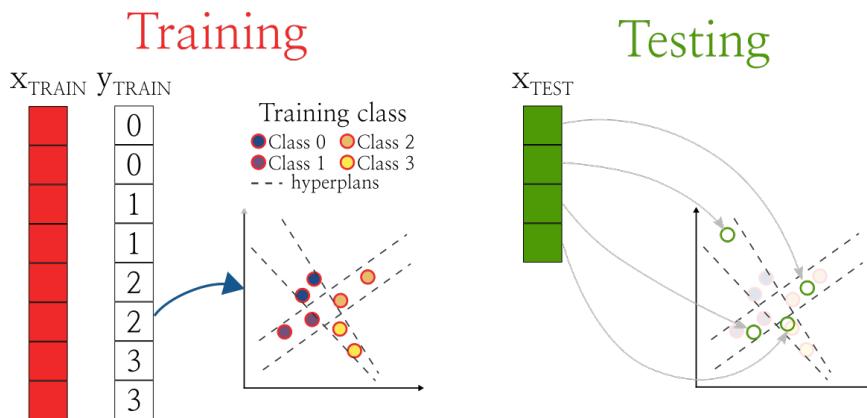


FIGURE 3.10 – Entraînement puis test d'un classifieur linéaire

3.2.4 Évaluation de la performance de décodage

La question qui se pose maintenant, c'est comment évaluer la performance de décodage. Pour cela, on peut par exemple utiliser le *Decoding accuracy* ou le *roc*

1. Decoding accuracy (DA)

L'utilisation (DA) est ce que l'on retrouve le plus fréquemment. Le calcul est simple, on compare les véritables labels avec les labels prédictifs par le classifieur. En faisant la somme des labels correctement prédictifs divisé par le nombre d'essais, on obtient un ratio qui correspond à l'acuité de décodage. Le plus souvent, ce ratio est ensuite exprimé en pourcentage. Le taux d'erreurs peut-être calculé en prenant $1 - DA$.

y_{TRUE}	$y_{PREDICTED}$
0	0
1	2
2	2
3	3
0	0
1	0
2	2
3	2
0	0
1	2
2	0
3	3

$$8/12 \Rightarrow 75\%$$

FIGURE 3.11 – Calcul de l'acuité de décodage

2. Receiver operating characteristic (ROC)

Une autre méthode pour évaluer la performance de décodage est l'utilisation de l'aire sous la courbe (AUC) ROC (Ling et al., 2003, Huang and Ling, 2005, Bradley, 1997). Celle-ci prend en compte le nombre d'essais correctement et incorrectement classifiés et pourrait donc prendre davantage de valeur possible comparé au Decoding accuracy .

3.2.5 Seuil de chance et évaluation statistique de la performance de décodage

De manière théorique, le seuil de chance est donné par $1/c$ où c est le nombre de classes. Par exemple, un problème à quatre classes donne un seuil de chance de 25%. Toutefois, ce seuil de chance est atteint pour un nombre de sample n infinis. En pratique, nous travaillons avec un nombre réduis de données, parfois même, avec très peu de sample. Dans ce cas, on peut obtenir des DA très élevés qui pourtant, ne sont pas pertinents. Les méthodes présentées ci-dessous ont pour but de trouver le seuil de chance associé à un jeu de donnée et de trouver pas la même occasion, la valeur p .

1. Loi binomiale

En faisant l'hypothèse que l'erreur de classification suit une distri-

bution binomiale cumulative, on peut utiliser la loi suivante pour en déduire la probabilité de prédire au moins z fois la classe c :

$$P(z) = \sum_{i=z}^n \binom{n}{i} \times \left(\frac{1}{c}\right)^i \times \left(\frac{c-1}{c}\right)^{n-1} \quad (3.10)$$

2. Permutation

Les permutations présentent l'avantage d'être calculées à partir des données (*data driven*). Ojala and Garriga (2010) nous renseigne sur les différents types de permutations possibles dans le cadre du décodage :

- (a) *Full permutation* : les données sont mélangés
- (b) *Shuffle y* : le vecteur de label est mélangé. C'est la procédure la plus fréquemment rencontrée.
- (c) *Intra-class shuffle* : les données sont mélangées à travers la dimension *features* (colonne) et ce, à l'intérieur de chaque classe.

Autant les méthodes (a) et (b) nous renseigne véritablement sur la consistance d'un décodage par rapport aux données, autant la méthode (c) donne des informations un peu différentes. En effet, en cas de décodage non-significatif, on pourra soit conclure qu'il n'y a pas de consistance dans les attributs à l'intérieur des classes, soit que le classifieur est incapable d'utiliser cette l'inter-dépendance. Ojala and Garriga (2010) précise que dans ce cas, il n'est pas nécessaire d'utiliser un classifieur compliqué et qu'un classifieur simple devrait suffire.

Cette partie est volontairement synthétique puisqu'elle a fait l'objet d'une publication scientifique (cf. II).

Un pipeline standard de classification est proposé en annexe (cf. A.2).

3.2.6 Du single au multi-features

Dans les sections précédentes, nous avons vu comment extraire des attributs de l'activité neuronale et comment les classifier. C'est ce que l'on appelle le *single feature* (SF), c'est-à-dire que l'on évalue la performance de chaque attribut séparément. Cette approche permet de constituer un set de features pertinents et répond à des questions neuro-scientifique. Cette démarche de SF a donc un but exploratoire.

La question que l'on peut maintenant se poser, c'est quelle performance de décodage puis-je obtenir si je combine ces attributs et dans quel cas est-ce utile ? C'est le multi-attributs (ou *multi-features* (MF)). Tout d'abord, le MF est utilisé lorsqu'il y a soit un désir soit un besoin de performances accrue. Par exemple, on utilisera le MF dans les compétitions de décodage ou tout simplement, pour une BCI où la performance est essentielle. Si l'on construit un système de bras robotisé piloté par activité neuronale, on comprend sans peine que celui-ci doit être le plus efficace possible et donc, le MF s'impose. Le dernier cas où l'on rencontre du MF, et ce n'est pas le cas le plus glorieux, c'est le cas où il y a un besoin de pallier à des résultats de SF assez faibles. La littérature expose des Decoding accuracy toujours plus hauts, des méthodes toujours plus complexes et donc, pour

publier correctement un article, il faut avoir des résultats au-moins aussi perspicaces.

Le multi-features c'est donc l'utilisation de multiples attributs pour aboutir à une classification et ce, sans sélection particulière. Individuellement, les attributs d'un même set n'auront pas la même performance. Certains seront des bons marqueurs et d'autres, n'ajouteront pas ou peu d'information. Donc en combinant ces features, il est probable que l'acuité de décodage soit moins bonne que la performance en attribut unique. Pour cela, on pourra donc utiliser des algorithmes de sélections de marqueurs (*feature selection*). Le but de cette sélection est de trouver dans un set d'attributs, un sous-ensemble dont la performance groupée est meilleure que la performance individuelle.

Cette sélection est une procédure exigeante où le risque de surapprentissage est grand. C'est la raison pour laquelle cette sélection doit être mise à l'intérieur d'une cross-validation . Donc on définit un set de *training* et de *testing* grâce à la validation croisée, puis sur le *training*, on lance la *feature selection*. On aboutit à un sous-ensemble de marqueurs qui va servir à entraîner le classifieur. Ensuite, on sélectionne ce subset dans le *testing* et on test le classifieur avec ce subset. Toute ceci étant enfin répété pour chaque *fold* de la cross-validation . A la vue de cette procédure, deux problèmes émergent :

- La sélection d'attributs se faisant à l'intérieur des folds de la cross-validation , on peut très bien aboutir à des listes d'attributs différentes. Pour obtenir une information finale, on pourra donc parler des attributs les plus fréquemment choisis. Par exemple, si la sélection se fait dans un cross-validation 10-folds, on pourra dire que le feature 1 a été choisi 7/10, le feature 2, 3/10...
- En fonction de la sélection choisie et de la cross-validation , le pipeline complet peut être très (très) lourd et long.

Les mécanismes de *feature selection* peuvent être regroupés en deux grandes familles (Guyon and Elisseeff, 2003, Liu et al., 2008, Das, 2001) : les *Filter methods* et les *Wrapper methods*.

Filter methods

Ces méthodes sont basées sur un critère et sont indépendantes du classifieur. Parmi elles, on retrouve des outils de corrélation, d'information mutuelle ou encore de statistiques. Ces derniers outils évaluent la contribution de chaque feature de manière indépendante sans tenir compte de la corrélation entre ces features. Pour résoudre ce problème, Yu and Liu (2004), Ding and Peng (2005) introduisent le *minimal-redundancy-maximal-relevance* qui en plus de trouver les features les plus pertinents, va permettre d'éliminer ceux qui sont redondants.

