Radboud University Nijmegen

Bachelor Thesis

Effective connectivity in the Brain – An analysis on MEG data using causal discovery

Author:
Robbert van der Gugten
robbert.gugten@student.ru.nl
s4137140

Supervised by:
Max Hinne & Marcel van Gerven

Abstract

Neuroimaging techniques have led to an increase of our knowledge of the brain, in particular with regard to connectivity between different brain regions. This study investigates how causal discovery is used for finding effective connectivity in task based MEG data. The PC (Peter-Clark) algorithm operates under the correctness of causal discovery and tests on simulated data show that it is feasible for finding effective connectivity on large sparse graphs. Data of two different tasks have been analysed for connectivity using the PC algorithm, a presented visual stimulus and the remembering of that visual stimulus. This study shows that differences in connectivity are found between the tasks, showing distinct different patterns.

Contents

1 Introduction

The brain is the most complex organ and is studied by the field of neuroscience. An important field in neuroscience is the study of brain connectivity, known as connectomics, in which one aims to map the connections in the brain. Connectomics is a recent field of study that originated in the early 1990's (Friston et al., 1993, Friston 1994). New neuroimaging techniques have led to fairly precise mappings of the brain. These mappings show how parts of the brain are connected, but not which parts stimulate other parts. In this paper we are not only interested in which brain regions are connected, but also the direction of the connections. We aim to find these directed connections by using causal discovery (Spirtes et al., 2000, Shalizi 2012) on MEG data. Similar work was conducted by Janssen & de Ruijter (2014) in which they investigated brain connectivity with causal discovery on resting state fMRI data.

There are various ways to analyse directionality in brain connectivity, many of these methods are time-dependent. An often used time-dependent method is Granger causality (Granger 1969) which is based on the observation that cause occurs prior to its effect. Granger causality occurs when patterns in time series data occur in an other time series after some time lag. This method works well with stable data. However brain data is noisy and therefore hard to interpret with Granger causality.

Causal discovery identifies cause and effect using conditional independence which makes the approach time independent. Time dependent analysis can be difficult since the exact duration and travel time of stimuli between brain regions is unknown. For MEG data this is less of an issue than for example fMRI data, because the distribution of oxygen in blood is not the same and therefore one activated region acquires oxygen faster than others, causing fMRI data to have time lag.

Regions of the brain can be analysed as a graph and connections between regions as directed edges. The MEG sensors are vertices in the graph and by analysing data of the vertices with causal discovery directed edges can be found between the vertices. The PC-algorithm is a method that applies causal discovery on large amounts of data. We will test whether the PC-algorithm is feasible for brain analysis and apply it on task based MEG data to investigate differences in brain connectivity between data of presented visual stimuli and data of remembering the visual stimuli.

1.1 Types of brain connectivity

The study of brain connections can be categorized into three different types(Friston 1994):

- **Functional connectivity:** Functional connectivity tests for statistical dependencies between neurophysiological events by measuring covariance or correlation. Functional connectivity is highly time-dependent and is often measured between all elements of the brain, despite the lack of a structural connection.

- **Structural connectivity:** Structural connectivity can be seen as mapping the brain according to its anatomical connections. This can be modeled using diffusion tensor

imaging (DTI) (Guye et al., 2008; Tournier et al., 2011) Structural connections are strongly related with functional connections, neuronal units can only be functionally connected if there is a structural relation between them (Cabral et al., 2012, van den Heuvel et al., 2010).

- **Effective connectivity:** "Effective connectivity refers explicitly to the influence that one neural system exerts over another"(Friston 1994). This influence can be found by measuring causal relations between neuronal units (Zhang et al., 2008). An advantage of finding causal relations is time-independency. For brain data time-dependent analysis methods are troublesome since there are fluctuations in the haemodynamic response (differences between neuronal activation and the measured BOLD response, (Zhang et al, 2008)). Effective connectivity research to causal relations could be an answer to that problem.

1.2   Previous research to effective connectivity

There are various ways to investigate effective connectivity; for example, structural equation modeling (SEM) (Büchel and Friston 1997; McIntosh and Gonzalez-Lima, 1994), Granger causality analysis (Eichler, 2005), dynamic causal modeling (Friston et al., 2003) and ancestral graph modeling (AG) (Bringmann et al, 2013).
Often used is the SEM method in which causal relations are measured using a combination of qualitative causal assumptions and statistical data. A problem with SEM models and most other effective connectivity analysis methods is the closed world assumption, in which unmeasured variables outside of the measured model are not considered to have a relation with variables inside of the measured model. The method of AGs however is able to model missing brain regions in a network model, and therefore performs better than SEM on models with missing regions. Granger causality methods often struggle with short and noisy datasets of many variables, typically in EEGs. A proposed method for this problem is partial conditioning on a small number of the most informative variables (Wu et al, 2013). Our proposed method for finding effective connectivity is causal discovery which is time-independent, handles noisy data and can indicate activity of brain regions outside the model. These are key motivational aspects to investigate the potential of using causal discovery on brain data.

1.3   Causal discovery

For model discovery and especially causal discovery it is hard to have a systematic way of producing DAGs from only data. There is a basic principle however that leads to the structure of a DAG $G = (V, E)$: for vertices $X, Y \in V$, if X and Y are dependent and there is no subset of other nodes that lead to an independence, then there is an edge E between X and Y (Shalizi, 2012). This is equal to: if X and Y are conditionally independent, then there is no edge between them. By testing the dependencies of every two nodes given all possible subsets $S \in V$, the structure of the graph can be derived. The directions of the edges can be

found by looking at all triples of the graph (X, Y and $Z \in V$, with edges $X - Y$ and $Y - Z$ but no edge between X and Z). The triples can be directed in three ways:

- $X \leftarrow Y \leftarrow Z$ (a chain)

- $X \leftarrow Y \rightarrow Z$ (a fork on Y)

- $X \rightarrow Y \leftarrow Z$ (a collision on Y)

To determine which of the above paths is correct we have to look at the conditional independence of X and Z. If $X \not\perp\!\!\!\perp Z \mid Y$, then X causes Y and Z causes Y, so this is a collision on Y. Such a triple with a collision on Y is also called a V-structure. If $X \perp\!\!\!\perp Z \mid Y$ we know that this must be either a fork or a chain. This condition also holds if $X \perp\!\!\!\perp Z \mid S$ (subset) and $Y \in S$. After testing for collisions some edges will be oriented. Using the orientation of these edges we can find if the triple is a fork or a chain. If $X \rightarrow Y$ $Y - Z$ and $X \perp\!\!\!\perp Z \mid Y$ we know it can not be collision and therefore must be a right chain.

The PC algorithms works by these principles, first all the conditional independences between nodes are calculated given subset S where S is increased in size starting with 0. Edges with conditional independences are removed, the direction of the remaining edges are determined by finding all collisions first, followed by checking for forks and chains using the information gathered from the collisions.

2 PC algorithm

2.1 Background

The PC algorithm is named after it's authors: Peter Spirtes and Clark Glymour (Spirtes et al., 2000). It is a modification of the SGS algorithm (Spirtes, Glymour and Scheines, 1990), in which its correctness follows from the principles of causal discovery as stated in previous section. The modification is necessary for computing large graphs because the SGS algorithm's complexity is exponential (2^n). The PC algorithm however is computationally feasible with complexity $\frac{n^2(n-1)^{k-1}}{(k-1)!}$, where n is the number of variables and k the cardinality. This complexity is a loose upper bound that assumes in the worst case for n and k that no edges are removed given a set of less then k. In practise this hardly ever occurs and leads to a better time complexity than the SGS algorithm. Since the resulting graph of MEG data has 273 nodes (which leads to a very big graph), the PC Algorithm is preferred for the analysis.

Starting from a full undirected graph G, the PC algorithm finds connections as stated in section 1.3 and directs the remaining connections.
The PC-algorithm does not take confounding variables into account. A confounding variable is an unobserved variable Z that correlates with X and Y, causing a measured bidirectional relation between X and Y (X stimulates Y and Y stimulates X). The PC-algorithm operates under the closed world assumption and is therefore not very suitable for brain data since hidden sources may be the cause of data acquired at the MEG sensors (Ramb et al., 2013). For our research we will therefore use a modification of the PC-algorithm (Spirtes et al., 2000) that does take confounding variables into account. The presence of a confounding variable that drives both X and Y is indicated as $X \leftrightarrow Y$. This means that there is a variable Z that causes both X and Y but has not been identified in the current graph.

