

## MI-based BCI – predicting movements with EEG

### הקדמה

התממשקות מוח-מחשב, או BCI (Brain-Computer Interface), הינה מערכת מבוססת מחשב המשיגה אותות מוחיים, שולפת מאפיינים אינפורמטיביים ומייצרת אותות מהידע שחולץ מאותם מאפיינים, לטובת תפעול מכשור חיצוני (Hossain et al., 2015).

Motor Imagery (MI) הוא תוצאת גישת המודעות לתוכן של כוונת התנועה, כאשר זו מבוצעת באופן לא מודע בזמן הכנת התנועה (movement preparation – MP) (Jeanneord, 1994, 1995). הוא הגיע למסקנה כי MI מודע ו-MP לא-מודע חולקים מכניזמים מוחיים ומתפקדים בצורה דומה.

על מנת לקרוא, לזהות ולנתח את הפעילות המוחית המאופיינת בדפוסים מרחביים כלשהם ב-MI מבוסס BCI, נתוני שיטת הרישום EEG (Electroencephalogram) ממספר ערוצים הם הכרחיים (Jin et al., 2019). עבודה זו תתמקד בניתוח נתוני EEG שנאספו במסגרת מחקר BCI מבוסס MI, למטרת זיהוי דפוסים מרחביים המסוגלים להבחין בין MI של יד ימין ל-MI של יד שמאל. נחלץ 32 מאפיינים מתוך שתי אלקטרודות (C3 ו-C4) במטרה להצליח ולנבא באמצעות שיטת LDA מתי הנבדק מדמיין פעולה של יד ימין, ומתי יד שמאל.

### מטרת התרגיל

במחקר זה התקבלו 160 דגימות. 128 מהן מתויגות ומשמשות ללמידה ו-32 אינן מתויגות ונועדו למבחן. כל דגימה היא הקלטת EEG בשתי אלקטרודות של ניסוי באורך 6 שניות. הניסוי מורכב משני חלקים – בחלק הראשון מופיע גירוי ויזואלי בין השנייה הראשונה ל-2.25 שניות והעיד על כיוון מסוים. החל מהשנייה ה-2.25 ועד לסוף הניסוי (6 שניות) על הנבדק היה לדמיין תנועה של היד בכיוון המסומן. הנתונים עברו עיבוד ראשוני, והוצגו בדרכים שונות על מנת לזהות מאפיינים איכותיים שניתן לזהות בעין אנושית המפרידים בין הקטגוריות השונות. לאחר איסוף המאפיינים, וסינון המאפיינים הרלוונטיים פחות באמצעות שיטת NCA (neighborhood component analysis) – יפורט בהמשך), אימנו את האלגוריתם לבצע את ההפרדה הליניארית בין 2 הקטגוריות בעזרת שיטת ה-LDA (Linear Discriminant Analysis – יפורט בהמשך).

על מנת לוודא כי איננו מבצעים התאמת-יתר (overfitting) – שגיאת אימון נמוכה בעוד ששגיאת ההכללה גבוהה – ביצענו תיקוף-צולב (cross-validation) מסוג k-fold. סוג זה של תיקוף-צולב מחלק את פרמוטציית הנתונים לא קבוצות, מאמן את האלגוריתם k פעמים על k-1 קבוצות ובודק את שגיאת התשריר (validation) על הקבוצה הא. בכל אימון, קבוצת התיקוף משתנה, כך שכל קבוצה הינה קבוצת תיקוף פעם אחת ונכללת בקבוצת האימון k-1 פעמים.

## שיטה

לטובת בחירת המאפיינים המבחין ביותר, נשתמש בשיטת NCA. שיטת ניתוח נתונים זו, בוחנת קשרים הסתברותיים בין נקודות שונות ותיגון, ומחשבת את ההסתברות שנקודה מסוימת תהיה שייכת לקטגוריה מסוימת, בעזרת ההסתברויות של הנקודות ה"שכנות". כלומר, השתייכות נקודה לקטגוריה מסוימת (מעבר לתיגה המקורי), נקבעת בצורה אקראית על ידי בחירת נקודה אחרת בהסתברות מסוימת (מוגדרת על ידי softmax של המרחק האוקלידי בין הנקודות<sup>1</sup>), ירושת התיג שלה וחישובים נוספים (Goldberger, Hinton, Roweis, & Salakhutdinov, 2005).

LDA הינו אלגוריתם המשתמש במאפיינים לטובת חילוק נקודות (גם רב-ממדיות) לקבוצות, באמצעות מציאת וקטור ההיטל המבחין ביותר, כך שהשונות התוך-קבוצתית בכל קבוצה תהא מינימלית בעוד שהשונות הבין-קבוצתית תהא מקסימלית. כשמו כן הוא, מפריד בצורה לינארית בין המאפיינים בהם נעשה שימוש. אלגוריתם דומה אך ההפרדה שלו נעשית בצורה שאינה לינארית, נקרא QDA (Quadratic Discriminant Analysis). ל LDA שימושים נרחבים, החל מזיהוי פנים (Zhao et al., 1998) ועד לסיווג שלבים ראשוניים בסרטן השד (Conceição et al., 2011). באמצעות אלגוריתם זה אימנו את האלגוריתם שלנו, והעמדנו את התוצאות שלנו במבחן "העולם האמיתי" (סט נתונים שלא השתמשנו בו אלא למטרה זו בלבד).

## קוד

מטרתנו הראשונית הייתה תיג נכון של הנתונים ל 2 קבוצות – ימין ושמאל. לצורך כך, בעזרת הפונקציות regexp, isempty, cellfun ו-find, יצרנו שני וקטורים בינאריים המייצגים את צעדי הניסוי (trials) בהם הוצג לבדקים חץ המצביע ימינה או שמאלה. באמצעות וקטורים אלו, ופונקציית find, חילצנו את המיקומים של אותם צעדי ניסוי. בנוסף, יצרנו וקטור שמכיל את כל צעדי הניסוי המתויגים. כמו כן, באמצעות פונקציית cellstr, יצרנו וקטור-תא כך שיכיל את התיג 'left' או 'right', במיקומים שחילצנו קודם לכן.

```
leftID = find(~cellfun(@isempty, regexp(P_C_S.attributename, 'LEFT')));
rightID = find(~cellfun(@isempty, regexp(P_C_S.attributename, 'RIGHT')));
% get the indices of each condition using the binary vector
leftLoc = find(P_C_S.attribute(leftID,:) == 1);
rightLoc = find(P_C_S.attribute(rightID,:) == 1);
% get the indices of all labeled data signals
allLoc = sort([leftLoc rightLoc]);
% create label vector
label = cell(length(allLoc),1);
label(leftLoc) = cellstr('left');
label(rightLoc) = cellstr('right');
```

<sup>1</sup> היחס בין אקספוננט המרחק האוקלידי של שתי הנקודות לסכום האקספוננטים של המרחק האוקלידי בין הנקודה הנבחרת לשאר הנקודות.

