

École polytechnique de Louvain

Cortex entorhinal dans la maladie d'Alzheimer préclinique

Author: **Romain PETIT**

Supervisors: **Benoît MACQ, Laurence DRICOT**

Readers: **Lise COLMANT, Nicolas DELINTE, Cyril DE BODT**

Academic year 2022–2023

Master [120] : ingénieur civil biomédical

Table des matières

1	Introduction	3
2	État de l'art	5
2.1	Découverte in-vivo	5
2.2	Maladie d'Alzheimer	8
2.3	Les Grid-cells chez l'humain	9
2.4	AppleGame	9
2.4.1	Mécanismes compensatoires	10
2.4.2	Étude de la performance	11
2.4.3	Influence de l'âge	12
2.4.4	Activations des différentes régions cérébrales	12
2.4.5	Magnitude des GLRs	13
3	Méthodes	14
3.1	Apple GAME	14
3.1.1	Tâche expérimentale	14
3.1.2	Recrutement	16
3.2	Data Preprocessing	17
3.2.1	Anatomical Preprocessing	17
3.2.2	Functional Preprocessing	18
3.2.3	Spatial smoothing	19
3.2.4	Coregistration	20
3.2.5	Liaison du protocole	20
3.3	Structure de stockage des données	21
3.4	Données du jeu	22
3.5	Région d'intérêt	23
3.6	Analyse des Grids-cells	23
3.6.1	Orientation angulaire moyenne	24
3.6.2	Magnitude des Grids-cells	25
3.6.3	Choix des analyses	25
3.6.4	Validation statistique	26
3.7	Programme GridCat	27
3.7.1	Données en entrée	29
3.7.2	Spécifications	29
3.7.3	GLM1	30
3.7.4	GLM2	30
3.7.5	Outils	31
4	Résultats	32
4.1	Scripts	32
4.1.1	Compatibilité BIDS	32
4.1.2	Data Analysis Manager	33

4.3.3	Stabilité d'orientation des voxels	44
4.3.4	Amplitude de réponse des Grids-cells	47
5	Discussion	50
5.1	Scripts	50
5.2	Masques MNI	52
5.3	GridCat	53
5.4	Grids-cells	54
5.4.1	Orientation de la grille	54
5.4.2	Stabilité d'orientation des voxels	55
5.4.3	Amplitude de réponse des Grids-cells	55
6	Conclusion	57
7	Remerciements	59

Chapitre 1

Introduction

La navigation spatiale a depuis longtemps, captivé l'intérêt des chercheurs en neurosciences, tant pour sa complexité que pour ses implications potentielles dans divers domaines. Ce mémoire se concentre sur son rôle dans la maladie d'Alzheimer avant l'apparition des premiers symptômes (stade préclinique), en explorant les mécanismes neuronaux sous-jacents.

La capacité à retrouver son chemin dans des environnements divers est basée sur un réseau neuronal complexe, comprenant des régions du cerveau qui jouent des rôles spécifiques dans le traitement de l'information spatiale. Parmi ces régions, on retrouve le cortex parahippocampique et entorhinal ainsi que l'hippocampe [1].

La désorientation spatiale est un des symptômes précoces de la maladie d'Alzheimer [2, 3]. Cette maladie est caractérisée par une perte progressive de la mémoire, de la cognition et représente un défi majeur pour la santé publique. Des études ont montré que des patients porteurs de l'allèle APOE- ϵ 4, un facteur de risque génétique majeur de la maladie d'Alzheimer, présentent des anomalies dans le cortex entorhinal [4]. Cette région du cerveau est constituée notamment de Grids-cells, qui présentent une organisation spatiale unique en forme de grille hexagonale. Ce mémoire vise à approfondir le rôle de ces cellules et leur lien avec la maladie d'Alzheimer. L'étude repose sur la comparaison du signal des Grids-cells entre les jeunes, les personnes âgées, les participants porteurs et non-porteurs de l'allèle APOE- ϵ 4.

Pour atteindre cet objectif, l'imagerie par résonance magnétique fonctionnelle (IRMf) est utilisée. Elle enregistre l'activité cérébrale du participant pendant qu'il effectue une tâche de mémorisation et de localisation d'un objet dans un espace virtuel. Ce type de tâche est fréquemment réalisé lors de l'analyse des Grids-cells. C'est d'ailleurs, l'adaptation d'un jeu virtuel existant, appelé l'AppleGame qui

est utilisé dans ce mémoire [5]. Afin d'analyser au mieux les données acquises, ce mémoire se propose d'automatiser les différentes étapes de traitement des données d'IRM. Dans le laboratoire, ces tâches doivent être effectuées manuellement pour chaque patient et chaque run. Le but est donc de fournir des scripts python qui communiqueront avec le programme BrainVoyager afin de traiter les images. La solution proposée se doit d'être facile à prendre en main et à adapter à d'autres expériences de traitement d'IRMf.

Les résultats de cette recherche pourraient avoir des implications significatives pour la compréhension des troubles neurodégénératifs et la détection de la maladie d'Alzheimer. En effet, la mise en évidence, de différences de signal des Grids-cells entre les différentes catégories de personnes étudiées, pourraient aider à un diagnostic précoce. C'est un enjeu majeur pour le développement d'approches thérapeutiques pour la maladie d'Alzheimer.

Chapitre 2

État de l'art

La navigation spatiale a été au coeur de plusieurs recherches afin d'en comprendre son fonctionnement et ses acteurs principaux. Le fait de pouvoir retrouver son chemin repose sur un réseau neuronal divisé en plusieurs modules. Les rôles des différentes régions du cerveau ont été identifiés par l'existence de cellules de lieu et de cellules de direction de la tête. Elles se situent respectivement dans l'hippocampe et dans le cortex parahippocampique (Visibles à la Figure 2.1). En ce qui concerne les cellules lieu, elles répondent aux données spatiales comme les mouvements directionnels, les translations ainsi que les repères. Cela montre que l'hippocampe joue un rôle spécifique au contexte. Il est donc légitime de penser que les informations indépendantes du contexte sont traitées en amont de l'hippocampe. Cela peut se faire par un algorithme transformant la position en une représentation métrique, orientée et indépendante du contexte. Si ce type de carte existe, elle pourrait se situer dans la région dorsocaudale du cortex entorhinal médian (dMEC) où il a été démontré, chez le rat, que la localisation y est représentée avec précision [1].

2.1 Découverte in-vivo

C'est en partant de cette supposition que des chercheurs ont étudié l'activité des neurones dans le dMEC chez le rat [1].

L'analyse a été réalisée chez les rongeurs car ces derniers présentent un hippocampe et un cortex entorhinal disproportionnellement grands par rapport aux autres régions de leur cerveau. En effet, cette caractéristique facilite l'expérimentation in vivo [7].

A l'aide d'électrodes, l'activité des neurones de la seconde couche du dMEC, est mesurée directement lorsque les rats se déplacent dans de grands enclos (Voir Figure 2.2.a,b).

Cette expérience a permis d'illustrer une organisation spatiale des champs de tir.

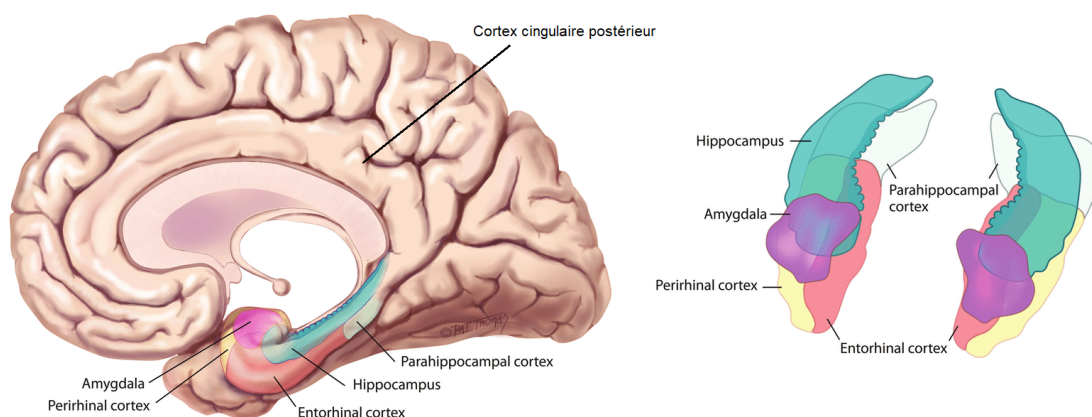


FIGURE 2.1 – Représentation des différentes régions impliquées dans la navigation avec l’hippocampe (en bleu), le cortex parahippocampique (en blanc) et le cortex entorhinal (en rose) [6].

En effet, dans chaque neurone, le champ forme une grille de triangles équilatéraux qui couvre la surface d’enregistrement. Chaque sommet de triangle correspond aux pics des taux de tirs (firing rate).

Ce type de neurone, présentant une organisation en grille, est donc appelé Grids-cell. Une analyse d’autocorrélation a montré que l’unité de la grille est un triangle équilatéral et que le centre de l’autocorrélogramme est entouré de six pics équidistants équivalents aux sommets d’un hexagone (Voir Figure 2.2.c). La présence répétitive d’hexagones à des distances multiples montre la régularité du modèle à travers l’ensemble du champ. Comme le montre la Figure 2.2.d, la taille n’a pas d’influence sur la structure en grille. C’est également le cas pour la forme et les bordures de l’enclos. À la différence des cellules de grille, les cellules de lieu possèdent un seul et unique champ de tir (firing field).

Les Grids-cells enregistrées par la même électrode (Nombre après la lettre t sur la Figure 2.2.b) présentent les mêmes propriétés métriques. Premièrement, l’espacement est la distance entre le pic central et les sommets de l’hexagone intérieur dans l’autocorrélogramme des Grids-cells. Deuxièmement, l’orientation est l’angle entre la ligne de référence et un vecteur vers le sommet le plus proche de l’hexagone intérieur dans le sens inverse des aiguilles d’une montre.

L’influence de la densité des Grids-cells a également été mise en évidence en comparant la région dorsale et la région ventrale du cortex entorhinal chez le rat. La région dorsale, plus dense en Grids-cells, se caractérise par des pics de firing rate plus importants et des champs individuels de taille plus réduite. L’inverse est observé pour la région ventrale à plus faible densité de Grids-cells.

Après avoir mis en évidence l’existence des Grid-cells, il est important d’étudier les différents facteurs qui peuvent les influencer. En ce qui les concerne, les termes

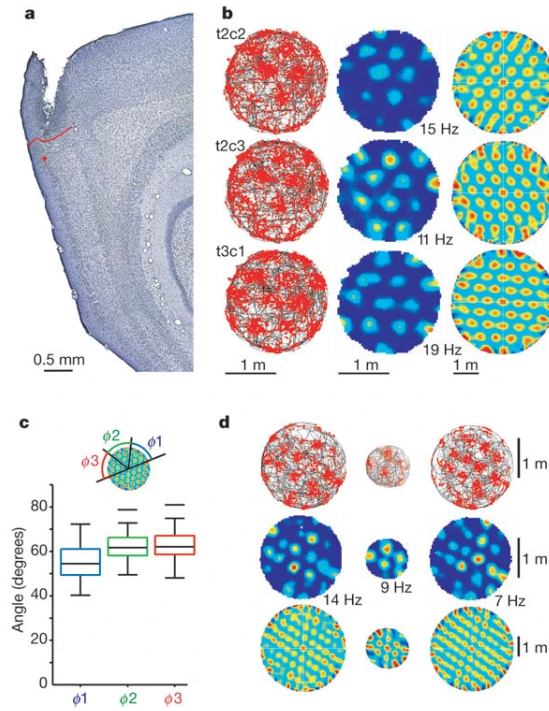


FIGURE 2.2 – **a** : Point rouge=Lieu d'enregistrement dans la couche II du dMEC. **b** : Gauche : Les firing rates de trois cellules simultanées. En noir, la trajectoire du rat avec en rouge, les firing fields superposés. Milieu : Rate map avec un code couleur et l'indication du peak rate (Min couleur=Bleu and Max couleur=Rouge). Droite : Autocorrélation spatiale pour chaque rate map. **c** : Distribution des 3 angles (ϕ_i) entre le pic central de l'autocorrélogramme et les sommets d'un hexagone défini par les six pics les plus proches. **d** : Mêmes analyses qu'en b montrant l'invariabilité de la structure triangulaire avec différentes tailles d'enceintes [1].

de repères allothetic (externes) et idiothetic (propres mouvements du rat) reviennent souvent dans la littérature. Il a été démontré, que les repères allothetic avaient une influence sur l'orientation de la grille. En effet, en faisant pivoter une carte de repères sur un mur, l'orientation de la grille pivote également. Cela suggère que la phase et l'orientation sont déterminées par ces repères. Après une navigation dans l'obscurité totale, il a été conclut que la présence de la grille en elle-même, n'est pas due à des repères sinon elle aurait disparu dans l'obscurité. Cependant, une légère dispersion des sommets est visible : les repères servent donc de référence pour l'alignement de la grille. Dans un environnement nouveau, la grille est présente dès le départ. L'orientation et la phase sont de nouveaux basées sur des repères externes. La structure spatiale peut donc être déterminée par des mouvements (repères idiothetic) tandis que l'orientation et la phase reposent sur des repères spécifiques à l'environnement (repères allothetic) [1].

En quoi la découverte de ces cellules de grille et leur rôle dans la navigation spatiale est-elle pertinente ? Cette question est expliquée ci-dessous en étudiant le lien entre ces cellules et la maladie d'Alzheimer chez l'homme.

2.2 Maladie d'Alzheimer

D'après les chiffres d'Alzheimer Belgique, en 2018, il y avait environ 212 000 personnes qui souffraient de démence en Belgique. Parmi elles, on dénombrait 137 800 personnes qui souffraient d'Alzheimer. Cette maladie représentait donc à elle seule près de 67% des démences en Belgique [8].

Cette maladie est caractérisée par la perte de mémoire et la désorientation spatiale. Actuellement, le manque de traitements efficaces est caractéristique de cette pathologie car ils interviennent à un stade trop avancé. Un des défis est de donc de détecter des signes de la maladie le plus tôt possible.

Pour cela, des études basées sur des patients porteurs du gène APOE- ϵ 4 ont été menées. En effet, il a été prouvé que celui-ci, est le gène avec le facteur de risque le plus important. De plus, les patients porteurs d'un seul allèle ont 3 fois plus de chance de développer la maladie et ceux qui en portent deux ont 10 fois plus de risques. Des changements histopathologiques ainsi que des anomalies de tau ont été observés dans le cortex entorhinal chez des jeunes patients (moins de 30 ans) et porteurs de l'allèle APOE- ϵ 4 [4].

On sait que les Grids-cells jouent un rôle dans la correction du mouvement et dans le processus d'intégration du chemin (Path Integration-PI). Ce dernier s'appuie sur les positions précédentes, la direction de la tête et la vitesse afin de déterminer la trajectoire vers l'objectif. Un dysfonctionnement des Grids-cells pourrait donc expliquer la désorientation des sujets atteints de la maladie d'Alzheimer [4]. Afin de le vérifier, il faut étudier leur signal chez l'humain.

2.3 Les Grid-cells chez l’humain

La mise en évidence du signal des Grids-cells chez l’homme ne se fait pas à l’aide d’électrodes comme chez le rat. Une méthode non-invasive est utilisée, à savoir l’imagerie par résonance magnétique fonctionnelle (IRMf). Le Blood-oxygenation-level-dependent (BOLD) signal est mesuré dans le cortex entorhinal afin d’y détecter les Grids-cells. Contrairement aux électrodes, c’est une mesure indirecte : on parle donc de grid-cell-like representations (GLRs).

Le type de tâche réalisé pour mettre en évidence les GLRs est basé sur la localisation et mémorisation d’un objet dans un environnement virtuel [4]. La tâche commence par la présentation d’un objet au patient. Celui-ci va disparaître quand le patient arrive à son emplacement. Parfois, le patient doit effectuer des déplacements supplémentaires (vers un arbre, ...) afin de complexifier la tâche. Finalement, il doit, soit reposer l’objet à l’endroit où il l’a pris ou retourner à sa position initiale dans le jeu. La performance dépend de la distance entre la position correcte et la position estimée par le patient. Bellmund et al.[9] ont accentué l’analyse en utilisant un casque de réalité virtuelle pour représenter une ville. Après une tâche d’entraînement, il est demandé aux patients d’imaginer la direction entre deux bâtiments. On peut remarquer que dans cette simulation, les sujets imaginent un mouvement mais ne le réalisent pas. La période d’imagination de ce vecteur directionnel a servi pour analyser l’influence des Grids-cells. L’analyse de leur signal a été réalisée dans le cortex entorhinal postéro-médial (pmEC) qui équivaut au cortex entorhinal médian (chez le rat) où la densité des Grids-cells est la plus élevée [1]. Un pattern récurrent y a été observé pour une périodicité de 60° , ce qui n’est pas le cas dans le cortex entorhinal antéro-latéral (alEC). Ce dernier est l’homologue du latéral EC chez les rongeurs qui ne contient pas de Grids-cells [1]. Cela démontre également, le rôle des cellules lors de mouvements imaginés et leur présence dans l’obscurité montré ci-dessus avec les rats [9].