Pour terminer, ces méthodes sont effectivement indépendantes de l'algorithme de classification mais elles peuvent s'avérer optimales pour tel ou tel classifieur (ex : l'utilisation du critère de Fisher pour filtrer les features est très performant lorsqu'il est ensuite associé au Linear Discriminant Analysis (Duda et al., 2001)).

Wrapper methods

Contrairement aux méthodes de filtrage, les *wrapper* utilisent le classifieur comme outil de sélection. Le premier inconvénient que l'on peut d'ors et déjà leur reprocher, c'est que le résultat final sera donc classifier-dépendant, donc difficile pour la généralisation.

Parmi ces *Wrapper methods*, on peut citer :

1. Sélection exhaustive : on teste toutes les combinaisons de features possibles puis on sélectionne la meilleure. Procédure qui ne peut être faisable qu'en présence d'un jeu de données particulièrement restreint.
2. Sélection sur la statistique de décodage : on utilise le classifieur pour évaluer l'acuité de décodage de chaque feature séparément pour en déduire une valeur p (cf : 3.2.5). Enfin, on sélectionne les features dont la valeur p est inférieure à un seuil désiré.
3. Sélection séquentielle : processus où l'on va ajouter/enlever des features de manière séquentielle jusqu'à atteindre un décodage optimal. Ce type de sélection se fait suivant deux directions :
 - (a) *Forward feature selection* (FFS) : la première étape consiste à évaluer la performance de chaque attribut. On sélectionne le meilleur que l'on va ensuite combiner en couple avec tout les features restant. On sélectionne le meilleur couple puis on teste les combinaisons des meilleures triplettes... On continue tant que la performance s'améliore. Si le DA d'une étape i est inférieur au DA de l'étape $i - 1$, on considère le nouveau subset de features à $i - 1$.

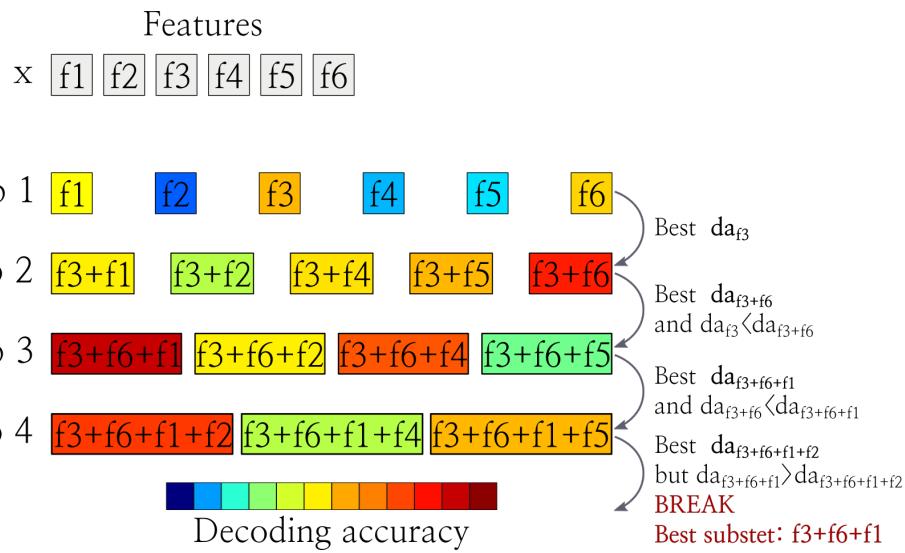


FIGURE 3.12 – Exemple d'une *Forward feature selection* appliquée sur six features

-
- (b) *Backward feature elimination* (BFE) : la philosophie est la même que pour un *forward*. On classifie d'abord les N features pris ensemble, puis on enlève à tour de rôle chaque marqueur. On sélectionne le subset composé de $N - 1$ features ayant fourni

le meilleur résultat, puis on enlève de nouveau chaque feature... L'algorithme s'arrête de la même façon que le *forward*.

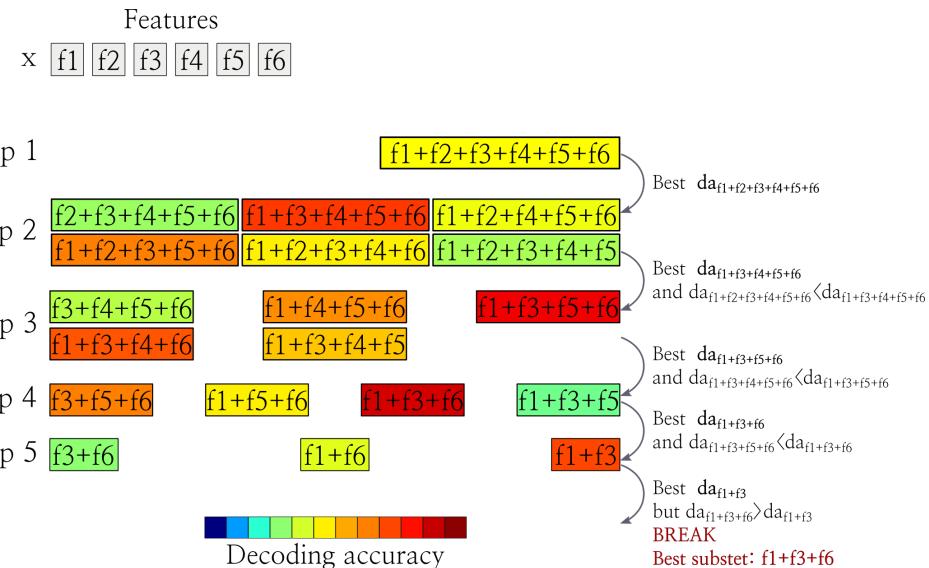


FIGURE 3.13 – Exemple d'une Backward feature elimination appliquée sur six features

De manière générale, il est rapporté que la FFS converge plus rapidement que la BFE (Guyon and Elisseeff, 2003). Toutefois, la FFS tombe plus facilement dans des minimums locaux et donc, mène à un décodage moins bon. En effet, la *forward* sélectionne pas-à-pas les meilleurs attributs, elle est donc moins ensembliste que la *backward*.

Les méthodes de filtrage demandent moins de ressources et représentent donc un premier choix pour les larges sets de données. En revanche, elles peuvent ne pas déceler les phénomènes de complémentarité entre features. Pour cette dernière raison, les méthodes de wrapper fournissent en général de meilleurs résultats (Chai and Domeniconi, 2004).

3.2.7 Généralisation temporelle

L'introduction du *single-feature* faite plus haut était une présentation générique, c'est-à-dire que celle-ci est vraie quelque soit les features étudiés. On pourra donc classifier des attributs de puissance, de PAC, de phase, d'entropie... On peut également envisager l'étude un seul marqueur mais dans sa dimension temporelle. En effet, cela consiste à entraîner et tester un classifieur à différents instants temporels pour voir si le décodage varie dans le temps. Une des limitations de cette utilisation d'un classifieur est que, à chaque instant, celui-ci change. Donc on ne peut inférer aucune généralisation. Pour envisager une généralisation, il faut entraîner le classifieur à un instant puis le tester à travers toute la dimension temporelle restante. Dans ce cas, on pourra parler de généralisation mais reste encore le problème du choix de l'instant temporel qui servira à entraîner le prédicteur.

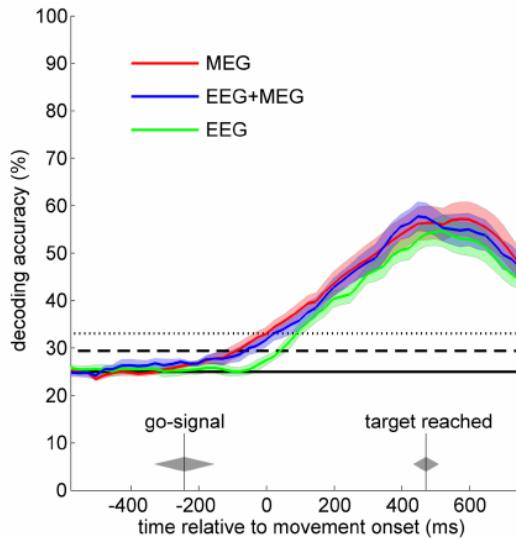


FIGURE 3.14 – Exemple de décodage temporel (Waldert et al., 2008). Ici, l'auteur décide 4-directions de mouvements de la main dans le temps. A chaque instant, un classifieur est créé, entraîné puis testé à ce même instant.

Pour répondre à cette question, King and Dehaene (2014) introduisent une idée particulièrement esthétique visant à généraliser le comportement d'un classifieur à travers le temps. Sans trop de surprise, ils ont nommé cette méthode la *généralisation temporelle*. Elle permet de répondre à deux limitations :

- Comment généraliser le comportement d'un classifieur lors d'une étude temporelle ?
- Comment choisir l'instant qui servira à entraîner le classifieur et quel impact ce choix aura-t-il sur le reste du décodage temporel ?