הנתונים נשמרו במערך תלת-מימדי (P\_C\_S.data) כך שהמימד הראשון מעיד על מספר צעד הניסוי, המימד השני מייצג את מספר הדגימה והמימד השלישי מייצג את האלקטרודה. הindex הראשון במימד השלישי מייצג את אלקטרודה C3 והאינדקס השני מייצג את אלקטרודה C4. בסך הכל (ראה טבלה 1). הנתונים חולצו ל-3 מבני נתונים – מערך הנתונים השלם (בשלושה מימדים) ולפי אלקטרודה - על מנת לחלץ את המאפיינים מהנתונים. הפרמטרים בהם השתמשנו בניסוי, ערכם ותפקידם, מפורטים בטבלה 1.

```
C3 = 1;
C4 = 2;

% extract all labeled data and by electrodes
All_Data = P_C_S.data(allLoc, :, [C3 C4]);
C3_Data = P_C_S.data(allLoc, :, C3);
C4_Data = P_C_S.data(allLoc, :, C4);
```

טבלה 1:

Variable name	Variable value(s)		Variable name	Variable value(s)	
C3	1		VarWind	192	
C4	2		area_std_inter	137-175	
visualize	0/1 (Boolean)		partTH	0.45	
simulation	0/1 (Boolean)		fullTH	0.55	
fs	128		dimComp	3	
Hz	[0.5, 0.6, 0.7, .... .... 39.8, 39.9, 40]		k_fold	9	
queEndTP	288		k_bin	15	
nGraphs	20		n_iter	500	
Bin_edges	[-4, -3.8, -3.6 .... ..... 3.6, 3.8, 4]		Font	'axes'	18
window	160			'title'	16
overlap	158			'sgtitle'	18
FeatNum	12			'legend'	12
				'tick'	12
				'colorlabel'	16

בעזרת פונקציית calcPower, חישבנו את הכוח החלקי – ה-power spectra הממוצע של הזמנים בניסוי בהם הנבדק התבקש לדמיין את תנועת אחת מגפיו (2.25 עד 6 שניות) לטובת תצוגה, עבור כל תנאי ואלקטרודה. השימוש בפונקציה נעשה בעזרת הקלטים הבאים: נתוני ה-EEG, גודל חלון לחישוב העוצמה, גודל החפיפה לחישוב העוצמה, וקטור התדרים, תדר הדגימה, משתנה בוליאני (המעיד על מיוצע (1) או לא (0)) ומשתנה המעיד על האינדקס ממנו חישוב העוצמה מתחיל.

במידה וערך המיצוע הוא אמת, הנתונים שחוזרים הם מבנה בעל שני שדות, ממוצע וסטיית תקן, שיחידות הערכים המופיעים בהם הוא בדציבלים.

```
% set mean boolean variable for calcPower function
Mean = 1;

% calculate mean power spectra and standard-deviation of each condition
% with self-written function
partialPower = struct(...
'RC3', calcPower(C3_Data(rightLoc,:), window, overlap, Hz, fs, Mean, ...
queEndTP), ...
'RC4', calcPower(C4_Data(rightLoc,:), window, overlap, Hz, fs, Mean, ...
queEndTP), ...
'LC3', calcPower(C3_Data(leftLoc,:), window, overlap, Hz, fs, Mean, ...
queEndTP), ...
'LC4', calcPower(C4_Data(leftLoc,:), window, overlap, Hz, fs, Mean, ...
queEndTP));
```

הפונקציה calcPower עצמה, המחזירה רכיב פלט שנקרא Array, מגדירה משתנים להגדרות ברירת מחדל במידה ואין מספיק רכיבי קלט. לאחר מכן מחשבת את העוצמה ובהתאם למשתנה הבוליאני Mean מחזירה את הנתונים כמו שהם או ממירה את העוצמה לדציבלית ושומרת את הממוצע וסטיית התקן.

```
if nargin < 6
    endTime = size(Data,2);
    startTime = 1;
    Mean = 0;
elseif nargin < 7
    endTime = size(Data,2);
    startTime = 1;
elseif nargin < 8
    endTime = size(Data,2);
end

% check startTime validity
% if equal 0, meant from the top -> equal 1
if startTime == 0
    startTime = 1;
end

% power spectra calculation
Power = pwelch(Data(:,startTime:endTime)', window, overlap, Hz, fs);

% create return variable
if Mean
    Power = pow2db(Power);
    Array.std = std(Power,0,2)';
    Array.mean = mean(Power,2)';
else
    Array = Power;
end
```

לאחר מכן, עברנו לשלב התצוגה, שמוציא גרפים רק במידה והמשתנה הבוליאני visualize בעל ערך אמת (1). בשלב התצוגה התחלנו בלהציג את 20 אותות EEG הגולמיים עבור כל תנאי, בעזרת פונקציית plotRand, המקבלת את מספר הגרפים שעליה להוציא מכל תנאי, את המידע

ממטריצת הנתונים השלמה של כל צעדי הניסוי בכל אחד מהתנאים השונים, אינדקסים המייצגים את האלקטרודות בממד השלישי, תדר דגימה ומבנה המכיל את הפונטים הרצויים.

```
plotRand(nGraphs, All_Data(rightLoc,:), All_Data(leftLoc,:), C3, C4, ...
fs, Font);
```

הפונקציה עצמה מחשבת את מיקום תווית ציר ה-Y, את אורך כל מערך נתונים (מספר צעדי הניסוי) ואז מגרילה את הנתונים אשר יוצגו בגרף (ראה תמונות 2 ו-3).

```
% set location of ylabel
Line = nGraphs/4;

if Line/2 == round(Line/2)
    yLab = nGraphs/2+1;
else
    yLab = nGraphs/2-1;
end
% get length of each data set
LnLeft = size(LData,1);
LnRight = size(RData,1);

% choose random trials from each condition
RandTrials = randi(LnLeft, nGraphs);
LrandTrials = randi(LnRight, nGraphs);
```