2.4 AppleGame

L’Apple Game est une tâche de localisation et mémorisation d’un objet dans un environnement virtuel [5]. Cela a permis d’étudier la variation de l’intégration du chemin (PI). Le participant doit tout d’abord récupérer un panier vide ("Start location"). La position initiale est également celle où le panier doit être reposé ("Goal location"). Ensuite, il se déplace d’arbres en arbres ("Outgoing phase") jusqu’à arriver à un arbre avec une pomme ("Retrieve location"). Il doit alors, aller reposer le panier avec la pomme ("Incoming phase"). Afin d’étudier l’influence des repères, trois situations ont été simulées. La première, sans repères est appelée "pure PI" (PPI). La deuxième est caractérisée par la présence de frontières : Boundary-

supported PI (BPI). La dernière situation se déroule avec la présence de repères : Landmark-supported PI (LPI). Une représentation du jeu ainsi que des différentes situations décrites précédemment sont visibles à la figure 2.3. Le patient reçoit ensuite un feedback calculé en fonction de la différence de position entre celle de départ et celle où le panier a été reposé.

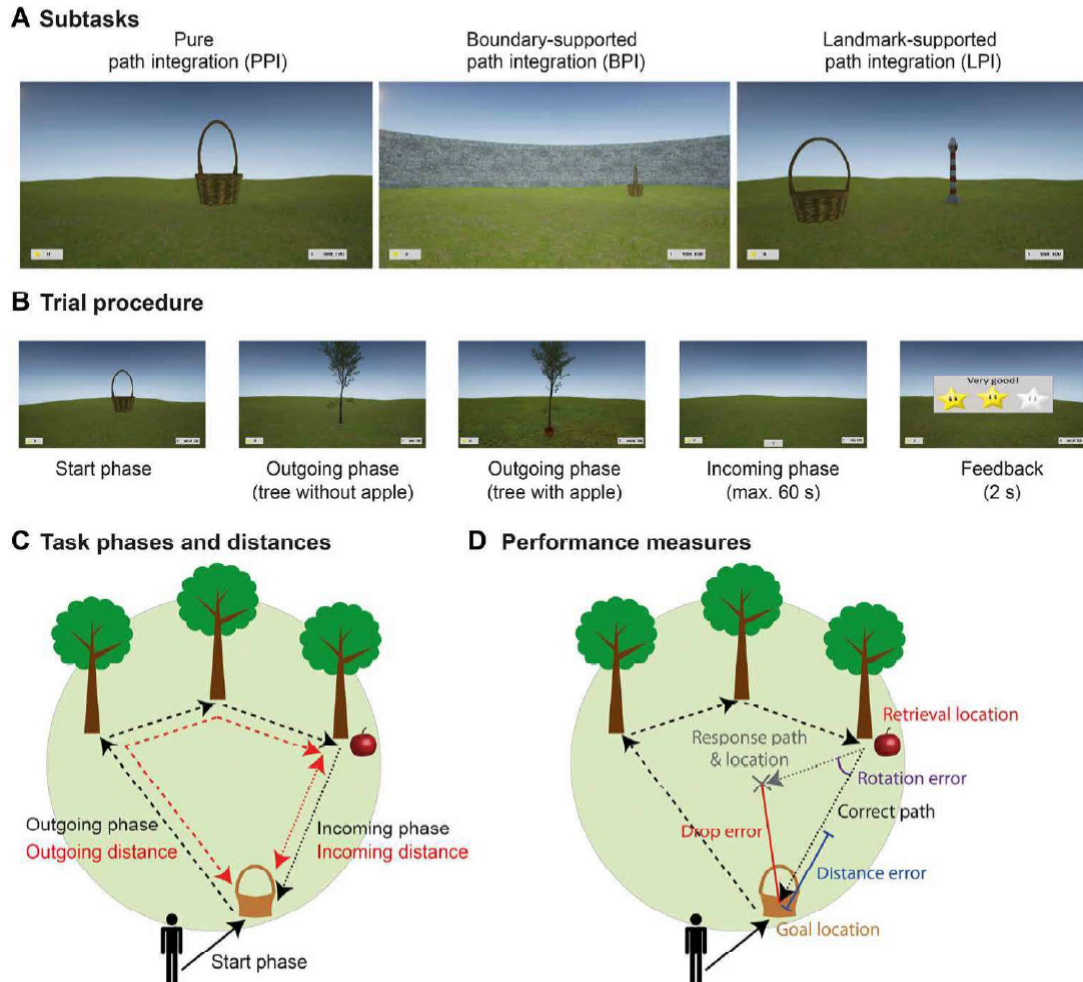


FIGURE 2.3 – Représentation de l’environnement virtuel dans l’Apple Game

Cette tâche est introduite brièvement ici car c’est une adaptation de celle-ci qui est étudiée dans ce mémoire.

2.4.1 Mécanismes compensatoires

La difficulté dans la détection de l’Alzheimer préclinique, est l’existence de mécanismes de compensation. En effet, ceux-ci vont dans un premier temps, prendre

le rôle de la région défectueuse. Ceci explique qu'il n'y a pas d'apparition de symptômes au début. En parallèle du cortex entorhinal, il faut donc analyser l'activité d'autres régions.

En comparant les performances des jeunes porteurs et non-porteurs de l'allèle APOE- ϵ 4, l'existence de mécanismes de compensation a pu être mise en évidence. A performances égales, une réduction des GLRs est remarquée chez les jeunes à risque [4]. Cette diminution d'activité dans le cortex entorhinal est accompagnée de l'augmentation de celle de l'hippocampe (HC). Son activité lors de tâche de navigation est donc négativement corrélée avec les GLRs dans le groupe à risque. L'hippocampe peut donc servir de mécanisme de compensation [5].

La comparaison des sous-tâches LPI et PPI, a mis en évidence une augmentation d'activité dans une autre région du cerveau, à savoir dans le cortex cingulaire/retrospinal postérieur (PC), en présence de repères spatiaux [5].

2.4.2 Étude de la performance

La performance est évaluée en mesurant la différence de distance entre la position initiale et la "Goal location".

La performance est comparée entre les 3 situations. Il en ressort qu'elle est impactée par la présence de repères. En effet, les participants sont plus précis en présence du phare (LPI). Les frontières (BPI) servent également de repères mais ont moins d'impact. L'absence totale de repères (PPI) augmente encore l'erreur de localisation.

Il y a également une "Incoming phase" plus longue pour les PPI. Cela signifie que les participants prennent plus de temps pour aller redéposer le panier.

L'absence de repères impacte donc la précision ainsi que la durée avant de déposer le panier.

Effets de l'APOE

Un déficit d'intégration du chemin est mis en évidence dans le groupe à risque.

Tout d'abord, les performances sont inférieures en PPI par rapport au BPI et LPI.

De plus, leur façon de se déplacer est différente du groupe contrôle. Les participants à risque ont tendance à naviguer proche du phare et loin des frontières. Il est également souligné qu'au plus la "goal location" est éloignée du phare, au plus les performances déclinent. Cela montre qu'ils utilisent les repères comme compensation.

Les membres du groupe à risque présentent donc un déficit de PI en absence de repère.

Effets du volume des régions cérébrales

Le volume de matière grise utilisée pour l'intégration du chemin est mesuré pour le cortex entorhinal, l'hippocampe et le cortex cingulaire/retrospinal postérieur (PC).

Une corrélation importante est mise en évidence entre le volume de l'EC et une "incoming distance" chez les personnes à risque. Cette relation montre l'influence de l'EC et laisse penser que s'il se dégrade, les erreurs lors de 'l'incoming phase' seront davantage visibles. Cette constatation n'est pas vérifiée pour "l'Outgoing phase" et pour une plus courte "Incoming phase".

En ce qui concerne l'HC et le PC, aucun effet du volume n'a été détecté.

2.4.3 Influence de l'âge

Dans cette section, l'influence de l'âge sur la performance et sur les repères est étudiée.

Pour ce qui est des jeunes, il n'y a pas de différences de performance entre le groupe contrôle et à risque pour les 3 situations. Pour les participants âgés, ceux à risque, performant mal en PPI. Ces résultats mettent en évidence de manière évidente l'impact de l'allèle APOE- $\epsilon 4$ avec l'avancée en âge, ce qui pourrait être en relation avec la pathologie préclinique de la maladie d'Alzheimer.

2.4.4 Activations des différentes régions cérébrales

Outgoing phase

Durant cette phase, on remarque la désactivation de l'EC. C'est le même constat pour l'HC excepté à proximité de la "Goal location".

Incoming phase

Le rôle de l'EC et de l'HC, dans l'intégration du chemin, est vérifié par l'activation de ces deux régions lors de cette phase.

LPI

Lors de cette sous-tâche, une augmentation de l'activité dans le PC est observée. Cela témoigne de son rôle de traitement des repères spatiaux. Cette région peut donc être recrutée comme compensateur suite à une réduction des GLRs.

2.4.5 Magnitude des GLRs

La magnitude des GLRs n'a pas d'effets significatifs sur la performance. Par contre, elle varie en fonction des sous-tâches. En effet, elle prédit la performance pour la sous-tâche PPI. Cela signifie que les Grids-cells gèrent l'intégration du chemin en absence de repères spatiaux.

Ces différentes découvertes suggèrent que le dysfonctionnement des Grids-cells pourrait être utilisé comme biomarqueur chez les jeunes à risque [5]. En effet, l'intégration du chemin est assuré par les Grids-cells en l'absence de repères. Il faut donc analyser les GLRs dans le cortex entorhinal postéro-médial, en l'absence de repères spatiaux. Cela permet d'éviter l'intervention des mécanismes de compensation [1, 5].

Chapitre 3

Méthodes

3.1 Apple GAME

Le protocole suivi lors de ce mémoire est une adaptation de l'AppleGame décrit dans l'état de l'art [5]. Il s'agit d'une tâche d'intégration du chemin, au cours de laquelle, les participants doivent retenir le chemin parcouru afin de pouvoir revenir au point de départ. Afin de pouvoir procéder à une analyse approfondie, il est important de parler des différentes tâches exécutées par ces personnes ainsi que du recrutement.

3.1.1 Tâche expérimentale

La tâche réalisée par les volontaires lors de ce mémoire est une adaptation de l'Apple Game décrit ci-dessus [5]. Le jeu consiste en une navigation dans un espace virtuel à l'aide d'un joystick. Ce dernier permet de réaliser des rotations et des translations, où à chaque instant la direction de mouvement correspond à la direction de la tête. Si l'on souhaite reculer, il faut réaliser une rotation de 180° et puis avancer. Les différents composants de l'interface virtuelle sont : un environnement assimilé à une plaine avec un ciel bleu, un panier pour collecter les pommes, un ou deux arbres et un phare. L'unité de mesure du jeu est le mètre virtuel (vm). La carte est un carré de 6400vm de côté et l'origine du repère est placé au milieu du carré.

Etapes du jeu

Le jeu est divisé en différentes phases (Visibles à la Figure 3.1.A) en fonction de l'avancement et des composants présents :

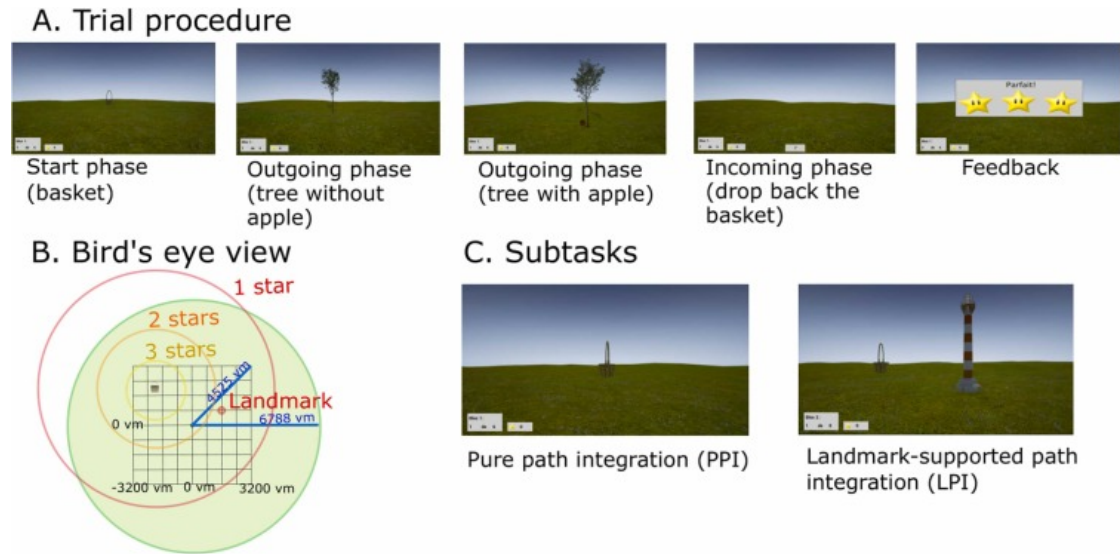


FIGURE 3.1 – **A** : Déroulement d’une partie avec les différentes étapes illustrées. **B** : Illustration de la méthode de notation avec le cercle rouge représentant la note d’1 étoile, le cercle orange la 2 étoiles et le cercle jaune la 3 étoiles. La position initiale du panier est représentée par un point gris. **C** : 2 sous-tâches sont disponibles : *Pure path integration (PPI)* : absence du phare et *Landmark-supported path integration (LPI)* : si présence du phare comme repère [10].

1. **Start phase** : Au début du jeu, un panier vide apparaît. Le joueur doit se déplacer afin d’aller le chercher. Il est important que le joueur mémorise la position du panier car le but final sera de le replacer à son emplacement initial le plus précisément possible.
2. **Outgoing phase** : Cette phase démarre avec l’apparition d’un arbre. Si une pomme se situe à son pied, le joueur doit se déplacer pour aller collecter la pomme. Dans le cas contraire, il se rend à l’emplacement de l’arbre et à son arrivée, celui-ci disparaîtra pour laisser place à deuxième arbre avec une pomme à un autre emplacement. Il faut alors aller chercher la pomme.
3. **Incoming phase** : Cette étape correspond au remplacement du panier le plus proche sa position initiale.
4. **Feedback** : En fonction de la différence de distance entre la position finale du panier et sa position d’origine, le joueur reçoit entre 0 et 3 étoiles (Voir Figure 3.1.B).

Comme expliqué ci-dessus, certaines parties diffèrent de par l’apparition d’un ou deux arbres lors du jeu. Il y a aussi une autre différence qui est la présence ou

pas d'un phare comme repère. Cela mène à 2 sous-tâches distinctes, à savoir, "Pure path integration (PPI)" en cas d'absence du phare et "Landmark-supported path integration (LPI)" si présence du phare comme repère. Ces deux situations sont illustrées à la figure 3.1.C. On remarque que, contrairement à l'article [5], il y aura au maximum 2 arbres et seulement un phare comme possible repère. La raison de cette simplification a pour objectif de réduire le temps d'acquisition par session tout en essayant de mettre en évidence l'activité des Grids-cells. De plus, les essais avec 3 arbres se sont révélés très compliqués à réaliser pour les personnes âgées.

Déroulement de l'acquisition

Le déroulement d'une session est divisée en 2 parties distinctes. La première, servant d'entraînement, est réalisée en dehors de l'IRM et n'est pas analysée. La deuxième partie se déroule dans l'IRM et 4 à 6 blocs sont acquis selon la motivation du participant. Chaque bloc est constitué de huit essais. Deux types d'essais sont proposés, LPI et PPI, comme décrit ci-dessus. Ces deux types sont équitablement répartis, à savoir, 4 LPI et 4 PPI par run. Les essais sont également caractérisés par la présence d'un ou deux arbres. En effet, soit l'arbre apparaît directement avec une pomme ou alors il est précédé par un arbre distracteur (sans pomme). La présence de distraction a pour but d'augmenter la difficulté d'intégration du chemin.

Types de données acquises

L'activité du cerveau, lors de la tâche de navigation, est enregistrée à l'aide de l'IRM fonctionnelle. Une IRM 3D anatomique est également réalisée pour chaque participant. Elle permet, après coregistration avec la IRMf, d'avoir une image anatomique précise afin de localiser les activations. Pour ce qui est des données du jeu, elles sont enregistrées sous forme de logfiles et seront développées dans la section 3.4 ci-dessous.

Les données analysées dans ce mémoire ont été acquises par Lise Colmant selon le protocole AppleGame expliqué ci-dessus. La première étape est donc de procéder au traitement des données avant de les analyser.

3.1.2 Recrutement

L'étude a commencé en Octobre 2020 et a pris fin en Decembre 2022. Pour cela, 279 personnes volontaires entre 18 et 80 ans ont été recrutées pour réaliser la tâche en dehors de l'IRM. La majorité des patients recrutés ont plus de 50 ans. De plus, le risque génétique de développer la maladie d'Alzheimer a également été estimé par le génotypage de l'APOE. Ce séquençage a mené à la classification des patients

en deux groupes. Le groupe à risque est celui porteur de l'allèle $\epsilon 4$ ($\epsilon 3\epsilon 4$ ou $\epsilon 4\epsilon 4$) tandis que le groupe contrôle regroupe les non-porteurs de l'allèle $\epsilon 4$ ($\epsilon 2\epsilon 3$ ou $\epsilon 3\epsilon 3$). A noter que les patients porteurs de $\epsilon 2\epsilon 4$ et $\epsilon 2\epsilon 2$ ont été exclus de l'étude car ils possèdent un risque intermédiaire de développer la maladie [10].

Afin d'assurer une bonne répartition entre les deux sous-groupes génétiques, différents aspects ont été évalués statistiquement à savoir l'âge, l'expérience des jeux vidéos ainsi que l'éducation. Le test "Mini-Mental State Examination (MMSE)" a également été réalisé par les volontaires de plus de 50 ans. Celui-ci permet de s'assurer que les personnes âgées ne possèdent pas de troubles cognitifs [11].