La méthodologie consiste à prendre un instant i , entraîner le classifieur et tester celui-ci sur tout les instants. Puis, on prend l'instant suivant $i + 1$, on entraîne un nouveau classifieur et on teste... Et on répète cette procédure pour tout les instants. On obtient ainsi une représentation 2D où, par convention, l'axe des ordonnées matérialise l'endroit où le classifieur a été entraîné (*Training time*) et l'axe des abscisses pour tester ce prédicteur sur le reste de la dimension temporelle (*Generalization time*). La couleur permettra de signaler la performance de décodage.

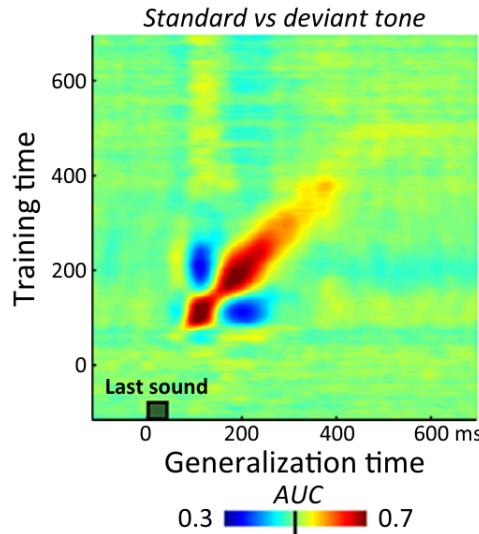


FIGURE 3.15 – Exemple de généralisation temporelle (King and Dehaene, 2014)

3.3 CONFIGURATION POUR DÉBUTER

Dans la jungle des méthodes, il peut parfois être difficile de s'y retrouver. Cette section a pour objectif de fournir une liste de méthodes conseillées pour débuter. Rien ne dit que ce sont les meilleures méthodes mais elles ont le mérite d'avoir fait leurs preuves, que ce soit dans cette thèse et surtout, dans la littérature. Gardons à l'esprit que les meilleures méthodes dépendent des données mais certaines, sont plus polyvalentes.

Couplage phase-amplitude :

Kullback-Leibler divergence avec swapping des essais de phase et d'amplitude (Tort et al., 2010)

Classification :

1. Cross-validation : 10-folds
2. Classifieur : Support Vector Machine avec noyau linéaire (Vladimir and Vapnik, 1995, Lotte et al., 2007)
3. Évaluation statistique : permutations (Ojala and Garriga, 2010, Combrisson and Jerbi, 2015)
4. Multi-features : *Backward feature elimination* (Guyon and Elisseeff, 2003)

Il y a aussi des interprétations neuroscientifiques possibles. Comme le fait d'entraîner dans une phase et de tester dans une autre. Si ça marche, ça veut aussi dire que ces phases partagent des corrélats neuronaux. A voir si j'inclue ceci.

3.4 IMPLÉMENTATION DES MÉTHODES PRÉSENTÉES

A force d'implémenter différentes méthodes afin de les tester, il se trouve que cet amas de fonction a formé assez naturellement une *toolbox*. Avec un peu de mise en forme, une uniformisation à travers les fonctions et la constitution d'une aide, nous avons décidé de mettre cet ensemble

d'outils *online* et en libre accès.

La toolbox s'appelle **brainpipe**, entièrement codée en Python. Le choix de python a également été mûrement réfléchis. La première version était en Matlab. Toutefois, Python présente les avantages d'être libre, beaucoup plus souple et intuitif et bien plus puissant. De plus, la communauté s'est appropriée ce langage et est particulièrement réactive. Quant à la documentation, elle est excellente et permet d'apprivoiser ce langage très rapidement. Autre point fort qui est une conséquence du libre, c'est la multitudes d'outils existants. Par exemple, pour la classification, il existe *scikit-learn* dont la qualité est sans comparaison possible avec Matlab. Enfin, les derniers outils à la pointe (tel que *tensorflow* de Google) ne seront jamais proposés en Matlab. Le Julia (qui est un autre langage de programmation) semble particulièrement prometteur mais, étant très jeune et ne disposant que d'une petite communauté, il a été écarté. Autre point, les outils que nous avons présenté dans cette partie méthode peuvent nécessiter énormément de ressources. **brainpipe** a donc entièrement été développé pour le calcul parallèle qui est intégré par défaut. A titre de comparaison, le calcul du PAC sur tout un jeu de données a pris une semaine Matlab, contre une journée en Python.

brainpipe est dédiée à la classification de signaux cérébraux. La construction globale peut être résumée avec les points suivants :

- *Pre-processing* : pré-traitement de données (filtrage, référencement, informations anatomiques en fonction de coordonnées MNI...)
- *Features* : extraction d'une large variété de features (amplitude, puissance, phase, PAC, PLV, ERPAC, entropie...)
- *Classification* des signatures cérébrales : repose sur *scikit-learn* pour l'implémentation des algorithmes de classification. **brainpipe** sert à faire le lien entre l'analyse de données neuro-scientifiques et le *machine-learning*. Il y a également un module dédié au *multi-features*.
- *Statistics* : tous les attributs disposent des méthodes statistiques ainsi que pour la classification et la correction multiple.
- *Visualization* : enfin, un ensemble d'outils de visualisation sont également disponibles afin de faciliter les représentations graphiques.

Liens vers **brainpipe** :

- Téléchargement : <https://github.com/EtienneCmb/brainpipe>
 - Documentation en ligne : <https://etiennecmb.github.io/>
1. *ipywksp* : pour les personnes désirant ou utilisant déjà python, un environnement de développement particulièrement agréable et puissant s'impose de plus en plus. Ce projet s'appelle Jupyter. Pour l'introduire brièvement, c'est un notebook dans lequel le code est intégré dans des cellules qui peuvent être lancées individuellement. Une des lacunes de ce notebook pour les utilisateurs Matlab, c'est le manque d'un *workspace* pour voir les variables en cours. *ipywksp* est un petit module, également développé dans cette thèse qui vient combler ce manque. Toujours en libre accès : <https://github.com/EtienneCmb/ipywksp>

2. *visbrain* : enfin, un dernier module Python qui est en cours de développement mais qui proposera une interface graphique pour toute visualisation nécessitant un cerveau MNI 3D (connectivité, projection corticale...). Il existe de nombreux logiciels existants déjà. Le but de celui-ci sera de permettre une intégration totale avec **brainpipe**. Le projet débute, mais le lien suivant permet de suivre son évolution :
<https://github.com/EtienneCmb/visbrain>

DONNÉES EXPÉRIMENTALES

4

Durant cette thèse, l'exploration s'est faite chez l'Homme par le biais, principalement, de données de type Stéréoélectroencéphalographie (SEEG). Ces données rares de très grande qualité (cf. 4.1.2) ont été acquises avant le début de la thèse, ce qui a permis de rentrer dans le vif du sujet très rapidement, après une période d'acclimatation aux différents traitements, propres à ce type d'enregistrement. D'autres types tel que l'EEG, la MEG ou les micro-électrodes ont également été approchés mais de manière ponctuelle, comme ce fut le cas dans l'Étude 1 (cf. II) ou dans différentes collaborations. Toutefois, étant donné que le temps consacré à ces données ne représente qu'une faible portion du travail total, nous allons ici nous concentrer uniquement sur l'intra.

Pour commencer, nous verrons ce que l'analyse de la Stéréoélectroencéphalographie a de particulier (richesse des données, qualité, les traitements associés, les avantages et les limitations). Enfin, nous verrons concrètement les enregistrements qui ont été utilisés dans le cadre de cette thèse.

4.1 DONNÉES INTRACRÂNIENNES

4.1.1 Acquisition

La première question que l'on est en droit de se poser, c'est comment est-il possible de travailler, chez l'Homme, avec des enregistrements qui nécessitent une implantation invasive, c'est-à-dire dans le cortex ? Certaines personnes présentent des formes agressives d'épilepsies, pouvant s'avérer pharmacorésistantes. En fonction de la localisation du foyer épileptogène, les méfaits engendrés par les décharges épileptiques peuvent être variés. Dans ce cas, il est nécessaire de localiser ce foyer avec, de préférence, des techniques non-invasives telles que l'EEG ou la MEG. Mais si ces dernières ne permettent pas une localisation précise le patient sera implanté avec des macro-électrodes comme la SEEG pour tenter de localiser puis d'enlever ce foyer par intervention chirurgicale. Cette implantation a un second objectif, déterminer quel est le rôle fonctionnel de la structure lésée (rôle moteur, langage, vision...). C'est dans ce contexte que les chercheurs proposent au patient de participer à une étude scientifique.

4.1.2 Avantages et limitations des données intracrâniennes

Le paragraphe précédent met en exergue la rareté de ces données. De plus, ce type d'acquisition enregistre l'activité cérébrale d'une population

relativement restreinte de neurones. En conséquence, on peut espérer que ce petit groupe s'active dans des processus précis et ainsi, étudier des phénomènes fins. Enfin, le rapport signal sur bruit (RSB) de la SEEG est excellent, ce qui doit permettre l'étude de processus, même en essais unique là où d'autres types d'enregistrements auront besoin d'une large banque d'essais avant de pouvoir constater l'émergence d'un phénomène.