המשכנו בהצגת ארבע ספקטרוגרמות המציגות את מיצוע הכוח בדציבלים מעבר לצעדי הניסוי, בעבור כל אלקטרודה בכל תנאי. בנוסף, הצגנו שתי ספקטרוגרמות כך שהפחתנו את תנאי שמאל מתנאי ימין בעבור כל אלקטרודה, באמצעות פונקציית dispSpectro. הפונקציה מקבלת ארבעה מערכי נתונים עבור ארבעה שילובי תנאי-אלקטרודה, גודל חלון לחישוב הספקטרוגרמה, גודל חלון החפיפה, וקטור תדרים, תדר דגימה ומבנה המכיל את הפונטים הרצויים.

```
dispSpectro(C4_Data(rightLoc,:), C3_Data(rightLoc,:), ...
C4_Data(leftLoc,:), C3_Data(leftLoc,:), window, overlap, Hz, fs, Font);
```

הפונקציה עצמה מחשבת את מספר צעדי הניסוי ואת גודל חלונות תוצר הספקטרוגרמה לטובת הקצאה של מערכים לפני הלולאה (מוצגים רק מערכי הניסוי של C4 ו-C3 ימין לצורך הדגמה). הלולאה רצה כמספר צעדי הניסוי שהוקלטו ומחשבת ספקטרוגרמה עבור כל צעד ניסוי בכל אחד מהתנאים (לשם נוחות, מוצג רק תנאי ימין אלקטרודה C4), המקדמים המרוכבים אותם אנו מקבלים עוברים המרה ליחידות של power spectra שלאחר מכן עוברות המרה לדציבלים. בסוף הלולאה מחושב הממוצע, מעבר לכל צעדי הניסוי, ועוברת לתצוגתם (ראה תמונות 4 ו-5).

```

N_trial = size(DataRC4, 1);

% allocate arraies according to one result
[~, freqVec, TVec] = spectrogram(DataRC4(1,:), window, overlap, Hz, fs,
'yaxis');
RC4spec = zeros(length(freqVec), length(TVec), N_trial);
RC3spec = zeros(size(RC4spec));

for n = 1:N_trial
    % get the spectrogram complex coefficients for each condition and trial
    RC4spec(:, :, n) = spectrogram(DataRC4(n,:), window, overlap, Hz, fs,
'yaxis');

    % translate into power
    RC4spec(:, :, n) = abs(RC4spec(:, :, n)).^2;

    % turn units into dB
    RC4spec(:, :, n) = 10*log10(RC4spec(:, :, n));
end
% calculate mean over trials
RC4spec = mean(RC4spec, 3);

```

לבסוף, בעזרת פונקציית `plotPower`, הצגנו את הכוח החלקי עבור כל אלקטרודה כך שהפחתנו את תנאי שמאל מתנאי ימין. הפונקציה מקבלת שני מבני נתונים, המכילים את ממוצע העוצמה וסטיית התקן שלה, עבור כל אחד מהתנאים של אלקטרודה מסוימת, את וקטור התדרים, תיג של איזו אלקטרודה זו ומבנה המכיל את הפונטים הרצויים.

```

figure('units','centimeters','Position',[3.5 4.5 16 10]);hold on;
subplot(2,1,1); plotPower(partialPower.RC3, partialPower.LC3, Hz, 'C3', ...
    Font);
subplot(2,1,2); plotPower(partialPower.RC4, partialPower.LC4, Hz, 'C4', ...
    Font);
shg;

```

הפונקציה עצמה מחשבת את אורך וקטור התדרים, ומחברת בינו ובין וקטור ההפוך לו, ליצירת וקטור פולינדרומי, הדרוש לפונקציית `fill`. לאחר מכן, מחשבת את וקטור ה-Y הראשון (הממוצע ועוד סטיית התקן) ואת הוקטור שישלים אותו לטובת הפונקציה `fill`, הממוצע פחות סטיית התקן. לאחר מכן, הפונקציה מפעילה את פונקציית `fill` עם הערכים שחושבו, לטובת יצירת שטח שייצג את סטיית התקן ובפונקציה `plot` ליצירת וקטור הממוצעים. החישוב והתצוגה התבצעו לשני מבני הנתונים שנקלטו (ראה תמונה 1).

```

% calculate the length of the frequency vector
LNHz = length(Hz);
% calculate x-axis values of standard deviation to be filled
x = (1:LNHz);
x2 = [x, fliplr(x)];
% calculate y-axis values of standard deviation of Data to be filled
curve1Data = (Data.mean + Data.std);
curve2Data = (Data.mean - Data.std);
inBetweenData = [curve2Data, fliplr(curve1Data)];
% calculate y-axis values of standard deviation of subData to be filled
curve1subData = (subData.mean + subData.std);
curve2subData = (subData.mean - subData.std);
inBetweenSubData = [curve2subData, fliplr(curve1subData)];

```

```
% plot
fill(x2, inBetweenData, 'g', 'FaceAlpha', 0.2, 'EdgeColor', 'none'); hold
on;
plot(x, Data.mean, 'r', 'LineWidth', 2); hold on;
fill(x2, inBetweenSubData, 'b', 'FaceAlpha', 0.2, 'EdgeColor', 'none'); hold
on;
plot(x, subData.mean, 'k', 'LineWidth', 2);
```

כל התצוגות האלו (תמונות 1-5) נעשו בניסיון למצוא מאפיינים שיכולים להכיל בתוכם מידע רלוונטי להפרדה ולסיווג.

### חילוץ הנתונים

פונקציית featureExtract נכתבה לטובת חילוץ המאפיינים. קודם כל, חושב ה power spectra המלא עבור כל אלקטרודה והעוצמה המנורמלת של כל אלקטרודה. בנוסף, חושב הכוח החלקי (2.25-6 שניות) והכוח המנורמל עבור כל אלקטרודה, ללא מיצוע.

```
fullPower = struct(...
    'C3', calcPower(C3_Data(:, :), window, overlap, Hz, fs), ...
    'C4', calcPower(C4_Data(:, :), window, overlap, Hz, fs));
% normalize full power
PowerNorm.C3 = normPower(fullPower.C3);
PowerNorm.C4 = normPower(fullPower.C4);

Mean = 0;

partPow = struct(...
    'C3', calcPower(C3_Data(:, :), window, overlap, Hz, fs, Mean, queEndTP), ...
    'C4', calcPower(C4_Data(:, :), window, overlap, Hz, fs, Mean, queEndTP));
% normalize partial power
partNormPow.C3 = normPower(partPow.C3);
partNormPow.C4 = normPower(partPow.C4);
```