Un sous-groupe de ces participants a été recruté pour réaliser la tâche dans l'IRM (n=59). Dans ce mémoire, les sujets jeunes (<50ans), au nombre de 9, sont comparés à un groupe de 9 sujets âgés. Des informations complémentaires sur les participants sont représentées dans le tableau 3.1

TABLE 3.1 – Caractérisation de la population étudiée

Groupes	Moyenne d'âge	Écart-type d'âge	% Hommes
Jeunes	24.4	2.5	66%
Âgés	75.11	5.20	33%

3.2 Data Preprocessing

Un des objectifs de ce mémoire est d'automatiser les différentes étapes du traitement des images d'IRM fonctionnelle. Jusqu'à présent, ces étapes étaient réalisées manuellement, en cliquant directement dans le logiciel BrainVoyager. Avant d'analyser les résultats, celles-ci sont primordiales pour améliorer la qualité de l'image et permettre une meilleure analyse et comparaison entre sujets. Dans ce mémoire, ces différentes étapes sont celles classiquement réalisées dans le cas d'IRM et IRMf. Nous allons détailler ces différentes étapes pour les images anatomiques puis fonctionnelles.

3.2.1 Anatomical Preprocessing

Intensity inhomogeneity Correction(IIHC)

Cette étape a pour but de corriger les problèmes de variations d'intensité dans les images IRM. En effet, ces inhomogénéités peuvent conduire à des problèmes de segmentation. Les causes de celles-ci sont en général l'inhomogénéité du champ magnétique ou des artefacts d'imagerie. Cette correction est basée sur 3 étapes. Pour commencer, l'arrière plan est nettoyé, en éliminant les voxels de faible intensité.

Ensuite, un masque est appliqué pour extraire le cerveau, c'est à dire obtenir une image sans la boîte crânienne. Cette étape est importante car l'image anatomique acquise ne comprend pas seulement le cerveau mais également les tissus environnants. Enfin, à l'aide de polynômes de faible ordre, les fluctuations de basses fréquences sont estimées. Les voxels utilisés comme référence pour cette estimation sont ceux associés à la matière blanche car elle présente des niveaux d'intensité plus élevés par rapport aux autres tissus cérébraux dans la séquence T1. Les paramètres des polynômes sont estimés de manière itérative et fournissent donc un champ de biais qui est ensuite soustrait aux intensités originales. L'image corrigée possède donc une meilleure homogénéité des intensités.

Brain Normalization

Cela consiste en l'expression du cerveau dans un espace de référence. En effet, après normalisation, un point (x,y,z) est censé représenter une région similaire dans tous les cerveaux dans le même espace normalisé. Il est donc possible par après de pointer une région précise et la retrouver dans l'atlas associé. Cette étape est utile pour comparer plusieurs sujets en faisant des correspondances spatiales par exemple. Cette type de comparaison étant moins aisée dans l'espace natif, au vu des variations de tailles et de formes des cerveaux. Dans le cas de cette étude, une telle normalisation peut-être pertinente car la région correspond au cortex entorhinal pourra être définie par un masque unique pour tous les sujets normalisés. L'espace normalisé utilisé dans cette étude est l'espace Montreal Neurological Institute (MNI).

3.2.2 Functional Preprocessing

Pour l'IRMf, le préprocessing comprend les étapes suivantes :

Head motion correction

Le head motion correction consiste à corriger les différents mouvements de la tête pendant l'acquisition. En général, un volume fonctionnel de référence est choisi en tant que référence sur laquelle les autres volumes vont être alignés. Dans cette étude, le volume de référence est le premier de chaque run. Les mouvements de la tête sont assimilés à ceux d'un corps rigide. Ces derniers peuvent être définis par trois paramètres de translation (selon des axes x,y,z) et par trois paramètres de rotation (autour des axes x,y,z). Ces paramètres, exemple visible à la Figure 3.2, sont définis de manière itérative en alignant le volume courant par rapport au volume de référence. Dans BrainVoyager, la méthode utilisée est appelée Levenberg-Marquardt

qui est une méthode des moindres carrés non linéaires. Les paramètres sont définis lorsque qu'il n'y a plus d'amélioration d'une itération à une autre. Dans certains cas, si un sujet a trop bougé, il est rejeté de l'analyse. Un seuil de rejet est souvent fixé à un mouvement de plus de 1 à 2 voxels ou de plus de 5 millimètres. Les paramètres finaux sont ensuite utilisés pour générer les volumes finaux. Cela nécessite une interpolation spatiale pour calculer les valeurs du signal aux nouvelles positions. Cependant, il est important de noter que certaines déformations non-rigides se passent lors de l'enregistrement du cerveau. Cela peut-être liés, par exemple, à la compression des tissus, les mouvements vasculaires, les variations pulsatoires,... Ces influences peuvent-être atténuées par exemple en séparation le bruit du signal du signal en lui-même.

Slice Scan Time correction

Lors de l'enregistrement de l'IRMf, le cerveau n'est pas enregistré en une seule fois mais par tranches. Afin de diminuer la durée de l'enregistrement, plusieurs tranches sont enregistrées simultanément. L'acquisition par multibande est une technique d'accélération visant à exciter plusieurs tranches simultanément réduisant ainsi le temps d'acquisition. Dans notre cas, c'est un multibande de facteur 3, ce qui signifie que 3 tranches sont acquises simultanément. Comme on peut le voir sur la Figure 3.3, les tranches 1, 26 et 51 ont été acquises au premier pas de temps ($T=0$). On remarque que ce ne sont pas des tranches voisines qui ont été acquises simultanément. C'est pour cela qu'il faut procéder au Slice scan time correction. A noter, le temps de répétition (TR) est la durée nécessaire pour enregistrer l'entièreté du volume. Cela équivaut donc à la différence de temps entre l'enregistrement de la première et la dernière tranche. Dans cette étude, le TR est de 1.7 secondes.

High-Pass filtering

Le signal enregistré est caractérisé par des hautes fréquences qui correspondent aux activités liées aux stimulus et par des basses fréquences qui caractérisent le bruit physiologique et le bruit lié au scanner. Un filtre passe-haut est donc nécessaire pour éliminer le bruit.

3.2.3 Spatial smoothing

L'effet de ce lissage est comparable à un filtre passe-bas. Les hautes-fréquences sont éliminées en moyennant les voxels avec leurs voisins. Ce lissage permet de réduire le bruit mais a tendance à supprimer les détails et les variations rapides dans les données. Cela peut être utile pour simplifier des données complexes, mais peut aussi entraîner une perte d'informations importantes [12].

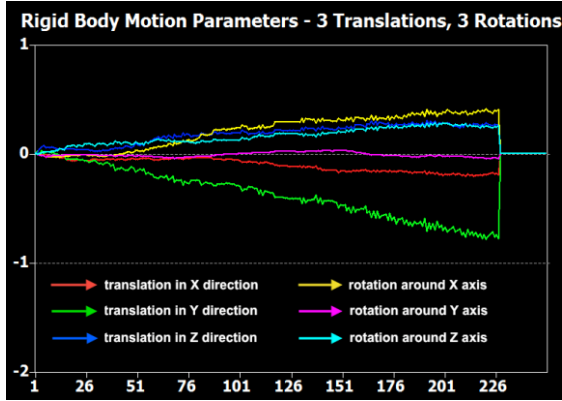


FIGURE 3.2 – Exemple d’estimation des 6 paramètres pour un run [12].



FIGURE 3.3 – Slice scan timing avec 3 tranches au même pas de temps ($T=0$) en rouge.

3.2.4 Coregistration

Après avoir traité les images anatomiques et fonctionnelles séparément, celles-ci sont alignées lors de la coregistration. Cette étape peut se faire dans l’espace du sujet (native) ou alors dans un espace de référence (Ex : MNI). Les différentes étapes de la coregistration sont :

- **Alignement initial (IA)** : Le but de cette étape est d’amener les images à se chevaucher en utilisant des rotations et des translations [12].
- **Ajustement de fin d’alignement (FA)** : Cette étape consiste à terminer l’alignement par méthode itérative. Deux techniques sont disponibles. La boundary-based registration (BBR) qui utilise les limites des structures anatomiques pour améliorer l’alignement. La deuxième méthode, plus basique, se base sur l’intensité des images pour ajuster l’alignement [12].
- **Volume Time Course (VTC)** : Un VTC représente l’évolution des activations cérébrales au cours du temps. Il est construit à partir des images anatomiques, des images fonctionnelles et des deux fichiers d’alignement (IA et FA) définis ci-dessus [12].

3.2.5 Liaison du protocole

La prochaine étape consiste à lier le protocole avec un VTC. Celle-ci permet d’aligner les événements du protocole avec les activations cérébrales du patient. La figure 3.4 permet de visualiser le lien entre le protocole et le VTC.

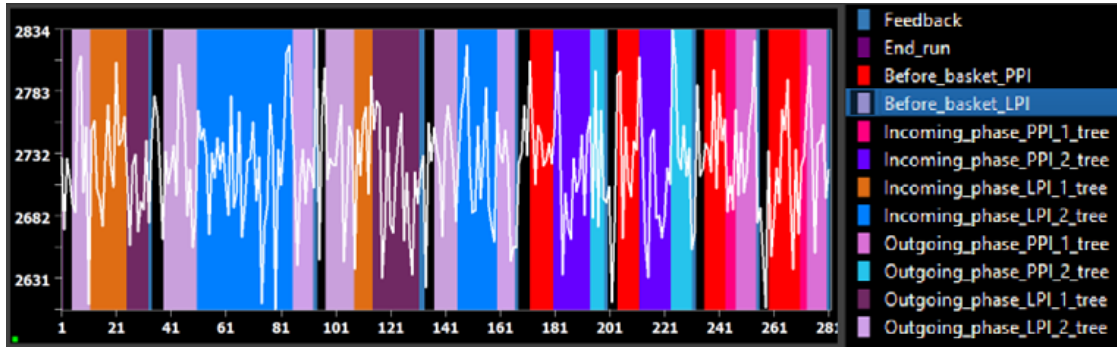


FIGURE 3.4 – Activité dans un voxel après avoir lié un VTC et un protocole. Les différents événements sont définis par la légende sur la droite de l'image. Ces couleurs sont superposées sur le signal et permettent donc d'observer l'activité cérébrale en lien avec une tâche spécifique.

3.3 Structure de stockage des données

Les différentes étapes nécessaires pour le traitement des images sont généralement les mêmes en IRMf, quelque soit la tâche utilisée. Les différents scripts développés lors de ce mémoire pourront donc être réutilisés par d'autres membres de l'UCLouvain. L'un des enjeux, est donc de réaliser des scripts clairs et bien détaillés. De plus, la structure de stockage ainsi que le nommage des fichiers sont très importants. Afin de faciliter l'utilisation de ceux-ci, un format universel de nommage et d'organisation des données de neuroimagerie appelé Brain Imaging Data Structure (BIDS) a été choisi. Dans la continuité des scripts, le format BIDS facilite le partage, les comparaisons et permet d'automatiser les analyses futures. De plus, dans les versions récentes de BrainVoyager, BIDS est aussi utilisé [12].

Un projet BIDS compatible (Voir Figure 3.5) est tout d'abord constitué d'un répertoire racine. Ce dernier contient un sous-dossier "rawdata" où sont stockées les données d'origine. Un deuxième sous-dossier appelé "derivatives" contient les données brutes converties au format NIfTI (Neuroimaging Informatics Technology Initiative). Ce dossier recueille donc tous les sous-dossiers relatifs aux différents traitements de données. A partir du moment où cette conversion en NIfTI est réalisée, il n'y a plus d'interactions avec les données d'origine et donc plus aucune influence de la structure de stockage d'origine. Les données brutes sont stockées dans des sous-dossiers spécifiques selon le sujet, la session et le type de données (anatomiques ou fonctionnelles). Pour ce qui est du nommage des fichiers, il se fait sous la forme $\langle key, value \rangle$, par exemple : sub-01 pour le sujet 1 [13, 14]. Dans le répertoire racine, on peut retrouver également des métadonnées (stockées en JSON ou TSV). Elles traitent de la description de l'étude (dataset_description.json), des participants (participants.tsv). Les différents scripts seront donc optimisés dans la

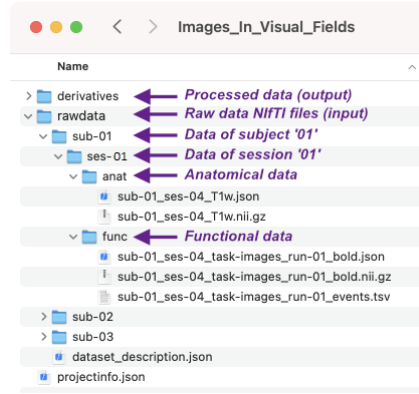


FIGURE 3.5 – Organisation du dossier racine selon le format BIDS [13].

mesure du possible avec l'utilisation de BrainVoyager.

3.4 Données du jeu

Les données du jeu servent de protocole et sont liées à l'imagerie fonctionnelle. Cela permet d'établir des liens entre certaines situations et l'évolution de l'activité cérébrale. Les données du jeu fournissent des informations à chaque pas de temps sur le jeu. Ces données sont extraites sous-forme de logfiles. Cependant, elles ne peuvent pas être directement liées au IRMF sous ce format.

Les données ont été traitées de manière suivante à l'aide de scripts python :

- Formatage des données du logfiles dans le format adéquat. Par exemple, les angles, les temps sont castés en double tandis que le numéro de la subtask, le phasetype sont castés en integer.
- Synchronisation des données du jeu avec l'enregistrement de la IRMF. En effet, l'enregistrement du jeu démarre avant l'IRM, il faut donc les réaligner.
- Calcul de la vitesse car le jeu ne fournit pas directement la vitesse mais les positions en X et Y du joueur. La vitesse peut donc être calculée sur base de ces deux composantes ainsi que de la différence de temps en les deux positions.

$$v = \frac{\sqrt{\Delta x^2 + \Delta y^2}}{\Delta t} \quad (3.1)$$

- Stockage sous-forme d'un dictionnaire pour avoir un accès facile aux différentes données (angles, vitesse, phasetype,...).
- Arrondi des angles au cinquième de degré près pour faciliter l'analyse future des Grids-cells. Le temps, exprimé en secondes, est aussi arrondi à 3 décimales

- pour plus de clarté.
- Déterminer les événements que l'on souhaite observer. Il y a des événements classiques comme une translation, une rotation (vitesse nulle) et la réception du feedback. En fonction de ce que l'on souhaite observer, il est possible de déterminer des événements avec la présence ou pas du phare par exemple.
- Écriture des données sous le format de sortie qui est un fichier texte (.txt).

3.5 Région d'intérêt

On sait que les Grids-cells sont localisées dans le cortex entorhinal [1]. Afin d'étudier uniquement l'activité cérébrale dans cette région, il faut appliquer sur les images fonctionnelles, un filtre qui sélectionne uniquement les voxels cibles. Dans ce cas-ci, cela consiste en un masque binaire où les voxels conservés ont la valeur 1 tandis que les autres, hors de la région cible, valent 0. Afin de faciliter la comparaison entre les différents participants, l'utilisation de masques en MNI a été choisie. Les masques du cortex entorhinal antéro-latéral (alEC) et postéro-médial (pmEC) en MNI sont utilisés [15]. Ces masques sont analysés et manipulés dans BrainVoyager et MRICron afin d'assurer leur bonne orientation et mise à l'échelle par rapport au VTC. L'article fournit 4 masques. En effet, chaque partie est divisée selon les deux voies d'entrées des informations dans le cortex entorhinal. Les sous-divisions représentent la partie connectée avec le cortex adjacent périrhinal (PRC) et le cortex parahippocampal (PHC). Les 4 masques disponibles sont : alEC_PRCpref_left, alEC_PRCpref_right, pmEC_PHCpref_left et pmEC_PHCpref_right. Les masques complets du pmEC et de l'alEC sont donc obtenus par fusion des sous-masques respectifs. En effet, les parties gauches et droites sont regroupées. Les figures 3.6 et 3.7 illustrent respectivement les deux masques, le postéro-médial et l'antéro-latéral [16, 17, 18].

3.6 Analyse des Grids-cells

Comme expliqué dans l'état de l'art, différentes études ont déjà mis en évidence l'activité des Grids-cells lors de tâches de navigation dans un milieu virtuel [4, 19]. Notre expérience étant semblable à celles-ci, la même technique d'analyse sera exécutée. Tout d'abord, les données sont divisées en deux-ensembles. Le premier sera l'ensemble d'estimation et le deuxième, celui de test. La méthode utilisée repose sur l'utilisation de deux modèles linéaires généralisés (GLM).

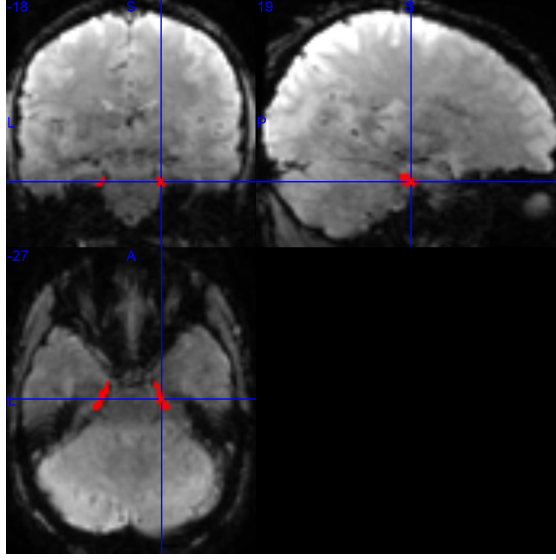


FIGURE 3.6 – Masque du cortex entorhinal postéro-médial (pmEC) superposé sur un volume d’imagerie fonctionnelle.

