

# Cahier de Tests

Modélisation, Visualisation et Comparaison de  
Connectomes

CONDETTE Clément

PRD 2019-2020

## Table des matières

I-	Graph tests .....	3
1)	Test_hospital : .....	3
2)	Test_edge_node : .....	3
3)	Test_xml : .....	3
4)	Test_csv : .....	3
5)	Test_from_graph_brodmann : .....	4
6)	Test_from_graph_hospital : .....	4
7)	Test_from_graph_harvardoxford : .....	4
8)	Test_from_graph_msdl : .....	4
9)	Test_calculations : .....	4
10)	Test_ReloadGraph : .....	5
11)	Test_SubGraph : .....	5
12)	Test_SymmetricDifference : .....	5
II-	Visualize tests .....	6
1)	Test_brodmann82view: .....	6
2)	Test_hospitalview: .....	6
3)	Test_msdlview: .....	6
4)	Test_harvardoxfordview: .....	6
5)	Test_visualizeLeft : .....	6
6)	Test_visualizeRight : .....	6

## I- Graph tests

Plusieurs tests ont été effectués sur les graphes générés à partir de connectomes. On teste ici la capacité de lecture des différents formats de fichiers et les opérations sur les graphes. A noter qu'en raison d'une absence de vérité terrain et à la difficulté de test, on assume que les graphes générés ont un nombre de sommets et d'arêtes correspondant aux fichiers d'entrée.

### 1) Test\_hospital :

Teste la création d'un graphe à partir d'une matrice et d'un atlas au format txt.

Ecrit ensuite le graphe créé dans un fichier graphml. Il est possible de comparer l'input avec le fichier graphml d'output pour vérifier la correspondance des données du graphe.

**Assert :** « **hospital.graphml** » n'existe pas au début et existe à la fin du test

### 2) Test\_edge\_node :

Teste la création d'un graphe à partir d'une matrice au format edge et d'un atlas au format node.

Ecrit ensuite le graphe créé dans un fichier graphml. Il est possible de comparer l'input avec le fichier graphml d'output pour vérifier la correspondance des données du graphe.

**Assert :** « **brodmann82.graphml** » n'existe pas au début et existe à la fin du test

### 3) Test\_xml :

Teste la création d'un graphe à partir d'une matrice au format txt et d'un atlas au format xml.

Ecrit ensuite le graphe créé dans un fichier graphml. Il est possible de comparer l'input avec le fichier graphml d'output pour vérifier la correspondance des données du graphe.

**Assert :** « **harvardoxford.graphml** » n'existe pas au début et existe à la fin du test

### 4) Test\_csv :

Teste la création d'un graphe à partir d'une matrice au format txt et d'un atlas au format csv.

Ecrit ensuite le graphe créé dans un fichier graphml. Il est possible de comparer l'input avec le fichier graphml d'output pour vérifier la correspondance des données du graphe.

**Assert :** « **msdl.graphml** » n'existe pas au début et existe à la fin du test

#### 5) Test\_from\_graph\_brodmann :

Teste la création d'un graphe à partir d'un fichier graphml.

Ecrit ensuite le graphe créé dans un nouveau fichier graphml.

**Assert :** « `brodmann82.graphml` » existe au début et est équivalent à l'output  
« `brodmann82_from_graphml.graphml` »

#### 6) Test\_from\_graph\_hospital :

Teste la création d'un graphe à partir d'un fichier graphml.

Ecrit ensuite le graphe créé dans un nouveau fichier graphml.

**Assert :** « `hospital.graphml` » existe au début et est équivalent à l'output  
« `hospital_from_graphml.graphml` »

#### 7) Test\_from\_graph\_harvardoxford :

Teste la création d'un graphe à partir d'un fichier graphml.

Ecrit ensuite le graphe créé dans un nouveau fichier graphml.

**Assert :** « `harvardoxford.graphml` » existe au début et est équivalent à l'output  
« `harvardoxford_from_graphml.graphml` »

#### 8) Test\_from\_graph\_msdl :

Teste la création d'un graphe à partir d'un fichier graphml.