לאחר כל חילוץ מאפיין, הופעלה פונקציה בשם labeling שתפקידה לעקוב אחר שמות המאפיינים ומספר המאפיינים הקיימים תחתיהם. הפונקציה מקבלת את מטריצת המאפיינים הנוכחי (לאחר הוספת המאפיין), את מספר המאפיינים הקיימים במטריצה, את מבנה התיוג המכיל את שמות המאפיינים ומספר המאפיינים הקיימים תחת שם זה, את מספר סוגי המאפיינים האחרון שחושב ווקטור-תא המכיל שרשרת בהם כתובים שמות המאפיינים. מופיעה כאן דוגמה לתיוג של מאפייני הממוצע ל-C3 ול-C4.

```
[LNFeat, FeatLabel, FeatIdx] = labeling(Feat, LNFeat, FeatLabel, ...
    FeatIdx, {'MEAN C3', 'MEAN C4'});
```

הפונקציה עצמה מחשבת את מספר התיוגים שנקלטו, את מספר המאפיינים לכל תיוג (מספר המאפיינים לכל תיוג חייב להיות שלם) ואז מעדכנת את מבנה התיוג בשם סוג המאפיין ואת

מספר המאפיינים לכל תיוג. לאחר מכן מעדכנת את מספר סוגי המאפיינים, מספר המאפיינים הכללי ומחזירה אותם ואת מבנה התיוג.

```
LN = size(str,2);

% get the number of features of each new feature type (HAS TO BE WHOLE!)
temp = (size(Feat,1) - LNFeat)/LN;
if temp ~= round(temp)
    error(['uneven proportions of features and labels.'; ...
        'each label has to have the same amount of features.']);
end

% add the label and amount of features for each feature type
for n = 1:LN
    FeatLabel(FeatIdx).name = cellstr(str{n});
    FeatLabel(FeatIdx).count = temp;
    FeatIdx = FeatIdx + 1;
end

% update the current amount of features
LNFeat = size(Feat,1);
```

המאפיינים הראשונים שחולצו הינם הממוצע וסטיית התקן של הכוח החלקי בדציבלים בעבור כל אלקטרודה. מתוצאות חישוב ממוצע הכוח החלקי וסטיית התקן (ראה תמונה 1), ניתן לראות הבדלים גדולים בין הממוצעים וסטיות התקן של העוצמה בטווח מסוים (14-18 Hz בערך) בין הקבוצות השונות, ולכן מאפיינים אלו בטווח זה, חולצו ראשונים. לטובת חילוץ זה, היה עלינו להמיר את יחידות העוצמה לדציבלים.

```
dBpartC3 = pow2db(partPow.C3);

dBpartC4 = pow2db(partPow.C4);

% extract mean of dB power spectra in area of interest
Feat = [Feat; ...
    mean(dBpartC3(area_std_interest,:)); ...
    mean(dBpartC4(area_std_interest,:))];

% extract std of dB power spectra in area of interest
Feat = [Feat; ...
    std(dBpartC3(area_std_interest,:)); ...
    std(dBpartC4(area_std_interest,:))];
```

המאפיין השלישי שחולץ הוא שונות האמפליטודה של הנתונים הגולמיים. בתצוגת 20 הנתונים הגולמיים עבור כל תנאי, גרף מתח כתלות בזמן המדידה מכל ערוץ (ראה תמונות 2 ו-3), צפינו כי בעבור תנאי ימין ישנה פעילות מוגברת בטווח הזמנים שבין כ- 4-6 שניות. לכן שיערנו כי מאפיין זה, מכיל בתוכו מידע שיעזור לאלגוריתם הLDA באימון הסיווג. חישוב השונות התבצע בעזרת פונקציית calcVAR, שמקבלת את הנתונים, גודל חלון חישוב והממד בו אנו רוצים לבצע את



החישוב, ומחזירה את המאפיינים עצמם, מערך המכיל את ערכי טווחי הזמן שבהן חושבו השוניות ומספר החלונות שבהם התבצע החישוב.

```
[VarC3, ~, ~] = calcVAR(C3_Data,VarWind,dim);

[VarC4, tempVAR, numOfWind] = calcVAR(C4_Data,VarWind,dim);

Feat = [Feat; VarC3; VarC4];
```

הפונקציה עצמה מחשבת את אורך האות, ופועלת כך שהחלון הראשון מתחיל באמצע אות הנתונים, מחשבת את שונות אות הנתונים לאורך חלון זה, מתקדמת בקפיצות של גודל החלון, מחשבת את השונות וחוזר חלילה, עד סוף המעבר על האות. נלקחו צעדי זהירות שמא החלון יהיה קטן מדי או שלילי (שהאינדקס הראשון גדול מגודל הנתונים).

```
LN = size(Data,2);

start = LN/2;

finish = start + Batch;

Times = floor(start/Batch);

variances = zeros(Times, size(Data,1));
% allocate variable for variance histogram
VAR = [];

% loop over windows
for n = 1:Times
    if start > LN
        break;
    end
    if finish > LN
        finish = LN;
    end
    % calculate the variance of the signal, in the given dimension
    variances(n,:) = var(Data(:,start:finish),0,dim)';
    % save time of windows
    VAR = [VAR; start finish];
    % update window indices
    start = start + Batch;
    finish = finish + Batch;
end
```

המאפיין הרביעי שחולץ הוא האנרגיה (השטח שנמצא מתחת לpower spectra). בהסתכלות על הספקטרוגרמות (ראה תמונה 4), חילצנו את אזורי העניין בהם בלטו ההבדלים המשמעותיים ביותר באנרגיה בין התנאים השונים בכל אלקטרודה (ראה טבלה 2). אזורי העניין מתייחסים לטווחי תדרים וטווחי זמנים ספציפיים. לצורך חישוב האנרגיה, השתמשנו בפונקציית bigBandPower, המקבלת מערך נתונים עבור אלקטרודה מסוימת, תדר דגימה ומערכים המכילים את טווחי תדרי העניין וזמני העניין, בהתאמה.

```
Feat = [Feat; ...
        bigBandPower(C4_Data', fs, Bands, Time); ...
        bigBandPower(C3_Data', fs, Bands, Time)];
```

הפונקציה עצמה רצה על מספר אזורי העניין שנקלטו ומחשבת את האנרגיה בעזרת הפונקציה `bandpower`, המקבלת את הסיגנל המקורי בטווח מסוים (או את `power spectran` בנוסף לעוד רכיב קלט שמעיד על כך, 'psd'), תדר הדגימה, ואת הזמנים המתאימים לאותו הטווח באזורי העניין ומחזירה סקלר המציין את השטח הכלוא מתחת לגרף `power spectran` בטווחי הזמנים והתדרים הרצויים.

```
LN = size(Freq,1);

% allocate a N-by-M array
Power = zeros(LN,size(Data,2));

% loop over rows in Freq
for n = 1:LN
    % for each range of calculation, calculate the bandpower
    Power(n,:) = bandpower(Data(Time(n,1):Time(n,2),:), fs, [Freq(n,1)
        Freq(n,2)]);
end
```