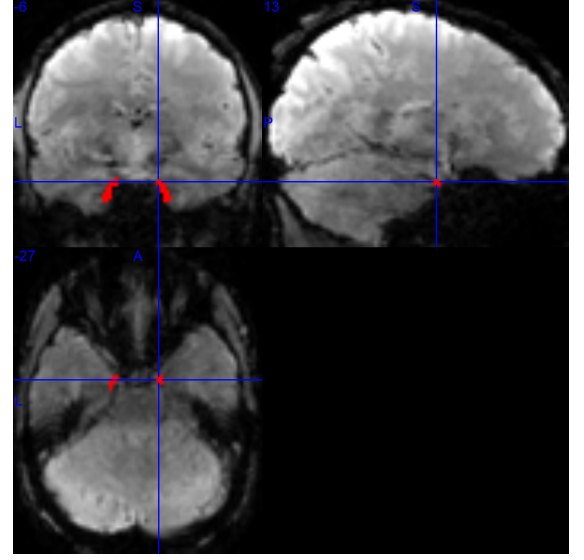


FIGURE 3.7 – Masque du cortex entorhinal antéro-latéral (alEC) superposé sur un volume d’imagerie fonctionnelle.

3.6.1 Orientation angulaire moyenne

Le premier GLM est utilisé pour estimer l’orientation angulaire de la grille relative pour chaque participant. Ce premier modèle comprend deux régresseurs paramétriques afin de représenter les mouvements de translations. Ces deux régresseurs sont :

$$\begin{cases} \beta_1 = \sin(\alpha_t * 6) \\ \beta_2 = \cos(\alpha_t * 6) \end{cases} \quad (3.2)$$

La variable α_t représente la direction du mouvement à l’instant t . Le facteur 6, présent dans les deux régresseurs de l’équation 3.2, est utilisé pour changer l’espace de référence (Voir Figure 3.8). En effet, après cette multiplication, les directions sont représentées dans un espace de 60° . Cela correspond à l’organisation sextuple d’une Grid-cell et les régresseurs seront donc plus sensibles à ce type de symétrie.

L’orientation moyenne de la grille est basée sur la moyenne des estimations de β_1 et β_2 pour tous les voxels de l’ensemble d’estimation. Elle peut-être obtenue par :

$$\varphi_{60^\circ} = \arctan\left(\frac{\bar{\beta}_1}{\bar{\beta}_2}\right) \quad (3.3)$$

En divisant l’orientation moyenne obtenue (φ_{60°) par 6, on repasse dans l’espace

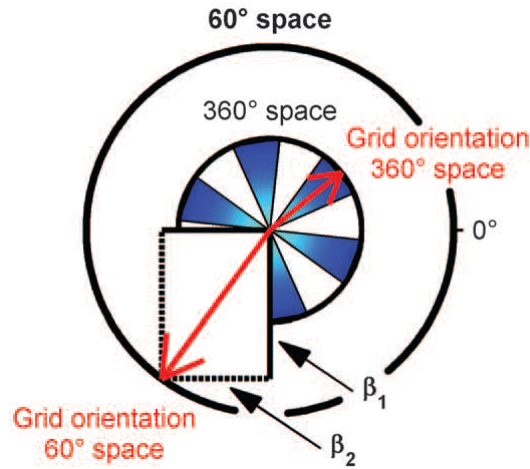


FIGURE 3.8 – Représentation du changement d’espace de 360° vers 60°, pour obtenir des régresseurs sensibles aux activations montrant une symétrie sextuple. La coloration bleu-blanche distingue les mouvements alignés et non-alignés [4]

de départ (360°).

3.6.2 Magnitude des Grids-cells

Un deuxième GLM est appliqué sur la deuxième moitié des données. Ce modèle est caractérisé par deux régresseurs. Le premier représente les mouvements alignés avec l’orientation moyenne de la grille, c’est à dire que la différence est de $\pm 15^\circ$. Le deuxième représente les mouvements non-alignés. La magnitude des Grids-cells est estimé par les moyennes de ces régresseurs. Cela signifie donc qu’une valeur positive indique une magnitude plus prononcée et inversement si la valeur est négative.

Un programme, nommé GridCat, a été développé pour analyser les Grids-cells [20]. Ce dernier a été choisi pour l’analyse de ce mémoire. Son fonctionnement est développé ci-dessous.

3.6.3 Choix des analyses

La première étape dans l’analyse des Grids-cells consiste à estimer l’orientation moyenne de la grille à l’aide du GLM1. Pour cela, la première moitié des données est utilisée. Seuls les participants dont une orientation cohérente a été obtenue sont utilisés pour la suite des analyses. La cohérence de l’orientation est évaluée à l’aide d’un test de Rayleigh.

Comparaison des sous-groupes

En fonction des résultats de l'orientation de la grille, des comparaisons entre sous-groupes peuvent être réalisées. Ces dernières peuvent être réalisées en fonction du sexe, de l'âge et la génétique (APOE).

Influences des repères

Afin de tester l'influence des repères lors de l'analyse des Grids-cells, on peut distinguer différentes situations :

- **LPI & PPI** : Dans ce cas, les deux GLMs sont réalisés sur l'ensemble des données. L'orientation de la grille ainsi que l'amplitude de réponse des Grids-cells sont évaluées sur les essais LPI et PPI.
- **LPI** : L'analyse est réalisée uniquement sur les données relatives aux essais LPI. C'est à dire que les images et les données du jeu relatives aux essais PPI ne sont pas prises en compte.
- **PPI** : Même analyse que la précédente mais en conservant uniquement les données relatives aux essais PPI.

Stabilité des orientation des voxels

La stabilité des voxels entre les runs est analysée. Une valeur de stabilité est extraite pour chaque comparaison entre deux runs. Ces valeurs permettent de procéder à une analyse de stabilité moyenne par sujet. Cette moyenne est ensuite comparée entre les trois situations décrites ci-dessus.

Amplitude de réponse des Grids-cells

L'analyse de la différence de réponse entre les mouvements alignés et non-alignés permet de mettre en évidence l'activité des Grids-cells. L'influence des repères est également analysée en comparant les trois situations.

3.6.4 Validation statistique

Dans le but de vérifier les résultats obtenus, une validation statistique est réalisée. Afin d'attester s'il existe des différences significatives entre les différents échantillons, il faut définir le seuil de signification (α). Il correspond à une probabilité de rejeter l'hypothèse nulle si $p\text{-valeur} \leq \alpha$. Dans ce mémoire, la valeur α choisie correspond au seuil le plus souvent utilisé : $\alpha = 0.05$. Cela correspond à une probabilité de 5% de rejeter l'hypothèse nulle alors que celle-ci est vraie [21].

Dans ce mémoire, les tests suivants sont utilisés :

- **Test t** : Le Test t est une méthode statistique utilisée pour comparer les moyennes de deux groupes présentant des différences significatives. Il évalue la différence entre les moyennes par rapport à la variabilité au sein des groupes. Après le calcul d'une statistique t, une p-valeur est obtenue, indiquant la probabilité que les résultats soient purement aléatoires. Si la valeur p est inférieure à α , on rejette l'hypothèse nulle (égalité des moyennes) en faveur d'une différence significative. Dans le cas contraire, aucune conclusion n'est tirée [21].
- **ANOVA** : L'analyse de variance (ANOVA) est une méthode statistique utilisée pour déterminer si les moyennes de plusieurs groupes sont statistiquement différentes. Elle compare les variations observées à l'intérieur de chaque groupe avec les variations entre ceux-ci. L'ANOVA ne permet pas d'identifier spécifiquement les groupes qui diffèrent, c'est pourquoi des tests post-hoc doivent être utilisés pour cela. Par exemple, des Tests t peuvent être réalisés pour déterminer si deux groupes diffèrent [21].
- **Rayleigh z test** : Ce test mesure la cohérence des données dans différentes directions. Si les données sont uniformément réparties, les vecteurs unitaires s'annuleront mutuellement, entraînant une faible valeur du test. Si les données sont fortement groupées dans certaines directions, le test de Rayleigh donne une valeur élevée. Une p-valeur, inférieure à α , indique que les données sont significativement non-uniformes. Dans ce cas, il existe une tendance directionnelle significative. Ce test est utilisé pour s'assurer de la cohérence d'orientation entre les voxels de la région d'intérêt [22].

3.7 Programme GridCat

Grid Code Analysis Toolbox (GridCat) est un programme open-source développé dans le cadre d'une thèse par le Dr M.Stangl [20]. Il permet d'analyser le grid code à partir d'IRMf acquises lors d'une tâche de navigation. Il reprend toutes les étapes, de l'estimation, en passant par les deux GLM, la génération des métriques et la visualisation des résultats. L'interface qui apparaît, à son lancement dans Matlab, est visible à la figure 3.9. Les différents champs à remplir sur cette figure sont développés ci-dessous.

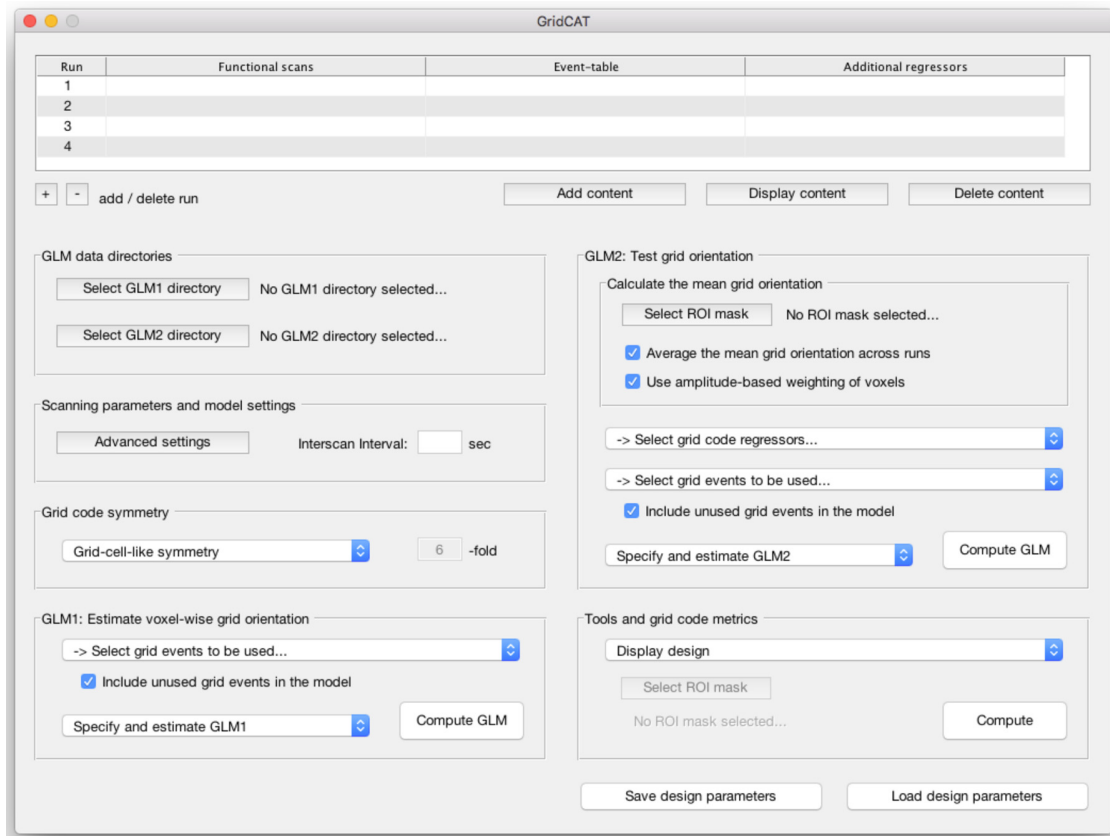


FIGURE 3.9 – Interface graphique Matlab du programme GridCat [20]. Il permet une analyse des Grids-cells à partir de données, paramètres et réglages spécifiques. Son fonctionnement repose sur l’analyse des données, deux GLM et la visualisation des résultats.

3.7.1 Données en entrée

Afin de procéder à une analyse spécifique à notre expérience, il faut introduire différentes données spécifiques à chaque run. L'interface du programme étant en anglais, les différentes variables le seront également.

- **Functional scans** : Les images fonctionnelles doivent être fournies sous le format Nifti. Chaque run est caractérisé par une série de Nifti 3D par volume. Cela implique un changement de format car les images coregistrées sont stockées sous forme de fichier VTC. Ceux-ci seront donc transformés en plusieurs 3D-Nifti. Le preprocessing ainsi que la normalisation MNI doivent être réalisés avant l'utilisation de GridCat.
- **Event-table** : Fichier texte (.txt) décrivant les différents événements du jeu, qui sont synchronisés avec l'enregistrement de l'imagerie fonctionnelle. En effet, chaque ligne correspond à un événement. Ce dernier est caractérisé par un type (translation, rotation,...), un temps de départ, une durée et l'angle de direction du joueur pendant cette période. Un exemple d'Event-table spécifique à la tâche de ce mémoire est visible dans la section 4.2.
- **Additional regressors** : Cette entrée est optionnelle et permet d'ajouter des régresseurs supplémentaires, qui seront pris en compte dans le GLM. Par exemple, un fichier .txt comprenant les paramètres de réalignement des images fonctionnelles peut être inclus.

3.7.2 Spécifications

GLM data directories

Deux dossiers de sortie doivent être définis, un pour chaque GLMs. C'est dans ces dossiers que seront stockés les différents régresseurs calculés lors de l'exécution des modèles.

Scanning parameters

Le temps de répétition (TR) spécifique aux données doit être renseigné en secondes dans la case *Interscan Interval*. Il est également possible de spécifier d'autres paramètres, par exemple, pour l'application d'un filtre passe-haut.

Grid code symmetry

Cette section permet de sélectionner différentes symétries à étudier. La symétrie par défaut, est celle des Grids-cells, la sextuple. Ce paramètre permet de mettre en évidence leur présence ou non, car en comparant différentes symétries sur un

même ensemble de données, la sextuple doit être plus présente en cas d'activation des Grids-cells.

3.7.3 GLM1

Afin de générer le premier modèle linéaire généralisé, il faut décider comment seront séparés les données en un ensemble d'estimation et de test. Pour cela, différents choix sont possibles : utiliser la première ou la deuxième moitié ; les événements pairs ou impairs ou uniquement les runs pairs ou impairs.

Les événements sélectionnés pour le GLM1 peuvent être inclus (via `Include unused grids events`) ou non dans l'estimation sans leur angle correspondant, ce qui implique qu'ils ne contribueront pas à l'estimation de l'orientation de la grille.

Après avoir fixé les différents paramètres, le GLM1 peut être exécuté. L'orientation de la grille par voxel est estimée et les régresseurs sont calculés et sauves dans le dossier GLM1. Il est important de noter que les voxels dont la valeur est *NaN* (*Not a number*), sont ceux pour lesquels l'orientation de la grille n'a pas pu être estimée.

3.7.4 GLM2

ROI mask

La région d'intérêt peut être sélectionnée à l'aide d'un fichier Nifti contenant un masque de la région en question. A noter que ce masque doit être binaire et de mêmes dimensions que les images de la fonctionnelle. Après avoir fixé le masque, l'orientation moyenne de la grille dans la ROI est calculée. Dans le cadre de ce mémoire, c'est à ce moment que le masque pmEC ou alEC est utilisé.

Grid code regressors

Différents types de régresseurs sont disponibles dans le programme. Ils sont tous en lien avec l'orientation moyenne de la grille estimée avec le GLM1. Cependant, ils diffèrent en ce qui concerne les comparaisons avec l'orientation moyenne ou les événements. Les trois types de régresseurs sont :

- **Parametric modulation regressor** : Les angles des événements sont modélisés via leur alignement par rapport à l'orientation moyenne. Le régresseur varie de -1 (opposition) à 1 (alignement).
- **Aligned/misaligned regressors (1 each)** : Contrairement au Parametric, ce régresseur est binaire. En effet, chaque angle d'événement est comparé à l'orientation moyenne. Si la différence n'excède pas 15° par rapport à celle-ci

ou à un multiple de 60° , l'angle est considéré comme *Aligned* ou *Misaligned* sinon.

- **Aligned/misaligned regressors (multiple)** : Le principe est le même que pour le régresseur "1 each" mise à part que dans ce cas-ci, la condition alignée sera représentée par plusieurs régresseurs qui sont les multiples de 60° ($0, 60^\circ, \dots$). Ces derniers correspondent aux pics de signal attendus lors d'une symétrie sextuple. Dans le même raisonnement, la condition non-alignée sera modélisée par les angles décalés ($30^\circ, 90^\circ, \dots$) correspondants à un signal faible.

Grid events

Cette sélection d'événements est généralement automatique car elle contient la partie non-utilisée dans le GLM1.

3.7.5 Outils

La dernière partie du programme permet d'illustrer les différents régresseurs et résultats calculés lors des deux GLMs. Il est notamment possible d'illustrer l'orientation calculée pour chaque voxel ou encore leur stabilité d'orientation entre deux runs. Différents graphiques sont présentés dans la section 4.3.