2.2 Pseudocode

```
 1 Start with the complete undirected Graph G;
 2 n = 0;
 3 while there exist an edge X − Y with Adjacencies(G,X)\Y of cardinality > n do
 4     for every X in G do
 5         for every Y in G do
 6             if there is an edge X − Y then
 7                 test for every subset S of Adjacencies(G,X)\Y of cardinality n if X is
                   conditionally independent of Y given S;
 8                 if X conditionally independent of Y given S then
 9                     remove edge X − Y in G;
10                     record S in Sepset(X,Y) and Sepset(Y,X);
11                 end
12             end
13         end
14     end
15     n = n + 1;
16 end
17 Orient every remaining edge X − Y in G as X o−o Y;
18 for every X in G do
19     if the pair X,Y and Y,Z are both adjacent in G but the pair X,Z is not adjacent in
       G then
20         if Y is not in Sepset(X,Z) then
21             orient X *−* Y *−* Z as X *→ Y ←* Z ;
22         end
23     end
24 end
25 while edges can be oriented do
26     for every triple A,B,C in G do
27         if A *→ B, B*−*C, A and C are not adjacent and there is no arrowhead
           pointing to B on B*−*C then
28             Orient B*−*C as B→C;
29         end
30         if there is an edge between A and B and a directed path from A to B then
31             Orient the edge as A*→B
32         end
33     end
34 end
```

Algorithm 1: Modified PC-algorithm

Orient $X * − *Y$ as $X* → Y$ means: leave the same mark on X and put an arrowhead towards Y.

2.3 Independence test

To test conditional independences between brain regions we use the Fisher Z-transformation. For vertices X and $Y \in V$ of graph $G = (V, E)$ and a subset $S \in V, X \perp\!\!\!\perp Y \mid S$ if:

$$\text{norminv}(\text{abs}(Z)) < (1 - \tfrac{\alpha}{2}), \qquad Z = \sqrt{n - |S| - 3} * 0.5 * \log(\tfrac{1+r}{1-r})$$

r is the partial correlation between X and Y given S, computed using the inversion of the covariance matrix. $r_{ij} = -\dfrac{P_{ij}}{\sqrt{P_{ii}P_{jj}}}$ where r_{ij} is the partial correlation of $X = V_i \in V$ and $Y = V_j \in V$ given S and P is the inverse of the covariance matrix. For our experiments and simulations $\alpha = 0.05$

2.4 Testing and evaluation of the PC-algorithm

The PC-algorithm is not deterministic, Spirtes et al. claim that both the structural part and the directional part of the algorithm are unstable, but in practice the structural part proved more reliable. The order in which edges are evaluated for their direction matters in the directional part, therefore a random ordering is needed. When using randomness however it is necessary to repeat the direction process and store the results. The mode of the results will be used to determine the correct direction. There are three things to check:
1) How accurate is the algorithms prediction? Does it perform better than chance?
2) How does it scale with the number of nodes n?
3) What is the least amount of iterations to direct the graph optimally?

As a check we compare the algorithm with an already written constraint version of the PC-algorithm from the BNT toolbox [1]. To test these points we use simulated data which is generated as follows (Donnachie, 2006):

- Start with a matrix A of size (n, n) where n is the desired number of nodes

- Determine which edges are connected by a Bernoulli random variable based on sparseness factor s, $A_{ij}= \text{Bern}(s)$.

- Model the partial correlations of the edges in the graph with a uniform distribution in the range [0.1, 1].

- Generate data for node $X_1 = \epsilon_1 \sim N(0, 1)$ and nodes $X_i = \sum\limits_{k-1}^{i-1} A_{ik}X_k + \epsilon_i$ i = 2,..,n

---

[1] The BNT toolbox, written by Murphy in 1997-2002 can be found on `https://code.google.com/p/bnt`. The used PC algorithm is learn_struct_pdag_pc_constrain.m

The directions of the edges in the simulated graph are always the same. $X_i \rightarrow Y_j$ (X causes Y) if there exists an edge $X - Y$ and $i > j$. In a picture it would look like this:
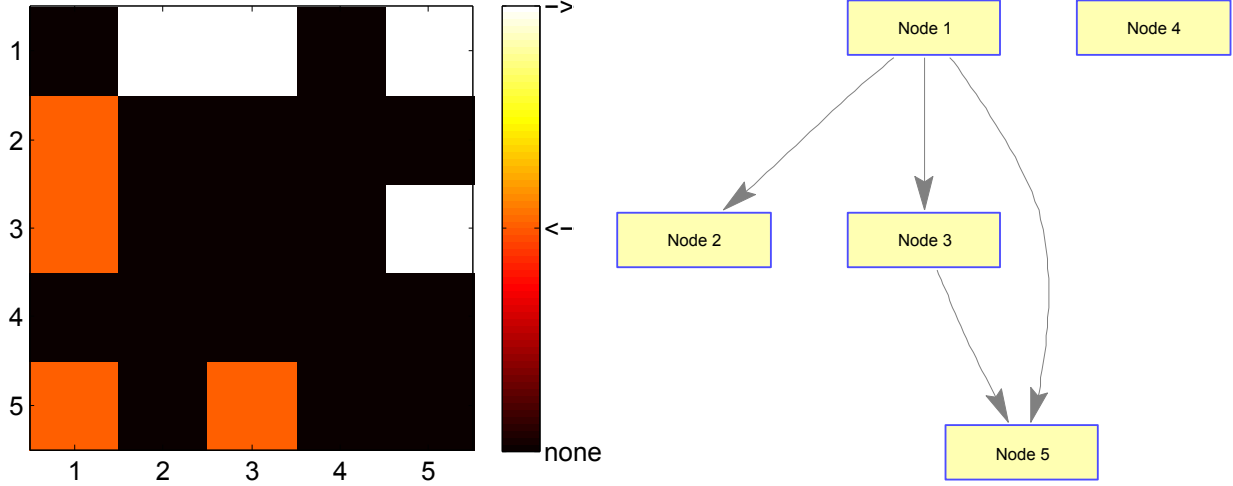

Figure 2.1: *Example of simulated data*

The white squares in the matrix represent a directed edge $X \rightarrow Y$. The orange squares represent an edge of the other direction and the black squares represent no edge. The sample size in the simulations is 100000. The graph in figure 2.1 shows the direction of the edges in a more readable way.

2.4.1 The algorithms prediction and comparison with BNT version

When predicting the edges of a graph, there are two different errors that can be made, a structural error and a directional error. For the structural errors there are also two types of errors, a false-positive and a false-negative.
False positive: There exists an edge E in the predicted graph pG but that edge does not exist in the real graph rG.
False negative: There exists an edge E in the real graph rG, but that edge does not exist in the predicted graph pG.

For the directional errors only the errors predicted for edges that exist in both the graphs rG and pG are taken into account. The edges x o-o y and x $\leftrightarrow$ y are counted as half an error since one side of the arrow is correct.
The number of errors are all normalized, we use the false discovery rate (fdr), false negative rate (fnr) and the directional errors divided by the total edges as the normalize functions.

False discovery rate: $\frac{falsepositives}{falsepositives+truepositives}$

False negative rate: $\frac{falsenegatives}{falsenegatives+truenegatives}$

As for brain data it is important that the fdr is very low, it is innocent if a connection between brain regions is not found, because we are not implying that there can not be a connection if no such connection is found in our research. However It can be troublesome if a wrong connection is found. In figure 2.2 are results of the modified PC algorithm compared with the BNT PC algorithm. The density and number of nodes differ in the plots. The error-rates are averaged over 25 iterations. The density p of the graphs start is in the range [0.03, 0.2], increasing with steps of 0.03.
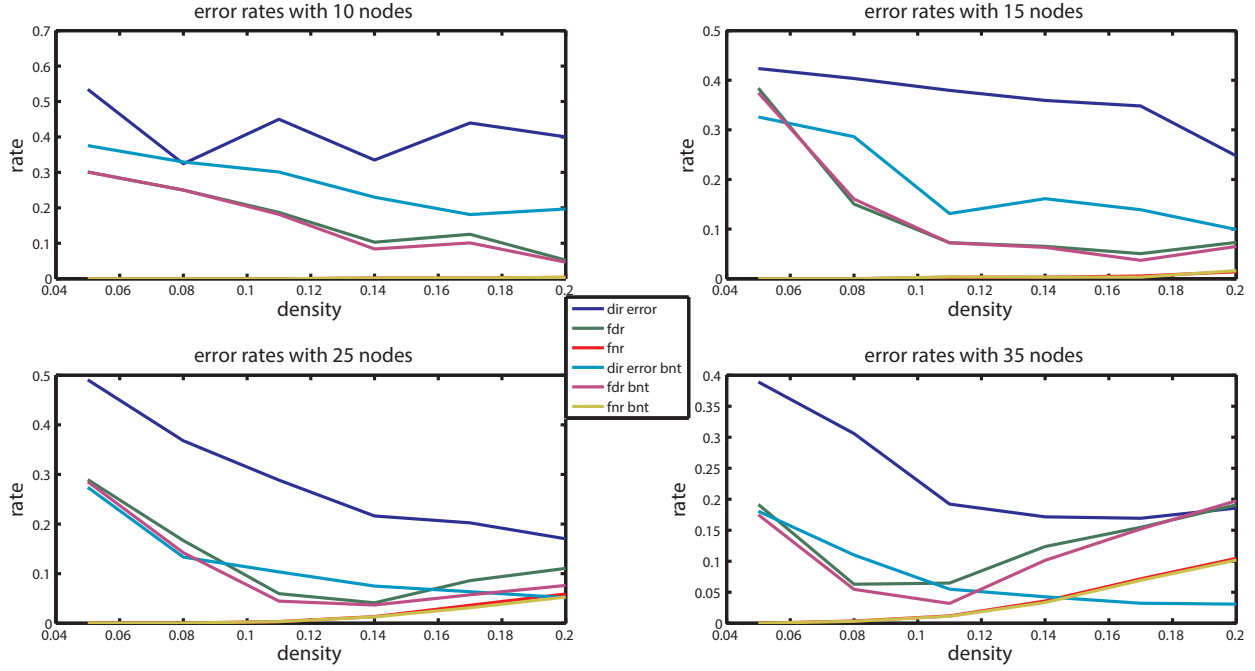