המאפיין הבא שחולץ הינו `spectral edge` – `index` שבו הסכימה המצטברת של השטח שמתחת לגרף העוצמה המנורמלת חוצה את הסף הרצוי. לאחר משחק קצר עם הפרמטרים, נמצא כי הספים המובילים למאפיינים התורמים להפרדה והסיווג הינם 0.55 ו-0.45 לעוצמה המנורמלת של הכוח המלא והכוח החלקי, בהתאמה (ניתן לראות פרמטרים בטבלה 1). הפונקציה בה השתמשנו היא `specEdge`, המקבלת את העוצמה המנורמלת וערך סף, ומחזירה את `index` האמור.

```
% calculate the spectral edge of the full normalized power
Feat = [Feat; ...
        specEdge(PowerNorm.C3, fullTH); ...
        specEdge(PowerNorm.C4, fullTH)];

% calculate the spectral edge of the partial normalized power
Feat = [Feat; ...
        specEdge(partNormPow.C3, partTH); ...
        specEdge(partNormPow.C4, partTH)];
```

הפונקציה עצמה מחשבת את הסכום המצטבר של העוצמה המנורמלת, יוצרת וקטור סף בינארי ומוצאת את הערך האחרון שהיה מתחת לסף.

```
sumIs = cumsum(Data, 1);
sumIs = sumIs < TH;

diffSum = diff(sumIs);

[spectralEdge, ~] = find(diffSum == -1);
spectralEdge = spectralEdge';
```

המאפיין הבא שחולץ, בעבור העוצמה המנורמלת המלאה והחלקית, הינו spectral moment. מאפיין זה מחושב על ידי הכפלת כל תדר בעוצמה המנורמלת שלו וסכימה מעבר לכל התדרים. ניתן לראות דוגמה לאופן החישוב לעוצמה המנורמלת המלאה, נעשה גם לחלקית.

```
% calculate the spectral moment of the full normalized power
Feat = [Feat; ...
        Hz*PowerNorm.C3; ...
        Hz*PowerNorm.C4];
```

לבסוף, המאפיינים האחרונים שחולצו הם המקדמים הלינאריים של התצוגה הלוגריתמית של העוצמה המלאה והעוצמה החלקית כפונקציה של לוגריתם וקטור התדרים, בעזרת פונקציית calcCoeff, המקבלת את העוצמה של אחת האלקטרודות ואת וקטור התדרים.

```
C3coeff = calcCoeff(fullPower.C3, Hz);

C4coeff = calcCoeff(fullPower.C4, Hz);

Feat = [Feat; C3coeff; C4coeff];
```

הפונקציה עצמה משתמשת בפונקציה polyfit, המקבלת את לוגריתם וקטור התדרים, לוגריתם וקטור הנתונים ומספר המייצג את הדרגה של הפולינום לו נרצה לבצע התאמה.

```
coeff = zeros(2, size(Data, 2));

% loop over windows
for n = 1:size(Data, 2)
    % extract two coefficients (polynom in order 1) for each window
    coeff(:,n) = polyfit(log(Hz), log(Data(:,n)), 1)';
end
```

## בחירת המאפיינים

באמצעות פונקציית בחירת המאפיינים fscnca, המבוססת על שיטת NCA, יצרנו וקטור חדש שערכיו הינם סדר המאפיינים לפי משקלם. השוואה בין התיוגים (המקורי והמושאל) האלו, מוסיפה (או לא) לפונקציית השגיאה, ובעזרת חישוב gradient descent, משנה את משקלות המאפיינים במטרה לצמצם את ההפסד בפונקציית השגיאה. כלומר, המאפיינים מסודרים לפי רמת תרומתם להסתברות באבחנת הנקודות השונות בין הקטגוריות השונות. את פירוט המאפיינים לפי סדר תרומתם ניתן לראות בנספח א.

```
select = fscnca(Feat', label);

[~,descendOrder] = sort(select.FeatureWeights, 'descend');
```

לאימון האלגוריתם לסיווג המאפיינים יצרנו שני וקטורים שמיועדים לשמור את אחוז דיוק סיווג התשריר, ואת אחוז טעות שגיאת האימון. כעת, ביצענו פרמוטציה של indices של צעדי הניסוי ויצרנו משתנה בשם k\_start שיהווה את index הראשון של קבוצת התשריר בכל צעד אימון.

```
ACC = zeros(1,k_fold);
err = zeros(1,k_fold);
% permute the data
perm = randperm(LN);
%first index of validation group
k_start = 1;
```

בשלב זה, התחלנו את התיקוף הצולב על ידי לולאה שרצה k\_fold פעמים (המשתנה שמעיד על מספר קבוצות הfold). חילצנו את indices של התשריר ושל האימון בעבור כל צעד אימון, וחישבנו את הניבוי של קבוצת התשריר ואת אחוז שגיאת האימון באמצעות הפונקציה classify. פונקציה זו מקבלת 3 נתונים – מטריצת המאפיינים של קבוצת התשריר (בעמודות), מטריצת המאפיינים של קבוצת האימון (בעמודות), ווקטור תיוג שאורכו כל מאפיין במטריצת המאפיינים. הפונקציה לומדת את הוקטור שמבצע LDA בצורה האופטימלית על ידי השוואה בין מטריצת המאפיינים של קבוצת האימון לתיוג המתאים להם בוקטור התיוג. כעת הפונקציה יודעת לנבא את שגיאת האימון שלה. לאחר מכן, הפונקציה מנבאת את סיווג קבוצת התשריר באמצעות וקטור זה. חישוב אחוז דיוק התשריר נעשה על ידי השוואת מבני תא בין ניבוי סיווג התשריר של פונקציית classify לבין התיוגים המקוריים.

```
for k = 1:k_fold
    % be aware that the last index of the validation group
    % isn't greater than the length of the data
    if k_start + k_bin - 1 < LN
        validx = perm(k_start:k_start + k_bin - 1);
        trnidx = perm;
        trnidx(k_start:k_start + k_bin - 1) = [];
    else
        validx = perm(k_start:LN);
        trnidx = perm;
        trnidx(k_start:LN) = [];
    end
    % get the validation prediction and error rate of training
    % features
    [ValAcc,err(k)] = ...
    classify(tempFeat(:,validx)',tempFeat(:,trnidx)',label(trnidx),'linear');
    % calculate prediction accuracy
    ACC(k) = mean(cellfun(@(a,b) strcmpi(a,b), ValAcc, label(validx)));
    % advance group
    k_start = k_start + k_bin;
end
```