Chapitre 4

Résultats

4.1 Scripts

Les différents scripts python ont été réalisés en collaboration avec Madame Judith Eck qui dirige le Support Desk de Brain Innovation. Après de nombreuses discussions, tests et modifications, les différentes étapes de traitements ont été codées de la manière suivante.

4.1.1 Compatibilité BIDS

Les deux premiers scripts servent à transformer la structure initiale, le format et le nommage des fichiers vers un projet BIDS compatible. Il est important de noter que ces deux scripts peuvent être exécutés sans utiliser BrainVoyager.

NIIfromDCMfolders.py

Ce script considère que les données sont arrangées selon les fichiers visibles à la Figure 4.1.a. Il peut être aisément adapté pour un autre nommage de fichier, par exemple, le patient pourrait-être identifié par "Patient-4" au lieu de "UCL2_4" dans l'exemple montré ici. En ce qui concerne l'organisation des fichiers, les données sont stockées dans un dossier rawdata ce qui est imposé par BIDS. On remarque également un regroupement des images en fonction de leur nature, à savoir anatomique (anat) ou fonctionnelle (func) et non plus par run. Le numéro du patient est rendu BIDS compatible en le renommant par "sub-04". Pour ce qui est du format, les Dicoms sont maintenant regroupés en un seul NIfTI par anatomie (Voir Figure 4.1.c) et un NIfTI par run dans le dossier func (Voir Figure 4.1.d). Pour traiter plusieurs patients à la fois, il suffit d'entrer leur numéro, dans ce cas-ci le 4 et le script détectera automatiquement le nombre de runs pour chaque patient.

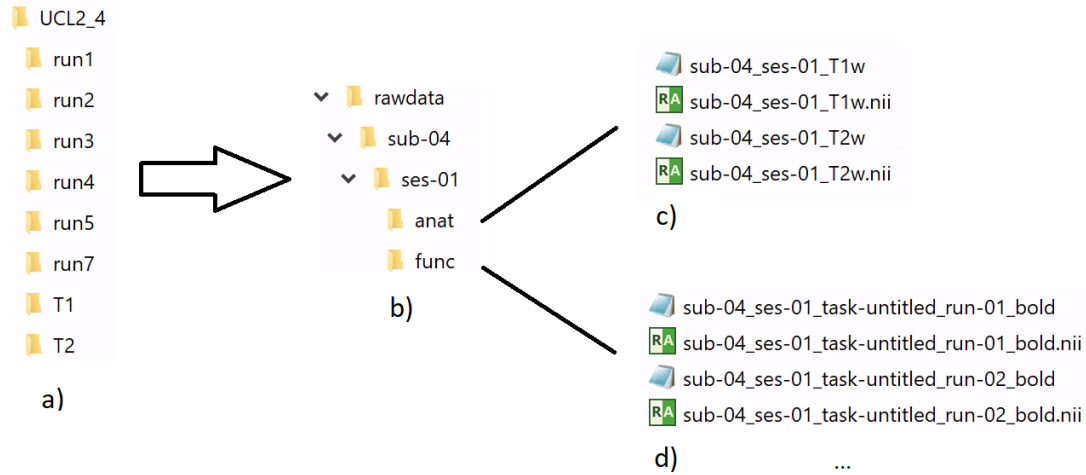


FIGURE 4.1 – Exécution du script *NIfTIfromDCMfolders.py* sur un sujet type (a). Les sous-dossiers créés (b) sont stockés dans le dossier rawdata. Après la conversion, les fichiers anatomiques (c) et fonctionnels (d) sont maintenant séparés et stockés dans le format NIfTi.

ConvertPRTsInCurrentFoldertoTSV.py

Ce script sert à transformer les protocoles dans le format adéquat du standard BIDS. Le format cible est un "Tab Separated Values" dit tsv. Le nom d'origine est conservé, seul le format est modifié.

4.1.2 Data Analysis Manager

Après avoir assuré la compatibilité BIDS du projet, le traitement des données peut commencer. Afin de se familiariser avec l'organisation du projet, il existe une interface dans BrainVoyager appelée Data Analysis Manager (voir Figure 4.2). Comme on peut le voir, celle-ci est très intuitive et permet aisément de visualiser les sujets, les étapes de traitements ainsi que les liens entre celles-ci. Chaque étape est assimilée à un workflow et est caractérisée par un type, un input et un output. L'avantage est que les workflows peuvent être soit exécutés directement grâce à l'interface soit par un script dans BrainVoyager. Cela permet donc de combiner et compléter l'interface avec des scripts plus complets, plus spécifiques comme ce sera le cas dans ce mémoire. Par exemple, les deux premiers workflows seront exécutés à l'aide de l'interface tandis que les autres seront réalisés par scripts en externe.

Les différents projets ainsi que leur liste de sujets sont visibles sur la gauche de l'image. A l'aide de la commande *Run All* ou *Run Selected* présentent au milieu de l'image, un workflow peut-être exécuté sur tous les sujets ou seulement sur une

sélection. Pour ce qui est des données, les input et les output peuvent être fixés à l'aide de la commande *Connect* sur la droite de l'image. Celle-ci, ouvrira une fenêtre permettant de choisir les données sources du workflow. Par exemple, on peut voir sur la Figure 4.2 que l'input pour l'anatomical preprocessing sera les T1w et que les output seront caractérisés par le suffix T1W auquel sera ajouté un suffix spécifique au preprocessing réalisé (IIHC,...).

ID	Type	Name	Input	Output	Subjects
6	Anatomical Preprocessing	anat-preprocessing	rawdata: T1w	T1w_(anatpp_6)	ALL
2	Anatomical Normalization	anat-normalization	workflow 6: T1w_(anatpp_6)	T1w_(anatpp_6)_(anatr_2)	ALL
3	Functional Preprocessing	func-preprocessing	rawdata: ses-01_task-untitled_bold	ses-01_task-untitled_bold_(funcpp_3)	ALL
4	Coregistration	func-to-anat-coreg	wf 3: ses-01_task-unt...wf 6: T1w_(anatpp_6)	ses-01_task-untitled...natpp_6)_(coreg-ia_4)	ALL
5	Functional Normalization	func-normalization	workflow 3: ses-01_ta...itled_bold_(funcpp_3)	ses-01_task-untitled..._(funcpp_3)_(funcn_5)	ALL

FIGURE 4.2 – Screen Capture of Data Analysis Manager for this thesis.

De plus, le traitement des données anatomiques y est facilement réalisable. Le preprocessing et la normalisation anatomiques seront donc réalisés à l'aide de cette interface.

Anatomical preprocessing

Comme le montre la figure 4.3, l'interface permet de fixer différents paramètres pour préprocesser les images anatomique. Après les avoir choisis, le workflow peut être exécuté sur un ou plusieurs sujets. Le résultat sera stocké dans un fichier cible respectant le format et le nommage BIDS (Voir Figure 4.4).

Anatomical normalization

Cette étape fonctionne de la même manière que l'étape du preprocessing. Elle permet d'effectuer une normalisation vers l'espace de référence souhaité qui doit être choisi dans les paramètres.

4.1.3 Functionnal Scripts

Cette section regroupe les différents scripts utilisés pour le traitement des données d'imagerie fonctionnelle.

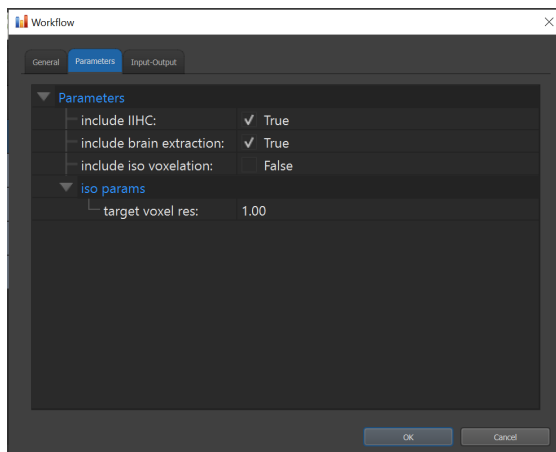


FIGURE 4.3 – Paramètres disponibles pour le preprocessing anatomique. On y retrouve la correction d’inhomogénéité (IIHC), l’extraction du cerveau ainsi que l’iso voxelation qui permet de transformer les voxels en voxels 1x1x1mm. Il est également possible de fixer la taille des voxels finaux via la variable **target voxel res**.^[12]

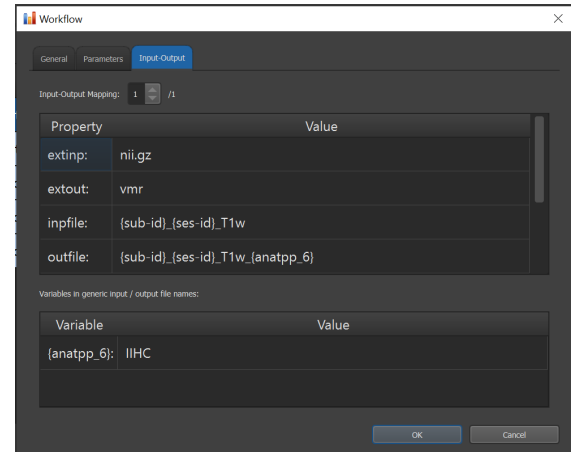


FIGURE 4.4 – Format et nommage des fichiers sources et finaux. Le nommage final(**anatpp_6**) dépend du preprocessing effectué, dans ce cas-ci on peut voir qu’il sera égal à IIHC car une correction d’inhomogénéité sera appliquée.

FMRsfromNII.py

Pour commencer, les fichiers .fmr sont créés à partir des fichiers NIfTis(.nii) pour chaque run de chaque patients. Afin d'assurer le bon fonctionnement du script, plusieurs variables doivent être fixées :

- **project_path** : Le chemin vers les données doit être indiqué dans cette variable.
- **subjects** : Tableau où il faut y indiquer le numéro de chaque sujet à analyser, le script détectera ensuite le nombre de runs effectués par sujet.
- **sessions** : Tableau qui regroupe le numéro de chaque session d'acquisition. Dans ce mémoire, tous les runs ont été acquis au cours d'une même et unique session. Le tableau est donc **sessions** = [1].
- **task_name** : Variable indiquant le nom de la tâche.

Les différents fichiers seront stockés dans des sous-dossiers spécifiques pour chaque patient, chaque session et selon la nature des données. Par exemple, les imageries fonctionnelles seront stockées selon ce chemin : *project_path+'derivatives/rawdata_bv/'+sub_id+'/' + ses_id+' /func/' + sub_id+'_' + ses_id+'_' + task_name+'_' + run_id+'_' + bold.fmr'*. On peut remarquer les différents sous-dossiers : derivatives, rawdata_bv et func, ceux-ci sont créés automatiquement s'ils sont inexistantes. Le dernier sous-dossier est func pour y stocker les fmr. Pour les images anatomiques, il existe un sous-dossier équivalent appelé anat. La présence des variables des sous-dossiers sub_id et ses_id montre que les images sont stockées en fonction du numéro de patient et de session. Par contre, tous les runs sont stockés dans le même dossier mais ils sont nommés différemment en fonction du numéro de run (run_id).

FMRPreprocessing.py

Afin de procéder au preprocessing des données, il faut définir les différentes étapes que l'on souhaite exécuter sur les images fonctionnelles acquises :

- **Motion correction** : Si l'on souhaite procéder à une correction de mouvement, il faut fixer le booléen **perform_moco** à True. Si cette correction s'effectue directement sur les images fonctionnelles d'origine(*_bold.fmr*), la variable **input_moco** est fixée à un String vide (' '). Sinon, cette variable doit être modifiée pour correspondre au nommage des images sélectionnées. Pour caractériser le type de motion correction, il faut définir plusieurs références. Pour commencer, le booléen **perform_intrasession** permet d'aligner plusieurs runs acquis lors d'une même session sur une même ré-

férence. En effet, la variable **moco_ref_volume** spécifie le volume et **moco_ref_run** le run de référence. Il est également possible de spécifier une tâche de référence avec la variable **moco_ref_run** si le contexte s’y prête.

- **Slice scan Time correction** : Le booléen **perform_ssc** doit être égal à True pour exécuter la correction. Lorsque l’on exécute plusieurs corrections à la suite, il faut faire attention d’adapter la variable **input_ssc**. Par exemple, si au préalable, la correction des mouvements a déjà été réalisée, il faut y indiquer *_3DMCTS*. Cela permet d’effectuer la correction de slice sur les images dont les mouvements ont déjà été corrigés.
- **High Pass Filtering** : Le booléen **perform_hpf** doit être égal à True pour exécuter le filtrage. A nouveau, il faut faire attention d’adapter la variable **input_hpf**. Si les corrections des mouvements ainsi et que de l’ordre des slices ont déjà été effectuées, la variable devient *_3DMCTS_SCCTBL*. Par contre, si seule la correction des mouvements a déjà été réalisée, la variable est égale à *_3DMCTS*. Enfin, la variable **n_cycles** permet de spécifier le nombre de cycles
- **Spatial Smoothing** : Comme précédemment, il faut adapter le booléen **perform_smoothing** ainsi que la variable **input_smoothing** en conséquence. Deux variables supplémentaires **gauss_fwhm** et **fwhm_unit** caractérisent respectivement the full-width at half maximum (fwhm) et ses unités.

GetPlots.py

Ce script permet de générer le graphique représentant les mouvements de la tête. Son analyse est utile pour rejeter les sujets qui ont trop bougé. Les paramètres dépassant les thresholds, fixés au préalable, ne sont pas inclus dans l’analyse. Dans notre cas, un threshold de 4mm et de 4 degrés sont utilisés pour les translations et pour les rotations. Lors de l’exécution de ce script, une fenêtre s’ouvre et il suffit de sélectionner le patient ou le run pour lequel une analyse de mouvement est souhaitée.

4.1.4 Coregistration

Les étapes de coregistration sont divisées en trois scripts. Le premier permet de calculer les transformations nécessaires pour aligner les données fonctionnelles sur les données anatomiques. Le second permet la création des VTCs et le dernier permet, si nécessaire, de réaliser un préprocessing sur les VTCs. Les différents paramètres et variables sont décrites ci-dessous :

Alignment.py

On y retrouve les paramètres classiques déjà développés ci-dessus : `project_path`, `subjects`, `sessions` et `task_name`. Après cela, on retrouve les chemins vers les différents dossiers où se trouvent les ressources nécessaires :

- **anat_prep** : Chemin vers le dossier contenant les données anatomiques préprocessées.
- **func_prep** : Chemin vers le dossier contenant les données fonctionnelles préprocessées.
- **coreg_out** : Chemin vers le dossier de sortie pour les fichiers créés.

Ensuite, les différents suffix sont définis. Si l'analyse réalisée est semblable à celle de ce mémoire et que le nommage des fichiers a été respecté, il se peut que les variables ne doivent pas être modifiées.

- **anat_suffix** : Suffix représentant les données anatomiques. Par exemple, `_T1w_IIHC.vmr` si les images anatomiques ont été soumis à une correction d'inhomogénéité.
- **func_suffix** : Suffix représentant les données fonctionnelles. Par exemple, `_bold_3DMCTS_SCCTBL.fmr` si la correction des mouvements(3DMCTS) et le slice scan time correction(SCCTBL) ont été réalisés lors du preprocessing fonctionnel.

Pour finir, il reste à définir si BBR l'on souhaite utiliser pour coregistrer. Si c'est le cas, il faut fixer la variable `use_BBR` à True.

Le résultat fourni est la création d'un fichier d'alignement initial (IA.trf) et final (FA.trf). Le nom complet des fichiers reprend le nom de la fonctionnelle ainsi que de l'anatomique sur laquelle elle a été alignée. Un fichier trf permet d'aligner des images anatomiques avec des images fonctionnelles. On y retrouve les informations sur les translations, les rotations et éventuellement les échelles nécessaires pour l'alignement.

VTC_creation.py

Après avoir calculé les alignements, la création des VTCs peut être réalisée. Deux chemins supplémentaires peuvent être nécessaires :

- **vtc_folder** : Chemin vers le dossier cible pour stocker les VTCs en MNI.
- **vtc_native** : Chemin vers le dossier cible pour stocker les VTCs dans l'espace native.

Le choix de l'espace cible se réalise grâce à la variable **mni** qui est fixée à True si l'espace MNI est choisi et false pour le native space. Si un seul des deux espaces est choisi, il n'est donc pas nécessaire de référencer l'autre chemin et cette variable peut-être mise en commentaire.

La caractérisation de la coregistration se fait à l'aide des variables suivantes :

- **res** : Ce paramètre permet de fixer la résolution en mm^2 . En général, trois valeurs sont possibles à savoir 1, 2 et 3 mm^2 . La valeur adéquate est choisie en fonction de la taille des voxels de la fonctionnelle. Dans notre cas, **res=2**.
- **interpol** : Ce paramètre définit le choix de la technique d'interpolation. La valeur 1 équivaut à une interpolation Nearest neighbor. L'interpolation trilineaire correspond à la valeur 2. Et enfin, pour une interpolation sinc, il faut choisir la valeur 3 [12].
- **coreg_BBR** : A fixer à True, si l'on souhaite utiliser BBR pour la coregistration, false sinon.
- **use_intrasesscoreg** : La valeur True indique que la coregistration intra session sera réalisée. C'est à dire que les images acquises au sein d'une même session pour un même sujet seront alignées.
- **intensity_thresh** : Option recommandée pour les données d'imagerie fonctionnelle. En fixant ce seuil d'intensité à 100, cela permet de détecter efficacement les valeurs d'intensité des données fonctionnelles dans le tissu cérébral. En effet, celles-ci sont supérieures à 100.