Figure 2.2: *PC test, compared with BNT version*

What can we conclude from these plots? Note that the number of false negatives look very low, but in bigger graphs it is usually higher than the number of false positives since it is normalized and the density of the graph is 0.1. The directional errors of our implementation are higher than in the BNT implementation. An explanation can be seen in figure 2.3 where the nodes are shuffled after assigning edges and where a comparison is made with the PC algorithm without latent variables. The influence of the density and amount of nodes is clearly visible here. The directed error rates are decreasing with n and p, the structural errors are harder to interpret, a better overview can be seen in 2.4 where the errors increase with n.

As for the question "Is the algorithms prediction better than chance"? We can conclude here that it is. If the edges would be assigned by chance and the density is known then the chance of assigning a correct edge equals density. This leads to an fdr of 1-density. With a graph of 0.1 density the chance algorithm acquires an fdr of 0.9, which is way higher than the average

of around 0.1 as seen in the figure.

If the density is unknown then we can say on average the algorithm will assign an edge with a chance of 0.5. For a graph of density 0.1 the expected fdr is then approximately $\frac{0.45}{0.45+0.05} = 0.9$ which is also higher than the fdr of the PC algorithm.

Why does the BNT version perform better on directing the edges? An answer comes by looking at the code of the BNT version,it shows there that the implementation does not use a random order to assign directions. This is odd since the PC algorithm depends on the order of which the nodes are checked for V-structures, directed paths etc. The simulated data is also ordered so that could mean that the BNT implementation performs well because of the lucky ordering of nodes (see figure 2.3). Another reason is that the modified version of the PC algorithm is not correct in general (Spirtes, Glymour and Scheines, 1990). We implemented the PC algorithm with causal sufficiency to compare the directed error rates with the causal insufficient PC algorithm and the BNT version. The nodes were shuffled after the initialisation to investigate the issue of the random ordering. The results are shown in figure 2.3, the errorbars are +.5 and -.5 standard deviation of that points data.
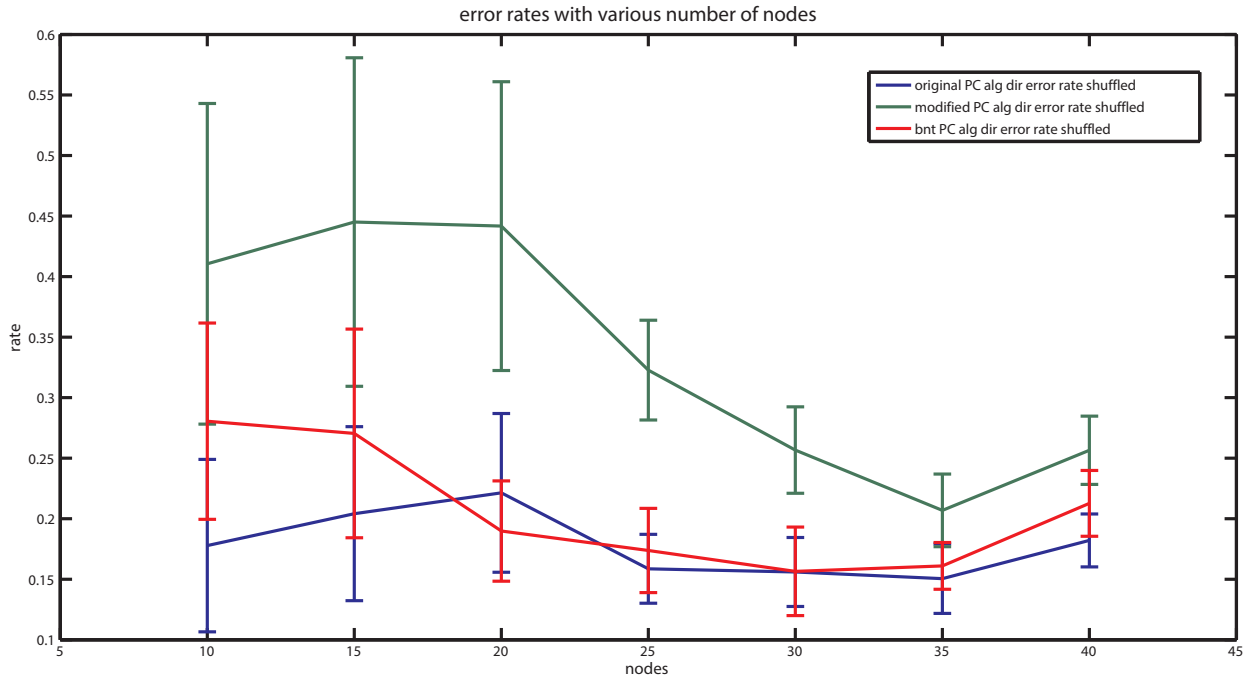

Figure 2.3: *Shuffle test and comparison with normal PC algorithm*

It is clear that the shuffled BNT version (red line) has a higher directional error rate than the unshuffled BNT version as seen in figure 2.2, so it looks like the ordering of the edges in the BNT version does matter. Most of the ordering of connections in the brain is not known and therefore the shuffled version is a better representation of the error rates.

The BNT version and the original PC algorithm have approximately the same error rate which also explains directed error rate difference seen in figure 2.2. The different error rates

12

are converging with n increasing, so for our research the incorrectness of the causal insufficient version is not worrying.

2.4.2 Test for large amounts of nodes

To test whether the algorithm produces decent results for large amounts of nodes we use the fdr and fnr as the structural error measures and the directed error rate for the directional errors. The edges are distributed randomly with a density of 0.1 and the results are averaged over 10 iterations. Results are shown in figure 2.4.
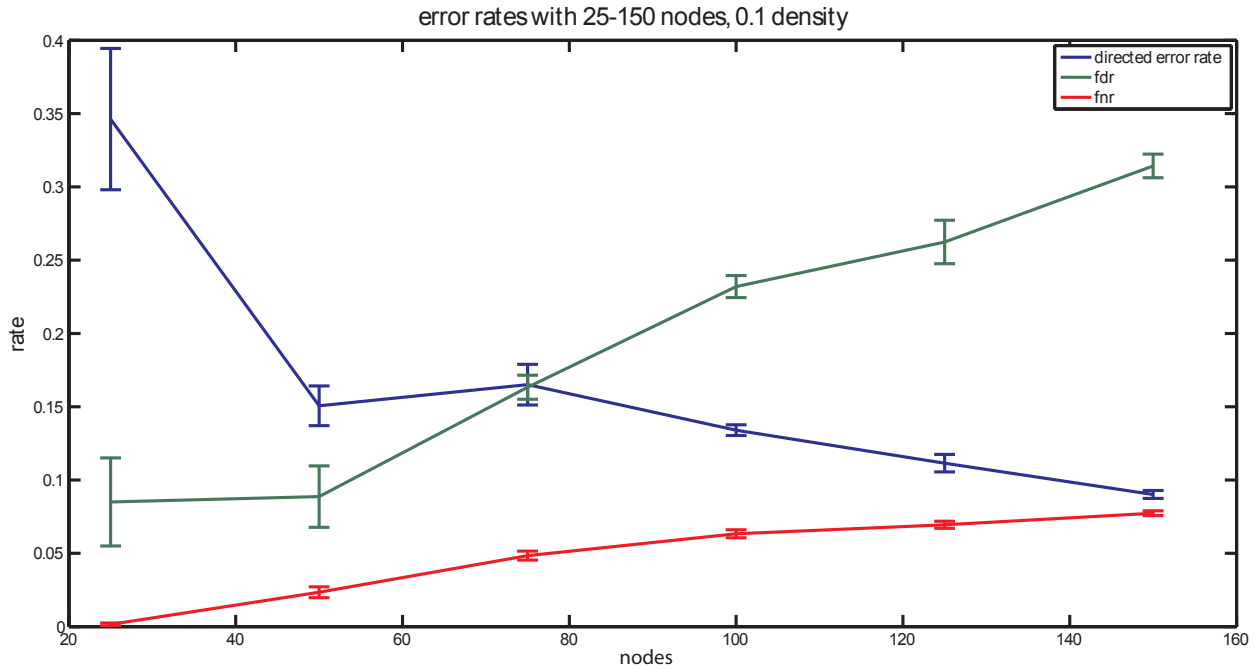