Ecrit ensuite le graphe créé dans un nouveau fichier graphml.

**Assert :** « `msdl.graphml` » existe au début et est équivalent à l'output  
« `msdl_from_graphml.graphml` »

#### 9) Test\_calculations :

Teste les calculs des différents métriques calculables sur les graphes de connectomes.

Au moment de l'écriture du cahier de test, les métriques calculées sont les suivantes :

- Degré des sommets
- Force des sommets
- Coefficient de clustering
- Centralité
- Plus court chemin moyen
- Efficacité du graphe

Ces calculs sont effectués sur un graphe simple où les calculs des fonctions sont comparés avec des valeurs précalculées pour ce graphe.

**Assert :** les valeurs précalculées sont égales aux valeurs obtenues par le biais des fonctions de calcul

#### 10)Test\_ReloadGraph :

Teste la capacité à recharger un graphe à partir du fichier d'entrée avec de nouvelles options.

**Assert :** Le graphe lu dans le fichier d'entrée n'est pas vide

Le graphe lu dans le fichier d'entrée n'a pas d'attribut efficacité

Le graphe après modification des options et réécriture n'est pas vide

Les deux graphes ont les mêmes arêtes et sommets

Le graphe après modification des options réécriture a un attribut efficacité

#### 11)Test\_SubGraph :

Teste la création de sous graphes représentant l'hémisphère gauche et droit du connectome. Cependant, le graphe total n'est pas la somme des deux sous-graphes, puisque l'on perd les arêtes entre les sommets des deux hémisphères.

**Assert :** Le fichier d'entrée existe

Le graphe créé à partir de ce fichier n'est pas nul

Le sous-graphe gauche n'est pas nul

Le sous-graphe droit n'est pas nul

L'union des sous-graphe n'est pas nul

#### 12)Test\_SymmetricDifference :

Teste de l'algorithme SymmetricDifference de networkx sur deux graphes similaires. Cet algorithme crée un graphe composé des arêtes qui sont présentes dans un des deux graphes mais pas les deux.

**Assert :** Les graphes lus dans les fichiers ne sont pas nuls

Le graphe résultant des arêtes présente dans un des deux graphes n'est pas nul

Les graphes comparés et le résultat de la comparaison sont visualisables

## II- Visualize tests

Ces tests ont pour but de vérifier que la visualisation de différents graphes fonctionne.

### 1) Test\_brodmann82view:

Teste la visualisation d'un graphe à partir d'une matrice au format edge et d'un atlas au format node.

On peut ensuite vérifier les attributs à l'œil nu.

**Assert** : La fonction de visualisation retourne True

### 2) Test\_hospitalview:

Teste la visualisation d'un graphe à partir d'une matrice au format txt et d'un atlas au format txt.

On peut ensuite vérifier les attributs à l'œil nu.

**Assert** : La fonction de visualisation retourne True

### 3) Test\_msdlview:

Teste la visualisation d'un graphe à partir d'une matrice au format txt et d'un atlas au format csv.

On peut ensuite vérifier les attributs à l'œil nu.

**Assert** : La fonction de visualisation retourne True

### 4) Test\_harvardoxfordview:

Teste la visualisation d'un graphe à partir d'un fichier graphml.

On peut ensuite vérifier les attributs à l'œil nu.

**Assert** : La fonction de visualisation retourne True

### 5) Test\_visualizeLeft :

Teste l'option « left » de la visualisation qui vise à ne visualiser que l'hémisphère gauche du cerveau.

On peut ensuite vérifier les attributs à l'œil nu.

**Assert** : La visualisation du graphe d'entrée en option « left » fonctionne

### 6) Test\_visualizeRight :

Teste l'option « right » de la visualisation qui vise à ne visualiser que l'hémisphère gauche du cerveau.

On peut ensuite vérifier les attributs à l'œil nu.

**Assert** : La visualisation du graphe d'entrée en option « right » fonctionne