En ce qui concerne les accès aux données, il faut à nouveau indiquer les suffix pour la fmr et l'anatomique (comme ci-dessus pour Alignement.py). Les suffix des alignements doivent être également renseignés. A noter que si la coregistration BBR est utilisée le suffix de l'alignement final passera de *_FA.trf* à *_BBR_FA.trf*.

L'interface Data Analysis Manager fournit une bonne visualisation générale du projet. En effet, à partir de l'option Data flow, on obtient la figure 4.5 qui illustre les différents liens entre les étapes exécutées.

VTC_preprocessing.py

Il se peut que certaines imperfections doivent être corrigées dans les VTCs. Pour cela, plusieurs étapes de preprocessing peuvent être réalisées :

- **Temporal High pass filtering**
- **Temporal smoothing**
- **Spatial smoothing**

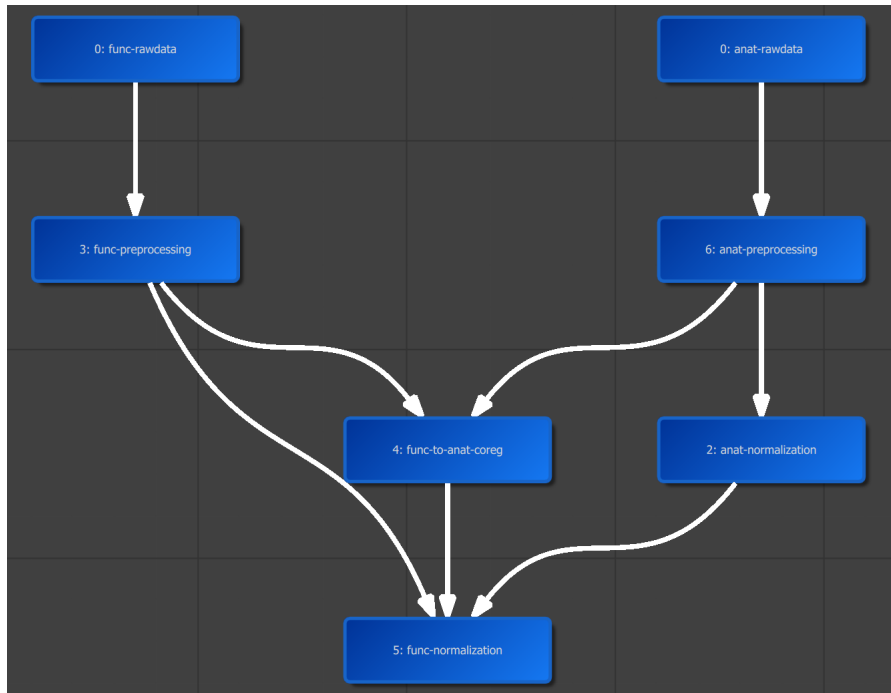


FIGURE 4.5 – **Data Flow Visualization** : Mind map showing the links between the various scripts, from the source data to the creation of the VTCs.

Comme expliqué dans le fmri preprocessing, il faut fixer les différentes variables liées au filtrage et au smoothing.

LinkPrt.py

Après la création des VTCs, ce script permet d’y lier un protocole. A nouveau, il faut spécifier les variables classiques telles que `project_path`, `subjects`, `sessions` et `task_name`. Afin que le script puisse lier les protocoles aux vtc, il faut spécifier les variables suivantes :

- **file_ending** : Suffix représentant le vtc. Par exemple, pour ce mémoire : `'_bold_3DMCTS_SCCTBL_256_sinc_2x1.0_MNI.vtc'`.
- **anat_folder** : Chemin vers le dossier contenant la T1.
- **anat_ending** : Suffix représentant la T1. Par exemple, dans le cas d’une T1 en MNI avec IHC : `'_T1w_IHC_MNI.vmr'`.
- **vtc_folder** : Chemin vers le dossier contenant les vtc.
- **prt_folder** : Chemin vers le dossier contenant les protocoles (.prt).

4.2 Protocole

Le script crée un fichier Excel intermédiaire où l'on retrouve toutes les données du fichier original castées dans le format adéquat (Voir Figure 4.6). Les variables supplémentaires, comme la vitesse, la variation angulaire y sont également sauveés.

time	dt	v	dyaw	yaw	phasetype	feedback	x	vx	y	vy	z	subtask
0	0,196	0	0	180	0	-1	0	0	0	0	203,91	1
0,196	0,199	0	0	180	0	-1	0	0	0	0	206,474	1
0,395	0,2	0	0	180	0	-1	0	0	0	0	206,474	1
0,595	0,2	0	0	180	0	-1	0	0	0	0	206,474	1
0,795	0,2	0	0	180	0	-1	0	0	0	0	206,474	1
0,995	0,2	0	0	180	0	-1	0	0	0	0	206,474	1
1,195	0,2	0	0	180	0	-1	0	0	0	0	206,474	1
1,395	0,2	0	0	180	0	-1	0	0	0	0	206,474	1
1,595	0,2	0	0	180	0	-1	0	0	0	0	206,474	1
1,795	0,2	0	0	180	0	-1	0	0	0	0	206,474	1
1,995	0,2	0	0	180	0	-1	0	0	0	0	206,474	1

FIGURE 4.6 – Fichier Excel intermédiaire utilisé pour la création d'un fichier protocole. On y retrouve les variables classiques telles que : *time* (temps), *v* (vitesse), *yaw* (position angulaire). La variable *subtask* permet d'identifier la sous-tâche réalisée à ce temps. Si la valeur est 3, c'est une tâche avec le phare (LPI). En absence de repères (PPI), sa valeur est 1. Cette variable est donc utilisée pour distinguer les translations entre les figures 4.7 et 4.8.

Ensuite, ce fichier Excel est structuré vers un fichier de sortie en fonction de son utilisation future. Par exemple, dans ce mémoire, il doit être reconnu par le programme GridCat et avoir le format illustré à la figure 4.7. Une version plus détaillée est visible à la figure 4.8 car les situations avec (LPI) ou sans phare (PPI) sont dissociées. Il est donc possible d'étudier les différences d'activations des Grids-cells en présence de repères ou non.

4.3 Grids-cells

Dans cette section, les résultats obtenus avec le programme GridCat sont présentés.

4.3.1 Orientation de la grille

La première étape dans la détection des Grids-cells est d'estimer une orientation moyenne de la grille à partir de l'orientation de chaque voxel. Cette étape est réalisée lors du GLM1.

L'estimation de l'orientation est performée sur les 9 participants jeunes et sur 9 participants âgés. Si l'orientation d'un voxel n'a pas pu être estimée, sa valeur est

```

vnull;6.795;7.6;180.0
translation;14.395;1.0;290.0
translation;15.395;0.2;285.0
translation;15.595;0.2;280.0
translation;15.795;1.8;275.0
vnull;17.595;4.2;275.0
translation;21.795;2.0;150.0
translation;23.795;0.8;145.0
translation;24.595;0.2;140.0
translation;24.795;1.401;135.0
vnull;26.196;7.999;135.0
translation;34.195;5.4;285.0
feedback;39.595;2.0;285.0

```

FIGURE 4.7 – Tableau d’événements sous forme de Fichier txt compatible avec GridCat.

```

vnull;6.795;7.6;180.0
translationPPI;14.395;1.0;290.0
translationPPI;15.395;0.2;285.0
translationPPI;15.595;0.2;280.0
translationPPI;15.795;1.8;275.0
vnull;17.595;4.2;275.0
translationPPI;21.795;2.0;150.0
translationPPI;23.795;0.8;145.0
translationPPI;24.595;0.2;140.0
translationPPI;24.795;1.401;135.0
vnull;26.196;7.999;135.0
translationPPI;34.195;5.4;285.0
feedback;39.595;2.0;285.0

```

FIGURE 4.8 – Tableau d’événements sous forme de Fichier txt compatible avec GridCat. Contrairement à la figure de gauche, les événements de translation sont distinguées en fonction de la présence de repères ou non.

égale à *Not a Number (NaN)*. Dans GridCat, l’orientation estimée de chaque voxel est illustrée par un histogramme polaire. La figure 4.9 montre l’orientation moyenne estimée à l’aide d’une flèche verte. Cette flèche est la moyenne des histogrammes bleus. En effet, chaque histogramme représente les voxels partageant la même orientation. La figure 4.10 montre l’histogramme d’un participant où, seulement 3 voxels sur les 88 correspondants au pmEC, ne sont pas des NaN. Contrairement à la figure 4.9, les histogrammes ne présentent pas une cohérence d’orientation et cela est vérifié par une p-valeur égale à 0.196.

La comparaison du pourcentage de NaN, entre les participants jeunes et vieux, est illustrée à la figure 4.11. Sur ce boxplot, deux valeurs sont situées hors des limites pour les jeunes (cercle vert) et une pour les personnes âgées (cercle rouge). Dans le cas des jeunes, les p-valeurs associées sont supérieures à 0.05. Cela signifie que l’orientation estimée de la grille n’est pas cohérente. Pour ce qui est de la valeur hors-limites chez les personnes âgées, la p-valeur est inférieure à 0.05. Celle-ci, correspond donc à un participant dont l’orientation de la grille a pu être estimée. C’est en fait l’histogramme de ce dernier qui est illustré à la figure 4.9.

La suite des analyses se poursuit uniquement avec les participants dont l’orientation de la grille est cohérente. C’est le cas pour 1 seul participant sur les 18 analysés. Les résultats suivants proviennent donc uniquement de ce dernier.

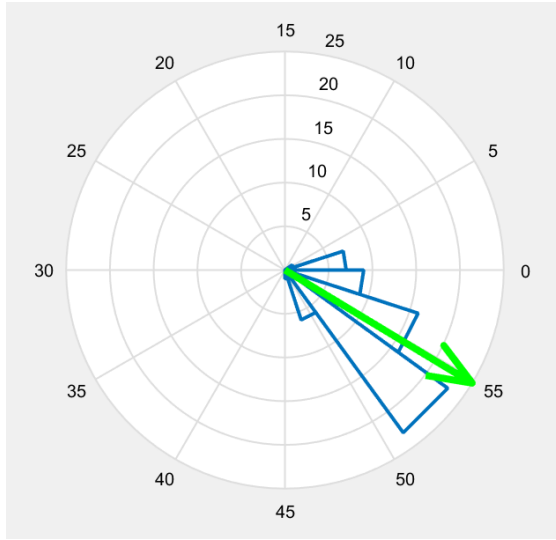


FIGURE 4.9 – Histogramme polaire montrant la cohérence entre les voxels dans le pmEC. L'orientation moyenne de la grille est représentée par la flèche verte. Sa cohérence est vérifiée à l'aide d'un test de Rayleigh calculé directement dans GridCat (p-valeur<0.001).

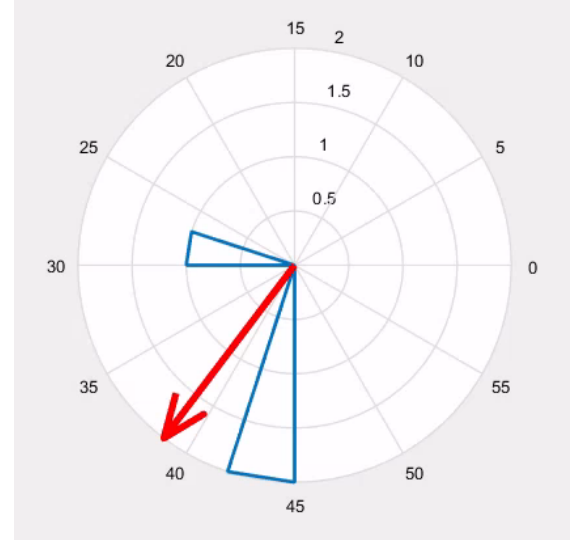


FIGURE 4.10 – Histogramme polaire montrant la non-cohérence entre les voxels dans le pmEC. L'orientation moyenne de la grille (flèche rouge) n'est pas cohérente (p-valeur=0.196).

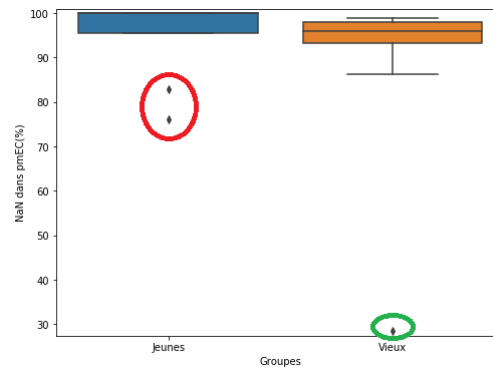
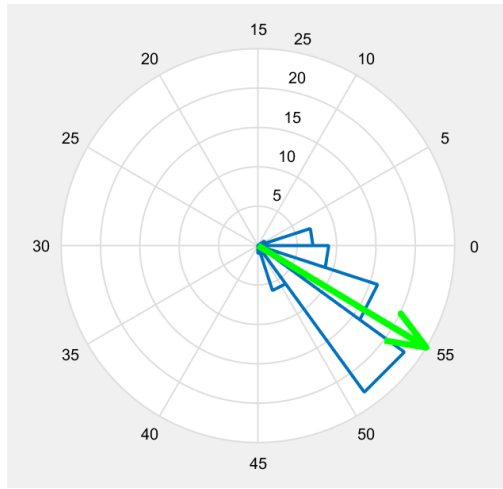


FIGURE 4.11 – Boxplot qui compare la proportion de NaN dans le pmEC entre les sujets jeunes et vieux. Les deux cercles de couleurs mettent en évidence les valeurs hors-limites.

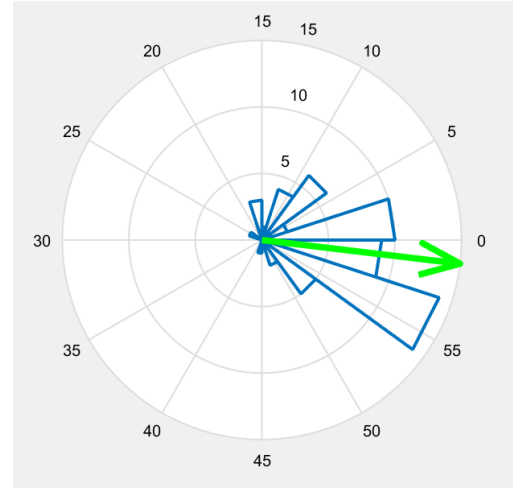
4.3.2 Comparaison entre pmEC et alEC

Orientation de la grille

La cohérence d'orientation peut être comparée entre plusieurs ROIs en introduisant les deux masques (pmEC et alEC), tour à tour, dans le programme. Un histogramme polaire est fourni, pour chaque région, accompagné du nombre de NaN, de l'orientation moyenne (flèche) et d'une p-valeur attestant de la cohérence entre voxels. La comparaison entre les deux régions est visible à la figure 4.12.¹ Dans les deux cas, une cohérence d'orientation a été observée entre les voxels avec des p-valeurs inférieures à 0.05. Une orientation a pu être estimée pour 72% des voxels dans le pmEC (Figure 4.12a) contre 25% dans l'alEC (Figure 4.12b).



(a) Cohérence de l'orientation estimée de la grille dans le pmEC. Une orientation a pu être estimée pour 72% des voxels.



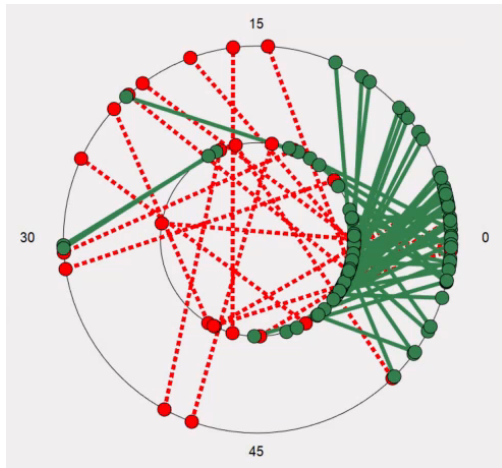
(b) Cohérence de l'orientation estimée de la grille dans l'alEC. Une orientation a pu être estimée pour 25% des voxels.

FIGURE 4.12 – Comparaison de l'orientation estimée de la grille entre le pmEC et l'alEC.

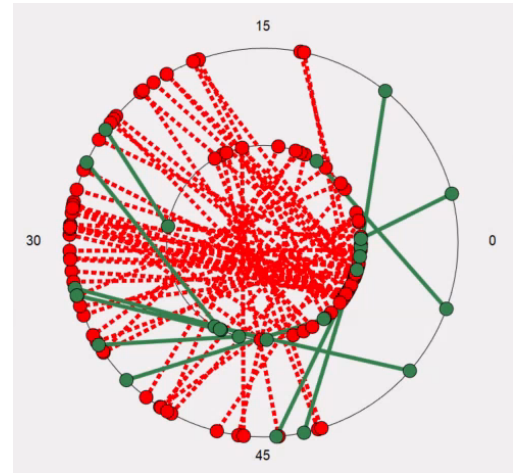
4.3.3 Stabilité d'orientation des voxels

La stabilité des orientations des voxels, entre deux runs, peut être observée directement dans Gridcat. On remarque que la stabilité est supérieure sur la figure 4.13a par rapport à la figure 4.13b.