Figure 2.4: *Node test*

The directed error rates are descending with n ascending. For our experiment $n = 272$ so the directed error rate will be low. The fdr however is ascending with n and presumably rising linearly. With $n = 272$ we can expect an fdr of approximately 0.5 if we follow the trend in the figure and assume it rises linearly. This might be too high to determine the right connections in the brain. The density of this graph is 0.1 which is partly the cause of such a high fdr. To skip forward a little, we need to know the density of the resulting graph of the real data to test the error rates with the correct density. For the first subjects this is:

Target subject 1: **0.0205**, Delay subject 1: **0.0212**
Target subject 2: **0.0235**, Delay subject 2: **0.0237**

The average density is 0.0222. To test the expected fdr for our real data we can use the average density as input for the simulated data. $n = 272$ and the density is 0.03, an overestimation of

0.22 since the resulting graph will have an expected density of less than the input. Figure 2.5 shows the fdrs in a boxplot.
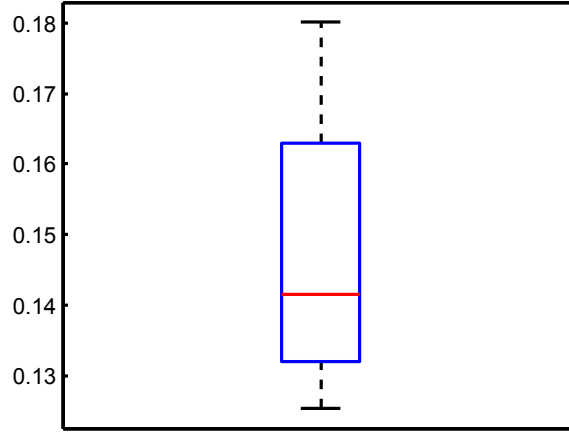

Figure 2.5: *Fdr test on 272 by 272 graphs, N = 10*

The average density is 0.021, so the predicted fdr is a good representation of the resulting fdr for the real data. The average fdr in the test results is 0.1465, this is a big improvement comparing to the expected fdr of 0.5 with 0.1 density graphs. There is still a chance that a predicted edge is wrong, but this chance has become a lot smaller.

2.4.3 The optimal number of iterations for directing edges

Since the order of which we check the nodes for directions influences the resulting expected direction, we use a random order. This means however that to determine the direction of an edge, a certain number of iterations is needed. The direction then can be determined by 3 methods:

- The mean of the iterations – Since our data is not numeric the mean is not a good method. For example if we have 100 edges with value 6 ($\leftrightarrow$) and 50 edges with value 2 (o$\rightarrow$) would result in an edge of value 5($\leftarrow$) which is clearly the wrong direction.

- The median of the iterations – This is also dangerous because it is possible that there are a few edges of for example value 3 in the middle but there are a lot of edges of different values left and right, which still leads to an edge of value 3 that is probably not correct.

- The mode of the iterations – This is probably the best option since it outputs the value of the direction that was most common in all of the iterations. This value can be insignificant so it is important to calculate the number of times in which the chosen direction occurs in the total set, divided by the number of iterations. This way we get an indication of how trustworthy the chosen direction is.

For the experiments and tests we use the mode. The question remains: how much iterations are needed for a good result? This is tested in figure 2.6 with a graph of 50 nodes and 0.1 density. The results are averaged over 10 different graphs.
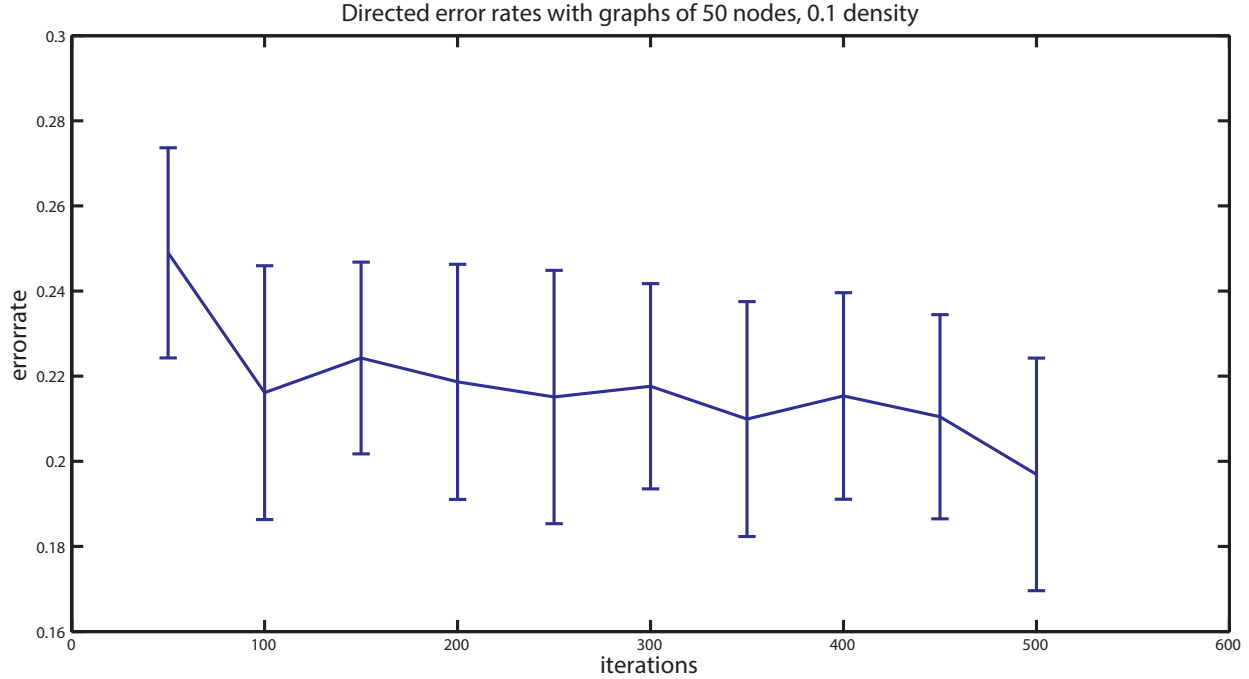

Figure 2.6: *Iteration test*

After 100 iterations the error-rates become somewhat stable, which is $2n$. The differences are little and the error bars are relatively small so any number of iterations $> 2n$ is acceptable. The MEG data results in a graph of $n = 272$, so as a minimum we need 544 iterations. For our experiments we use 1000 iterations.

2.4.4   Relation results and occurrences

Can we determine a faulty edge by the number of it's occurrences? For a graph of 50 nodes with 0.1 density this is tested by averaging the number of occurrences of correct and falsely oriented edges (hereby undirected or bidirectional edges are ignored). The number of occurrences are divided by the total edges and then averaged over 10 iterations.
Correct oriented = **0.390** and standard deviaton = **0.012**
Falsely oriented = **0.329** and standard deviaton = **0.037**
There is a difference between the two, however relatively small. It is wise when determining the correctness of an oriented edge to look at the number of occurrences, the higher the safer the direction is correct.

3 Data

The data for the experiments is task based, sensor based MEG data. Some of the subjects had a malfunctioning sensor, therefore are some connection matrices 272x272 instead of 273x273. The task of the subject was as follows:

The subjects were shown a face, object or letter during target period 1 for 0.5 sec. This was followed by some random noise for 0.2 sec and after that the first delay period started where the subjects needed to remember the target for 2 seconds.