La SEEG présente toutefois quelques limitations que l'on peut nuancer. Le problème majeur est certainement la généralisation d'un phénomène ou la reproductibilité à travers les sujets. La pathologie est propre à chaque patient, donc son implantation aussi. Ce qui signifie qu'il n'y a aucune chance que plusieurs sujets présentent rigoureusement la même implantation. Pour contourner cette limitation, on pourra utiliser :

- Des régions d'intérêt (ROI) : on va regrouper les électrodes des différents sujets par "proximité" en faisant l'hypothèse que celles-ci s'activent de façon similaire face à un processus. Ces ROI pourront être par exemple les gyrus ou les aires de Brodmann. Bien sûr, ce que l'on gagne en généralisation, on le perd en précision.
- Projection corticale : autour de chaque électrode, on définit une sphère d'intérêt (généralement 10 millimètres de rayon) puis on prend l'intersection de ces sphères avec la surface du cerveau. Cette technique permet une visualisation des activités proches de la surface à travers les sujets mais on perd de la lisibilité sur ce qui se passe en profondeur.

Une utilisation combinée des ROI et de la projection corticale permet de palier, au moins partiellement, au problème de reproductibilité inter-sujets. Autre limitation, on ne dispose que d'une couverture partielle puisque le neuro-chirurgien implante une quantité limitée d'électrodes. Ce dernier point est traité en augmentant le nombre de sujets. Enfin, la dernière limitation que l'on soulèvera ici, concerne le fait de travailler sur un cerveau "malade" empêchant donc une généralisation à des sujets sains. On limite ce problème par un ensemble de prétraitements (Jerbi et al., 2009) décrits dans le prochain paragraphe.

Un dernier point que l'on peut argumenter à la fois comme avantage ou limitation, c'est de ne pas pouvoir contrôler l'implantation pour étudier un phénomène précis. Par exemple, si l'on analyse l'encodage moteur, on s'attendrait à concentrer les efforts sur le cortex moteur primaire ou pré-moteur. Or l'implantation SEEG peut très bien contenir du frontal, du pariétal, du temporal. Là où finalement on peut considérer ça comme un avantage, c'est que l'on a accès à un ensemble de structures jugées non-primordiales mais dont l'ajout pourrait permettre la compréhension d'un processus de manière plus globale.

4.1.3 Prétraitements

Le premier prétraitement appliqué a été une réjection des sites bruités ou présentant une activité pathologique, c'est-à-dire des décharges épileptiques. C'est par cette inspection manuelle que l'on augmente le potentiel de généralisation aux sujets sains.

Autre prétraitement, les données peuvent être bipolarisées comme c'est le cas dans de nombreuses études (Bastin et al., 2016, Ossandon et al., 2011,

Jerbi et al., 2009). La bipolarisation part du principe que deux sites proches enregistrent des activités neuronales différentes mais que toute source de bruit, ou influence de sources lointaines, se retrouvera sur ces deux sites. La technique de bipolarisation consiste donc à soustraire les activités neuronales de sites proches ce qui a pour effet de supprimer la partie commune, le bruit. Par exemple prenons une électrode contenant les sites $k9$, $k10$, $k11$ et $k12$. Après bipolarisation, on considérera les sites matérialisés par $k10 - k9$, $k11 - k10$ et $k12 - k11$. Les bénéfices de la bipolarisation peuvent être résumés par :

1. Limitation des influences des sources lointaines et de la tension secteur (50hz)
2. Augmentation de la spécificité qui, pour un site bipolarisé, est estimée à 3mm (Kahane et al., 2006, Lachaux et al., 2003, Jerbi et al., 2009)

4.2 DONNÉES D'ÉTUDE

Trois jeux de données intracrâniaux ont été exploré :

1. *Center-out* : étude de l'encodage et du décodage des actions et des intentions motrices lors de mouvements de main
2. *Occulo* : étude des intentions et décision de mouvements oculaires
3. *Emotions* : étude de l'encodage des émotions

4.2.1 Données *Center-out*

C'est le jeu de données qui a été le plus largement exploité. En effet, celui-ci a servis à étudier l'encodage (cf. III) et le décodage (cf. IV) des actions motrices chez l'homme.

Descriptif des données

Six sujets (six femmes), implantés au département de l'épilepsie de l'hôpital de Grenoble ont donné leur consentement écrit pour passer l'expérience, sous la supervision du personnel médical. Le tableau 4.2.1 résume les détails clinique des différents sujets.

	Dominance	Age	Genre	Zone épileptique
P ₁	D	19	F	Frontal (RH)
P ₂	D	23	F	Frontal (LH)
P ₃	D	18	F	Frontal (RH)
P ₄	D	18	F	Frontal (RH)
P ₅	D	31	F	Insula (RH)
P ₆	D	24	F	Frontal (LH)
Moyenne : 22.17 ± 4.6				

FIGURE 4.1 – Détails cliniques des sujets ayant participé à la tâche *Center-out*

Matériel d'acquisition

De 12 à 15 multi-électrodes ont été implantées dans différentes structures. Chaque multi-électrode possède entre 10 et 15 sites mesurant 0.8mm et séparés de 1.5mm. La localisation anatomique des électrodes s'est faite en utilisant le schéma d'implantation (exemple en annexe A.4) et l'atlas proportionnel de Talairach et Tournoux (Talairach and Tournoux, 1993). La visualisation de la pré-implantation s'est faite par *IRM – 3D* et un *CT – scan* a été utilisé pour la post-implantation. Enfin, un *IRM* a également servi pour visualiser les électrodes implantées dans la matière blanche. Les coordonnées Talairach ont été déduites du *CT – scan* puis ont été transformées en MNI afin de pouvoir les superposer dans un cerveau standard (cf. figure ci-dessous).

Le système Micromed a été utilisé pour visionner l'acquisition de l'activité neuronale. Une électrode prise dans la matière blanche a été prise comme référence et un filtrage *passband* entre [0.1, 200hz] a également été effectué *online*. La fréquence d'échantillonnage est de 1024hz.

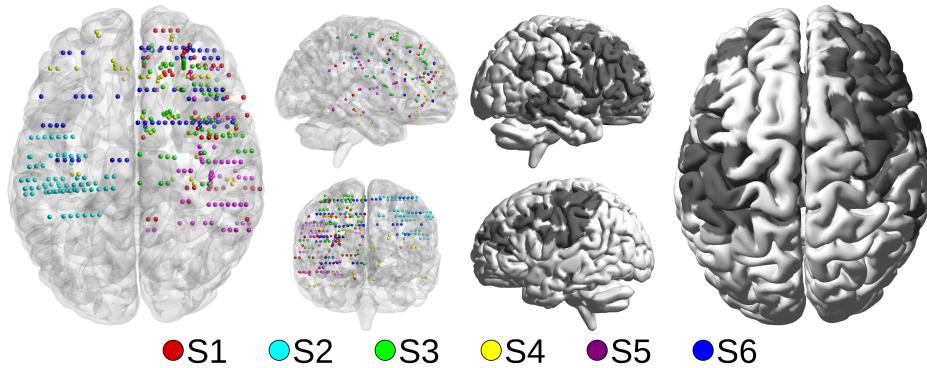


FIGURE 4.2 – *Implantation intracrâniale et couverture corticale de six sujets épileptiques ayant passés la tâche Center-out*

Descriptif de la tâche

La tâche est composée de trois phases :

1. Phase de repos : on demande au sujet de rester immobile pendant une durée de une seconde
2. Phase de préparation motrice (**CUE 1**) : une direction est imposée à l'écran (haut/bas/gauche/droite). On demande au sujet de se préparer pendant 1.5s à bouger la souris dans la direction imposée.
3. Phase d'exécution motrice : le sujet exécute le mouvement en bougeant la souris du centre à l'extrême de l'écran indiquée (environ 1.5s) puis de revenir vers le centre.

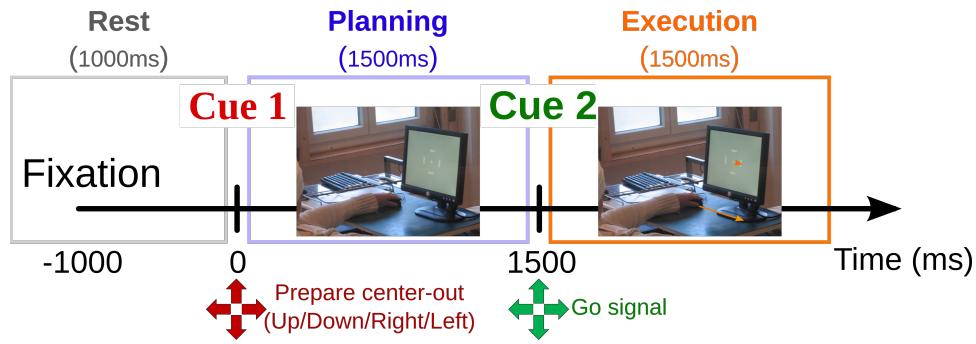


FIGURE 4.3 – Descriptif de la tâche Center-out

Cette tâche à conduit à deux études essentielles :

- L'étude de l'encodage des intentions motrices (cf. III) : on étudiera le décodage du repos vs préparation, repos vs exécution et de la préparation vs exécution
- L'étude du décodage des directions de mouvement (cf. IV) que ce soit pendant la préparation ou pendant l'exécution motrice.

4.2.2 Autres données

- Occulo : Données occulo mais un seul sujet donc pas super cool. De plus, les résultats sont un peu vieux et mériteraient de se pencher dessus une fois pour toute.
- Emotions : Ce serait pas mal que j'ai quelques résultats sur les émotions, histoire de montrer aussi un peu la diversité des données utilisées.

OUVERTURE

5

Nos contributions portent sur : ...

Le *premier chapitre* expose la problématique de la thèse.
Le *deuxième chapitre* présente en détail ...

etc.

Cette thèse a fait l'objet de divers travaux écrits : ...

Deuxième partie

Étude 1 : niveau de chance et évaluation statistique des résultats de classification par apprentissage supervisé

SOMMAIRE

5.1	PRÉSENTATION DE L'ÉTUDE	49
5.1.1	Contexte	49
5.1.2	Problématique	49
5.1.3	Résultats majeurs	49
5.2	ARTICLE	49
5.3	COMPLÉMENTS D'ÉTUDE	49
5.4	RÉSUMÉ DE L'ÉTUDE	55
5.5	ARTICLE	55
	CONCLUSION	55
5.6	RÉSUMÉ DE L'ÉTUDE	61
5.7	ARTICLE	61
	CONCLUSION	61
5.8	RÉSUMÉ DE L'ÉTUDE	67
5.9	ARTICLE	67
	CONCLUSION	67
5.10	RÉSUMÉ DE L'ÉTUDE	73
5.11	ARTICLE	73
	CONCLUSION	73

Sensibilisation à l'importance du nombre d'essais par exemple

5.1 PRÉSENTATION DE L'ÉTUDE**5.1.1 Contexte****5.1.2 Problématique****5.1.3 Résultats majeurs**

pourquoi cette étude ? Quelles questions ? - Seuil de chance théorique vs pratique ? - Impact sur des méthodes (cross-validation , classifieur) - Validation sur des données réelles (Intra MEG) - dédié aux étudiants - Fournit une toolbox pour reproduire les résultats

5.2 ARTICLE**5.3 COMPLÉMENTS D'ÉTUDE**

compléments sur les différents types de permutation (Ojala and Garriga, 2010)

Troisième partie

Étude 2 : encodage de l'intention et de l'exécution motrice

SOMMAIRE

5.1	PRÉSENTATION DE L'ÉTUDE	49
5.1.1	Contexte	49
5.1.2	Problématique	49
5.1.3	Résultats majeurs	49
5.2	ARTICLE	49
5.3	COMPLÉMENTS D'ÉTUDE	49
5.4	RÉSUMÉ DE L'ÉTUDE	55
5.5	ARTICLE	55
	CONCLUSION	55
5.6	RÉSUMÉ DE L'ÉTUDE	61
5.7	ARTICLE	61
	CONCLUSION	61
5.8	RÉSUMÉ DE L'ÉTUDE	67
5.9	ARTICLE	67
	CONCLUSION	67
5.10	RÉSUMÉ DE L'ÉTUDE	73
5.11	ARTICLE	73
	CONCLUSION	73

Ce chapitre introductif gnagnagna.
Pas obligatoire!

5.4 RÉSUMÉ DE L'ÉTUDE**5.5 ARTICLE****CONCLUSION DU CHAPITRE**

Ceci est la conclusion. Personnellement, je n'aime pas que la conclusion soit numéroté, mais je veux qu'elle apparaisse dans la table des matière, d'où la commande addcontentsline.

Quatrième partie

**Étude 3 : décodage des
directions de mouvement
pendant et avant l'exécution de
mouvement de membres
supérieurs**

SOMMAIRE

5.1	PRÉSENTATION DE L'ÉTUDE	49
5.1.1	Contexte	49
5.1.2	Problématique	49
5.1.3	Résultats majeurs	49
5.2	ARTICLE	49
5.3	COMPLÉMENTS D'ÉTUDE	49
5.4	RÉSUMÉ DE L'ÉTUDE	55
5.5	ARTICLE	55
	CONCLUSION	55
5.6	RÉSUMÉ DE L'ÉTUDE	61
5.7	ARTICLE	61
	CONCLUSION	61
5.8	RÉSUMÉ DE L'ÉTUDE	67
5.9	ARTICLE	67
	CONCLUSION	67
5.10	RÉSUMÉ DE L'ÉTUDE	73
5.11	ARTICLE	73
	CONCLUSION	73

Ce chapitre introductif gnagnagna.
Pas obligatoire!

5.6 RÉSUMÉ DE L'ÉTUDE

5.7 ARTICLE

CONCLUSION DU CHAPITRE

Ceci est la conclusion. Personnellement, je n'aime pas que la conclusion soit numéroté, mais je veux qu'elle apparaisse dans la table des matière, d'où la commande addcontentsline.

Cinquième partie

Étude 4 : optimisation des paramètres de la bande gamma

SOMMAIRE

5.1	PRÉSENTATION DE L'ÉTUDE	49
5.1.1	Contexte	49
5.1.2	Problématique	49
5.1.3	Résultats majeurs	49
5.2	ARTICLE	49
5.3	COMPLÉMENTS D'ÉTUDE	49
5.4	RÉSUMÉ DE L'ÉTUDE	55
5.5	ARTICLE	55
	CONCLUSION	55
5.6	RÉSUMÉ DE L'ÉTUDE	61
5.7	ARTICLE	61
	CONCLUSION	61
5.8	RÉSUMÉ DE L'ÉTUDE	67
5.9	ARTICLE	67
	CONCLUSION	67
5.10	RÉSUMÉ DE L'ÉTUDE	73
5.11	ARTICLE	73
	CONCLUSION	73

Ce chapitre introductif gnagnagna.
Pas obligatoire!

5.8 RÉSUMÉ DE L'ÉTUDE

5.9 ARTICLE

CONCLUSION DU CHAPITRE

Ceci est la conclusion. Personnellement, je n'aime pas que la conclusion soit numéroté, mais je veux qu'elle apparaisse dans la table des matière, d'où la commande addcontentsline.

Sixième partie

Étude 5 : décodage des émotions

SOMMAIRE

5.1	PRÉSENTATION DE L'ÉTUDE	49
5.1.1	Contexte	49
5.1.2	Problématique	49
5.1.3	Résultats majeurs	49
5.2	ARTICLE	49
5.3	COMPLÉMENTS D'ÉTUDE	49
5.4	RÉSUMÉ DE L'ÉTUDE	55
5.5	ARTICLE	55
	CONCLUSION	55
5.6	RÉSUMÉ DE L'ÉTUDE	61
5.7	ARTICLE	61
	CONCLUSION	61
5.8	RÉSUMÉ DE L'ÉTUDE	67
5.9	ARTICLE	67
	CONCLUSION	67
5.10	RÉSUMÉ DE L'ÉTUDE	73
5.11	ARTICLE	73
	CONCLUSION	73

Ce chapitre introductif gnagnagna.
Pas obligatoire!

5.10 RÉSUMÉ DE L'ÉTUDE

5.11 ARTICLE

CONCLUSION DU CHAPITRE

Ceci est la conclusion. Personnellement, je n'aime pas que la conclusion soit numéroté, mais je veux qu'elle apparaisse dans la table des matière, d'où la commande addcontentsline.

DISCUSSION GÉNÉRALE

Enfin : la conclusion générale!!!

Au cours de ce mémoire, nous avons développé un modèle ...

1. **Modélisation**

2. **Inférence statistique**

PERSPECTIVES

Dans la continuité directe de notre travail de thèse, nous pouvons ...

A ANNEXES

SOMMAIRE

A.1	COMPARATIF DE METHODES PAC (TORT ET AL., 2010)	77
A.2	PIPELINE STANDARD DE CLASSIFICATION	78
A.3	COMPARATIF DE CLASSIFIEURS (PEDREGOSA ET AL., 2011)	80
A.4	EXEMPLE DE SCHÉMA D'IMPLANTATION	82

Mettre juste
la page de
références
de brainpipe

A.1 COMPARATIF DE MÉTHODES PAC (TORT ET AL., 2010)

TABLE 1. *Summary of characteristics of the phase-amplitude coupling measures studied*

Phase-Amplitude Coupling Measure	Tolerance to Noise	Amplitude Independent	Sensitivity to Multimodality	Sensitivity to Modulation Width
Modulation index	Good	Yes	Good	Good
Heights ratio	Good	Yes	No discrimination	No
Mean vector length	Good	No	Restricted	Reasonable
Amplitude PSD	Low	No	Restricted	Good
Phase-locking value	Low	No*	Restricted	Low
Correlation measure	Low	No*	Restricted	Low
GLM measure	Low	No*	Restricted	Low
Coherence value	Low	No*	Restricted	Low

* Under the presence of noise.

FIGURE A.1 – Comparatif de méthodes PAC (Tort et al., 2010)

A.2 PIPELINE STANDARD DE CLASSIFICATION

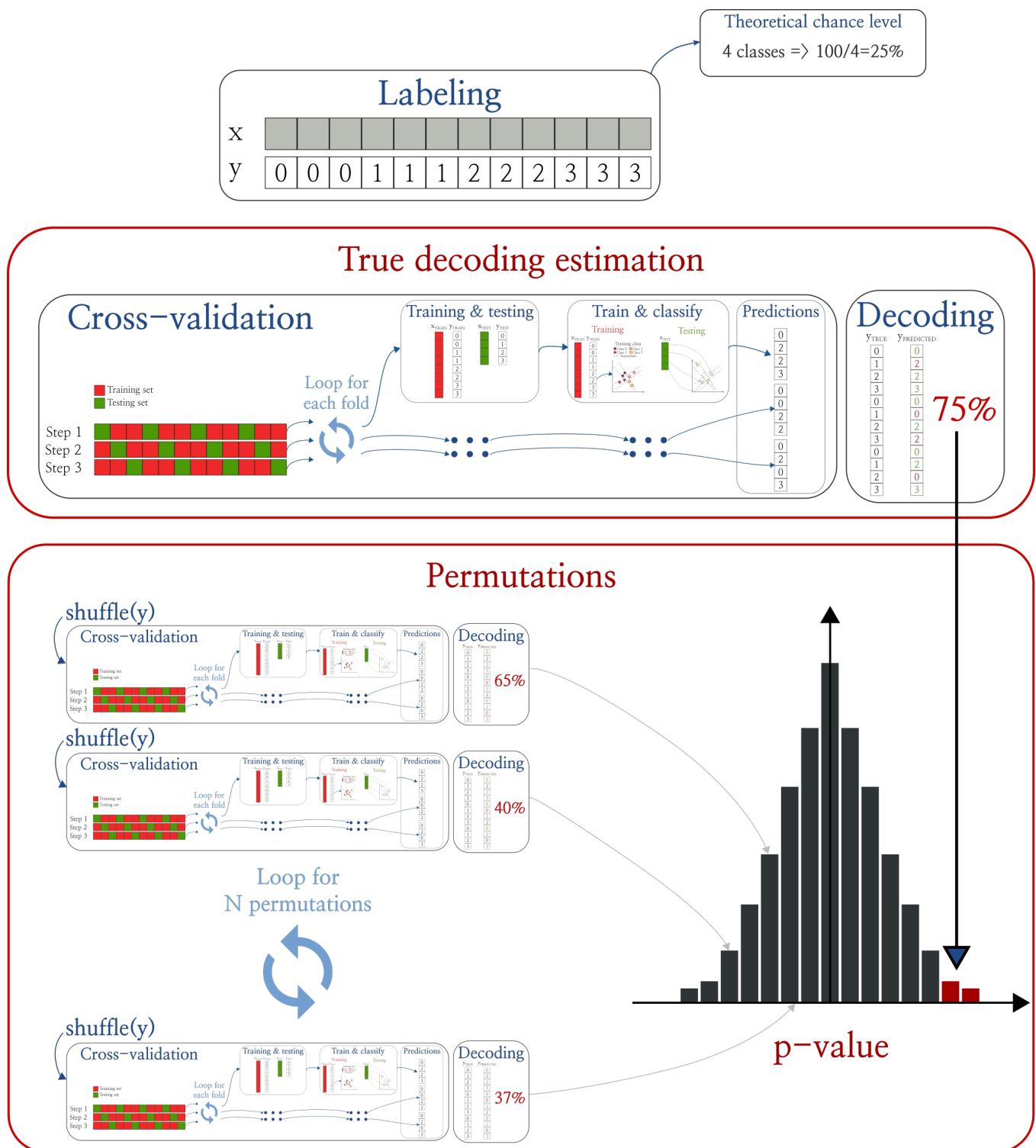


FIGURE A.2 – Pipeline standard de classification

A.3 COMPARATIF DE CLASSIFIERS (PEDREGOSA ET AL., 2011)

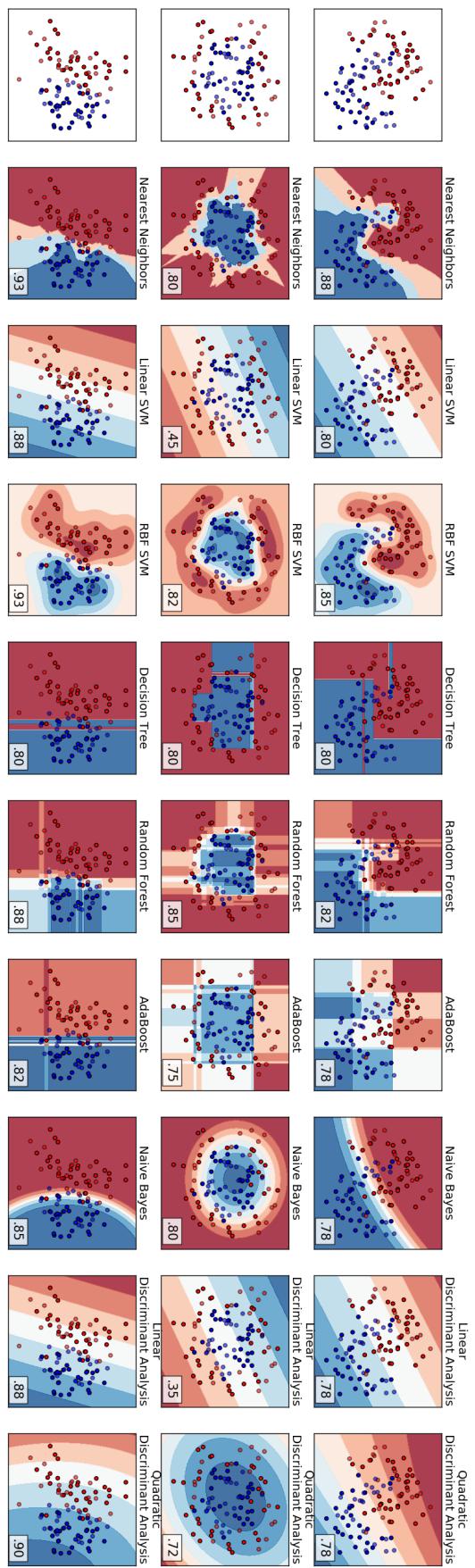


FIGURE A.3 – Comparatif de classifieurs (Pedregosa et al., 2011)

A.4 EXEMPLE DE SCHÉMA D'IMPLANTATION

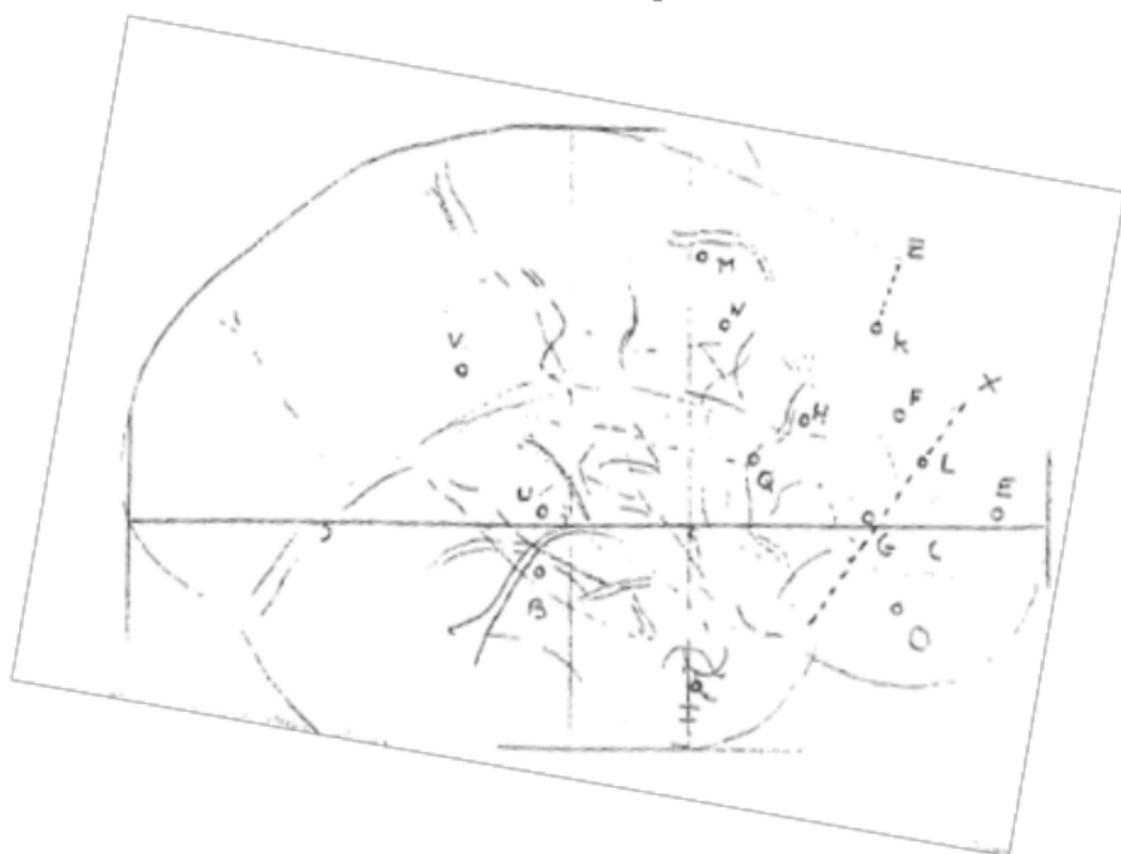
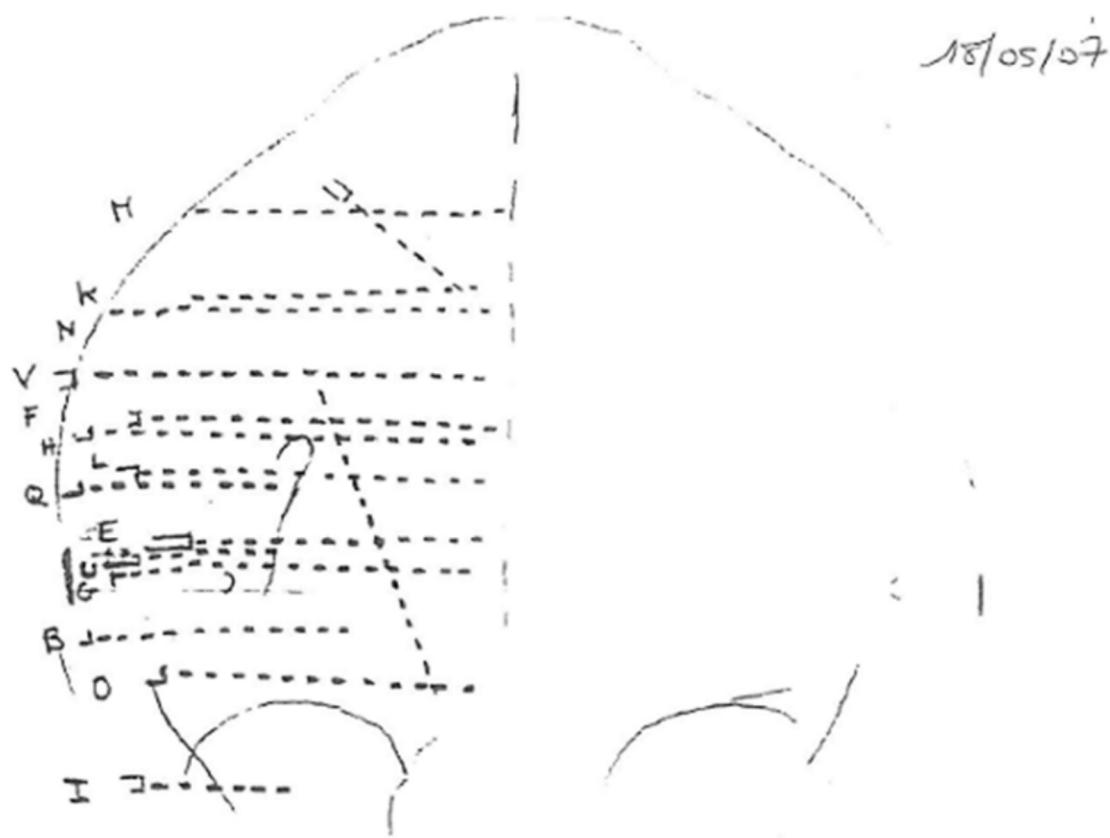


FIGURE A.4 – Exemple de schéma d'implantation

BIBLIOGRAPHIE

- Bahramisharif, A., van Gerven, M. A. J., Aarnoutse, E. J., Mercier, M. R., Schwartz, T. H., Foxe, J. J., Ramsey, N. F., and Jensen, O. (2013). Propagating Neocortical Gamma Bursts Are Coordinated by Traveling Alpha Waves. *Journal of Neuroscience*, 33(48) :18849–18854.
- Bastin, J., Deman, P., David, O., Gueguen, M., Benis, D., Minotti, L., Hoffman, D., Combrisson, E., Kujala, J., Perrone-Bertolotti, M., Kahane, P., Lachaux, J.-P., and Jerbi, K. (2016). Direct Recordings from Human Anterior Insula Reveal its Leading Role within the Error-Monitoring Network. *Cerebral Cortex*, page bhv352.
- Bekaert, M. H., Botte-Lecocq, C., Cabestaing, F., Rakotomamonjy, A., et al. (2009). Les interfaces Cerveau-Machine pour la palliation du handicap moteur sévère. *Sciences et Technologies pour le Handicap*, 3(1) :95–121.
- Berens, P. and others (2009). CircStat a MATLAB toolbox for circular statistics. *J Stat Softw*, 31(10) :1–21.
- Bertrand, O., Bohorquez, J., and Pernier, J. (1994). Time-frequency digital filtering based on an invertible wavelet transform : an application to evoked potentials. *IEEE Transactions on Biomedical Engineering*, 41(1) :77–88.
- Besserve, M., Jerbi, K., Laurent, F., Baillet, S., Martinerie, J., Garnero, L., et al. (2007). Classification methods for ongoing EEG and MEG signals. *Biol Res*, 40(4) :415–437.
- Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM.
- Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7) :1145–1159.
- Canolty, R. T., Edwards, E., Dalal, S. S., Soltani, M., Nagarajan, S. S., Kirsch, H. E., Berger, M. S., Barbaro, N. M., and Knight, R. T. (2006). High Gamma Power Is Phase-Locked to Theta Oscillations in Human Neocortex. *Science*, 313(5793) :1626–1628.
- Canolty, R. T. and Knight, R. T. (2010). The functional role of cross-frequency coupling.
- Chai, H. and Domeniconi, C. (2004). An evaluation of gene selection methods for multi-class microarray data classification. In *Proceedings of the*

- Second European Workshop on Data Mining and Text Mining in Bioinformatics, pages 3–10.
- Cohen, M. X. (2008). Assessing transient cross-frequency coupling in EEG data. *Journal of Neuroscience Methods*, 168(2) :494–499.
- Combrisson, E. and Jerbi, K. (2015). Exceeding chance level by chance : The caveat of theoretical chance levels in brain signal classification and statistical assessment of decoding accuracy. *Journal of Neuroscience Methods*, 250 :126–136.
- Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3) :273–297.
- Das, S. (2001). Filters, wrappers and a boosting-based hybrid for feature selection. In *ICML*, volume 1, pages 74–81. Citeseer.
- de Hemptinne, C., Ryapolova-Webb, E. S., Air, E. L., Garcia, P. A., Miller, K. J., Ojemann, J. G., Ostrem, J. L., Galifianakis, N. B., and Starr, P. A. (2013). Exaggerated phase–amplitude coupling in the primary motor cortex in Parkinson disease. *Proceedings of the National Academy of Sciences*.
- Demandt, E., Mehring, C., Vogt, K., Schulze-Bonhage, A., Aertsen, A., and Ball, T. (2012). Reaching Movement Onset- and End-Related Characteristics of EEG Spectral Power Modulations. *Frontiers in Neuroscience*, 6.
- Ding, C. and Peng, H. (2005). Minimum Redundancy Feature Selection from Microarray Gene Expression Data.
- Duda, R. O., Hart, P. E., and Stork, D. G. (2001). Pattern classification. 2nd. Edition. New York.
- Dvorak, D. and Fenton, A. A. (2014). Toward a proper estimation of phase–amplitude coupling in neural oscillations. *Journal of Neuroscience Methods*, 225 :42–56.
- Efron, B. and Tibshirani, R. J. (1994). *An introduction to the bootstrap*. CRC press.
- Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2) :179–188.
- Fix, E. and Hodges Jr, J. L. (1951). Discriminatory analysis-nonparametric discrimination : consistency properties. Technical report, DTIC Document.
- Fukunaga, K. (1990). Introduction to statistical pattern classification.
- Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3 :1157–1182.
- Hanakawa, T., Dimyan, M. A., and Hallett, M. (2008). Motor Planning, Imagery, and Execution in the Distributed Motor Network : A Time-Course Study with Functional MRI. *Cerebral Cortex*, 18(12) :2775–2788.