1. La figure 4.12a est identique à la figure 4.9. Elle est dupliquée pour faciliter la lisibilité lors des comparaisons.



(a) Stabilité des voxels entre le run1 et le run2 dans le pmEC. Stabilité=75%

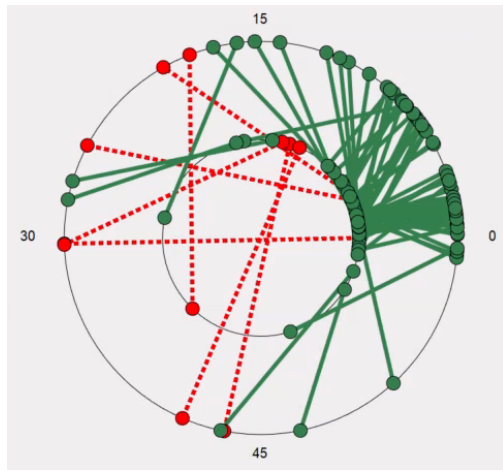


(b) Stabilité des voxels entre le run1 et le run3 dans le pmEC. Stabilité=19%

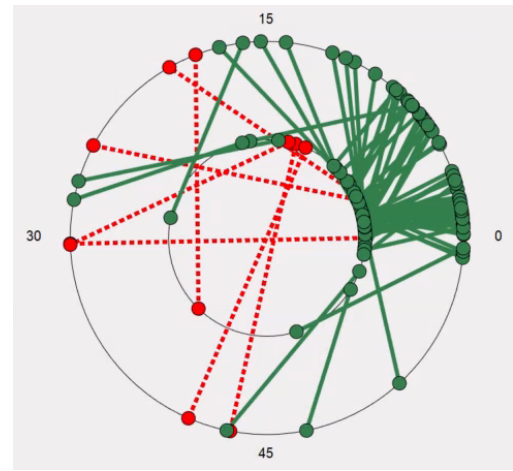
FIGURE 4.13 – Graphiques polaires montrant la cohérence de l'orientation des grilles entre les voxels, entre deux runs, dans la région d'intérêt choisie. Les deux anneaux noirs représentent les deux runs différents (anneau intérieur : run 1, anneau extérieur : run 2), et l'orientation pour chaque voxel est indiquée avec un point. Une ligne de couleur relie les orientations de chaque voxel. Les lignes vertes pleines indiquent les voxels avec des orientations stables, tandis que les lignes rouges en pointillés indiquent les voxels avec des orientations instables.

Influence des repères

Lors du GLM1, l'orientation de la grille est estimée sur la première moitié des données. L'autre moitié sert d'ensemble test. Or, lors des acquisitions, l'ordre des essais est de 4 LPI puis 4 PPI ou inversement. Ceci implique que soit l'orientation est estimée sur 4 LPI et évaluée sur 4 PPI ou inversement. Pour mettre en évidence l'influence des repères, une analyse est effectuée uniquement sur les LPI, en supprimant les images des essais PPI. La même démarche est réalisée pour les essais PPI. Les stabilités, entre le run1 et le run2 ainsi qu'entre le run1 et le run3, sont à nouveau comparées. On remarque sur les figures 4.14a et 4.14b que les voxels semblent plus stables entre les différents runs contrairement à ce qui a été observé à la figure 4.13.



(a) Stabilité des voxels entre le run1 et le run2 dans le pmEC. Stabilité=90%



(b) Stabilité des voxels entre le run1 et le run3 dans le pmEC. Stabilité=82%

FIGURE 4.14 – Comparaison de la stabilité des voxels entre deux runs avec les essais LPI uniquement.

La figure 4.15 contient 3 boxplots. Le premier représente les valeurs de stabilité entre runs pour les essais LPI et PPI confondus. Le deuxième représente la stabilité pour les LPI et le troisième pour les PPI. Pour ce participant, 5 runs ont été enregistrés. En comparant les runs, deux à deux, on obtient 10 valeurs de stabilité par boxplot. Pour s'assurer qu'il y a bien une différence de stabilité entre les 3 groupes, un test ANOVA est réalisé. La p-valeur obtenue est égale à 0.053. Afin d'attester la différence entre au moins deux des sous-ensembles, des Tests t sont réalisés deux à deux. Les résultats sont visibles dans le tableau 4.1. On remarque que lorsque l'analyse est seulement performée sur des essais LPI, la stabilité des voxels est meilleure. L'influence des repères, sur l'amplitude de réponse des Grids-cells, est également étudiée ci-dessous.

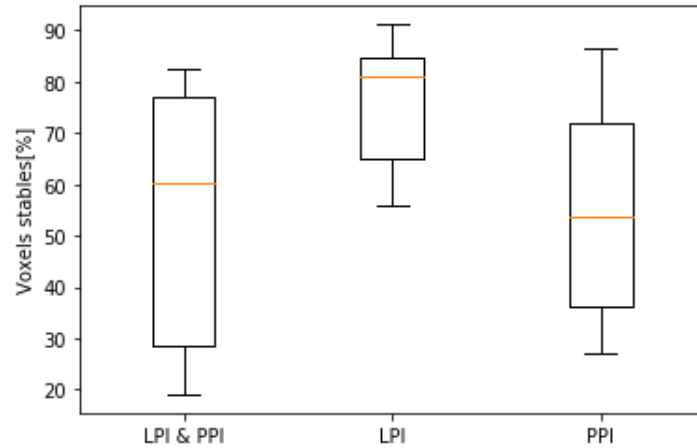


FIGURE 4.15 – Comparaison de la stabilité des voxels pour les orientations estimées pour tous les essais (LPI & PPI), pour ceux uniquement avec phare (LPI) et uniquement sans-phare (PPI).

TABLE 4.1 – Comparaison des p-valeurs entre les groupes LPI & PPI, LPI et PPI

Groupes	LPI & PPI	LPI	PPI
LPI & PPI	–	0.028	0.87
LPI		–	0.026
PPI			–

4.3.4 Amplitude de réponse des Grids-cells

Le programme fournit une valeur de réponse des Grids-cells en comparant les translations alignées et non-alignées avec l'orientation de la grille. Dans un premier temps, les événements avec phare et sans-phare ne sont pas séparés. Les amplitudes de réponse pour les 5 runs du participant agé sont visibles à la figure 4.16. Un Test t est réalisé afin de déterminer si la moyenne des amplitudes diffère de zéro. La p-valeur obtenue est 0.35. Il n'y a donc pas de différences d'amplitude de réponse entre les mouvements alignés et non-alignés.

Influence des repères

La même analyse peut être réalisée en distinguant les mouvements avec phare (LPI) et sans phare (PPI). Sur la figure 4.17, on peut voir les boxplots représentant l'amplitude de réponse pour les translations LPI et PPI. Un Test t a également été réalisé afin de conclure sur un éventuel impact de la présence ou absence de repères. La p-valeur associée est égale à 0.26, ce qui indique qu'il n'y a pas de différences d'amplitude de réponse des Grids-cells entre des mouvements alignés et non-alignés

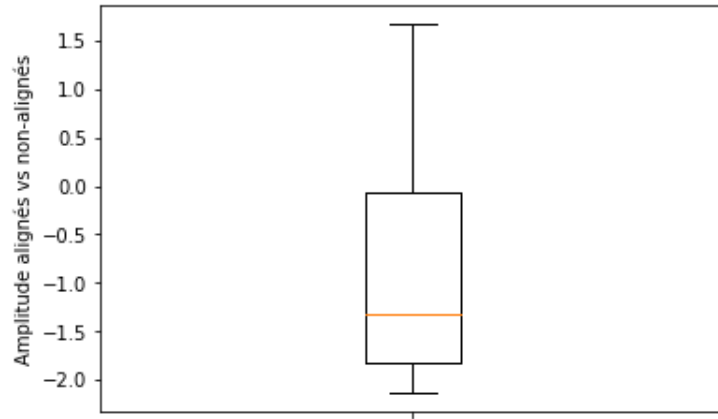


FIGURE 4.16 – Représentation des amplitudes de réponse des Grids-cells entre les mouvements alignés et non-alignés. Ce boxplot comprend 1 valeur par run pour le sujet âgé.

dans ce cas.

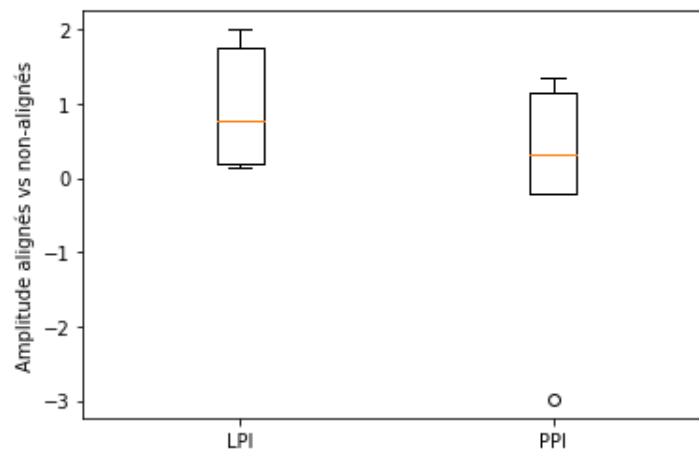


FIGURE 4.17 – Comparaison des amplitudes de réponse des Grids-cells entre les translations avec phare (LPI) et sans phare (PPI). Chaque boxplot comprend 5 valeurs représentant les 5 runs effectués par le participant.

Comme pour l'analyse de stabilité, l'amplitude de réponse des Grids-cells peut être évaluée sur deux analyses distinctes. La première est réalisée uniquement sur les essais LPI et la deuxième sur les essais PPI. La figure 4.18 montre la comparaison des réponses entre LPI et PPI. La p-valeur fournie par le Test t entre ces deux types d'essais est égale à 0.014. Il y a donc une différence significative entre les deux

types. Un Test t supplémentaire est réalisé si la moyenne des deux échantillons diffère de zéro. Les p-valeurs obtenues sont égales à 0.08 pour LPI et 0.10 pour PPI. Cela signifie qu'il n'y pas de réponse préférentielle à des mouvements alignés par rapport à des mouvements non-alignés.

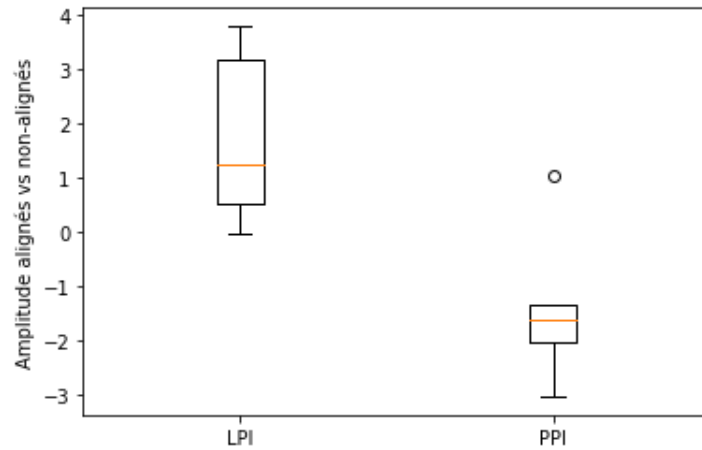


FIGURE 4.18 – Comparaison des amplitudes de réponse des Grids-cells entre les orientations estimées uniquement sur les essais avec phare (LPI) et ceux sans phare (PPI). Chaque boxplot comprend 5 valeurs représentant les 5 runs effectués par le participant

Chapitre 5

Discussion

5.1 Scripts

En résumé, le but principal est d'automatiser les étapes du traitement des images anatomiques et fonctionnelles.

La solution proposée repose sur l'utilisation du Data Analysis Manager de BrainVoyager ainsi que sur des scripts python. La marche à suivre se doit d'être claire et facilement utilisable par les autres membres du laboratoire. La seule condition à remplir pour pouvoir en bénéficier est de l'exécuter sur un ordinateur possédant BrainVoyager.

Comme ces différentes étapes sont très régulièrement réalisées dans le cadre d'imagerie fonctionnelle, il a été décidé d'utiliser le format de stockage BIDS. Ce standard permet un arrangement et un nommage identiques pour un même type de format quelque soit l'expérience réalisée. Cela facilite le partage ainsi que l'exécution des scripts.

Comme expliqué dans les résultats 4.1, le premier script fait la conversion vers le format BIDS et considère que les données sont initialement arrangés selon la figure 4.1.a. Cela implique que pour réutiliser les scripts, il faudra respecter cette organisation et ce nommage des fichiers. Il est légitime de se demander, pourquoi ne pas directement respecter le format BIDS. Lorsque que l'on observe la figure 4.1.d, on remarque que le nommage est assez long et complexe. Dans notre cas, le premier script contient les variables suivantes :

- **Session** : Numéro session.
- **task_name** : Nom de la tâche.

Lors de la conversion, il suffit de compléter les variables ci-dessus avec les valeurs

adéquates et le script réalisera le nommage et l'arrangement sans étape manuelle à exécuter au préalable.

Après ce changement de format, les scripts peuvent être exécutés l'un après l'autre sans modification du code source. En effet, le code source crée automatiquement les sous-dossiers nécessaires. Par exemple, pour stocker les données du preprocessing des IRMf, un dossier correspondant au preprocessing sera créé ainsi que tous les sous-dossiers par sujet.

En ce qui concerne les outils utilisés, il aurait été possible de réaliser uniquement des scripts, cependant l'utilisation du Data Analysis Manager a été ajoutée. En effet, cela permet une bonne visualisation de l'ensemble du projet. Il est possible d'afficher les liens entre les différents fichiers et les étapes (Figure 4.5). On peut également ajouter ou supprimer aisément un sujet en accord avec le format BIDS. Le contrôle du nommage via l'utilisation de différents suffixes peut être analysé et modifié si besoin.

Généralement, chaque script exécute une tâche bien spécifique. Le choix de créer plusieurs scripts est justifié car il n'est pas toujours nécessaire de réaliser toutes les étapes codées dans ce mémoire. De plus, en cas de problèmes d'exécution, il est plus aisé de localiser une erreur dans un code restreint. Enfin, comme les étapes de traitements impliquent des vérifications visuelles, cela permet une inspection plus régulière. C'est un avantage car la propagation d'erreurs de script en script est évitée.

Une autre solution envisageable aurait été de réaliser une fonction globale ou un code unique. Par exemple, en utilisant des booléens indiquant les étapes à réaliser ou non. Cela permet d'éviter la répétition de certaines variables par exemple le numéro des sujets. Actuellement, si l'on souhaite réaliser toutes les étapes sur une série de sujets, il faut indiquer le numéro des sujets dans chacun des scripts. Une piste d'amélioration serait donc de créer un code principal qui permettrait d'exécuter le traitement complet ou partiel. Ce dernier pourrait regrouper les variables communes à chaque script afin qu'elles ne soient complétées qu'à une seule reprise.

En ce qui concerne le stockage des données, l'efficacité du format BIDS peut se discuter. En effet, un avantage est qu'aucune manipulation n'est réalisée sur les données d'origine. A chaque début d'étape, les données nécessaires sont copiées dans le dossier cible. Par exemple, pour procéder au preprocessing, chaque IRMf est copiée et puis le traitement s'exécute sur celle-ci. C'est un réel avantage pour la sécurité des données et pour éviter toute suppression accidentelle ou corruption des données. En plus d'une copie des données, le script crée des fichiers pour chaque étape du preprocessing. Pour rappel, le preprocessing consiste en :

- Head motion correction(3DMC)
- Slice scan time correction(SCCTBL)
- High pass filtering (THPGLMF)

Dans ce cas, il y a un premier fichier créé après la correction des mouvements de la tête nommé *3DMC.fmr*. Ensuite, après la deuxième étape, le fichier *3DMC_SCCTBL.fmr* est enregistré. Finalement, le dernier fichier se nomme *3DMC_SCCTBL_THPGLMF.fmr*. Cela fait donc 3 fichiers .fmr supplémentaires par run mais cela permet une vérification des sous-étapes et une meilleure identification d’une erreur si besoin.

Par contre, cela augmente considérablement la quantité de données et par conséquent la capacité de stockage nécessaire. Par exemple, pour une cinquantaine de patients, un disque dur de 2To a été utilisé.

De plus, ce format met en avant les étapes plutôt que les sujets. Il est donc aisé de savoir quelles étapes ont déjà été effectuées en observant le nom des dossiers. Cependant, cela implique que pour vérifier l’avancement des traitements de données, il faut ouvrir chaque sous-dossier et vérifier si le numéro de patient en question s’y trouve.

Finalement, il est aussi question de limiter le traitement des données manuellement. Jusqu’à présent, ces dernières étaient réalisées en cliquant directement dans le logiciel BrainVoyager. C’est intéressant dans un premier temps de réaliser ces étapes manuellement afin de permettre une bonne compréhension et prise en main du logiciel. Cependant, cela est très répétitif et le risque de fausses manipulations est augmenté. Avec les scripts, il est possible de lancer une étape complète pour un ou plusieurs patients en indiquant leur identifiant. La détection des fichiers à utiliser, est faite par le script et il suffit de le laisser s’exécuter. C’est un réel avantage pour le laboratoire car peu d’ordinateurs possèdent une licence Brainvoyager et il faut donc régulièrement le partager. Dès à présent, chaque utilisateur peut configurer son script. Ensuite, il suffit de se connecter sur l’ordinateur adéquat et de lancer l’exécution. Il est en effet possible que plusieurs utilisateurs exécutent des scripts simultanément. Comme le preprocessing est une étape assez longue (environ 1h par run), il est intéressant de le lancer la nuit par exemple. Ayant expérimenté la connexion à distance, il est plus facile de manipuler l’interface python que de réaliser les étapes manuellement. En effet, il n’est pas rare que les dimensions de l’écran ne soient pas adaptées ou que la connexion se coupe inopinément.