After the first delay period another target was presented followed by random noise and delay. Then there was a cue, consisting of 1, 2 or 1+2. 1 means remember the first seen target, 2 remember the second and 1+2 remember both. After a delay period of 6 seconds the cued item was probed for one second, this was either the same item or an item of the same category. The subjects had to identify if the probe was equal or different than the cue. For the 1+2 cue target 1 or 2 were probed randomly. Feedback was presented after the decision of the subjects (Correct, Wrong, Early or Late). After an inter trial interval of 2 seconds it started all over. Every subject participated in 18x14 trials. See Figure 3.1 for the timeline.
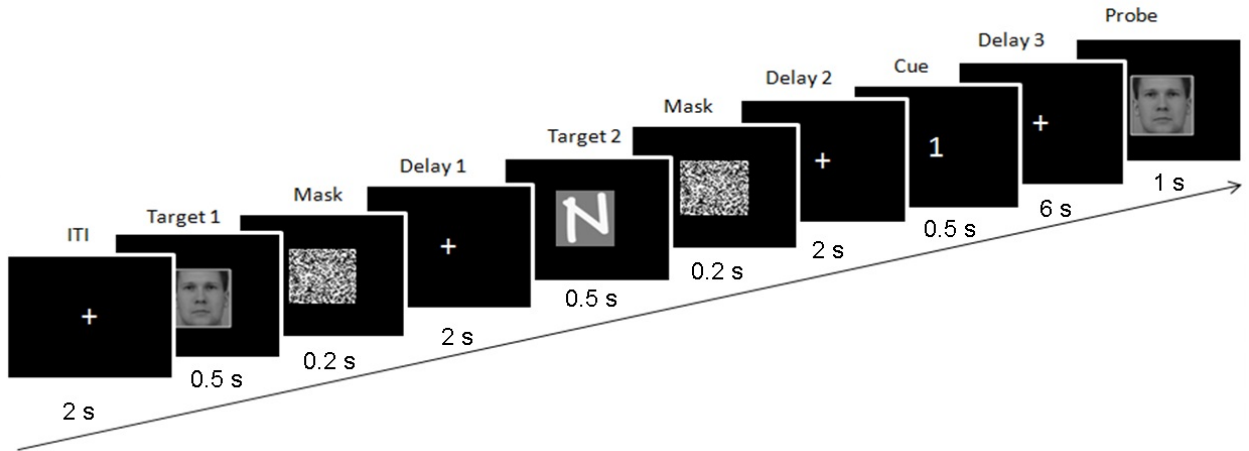

Figure 3.1: *Timeline of one trial*

We have two different datasets for the experiment, one consisting of the first perception period (target1) all trials of one subject and the other is from the first memory period (delay1) of all trials of one subject. We expect to see differences in these sets since perceiving and memorization are two very different cognitive tasks.

4 Experiments

The first target and delay periods of two subjects have been analysed using the PC algorithm. All the trials are analysed together, resulting in one graph. Both subjects had 272 working sensors with resulting sample sizes per sensor of:

Target subject 1: **99671**, Delay subject 1: **206329**
Target subject 2: **142437**, Delay subject 2: **216240**

4.1   Subject 1

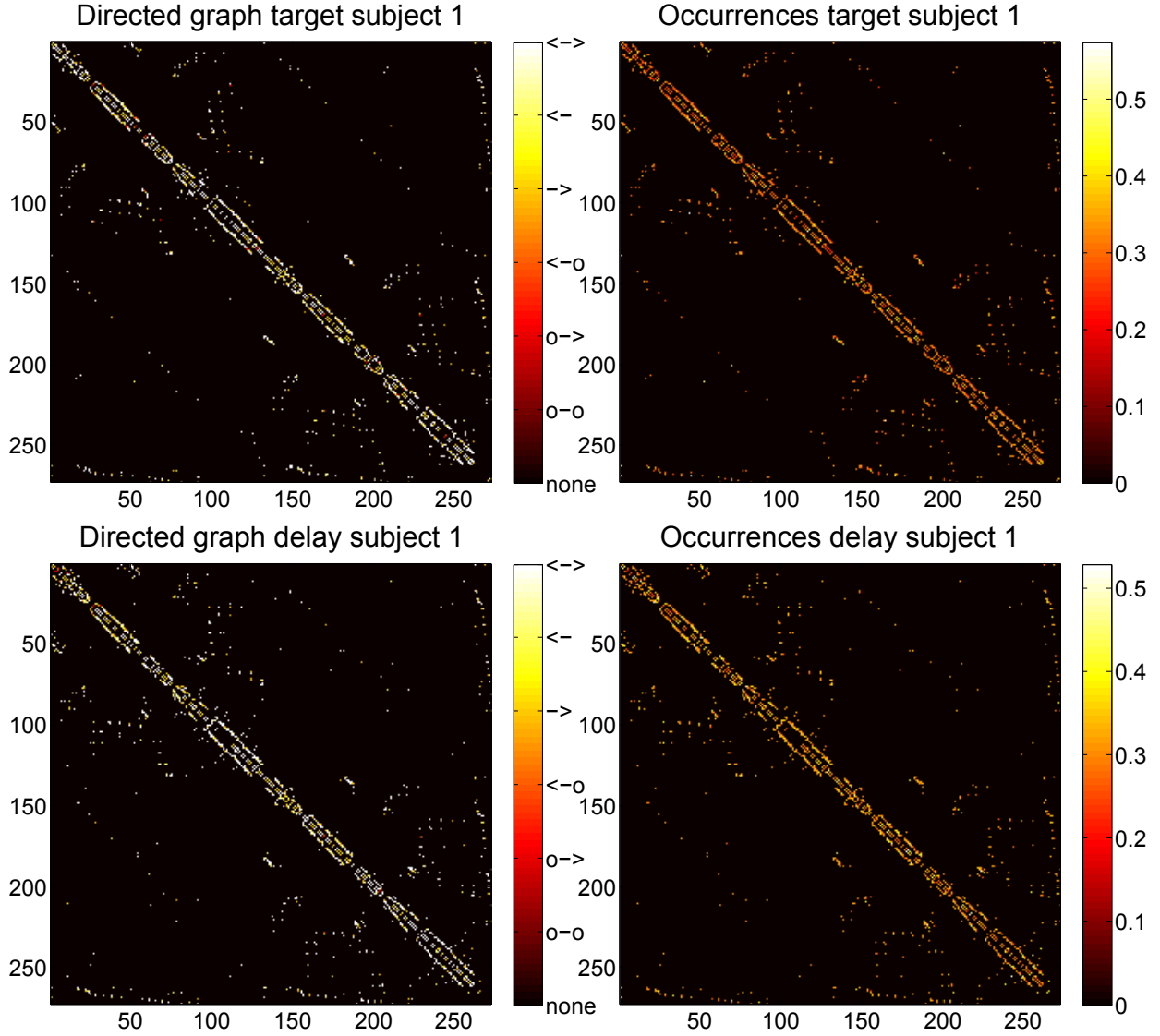The resulting directed graph and the occurrences per edge for subject 1 is shown in figure 4.1.

Figure 4.1: *Delay and target analysis subject 1, number of occurrences is divided by total*

Many of the edges occur in nodes close to each other, which makes sense since some brain signals are detected by multiple sensors in the same region. The number of confounding

variables ($\leftrightarrow$) is relatively high in both the target and delay period, and the occurrences are relatively low. For the number of occurrences, undirected edges do not exist in the full distribution so there are 5 remaining edges that can be assigned. The distribution by chance is 0.2 per edge. Occurrences higher than 0.4 (2 times chance) can be seen as high.

The focus of our experiment lies in the difference between the connections of the target period and the delay period. Edges that occur in both target and delay period have been discarded, figure 4.2 and 4.4 show the remaining connections of the target period and figure 4.3 and 4.5 the remaining connections of the delay period, using coordinates of the MEG sensors. The bigger the nodes, the more connections that node has.
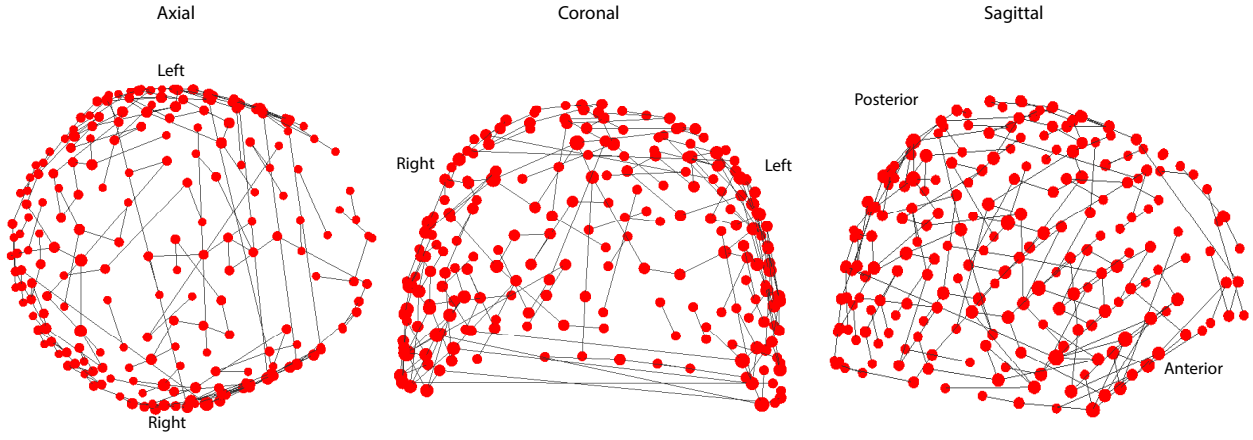

Figure 4.2: *Undirected connections of subject 1 that occur in the target period but not in the delay period*
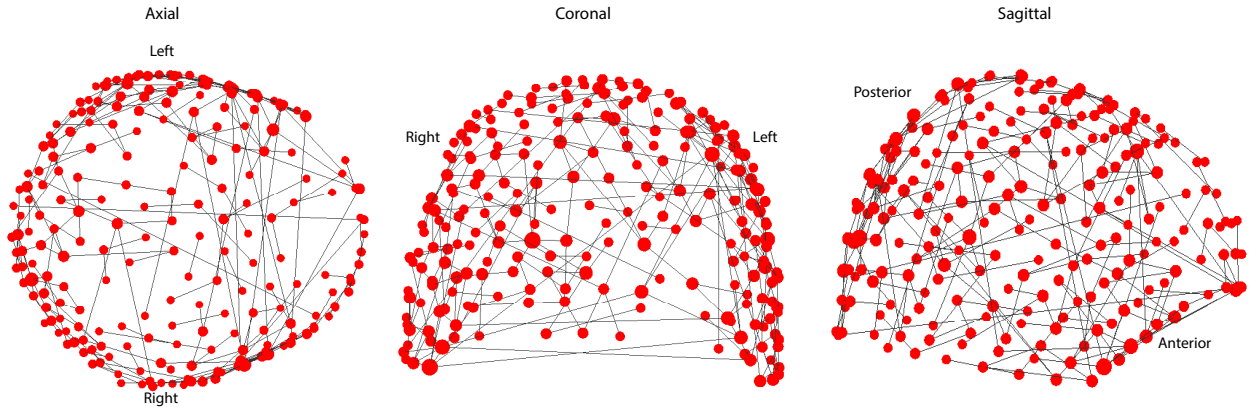

Figure 4.3: *Undirected connections of subject 1 that occur in the delay period but not in the target period*

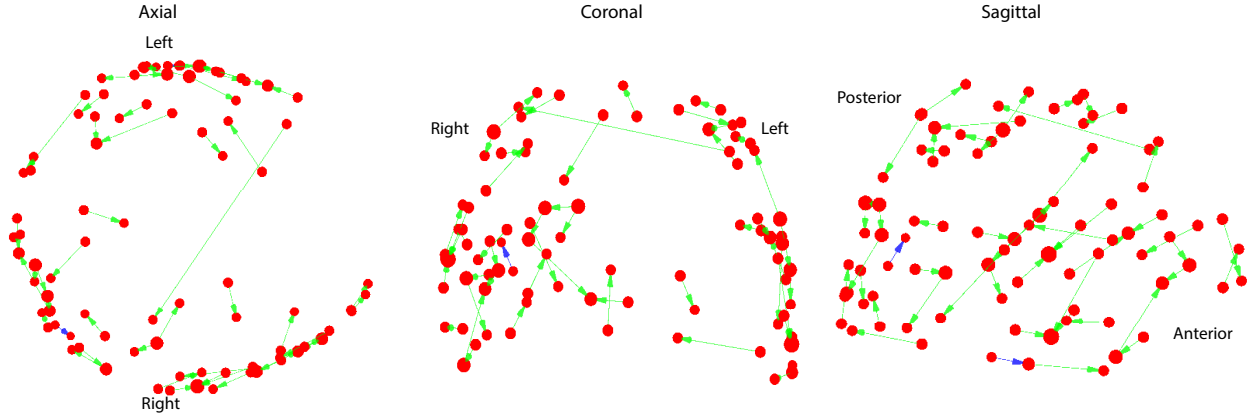Figure 4.4 and 4.5 show the direction of the edges, without confounding variables.

Figure 4.4: *Directed connections of subject 1 that occur in the target period but not in the delay period. Confounding variables have been removed.* Green = →, blue = o→
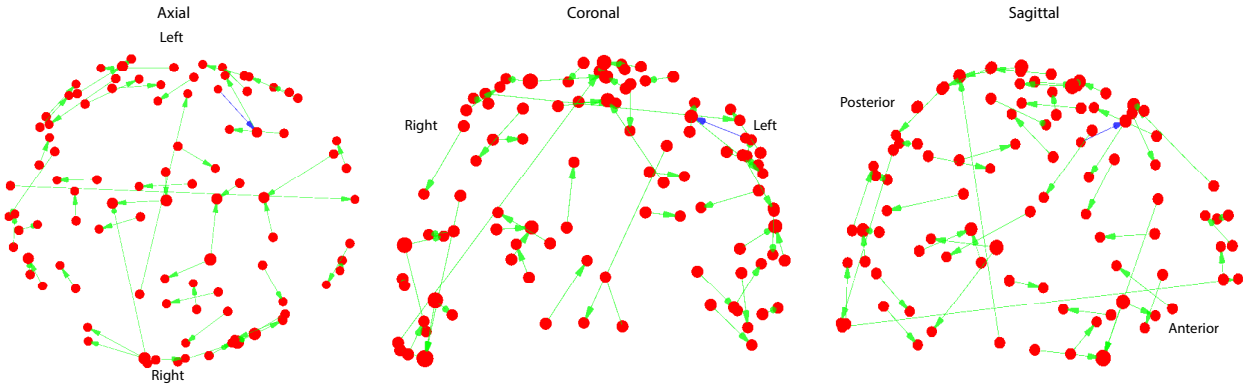

Figure 4.5: *Directed connections of subject 1 that occur in the delay period but not in the target period. Confounding variables have been removed.* Green = →, blue = o→

Less confounding variables are found in the delay period, more green arrows lead to a better view of the directions. What can we conclude from these connections? There are fewer edges in the target period and many of the edges occur on the sides at the temporal lobes. The temporal lobes have areas associated with vision (Ungerleider & Mischkin, 1982), especially with object recognition and interpretation of the visual stimuli. Since the subject in the target period was presented an image of an object, face or a letter the areas in the temporal lobes are likely to be activated. The connections in the target period seem therefore very plausible.

In the delay period the edges are still spread across the whole brain. The subjects had to memorize the image shown in the target period. Memorization activates multiple regions in the brain, including the prefrontal cortex, thus the edges in the delay period seem plausible as well.

As for the direction of the edges there are some chains formed in the target period at the

sides and back of the brain. In the delay period there are chains formed directed to the posterior part of the brain. Despite these found patterns it is hard to interpret why the edges are directed that way.

4.2   Subject 2

In figure 4.6 to 4.10 the results of the second subject are shown, many patterns in figure 4.6 are similar to figure 4.1 of subject 1.
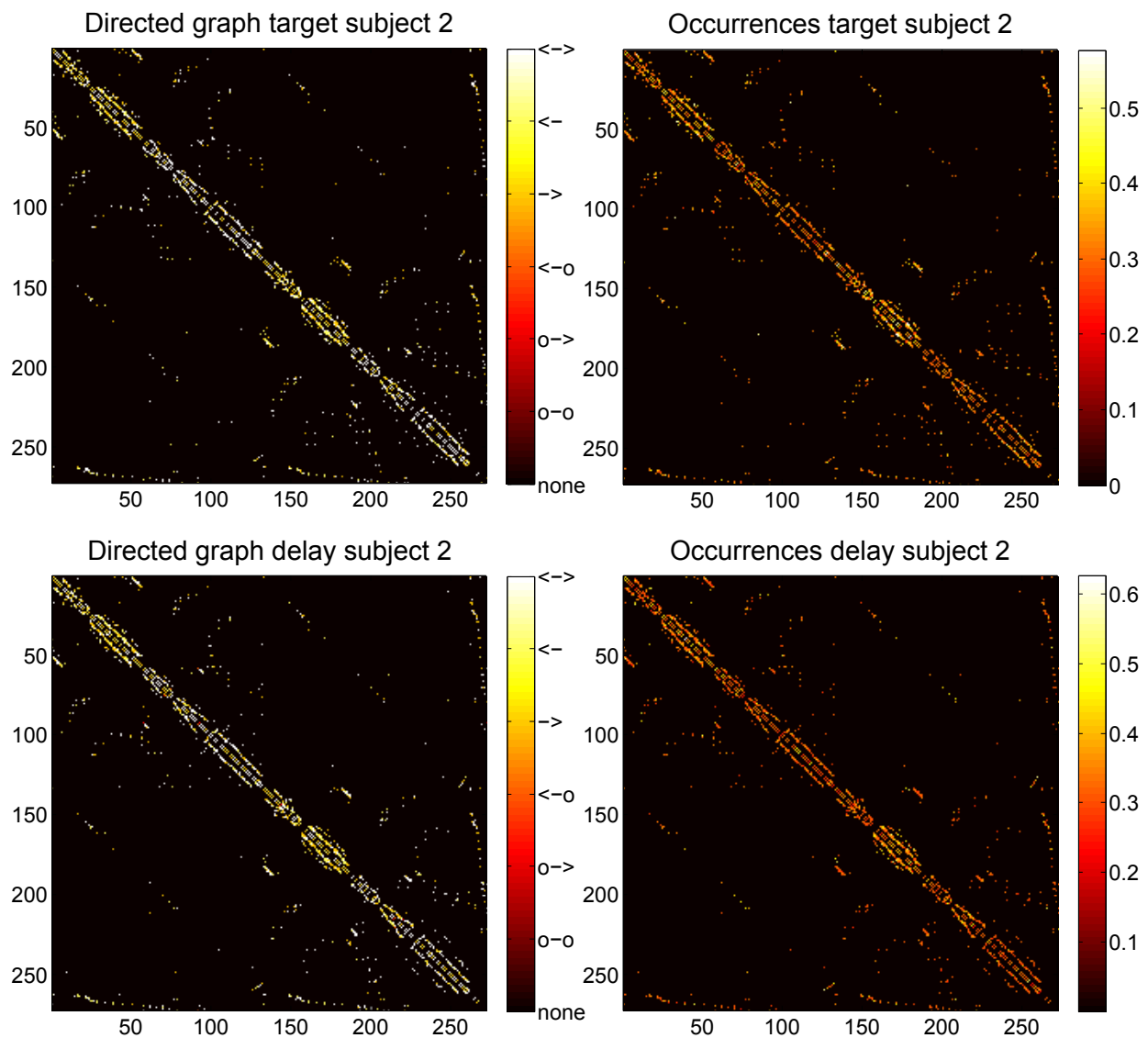

Figure 4.6: *Delay and target analysis subject 2, number of occurrences is divided by total*
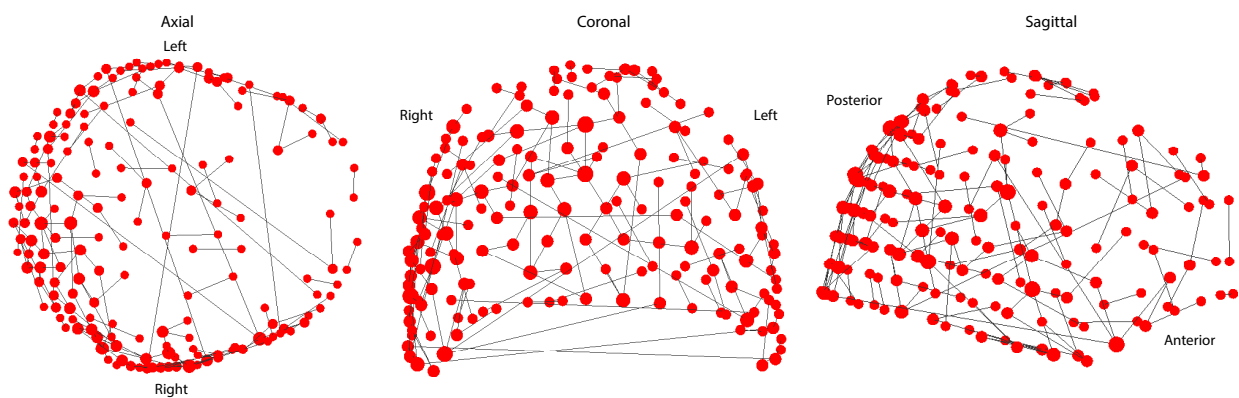
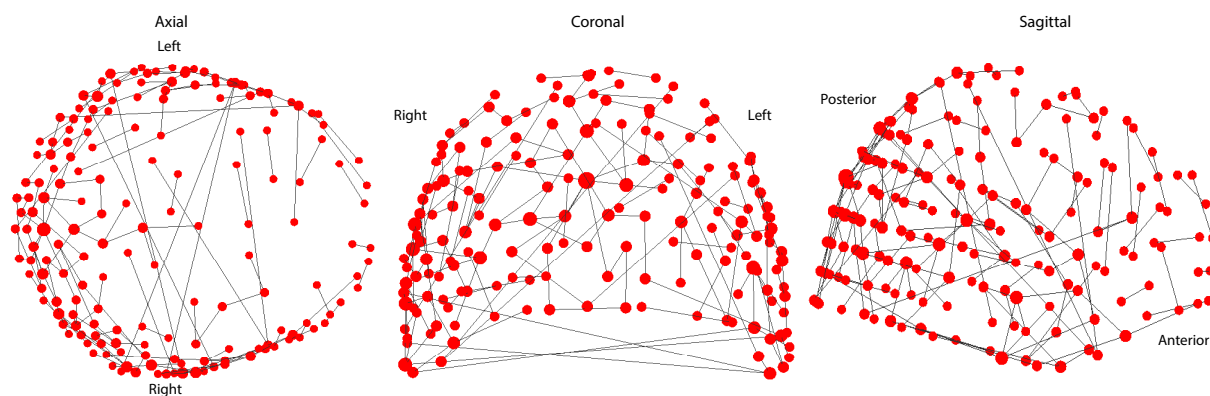**Figure 4.7:** *Undirected connections of subject 2 that occur in the target period but not in the delay period*

Figure 4.8: *Undirected connections of subject 2 that occur in the delay period but not in the target period*
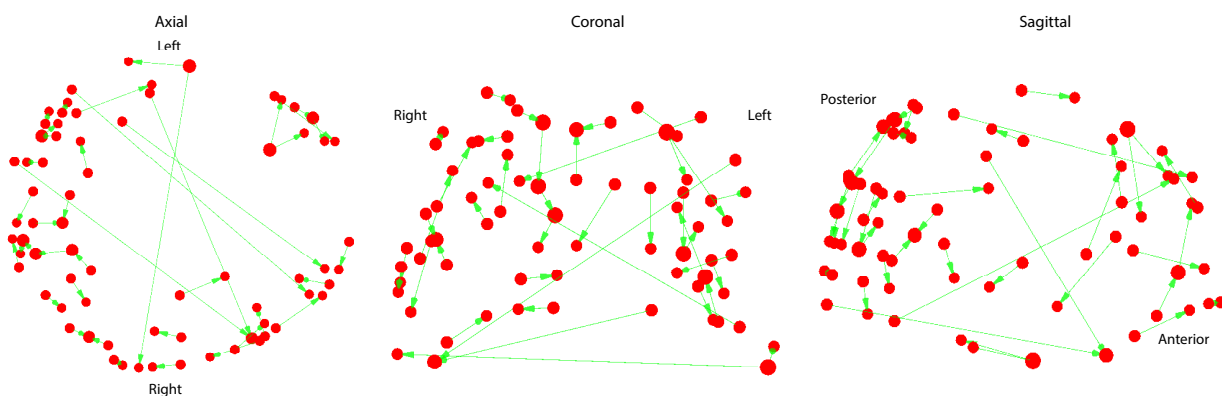

Figure 4.9: *Directed connections of subject 2 that occur in the target period but not in the delay period. Confounding variables have been removed.* Green = →, blue = o→
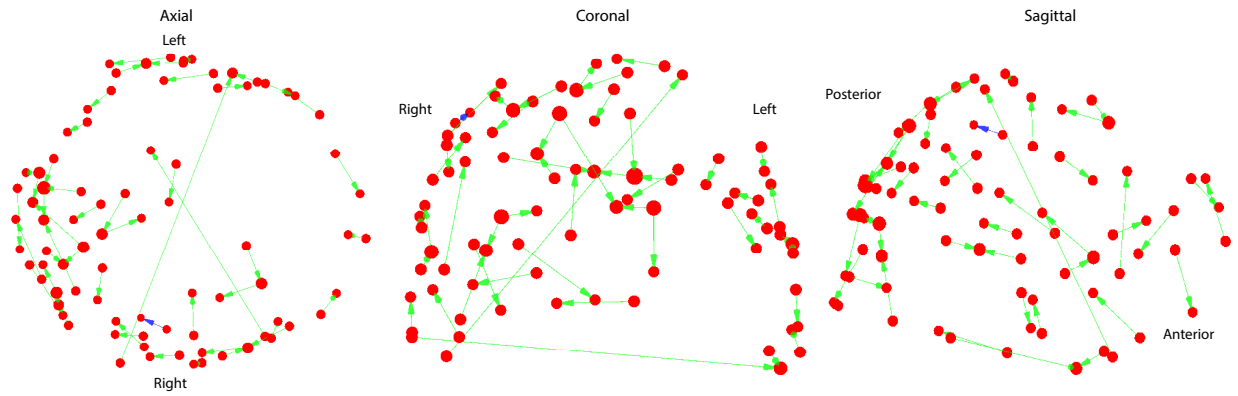
Figure 4.10: *Directed connections of subject 2 that occur in the delay period but not in the target period. Confounding variables have been removed. Green = →, blue = o→*

In the target period many cross-brain edges appear to be unidirectional which in the first subject were mostly bidirectional (and therefore removed due to the indication of a confounding variable). Again fewer edges are found in the target period but number of edges in the temporal lobes have decreased comparing with subject 1. Looking at figure 4.8 these edges still exist, so there are many confounding variables causing these connections. There are however increased connections at the edge of the temporal lobes and the occipital lobe. The occipital lobe is also associated with vision. Various edges are localized in the parietal lobe, which is associated mainly with integrating sensory information and manipulation of objects. Although it is odd that there are cross-brain edges, the results of the target period appear to be plausible because of the edges associated with vision. The brain is always running multiple processes which can account for the edges that are not associated with vision.

As for the delay period fewer connections on top of the brain are found than the first subject, but still more than the target period of the second subject. The same observation holds as for the first subject, however the evidence is less strong due to fewer connections.

5 Evaluation

5.1 Conclusion

In this paper we investigated the possibility of using causal discovery for finding effective connectivity in the brain. Tests on simulated data show that the PC algorithm, which its correctness follows from the basics of causal discovery, is feasible for finding effective connectivity on sparse graphs. The PC algorithm is applied on data of two subjects, consisting of target data and delay data. In the target period the subjects were shown an object, letter or face and in the delay period the subjects had to remember the image seen in the target period. The resulting predicted connectivity graphs show that differences are found between the two periods in both the subjects. The delay period graphs contained more connections that were spread across the brain, which indicates memory processes. The target period graphs contained edges mostly on the sides and back of the brain, where the visual areas lie.

5.2 Discussion

5.2.1 Simulation results

There are a few remarks considering the results of the tests on simulated data. The false discovery rate was still on average 0.14, which means that 14 of 100 edges predicted by the PC algorithm do not exist in the real graph (or in the brain). There is a method however to reduce the fdr, see section 5.2.1. The directional error rate is with few nodes fairly high but decreases when there is more information. An always returning problem using simulated data is if the simulated data is representative for the observed data. Brain data can contain noise from unwanted sources, like movement or off topic thoughts. Many of this unwanted data is filtered out during artefact checks, but there can still be influences of these sources. Randomness and noise have been added to the simulated data to represent these influences. If we assume that the simulated data is a good representation for real brain data, then the tests show that the PC algorithm is feasible for finding effective connectivity in the brain, taking into account that it is not error proof.

5.2.2 Differences in observed connections between target and delay period

The goal was to find differences between connections in the target and delay period of the task, these differences have been found and it can roughly be explained why these differences exist. But only two subjects have been analysed so it is more a speculation than explanation. The strength of the connections are not considered with the interpretation and only a global interpretation of the connections is given. A more detailed view can be obtained by analysing the individual connections, the brain areas associated with the nodes of the connections and the direction of the connections.

5.3 Further research

5.3.1 False discovery rate reduction

It is possible to reduce the false discovery rate in learning the structure of a DAG by using the PC_{fdr}-skeleton algorithm (Li & Wang 2009). This algorithm curbs the false discovery rate of the DAG's structure. The improvements were noticeable, however Li & Wang only tested the method on graphs with $n \leq 30$ and density around 0.1. To use the PC_{fdr}-skeleton algorithm on MEG data, simulations with a larger amount of nodes are necessary to determine whether the method is an improvement to our study.

5.3.2 Evaluate more subjects

Two subjects is not enough to determine whether found connections actually exist in our brain, especially not since determining the structure of the graph and directing edges both are not error proof. When evaluating more subjects all the found edges can be put into one graph, including the strength of the connections. With finding the same connections for different subjects the strength of that connections increases. An other option is to find similarities between trials of one subject. For our experiment there was one target and one delay period, evaluating more periods leads to a better understanding of that subjects connectivity in the brain.

References

[1] C. Büchel, K. Friston, *Modulation of connectivity in visual pathways by attention: cortical interactions evaluated with structural equation modeling and fMRI.* Cereb Cortex 7, 768-778, 1997.

[2] J. Cabral, E.Hugues, M.L. Kringelbach, G. Deco, *Modeling the outcome of structural disconnection on resting-state functional connectivity..* Neuroimage, 62(3), 1342-1353, 2012.

[3] E. Donnachie, *Graphical Models and the PC Algorithm.* Lecture notes Leipzig University, 2006.

[4] A. Eichler, *A graphical approach for evaluating effective connectivity in neural systems.* Philos Transact R Soc B 360, 953-967, 2005.

[5] K.J. Friston, *Functional and effective connectivity in neuroimaging: A synthesis.* Human Brain Mapping. 2(1-2), 56-78, 1994.

[6] K.J. Friston, C.D. Frith, P.F. Liddle, R.S.J. Frackowiak, et al., *Functional connectivity: the principal-component analysis of large (pet) data sets..* Journal of cerebral blood flow and metabolism, 13:5-5, 1993.

[7] K.J. Friston, L. Harrison, W. Penny, *Dynamic causal modelling.* Neuroimage 19, 1273-1302, 2003.

[8] C.W.J. Granger, *Investigating Causal Relations by Econometric Models and Cross-spectral Methods.* Econometrica 37 (3), 424-438, 1969.

[9] M. Guye, F. Bartolomei, J.P. Ranjeva, *Imaging structural and functional connectivity: towards a unified definition of human brain organization?* Curr Opin Neurol 24, 393-403, 2008.

[10] M.P. van den Heuvel, C.W. Mandl, R.S. Kahn, and H.E. Hulshoff Pol., *Functionally linked resting-state networks reflect the underlying structural connectivity architecture of the human brain.* Human Brain Mapping, 30(10), 3127-3141, 2009.

[11] R. Janssen & T. de Ruijter, *Causal Discovery methods for Effective Connectivity in Human Brains.* Radboud Univerity Nijmegen, January 17, 2014.

[12] J. Li, Z.J. Wang, *Controlling the False Discovery Rate of the Association/Causality Structure Learned with the PC Algorithm.* Journal of Machine Learning Research 10, 475-514, 2009.

[13] A.R. McIntosh, F. Gonzalez-Lima, *Structural equation modeling and its application to network analysis in functional brain imaging.* Hum Brain Mapp 2, 2-22, 1994.

[14] R. Ramb et al., *The impact of latent confounders in directed network analysis in neuroscience.* Phil. Trans. R. Soc., july 2013.

[15] C. R. Shalizi, *Advanced Data Analysis from an Elementary Point of View* 482-493, 2012.

[16] P. Spirtes, C. Glymour, and R. Scheines, *Causality from probability. Conference Proceedings: Advanced Computing for the Social Sciences.* Williamsburgh, VA, 1990.

[17] P. Spirtes, C. Glymour & R. Scheines, *Causation, prediction and search.* The MIT Press, volume 81, 165-167, 2000.

[18] J.D. Tournier, S. Mori, A. Leemans, *Diffusion tensor imaging and beyond.* Magn Reson Med 65, 1532-1556, 2011.

[19] L.G. Ungerleider, M.Mishkin, *Two cortical visual systems, in analysis of visual behaviour.* Edited by D.J. Cambridge, Massachusetts, MIT Press, 549-586, 1982.

[20] G. Wu, W. Liao, S. Stramaglia, H. Chen, D. Marinazzo, *Recovering directed networks in neuroimaging datasets using partially conditioned granger causality.* Brain connectivity 3, 294-301, 2013.

[21] D. Zhang, A.Z. Snyder, M.D. Fox, M.W. Sansbury, J.S. Shimony, M.E. Raichle, *Intrinsic functional relations between human cerebral cortex and thalamus.* J Neurophysiol 100, 1740-1748, 2008.