- Hochberg, L. R., Bacher, D., Jarosiewicz, B., Masse, N. Y., Simeral, J. D., Vogel, J., Haddadin, S., Liu, J., Cash, S. S., van der Smagt, P., and Donoghue, J. P. (2012). Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, 485(7398) :372–375.
- Huang, J. and Ling, C. X. (2005). Using AUC and accuracy in evaluating learning algorithms. *Knowledge and Data Engineering, IEEE Transactions on*, 17(3) :299–310.
- Hyafil, A., Giraud, A.-L., Fontolan, L., and Gutkin, B. (2015). Neural Cross-Frequency Coupling : Connecting Architectures, Mechanisms, and Functions. *Trends in Neurosciences*, 38(11) :725–740.
- Jerbi, K., Ossandón, T., Hamamé, C. M., Senova, S., Dalal, S. S., Jung, J., Minotti, L., Bertrand, O., Berthoz, A., Kahane, P., and Lachaux, J.-P. (2009). Task-related gamma-band dynamics from an intracerebral perspective : Review and implications for surface EEG and MEG. *Human Brain Mapping*, 30(6) :1758–1771.
- Jervis, B. W., Nichols, M. J., Johnson, T. E., Allen, E., and Hudson, N. R. (1983). A fundamental investigation of the composition of auditory evoked potentials. *IEEE transactions on bio-medical engineering*, 30(1) :43–50.
- Kahane, P., Landré, E., Minotti, L., Francione, S., and Ryvlin, P. (2006). The Bancaud and Talairach view on the epileptogenic zone : a working hypothesis. *Epileptic disorders : international epilepsy journal with videotape*, 8 :S16–26.
- King, J.-R. and Dehaene, S. (2014). Characterizing the dynamics of mental representations : the temporal generalization method. *Trends in Cognitive Sciences*, 18(4) :203–210.
- Lachaux, J., Rudrauf, D., and Kahane, P. (2003). Intracranial EEG and human brain mapping. *Journal of Physiology-Paris*, 97(4-6) :613–628.
- Lachaux, J.-P., Rodriguez, E., Martinerie, J., Varela, F. J., and others (1999). Measuring phase synchrony in brain signals. *Human brain mapping*, 8(4) :194–208.
- Lajnef, T., Chaibi, S., Ruby, P., Aguera, P.-E., Eichenlaub, J.-B., Samet, M., Kachouri, A., and Jerbi, K. (2015). Learning machines and sleeping brains : Automatic sleep stage classification using decision-tree multi-class support vector machines. *Journal of Neuroscience Methods*, 250 :94–105.
- Lakatos, P. (2005). An Oscillatory Hierarchy Controlling Neuronal Excitability and Stimulus Processing in the Auditory Cortex. *Journal of Neurophysiology*, 94(3) :1904–1911.
- Latinne, P., Debeir, O., and Decaestecker, C. (2001). Limiting the number of trees in random forests. In *International Workshop on Multiple Classifier Systems*, pages 178–187. Springer.

- Ling, C. X., Huang, J., and Zhang, H. (2003). AUC a statistically consistent and more discriminating measure than accuracy. In *IJCAI*, volume 3, pages 519–524.
- Liu, J., Ranka, S., and Kahveci, T. (2008). Classification and feature selection algorithms for multi-class CGH data. *Bioinformatics*, 24(13) :i86–i95.
- Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., Arnaldi, B., et al. (2007). A review of classification algorithms for EEG-based brain–computer interfaces. *Journal of neural engineering*, 4.
- Nakhnikian, A., Ito, S., Dwiel, L., Grasse, L., Rebec, G., Lauridsen, L., and Beggs, J. (2016). A novel cross-frequency coupling detection method using the generalized Morse wavelets. *Journal of Neuroscience Methods*, 269 :61–73.
- Ojala, M. and Garriga, G. C. (2010). Permutation tests for studying classifier performance. *The Journal of Machine Learning Research*, 11 :1833–1863.
- Ossandon, T., Jerbi, K., Vidal, J. R., Bayle, D. J., Henaff, M.-A., Jung, J., Minotti, L., Bertrand, O., Kahane, P., and Lachaux, J.-P. (2011). Transient Suppression of Broadband Gamma Power in the Default-Mode Network Is Correlated with Task Complexity and Subject Performance. *Journal of Neuroscience*, 31(41) :14521–14530.
- Ozkurt, T. E. (2012). Statistically Reliable and Fast Direct Estimation of Phase-Amplitude Cross-Frequency Coupling. *Biomedical Engineering, IEEE Transactions on*, 59(7) :1943–1950.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn : Machine Learning in Python. *Journal of Machine Learning Research*, 12 :2825–2830.
- Penny, W., Duzel, E., Miller, K., and Ojemann, J. (2008). Testing for nested oscillation. *Journal of Neuroscience Methods*, 174(1) :50–61.
- Rickert, J. (2005). Encoding of Movement Direction in Different Frequency Ranges of Motor Cortical Local Field Potentials. *Journal of Neuroscience*, 25(39) :8815–8824.
- Talairach, J. and Tournoux, P. (1993). *Referentially oriented cerebral MRI anatomy : an atlas of stereotaxic anatomical correlations for gray and white matter*. Thieme.
- Tallon-Baudry, C., Bertrand, O., Delpuech, C., and Pernier, J. (1997). Oscillatory γ -band (30–70 Hz) activity induced by a visual search task in humans. *The Journal of neuroscience*, 17(2) :722–734.
- Tort, A. B. L., Komorowski, R., Eichenbaum, H., and Kopell, N. (2010). Measuring Phase-Amplitude Coupling Between Neuronal Oscillations of Different Frequencies. *Journal of Neurophysiology*, 104(2) :1195–1210.

- Van Langhenhove, A., Bekaert, M. H., Cabestaing, F., N'Guyen, J. P., et al. (2008). Interfaces cerveau-ordinateur et rééducation fonctionnelle : étude de cas chez un patient hémiplasique. *Sciences et Technologies pour le Handicap*, 2(1) :41–54.
- Vidaurre, C., Krämer, N., Blankertz, B., and Schlögl, A. (2009). Time domain parameters as a feature for EEG-based brain-computer interfaces. *Neural Networks*, 22(9) :1313–1319.
- Vladimir, V. N. and Vapnik, V. (1995). The nature of statistical learning theory.
- Voytek, B., D'Esposito, M., Crone, N., and Knight, R. T. (2013). A method for event-related phase/amplitude coupling. *NeuroImage*, 64 :416–424.
- Waldert, S., Pistohl, T., Braun, C., Ball, T., Aertsen, A., and Mehring, C. (2009). A review on directional information in neural signals for brain-machine interfaces. *Journal of Physiology-Paris*, 103(3-5) :244–254.
- Waldert, S., Preissl, H., Demandt, E., Braun, C., Birbaumer, N., Aertsen, A., and Mehring, C. (2008). Hand Movement Direction Decoded from MEG and EEG. *Journal of Neuroscience*, 28(4) :1000–1008.
- Watrous, A. J., Deuker, L., Fell, J., and Axmacher, N. (2015). Phase-amplitude coupling supports phase coding in human ECoG. *eLife*, 4 :e07886.
- Weinberger, K. Q., Blitzer, J., and Saul, L. K. (2005). Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480.
- Wieland, M. and Pittore, M. (2014). Performance Evaluation of Machine Learning Algorithms for Urban Pattern Recognition from Multi-spectral Satellite Images. *Remote Sensing*, 6(4) :2912–2939.
- Worrell, G., Jerbi, K., Kobayashi, K., Lina, J., Zelmann, R., and Le Van Quyen, M. (2012). Recording and analysis techniques for high-frequency oscillations. *Progress in neurobiology*, 98(3) :265–278.
- Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z.-H., Steinbach, M., Hand, D. J., and Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1) :1–37.
- Yanagisawa, T., Hirata, M., Saitoh, Y., Kato, A., Shibuya, D., Kamitani, Y., and Yoshimine, T. (2009). Neural decoding using gyral and intrasulcal electrocorticograms. *NeuroImage*, 45(4) :1099–1106.
- Yanagisawa, T., Yamashita, O., Hirata, M., Kishima, H., Saitoh, Y., Goto, T., Yoshimine, T., and Kamitani, Y. (2012). Regulation of Motor Representation by Phase-Amplitude Coupling in the Sensorimotor Cortex. *The Journal of Neuroscience*, 32(44) :15467–15475.
- Yu, L. and Liu, H. (2004). Redundancy based feature selection for microarray data. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 737–742. ACM.

Titre Décodage des intentions et des représentations motrices chez l'homme : analyse multi-échelle et application aux interfaces cerveau-machine

Résumé Le résumé en français (\approx 1000 caractères)

Mots-clés Les mots-clés en français

Title Le titre en anglais

Abstract Le résumé en anglais (\approx 1000 caractères)

Keywords Les mots-clés en anglais