5.2 Masques MNI

L’utilisation des masques MNI permettent d’utiliser les mêmes masques pour chaque sujet. Cependant, les dimensions des masques fournis ne concordaient pas

avec celles des IRMf. La solution utilisée pour régler ce problème requiert plusieurs manipulations, des changements de format et n'est donc pas optimisée. Dans ce cas, l'automatisation de ces étapes n'a pas été implémentée car les opérations ne doivent être répétées qu'à 4 reprises. Après avoir adapté les dimensions des masques, le programme MRICron permet d'attester leur bonne superposition sur l'IRMf. Cependant, il s'agit d'une vérification visuelle et seuls les décalages flagrants peuvent être remarqués.

Comme expliqué dans le Protocole 3.5, le cortex entorhinal est divisé en deux parties, l'antéro-latérale (alEC) et la postéro-médiale (pmEC). Chaque sous-partie est également divisée en 2, la PRC et la PHC. C'est intéressant d'avoir déjà des subdivisions du cortex entorhinal car cela permet, si besoin, de comparer ces différentes parties. Dans ce cas-ci, c'est l'action inverse qui a été réalisée car seuls les masques alEC et pmEC sont utilisés. La fusion des deux parties, PRC et PHC, n'a pas posé problème car cela consiste en une addition de masques qui équivaut à l'addition de deux matrices en python.

A l'avenir, les analyses pourraient être performées dans l'espace du sujet. En effet, il est possible d'utiliser l'autosegmentation des régions du cortex disponibles dans FreeSurfer. Un fichier nommé *aparc+aseg.mgz*, contient la segmentation des régions et il est possible d'en extraire la région souhaitée dans l'espace du sujet. Il est évident que la coregistration doit aussi se réaliser dans ce même espace. Cette opération peut être réalisée aisément à l'aide du script *VTC_creation* 4.1.4 où la variable *mni* devra être fixée à False. Le choix de réaliser les analyses dans l'espace du sujet, implique qu'il y a donc 4 masques propres à chaque sujet. S'il est nécessaire de modifier les dimensions, comme c'est le cas plus haut, l'automatisation de ces étapes est primordiale afin d'éviter un nombre conséquent de manipulations.

5.3 GridCat

L'utilisation du programme GridCat lors de ce mémoire présente de nombreux avantages. Tout d'abord, c'est une programme open-source qui est facilement téléchargeable. Il est fourni avec un manuel et un exemple de données pour se familiariser à son utilisation. Le fait de devoir comprendre et utiliser une ressource créée par quelqu'un d'autre m'a permis de comprendre l'importance d'un manuel ainsi que la clarté des explications. En effet, par la création des scripts, ce mémoire m'a amené, personnellement, à développer une démarche qui doit pouvoir être reproduite sans mon intervention. La description des scripts a donc été impactée par l'utilisation de Gridcat. Cela se ressent par l'utilisation de plusieurs captures d'écran du Data Analysis Manager ainsi que la description précise des concepts et

des variables.

En ce qui concerne l'interface, elle est assez intuitive et le remplissage des différents champs de données est facile et rapide. De plus, le programme propose directement une série de graphiques qui ont été utilisés dans ce mémoire. Quelques analyses statistiques y sont également proposées. Cependant, celles-ci sont performées sur un seul sujet à la fois. Cela ne complique par pour autant les comparaisons entre sujets car tous les résultats sont stockés dans un dossier. Il est donc aisé des les réutiliser, en dehors du programme, pour d'autres analyses complémentaires.

Le principal défaut de ce programme est qu'il faut absolument respecter les formats d'entrées définis. Malheureusement, ceux-ci ne correspondent pas aux formats de sortie de BrainVoyager. Cela a donc nécessité une adaptation du format pour les VTC, le protocole ainsi que pour le masque. Ces différentes adaptations ont pris un temps conséquent. L'interprétation des erreurs, lors de l'insertion des données dans le programme, n'est pas toujours aisée. Après l'adaptation du format des données, il n'est pas évident de savoir si les données n'ont pas été corrompues. Par exemple, sur les sujets jeunes, il arrive assez souvent que le programme retourne une erreur. Dans ce cas, on pense que la conversion des données s'est mal déroulée. Après avoir analysé le code source, l'erreur provient en fait du patient. Cela se produit lorsque que l'orientation ne peut être estimée dans aucun des voxels. C'est donc en fait un résultat et pas une erreur liée aux données en elles-mêmes.

L'utilisation de ce programme permet un gain de temps pour l'analyse des Grids-cells mais la conversion de toutes les données a un peu bridé son efficacité.

5.4 Grids-cells

5.4.1 Orientation de la grille

La première étape d'analyse des Grids-cells consiste à estimer l'orientation moyenne de la grille. Sur les 18 patients analysés, seulement 3 patients présentent moins de 80% de NaN. Parmi ceux-ci, 1 patient présente une cohérence d'orientation entre les voxels. En effet, les deux autres p-valeurs sont supérieures à 0.05. Comme convenu dans le protocole, les analyses suivantes se poursuivent uniquement avec le participant âgé dont l'orientation moyenne de la grille a pu être estimée (Figure 4.9).

Il est clair que le fait d'analyser un seul patient, réduit considérablement les chances de pouvoir tirer des conclusions sur l'efficacité de la tâche. De plus, la

comparaison entre les différents sous-groupes est également impossible. Cependant, des analyses sont réalisées en comparant les 5 runs du participant.

Comparaison entre pmEC et aLEC

On remarque des différences entre les deux sous-régions du cortex entorhinal. Tout d'abord, une orientation cohérente de la grille a pu être estimée dans les deux régions. De plus, le pourcentage de voxels non-NaN est de 72% pour le pmEC et 25% pour l'aLEC. Cette observation est étonnante car dans la littérature, on évoque l'absence de Grids-cells dans l'aLEC [1]. Cela peut s'expliquer par la précision des masques utilisés. En effet, comme les sous-régions sont proches l'une de l'autre, il est possible que le masque aLEC contienne également des voxels du pmEC. La vérification d'une telle supposition n'est pas facilement réalisable visuellement car il faut être précis au voxel près.

5.4.2 Stabilité d'orientation des voxels

La stabilité est comparée avec 3 situations différentes : LPI & PPI, LPI et PPI. Le résultat du test ANOVA ne permet pas de conclure de confirmer une différence significative entre les 3 groupes. Cependant, les Tests t, deux à deux, montrent que les voxels sont plus stables avec les essais LPI comparés aux PPI et LPI & PPI. La stabilité temporelle est corrélée positivement avec les GLRs [19]. C'est pourquoi, ce résultat est interprété dans la section suivante avec l'amplitude de réponses des Grids-cells.

5.4.3 Amplitude de réponse des Grids-cells

La première situation (LPI & PPI) ne présente pas d'amplitude de réponse significative. En effet, la moyenne des amplitudes ne diffère pas significativement de 0. Cela signifie que lorsque l'on ne fait pas la différence entre les translations LPI et PPI, on ne peut conclure de la présence d'une réponse des Grids-cells. En faisant la distinction entre LPI et PPI, le résultat est le même (p-valeur >0.05).

En distinguant les deux types d'essais comme pour l'analyse de stabilité, on remarque qu'il n'y pas de différence d'amplitude de réponse entre les deux situations. De plus, chaque moyenne ne diffère pas significativement de zéro non plus. Cela signifie que l'on ne peut conclure sur la présence du signal des Grids-cells. En effet, celles-ci répondent préférentiellement lors de translations alignées avec l'orientation de la grille. Normalement, une réduction des GLRs est liée à un manque de stabilité temporelle [19]. Or, dans ce cas-ci, c'est l'inverse qui est observée dans le cas

LPI. De plus, il a été prouvé à l'aide de l'AppleGame original que les Grids-cells supportent l'intégration du chemin en absence de repères [5].

Finalement, on remarque que si une orientation de la grille est constatée, ce n'est pas pour autant qu'on observe l'activation des Grids-cells.

Chapitre 6

Conclusion

La détection de la maladie d'Alzheimer, avant l'apparition des premiers symptômes (stade préclinique), est un enjeu majeur dans le but de développer un traitement thérapeutique. Le dysfonctionnement des Grids-cells peut constituer un biomarqueur sensible du risque de maladie d'Alzheimer [5].

Ce mémoire repose sur deux tâches principales. La première est l'automatisation des étapes de traitements d'IRM. La solution est utilisée dans le cadre de ce mémoire mais se veut d'être disponible pour les membres du laboratoire. Elle doit également être utilisable, quelque soit la tâche réalisée dans l'IRM. Dans la continuité, les données d'IRMf sont traitées et analysées afin de mettre en évidence le signal des Grids-cells chez les jeunes.

Ce mémoire m'a permis de m'initier au travail de recherche. En effet, il a développé mes connaissances avec les Grids-cells, ainsi que la manipulation et le traitement de données d'imagerie médicale. Ce travail a enrichi mes compétences, façonnant mon esprit critique, renforçant ma capacité à poser un problème, et affinant la clarté avec laquelle je communique mes avancements.

L'automatisation des traitement des données d'IRM s'est traduit par l'utilisation du Data Analysis Manager de BrainVoyager ainsi que de scripts python pour le préprocessing fonctionnel et pour la coregistration. La décision de rendre ce travail accessible au laboratoire est renforcée à la fois par l'adoption du format BIDS et par une description détaillée du rôle et des variables de chaque script.

Ces scripts ont pu ensuite être directement utilisés pour traiter les images de 9 sujets jeunes et de 9 sujets âgés. L'analyse des Grids-cells a été effectuée, après adaptation du format des données, avec un programme open-source appelé GridCat. Malgré l'absence de détection du signal des Grids-cells sur 18 participants, le travail réalisé pour assurer le bon formatage des données servira pour analyser

les participants restants. En effet, en exécutant les étapes définies, d'autres sujets pourront être analysés à l'aide du programme GridCat.

Le chemin vers un traitement thérapeutique pour la maladie d'Alzheimer est encore long, mais l'arrivée n'en sera que plus belle.

Chapitre 7

Remerciements

Tout d'abord, je voudrais remercier particulièrement mes deux promoteurs, la Dr Laurence Dricot et le Dr Benoît Macq, de m'avoir permis de réaliser un mémoire sur un sujet passionnant ainsi que pour leur suivi régulier. Merci également à Lise Colmant pour sa disponibilité, son écoute et ses bons conseils tout au long de mon mémoire. Je lui souhaite une bonne continuation pour sa thèse. Merci également à tous les membres du laboratoire et tout spécialement à Vincent et Mélissa pour leur sympathie, leurs conseils et leurs encouragements. Finalement, je souhaite remercier ma famille, mes amis, pour avoir été à mes côtés au cours de mes études. Ils ont été ma principale source de motivation et je n'y serais pas arrivé sans eux.

Bibliographie

- [1] T. Hafting, M. Fyhn, S. Molden, M.-B. Moser, and E. I. Moser, “Microstructure of a spatial map in the entorhinal cortex,” *Nature*, vol. 436, pp. 801–6, 09 2005.
- [2] G. Coughlan, “Spatial navigation deficits — overlooked cognitive marker for preclinical alzheimer disease ?” *Nature Reviews Neurology*, vol. 14, p. 496–506, 2018. [Online]. Available : <https://doi.org/10.1038/s41582-018-0031-x>
- [3] K. M. Igarashi, “Entorhinal cortex dysfunction in alzheimer’s disease,” *Trends in Neurosciences*. [Online]. Available : <https://doi.org/10.1016/j.tins.2022.11.006>
- [4] L. Kunz, T. N. Schröder, H. Lee, C. Montag, B. Lachmann, R. Sariyska, M. Reuter, R. Stirnberg, T. Stöcker, P. C. Messing-Floeter, J. Fell, C. F. Doeller, and N. Axmacher, “Reduced grid-cell-like representations in adults at genetic risk for alzheimer’s disease,” *Science*, vol. 350, no. 6259, pp. 430–433, 2015. [Online]. Available : <https://www.science.org/doi/abs/10.1126/science.aac8128>
- [5] A. Bierbrauer, L. Kunz, C. A. Gomes, M. Luhmann, L. Deuker, S. Getzmann, E. Wascher, P. D. Gajewski, J. G. Hengstler, M. Fernandez-Alvarez, M. Atienza, D. M. Cammisuli, F. Bonatti, C. Pruneti, A. Percesepe, Y. Bellaali, B. Hanseeuw, B. A. Strange, J. L. Cantero, and N. Axmacher, “Unmasking selective path integration deficits in alzheimer’s disease risk carriers,” *Science Advances*, vol. 6, no. 35, p. eaba1394, 2020. [Online]. Available : <https://www.science.org/doi/abs/10.1126/sciadv.aba1394>
- [6] F. Raslau, I. Mark, A. Klein, J. Ulmer, V. Mathews, and L. Mark, “Memory part 2 : The role of the medial temporal lobe,” *AJNR. American journal of neuroradiology*, vol. 36, 11 2014.
- [7] M. Hasselmo, “Coding of space and time in the entorhinal cortex,” 2019. [Online]. Available : <https://www.youtube.com/watch?v=OdFJuKhtBWU&list=WL&index=9>
- [8] A. Belgique, “Alzheimer’s disease in numbers,” Website, 2023, [Accessed May 4, 2023]. [Online]. Available :

<https://alzheimer.be/la-maladie-dalzheimer/en-chiffres/#:~:text=Belgique,d%C3%A9mences%2C%20soit%20presque%20137.800%20personnes>.

- [9] J. L. Bellmund, L. Deuker, T. Navarro Schröder, and C. F. Doeller, “Grid-cell representations in mental simulation,” *eLife*, vol. 5, p. e17089, aug 2016. [Online]. Available : <https://doi.org/10.7554/eLife.17089>
- [10] L. Colmant, A. Bierbrauer, Y. Bellaali, L. Kunz, J. Van Dongen, K. Slegers, N. Axmacher, P. Lefèvre, and B. Hanseeuw, “Dissociating effects of aging and genetic risk of sporadic alzheimer’s disease on path integration,” *Neurobiology of Aging*, 2023. [Online]. Available : <https://www.sciencedirect.com/science/article/pii/S0197458023001677>
- [11] M. F. Folstein, S. E. Folstein, and P. R. McHugh, “Mini-mental state : A practical method for grading the cognitive state of patients for the clinician,” *Journal of Psychiatric Research*, vol. 12, no. 3, pp. 189–198, Nov 1975.
- [12] BrainVoyager, “BrainVoyager User Guide,” "WebPage", 2022. [Online]. Available : <https://download.brainvoyager.com/bv/doc/UsersGuide/BrainVoyagerUsersGuide.html>
- [13] C. Vanden Bulcke, M. Wynen, J. Detobel, F. La Rosa, M. Absinta, L. Dricot, B. Macq, M. Bach Cuadra, and P. Maggi, “Bmat : An open-source bids managing and analysis tool,” *NeuroImage : Clinical*, vol. 36, p. 103252, 2022. [Online]. Available : <https://www.sciencedirect.com/science/article/pii/S2213158222003175>
- [14] “Brain imaging data structure (bids),” <https://bids.neuroimaging.io/>, [Online; accessed June 1, 2023].
- [15] A. Maass, D. Berron, L. A. Libby, C. Ranganath, and E. Düzel, “Functional subregions of the human entorhinal cortex,” *eLife*, vol. 4, p. e06426, jun 2015. [Online]. Available : <https://doi.org/10.7554/eLife.06426>
- [16] J. J. Knierim, “Neural representations of location outside the hippocampus,” *Learning & Memory*, vol. 13, pp. 405–415, 2006.
- [17] N. M. van Strien, N. L. M. Cappaert, and M. P. Witter, “The anatomy of memory : an interactive overview of the parahippocampal-hippocampal network,” *Nature Reviews. Neuroscience*, vol. 10, pp. 272–282, 2009.
- [18] M. P. Witter, C. B. Canto, J. J. Couey, N. Koganezawa, and K. C. O’Reilly, “Architecture of spatial circuits in the hippocampal region,” *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, vol. 369, p. 20120515, 2014.
- [19] M. Stangl, J. Achtzehn, K. Huber, C. Dietrich, C. Tempelmann, and T. Wolbers, “Compromised grid-cell-like representations in old age as a key mechanism to explain age-related navigational deficits,” *Current*

- Biology*, vol. 28, no. 7, pp. 1108–1115.e6, 2018. [Online]. Available : <https://www.sciencedirect.com/science/article/pii/S0960982218302239>
- [20] M. Stangl, J. Shine, and T. Wolbers, “The gridcat : A toolbox for automated analysis of human grid cell codes in fmri,” *Frontiers in Neuroinformatics*, vol. 11, 2017. [Online]. Available : <https://www.frontiersin.org/articles/10.3389/fninf.2017.00047>
- [21] D. Hainaut and R. von Sachs, “Lfsab1105 - probabilité et statistiques,” École Polytechnique de Louvain, Université Catholique de Louvain, 2019, cours universitaire.
- [22] Astropy Collaboration. (2023) Rayleigh test of uniformity. [Online]. Available : <https://docs.astropy.org/en/stable/api/astropy.stats.rayleightest.html>

UNIVERSITÉ CATHOLIQUE DE LOUVAIN

École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl