%% Time frequency analysis

% 1. Load clean data Up1:28,21 Up2:14,15 Control3:9,10

% 2. Do freqanalysis WITHOUT cfg.keeptrials

load('W:\Angela\Data\Clean data\clean\_data\_P9\_SWS.mat');

cfg = [];

n = 35;

cfg.method = 'wavelet';

cfg.output = 'pow';

%cfg.keeptrials = 'yes';

cfg.foi = exp(linspace(log(5),log(80),n)); %by Leander

cfg.toi = -1:0.01:2.5;

cfg.width = exp(linspace(log(4),log(12),n)); %by Leander

TFA9 = ft\_freqanalysis(cfg, data\_clean);

% Saving

filename = 'TFA\_P9';

cd('W:\Angela\Data');

save(filename, 'TFA9') %4x PP CODE !!!!

%% Grandaverage

% Load if you have to

load('W:\Angela\Data\TFA\_P21');

load('W:\Angela\Data\TFA\_P28');

load('W:\Angela\Data\TFA\_P14');

load('W:\Angela\Data\TFA\_P15');

load('W:\Angela\Data\TFA\_P9');

load('W:\Angela\Data\TFA\_P10');

% Grand averagen

% cfg.keepindividual = YES!!

cfg = [];

cfg.keepindividual = 'yes';

[grandavg1] = ft\_freqgrandaverage(cfg, TFA28, TFA21);

[grandavg2] = ft\_freqgrandaverage(cfg, TFA14, TFA15);

[grandavg3] = ft\_freqgrandaverage(cfg, TFA9, TFA10);

% Saving

filename = 'grandavrg1\_28\_21';

cd('W:\Angela\Data');

save(filename, 'grandavg1')

filename = 'grandavrg2\_14\_15';

cd('W:\Angela\Data');

save(filename, 'grandavg2')

filename = 'grandavrg3\_9\_10';

cd('W:\Angela\Data');

save(filename, 'grandavg3')

%% Cluster calculation

cfg = [];

% cfg.channel = (default = 'all')

% cfg.latency = (default = 'all')

cfg.frequency = [5 8]; %Freq & avg over freq combined ! T 5-8

cfg.avgoverfreq = 'yes'; % B 12-30 12-20/21-30

cfg.method = 'montecarlo';

cfg.statistic = 'ft\_statfun\_depsamplesT';

cfg.correctm = 'cluster';

cfg.clusteralpha = 0.05;

cfg.clusterstatistic = 'maxsum'; %'wcm'; how to combine the single samples that belong to a cluster, 'maxsum', 'maxsize', 'wcm' (default = 'maxsum')

% option 'wcm' refers to 'weighted cluster mass', a statistic that combines cluster size and intensity; see Hayasaka & Nichols (2004) NeuroImage for details

cfg.clusterthreshold = 'nonparametric\_individual';

cfg.minnbchan = 2;

cfg.tail = 0;

cfg.clustertail = 0;

% Note that if you want to run a two-sided test, you have to split the critical alpha value by setting cfg.correcttail = 'alpha'; i.e. this sets cfg.alpha = 0.025, corresponding to a false alarm rate of 0.05 in a two-sided test.

cfg.alpha = 0.025;

cfg.numrandomization = 500;

% Specifies with which elecrodes clusters can be formed

neighbours = open('W:\Angela\waveguard\_neighbours\_064ch.mat');

neighbours = neighbours.waveguard\_neighbours\_064ch;

cfg.neighbours = neighbours;

subj = 2;

design = zeros(2,2\*subj);

for i = 1:subj

 design(1,i) = i;

end

for i = 1:subj

 design(1,subj+i) = i;

end

design(2,1:subj) = 1;

design(2,subj+1:2\*subj) = 2;

cfg.design = design;

cfg.uvar = 1;

cfg.ivar = 2;

[stat] = ft\_freqstatistics(cfg, grandavg1, grandavg3);

% Saving

filename = 'stat\_grndavg1\_3\_theta';

cd('W:\Angela\Data');

save(filename, 'stat')

% Under stat.negclusters and stat.posclusters you can see if you have sig

% clusters! prob is your p value!

%% Averaging over trials, add raw effect. Keeptrials no, dus geen middelen mogelijk

cfg = [];

grandavg1\_2 = ft\_freqdescriptives(cfg, grandavg1);

grandavg3\_2 = ft\_freqdescriptives(cfg, grandavg3);

%% Plotting stats

% Plotting significant cluster overlayed on the raw effect

% stat.raweffect = grandavg1\_2.powspctrm - grandavg3\_2.powspctrm;

cfg = [];

% cfg.zlim = [-4.6549e+13, 1.5640e+04]; maxmin

cfg.alpha = 0.025;

cfg.parameter = 'stat';

cfg.layout = 'waveguard\_layout\_064ch2.lay';

ft\_clusterplot(cfg, stat);