אם ערך המשתנה הבוליאני simulation הינו אמת – האלגוריתם יבצע סימולציה לצורך בחירת מספר המאפיינים המיטבי שמנבאים באחוז הדיוק הגבוה ביותר. לצורך כך, יצרנו 4 מערכים

לשמירת הנתונים – ממוצע וסטיית תקן התשריר, ממוצע וסטיית תקן ניבוי האימון. כעת, עבור כל מספר מאפיינים שונה הרצנו תיקוף-צולב מסוג k-fold 500 פעמים ומיצענו את תוצאות אלו עבור כל הרצה בכל אחד מהמערכים שהקצנו. לאחר סיום הסימולציה, אנו ממצעים מעבר ל-500 ההרצות עבור כל מספר מאפיינים, מוצאים את המספר שמבא באחוז הדיוק הגבוה ביותר ומציגים את התוצאות (ראה תמונה 5). עבור אותו מספר המאפיינים, אנו מדפיסים את הסטטיסטיקה התיאורית שלו (ראה טבלה 3).

```
if visualize
    % allocate arrays
    MEAN_Val = zeros(n_iter,size(Feat,1));
    STD_Val = zeros(n_iter,size(Feat,1));
    Train_meanPRED = zeros(n_iter,size(Feat,1));
    Train_stdPRED = zeros(n_iter,size(Feat,1));
    % for each number of features
    for b = 1:size(Feat,1)
        % run n_iter times
        for a = 1:n_iter
            % extract the relevant number of features, according to
            % previous calculation
            tempFeat = Feat(descendOrder(1:b),:);

            [K-fold algorithm. Previous code description]

            % calculate mean accuracy and standard deviation of training
            % and validation for each iteration of each number of features
            MEAN_Val(a,b) = (mean(ACC)*100);
            STD_Val(a,b) = (std(ACC)*100);
            Train_meanPRED(a,b) = (1 - mean(err))*100;
            Train_stdPRED(a,b) = std(err)*100;
        end
    end
end
```

לבסוף, במידה וערך המשתנה הבוליאני visualize הוא אמת, הפונקציה תחלק את ערכי המערך VAR (נתוני זמני החלונות בהם היה חישוב השונות) כך שהערכים יהיו ביחידות של שניות ולא index דגימה, מציגה את ההיסטוגרמות עבור כל המאפיינים באמצעות פונקציית plotHist, ומציגה את ההיטל הדו-ממדי והתלת-ממדי לפי שיטת הניתוח PCA בעזרת הפונקציה calcPCA.

```
VAR = VAR/fs;
% plot all features as distinguishing histograms
plotHist(Feat, rightLoc, leftLoc, bin_edges, FeatLabel, TimeInSec, ...
    Bands, VAR, Font);
% compute and plot PCA
calcPCA(Feat', dimComp, rightLoc, leftLoc, Font);
```

הפונקציה plotHist מקבלת את מטריצת המאפיינים, וקטור המעיד על צעדי הניסוי המתויגים לימין וקטור המעיד על צעדי הניסוי המתויגים לשמאל, וקטור המכיל את קצוות עמודות ההיסטוגרמות, מבנה תיג הנתונים, מערכי הזמנים וטווחי התדרים בהם השתמשנו בחילוף מאפייני האנרגיה, המערך בו שמורים זמני החלונות בהם התבצע חישוב ומבנה המכיל את הפונטים הרצויים.

הפונקציה עצמה, מחשבת את ציוני התקן של כל נתוני המאפיינים, את מספר חלונות התצוגה שייפתחו ומקצה משתנה שיהווה index להיסטוגרמות. כמו כן, בעזרת מבנה התיג, הפונקציה מבחינה בין חלונות תצוגה המכילים גרף אחד או יותר, ואז לפי כמות המאפיינים הקיימים תחת סוג מאפיין מסוים ושם, קובעת את גודל חלון התצוגה, כמה גרפים יהיו, את הכותרות שלהם והערכים שיופיעו.

```
Feat = zscore(Feat)';

n_hist = size(FeatLabel,2);

Idx = 1;

for n = 1:n_hist
    h = figure('units','centimeters','Position',[0 0 10 10]); hold on;
    % get number of plots to a single figure
    n_count = FeatLabel(n).count;
    % plot if one per figure
    if n_count == 1
        histogram(Feat(Idx,rightLoc),bin_edges);
        histogram(Feat(Idx,leftLoc),bin_edges);
        Idx = Idx +1;
        [...]
        [...]
    else
        % if more than one plot per figure
        % creates a super title
        sgtitle(FeatLabel(n).name, 'FontSize', Font.sgtitle);
        % run as the number of subplots
        for a = 1:n_count
            % create plot
            subplot(2,n_count/2,a); hold on;
            histogram(Feat(Idx,rightLoc),bin_edges);
            histogram(Feat(Idx,leftLoc),bin_edges);
            Idx = Idx +1;
            [...]
            [...]
            % add decription by name
            if contains(FeatLabel(n).name, 'bandpower')
                h.Position = [0 0 16 16];
                title(['Band: ' num2str(Bands(a,1)) '-'
                    num2str(Bands(a,2)) ...
                    ', Time: ' num2str(Time(a,1)) '-' num2str(Time(a,2))], ...
                    'FontSize', Font.title);
            end
            if contains(FeatLabel(n).name, 'VAR')
                h.Position = [0 0 16 12];
                title(['Amplitude Variance, experiment time: ' ...
                    num2str(VAR(a,1)) '-' num2str(VAR(a,2))], ...
                    'FontSize', Font.title);
            end
            if contains(FeatLabel(n).name, 'coefficients')
                if a == n_count
                    h.Position = [0 0 16 12];
                    title('linear constant coefficient of logarithmic plot',
                        'FontSize', Font.title);
                else
                    title(['linear ' num2str(n_count-a) ' order coefficient
                        of logarithmic plot'], ...
                        'FontSize', Font.title);
                end
            end
        end
    end
end
```

הפונקציה calcPCA מקבלת את מטריצת המאפיינים, הממד אליו אנו רוצים לדחוס (למרות שהתצוגה תהיה לא פחות מ2 ולא יותר מ3), וקטור המעיד על צעדי הניסוי המתויגים לימין, וקטור המעיד על צעדי הניסוי המתויגים לשמאל ואת מבנה הנתונים המכיל את הפונטים הרצויים. הפונקציה עצמה מחשבת את המקדמים המתאימים בסדר המתאים (הו"ע של הע"ע הכי גדולים ועד להכי קטנים = גודל הע"ע מבטא את השונות המוסברת של הנתונים בעזרת הו"ע) באמצעות פונקציית pca, מכפילה את הנתונים במספר הוקטורים הראשונים, התואמים לממד אליו אנו רוצים לדחוס, ואז מציגה את הנתונים. במידה והממד קטן מ2, לא יקרה דבר. אם הממד שווה ל2, תהיה תצוגה דו-ממדית בלבד של הנתונים. במידה והממד אליו דוחסים הינו 3 ומעלה, תהיה תצוגה דו-ממדית ותלת-ממדית של הנתונים.

```
if dimComp>2
    figure('units', 'centimeters', 'Position', [0 0 16 12]);
    sgtitle('PCA', 'FontSize', Font.sgttitle*2);
    subplot(1,2,1); hold on;
    scatter(DataPCAcamp(RLoc,1), DataPCAcamp(RLoc,2), 'r', 'fill');
    scatter(DataPCAcamp(LLoc,1), DataPCAcamp(LLoc,2), 'b', 'fill');
    [...]

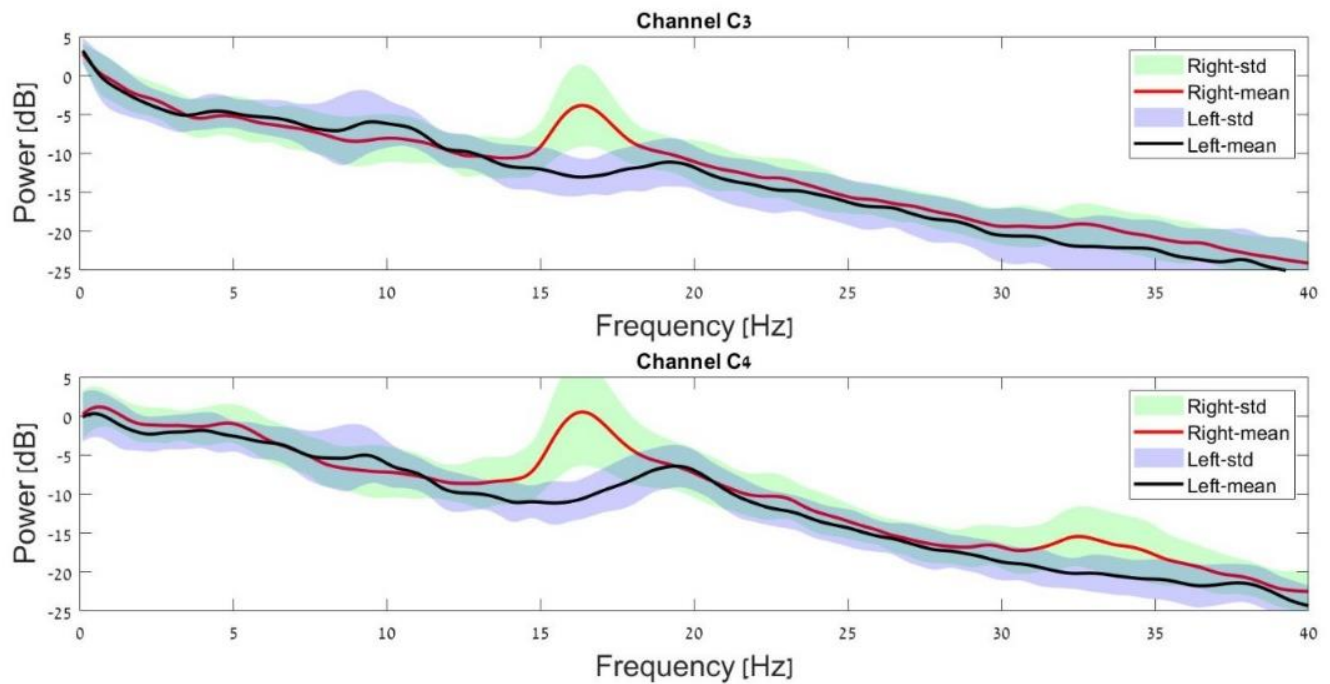
    subplot(1,2,2); hold on;
    scatter3(DataPCAcamp(RLoc,1), DataPCAcamp(RLoc,2), DataPCAcamp(RLoc,3),
             'r', 'fill');
    scatter3(DataPCAcamp(LLoc,1), DataPCAcamp(LLoc,2), DataPCAcamp(LLoc,3),
             'b', 'fill');
    [...]
elseif dimComp == 2
    % plot two dimensional PCA
    figure('units', 'centimeters', 'Position', [0 0 16 12]);
    hold on;
    scatter(DataPCAcamp(RLoc,1), DataPCAcamp(RLoc,2), 'r', 'fill');
    scatter(DataPCAcamp(LLoc,1), DataPCAcamp(LLoc,2), 'b', 'fill');
    [...]
end
```

## תוצאות

. תמונה 1:

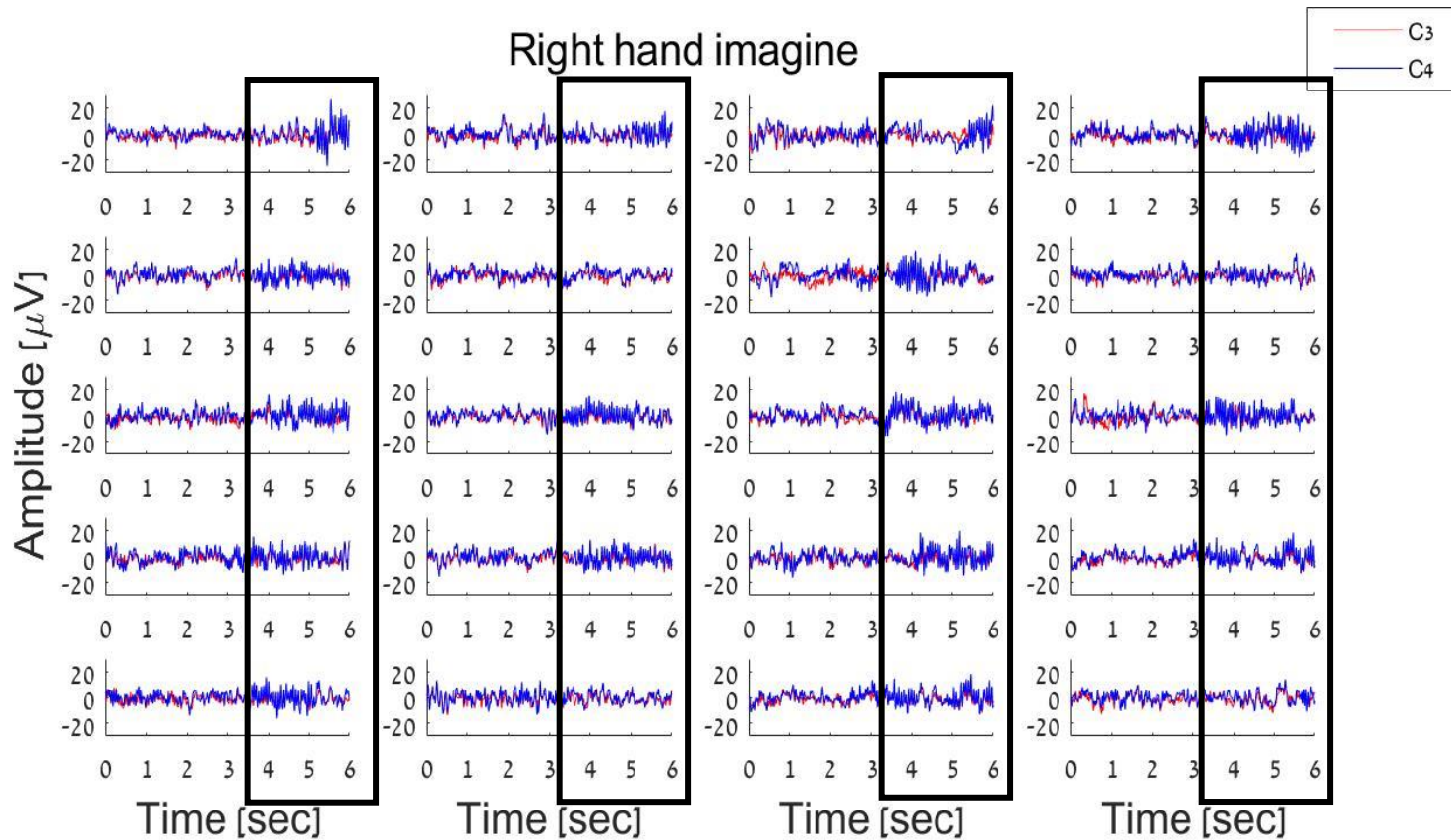
ממוצע העוצמה בדציבלים של כל אחד מתדרי המוח, בכל תנאי ניסוי (גרף אדום – יד ימין, גרף שחור – יד שמאל). כמו כן, בכל תנאי ניתן לראות את סטיות התקן של הממוצע (גרף ירוק – יד ימין, גרף כחול – יד שמאל). ציר ה-X הינו התדר וציר ה-Y הוא העוצמה בדציבלים.

Mean dB power spectra and std



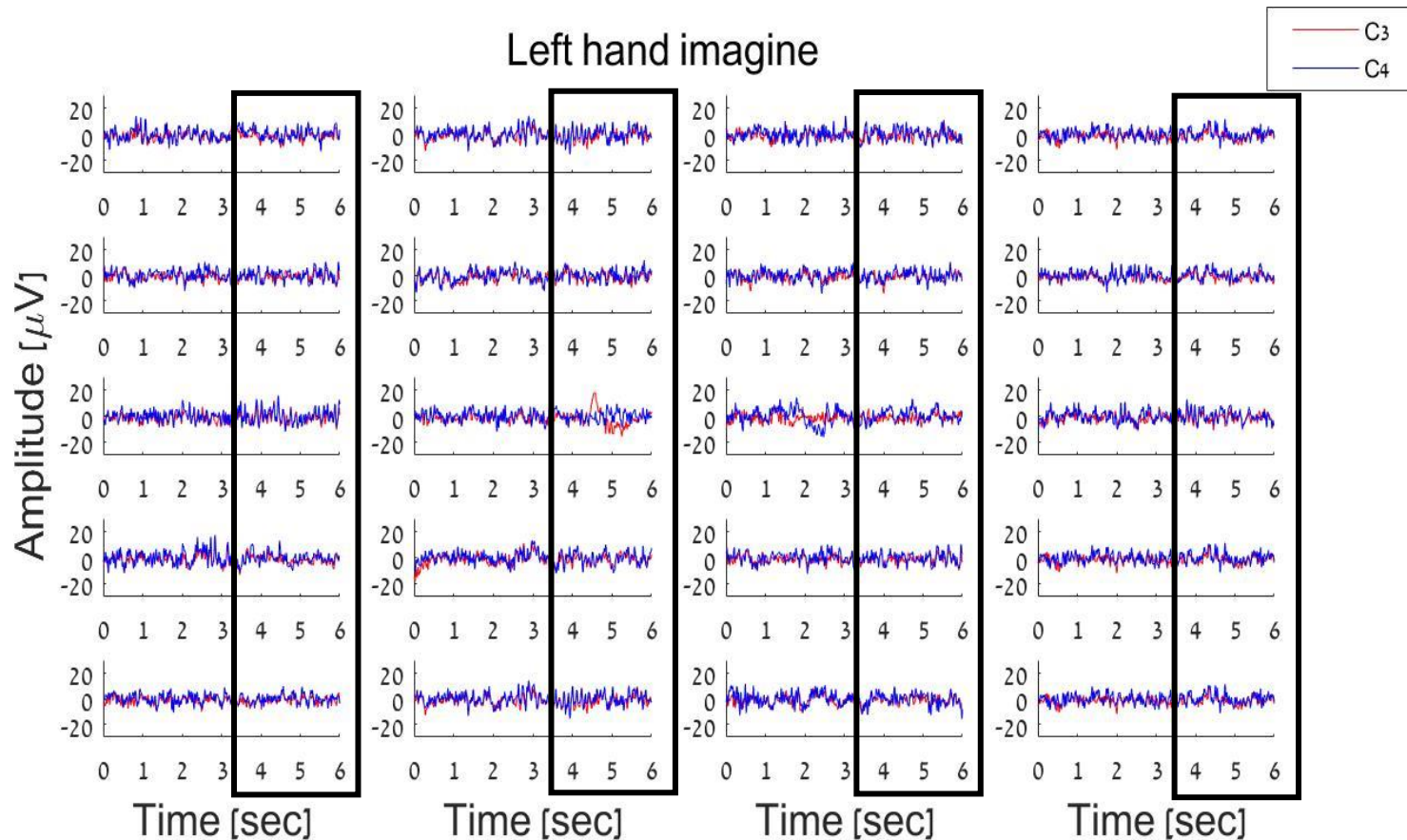


Right hand imagine



### תמונה 3:

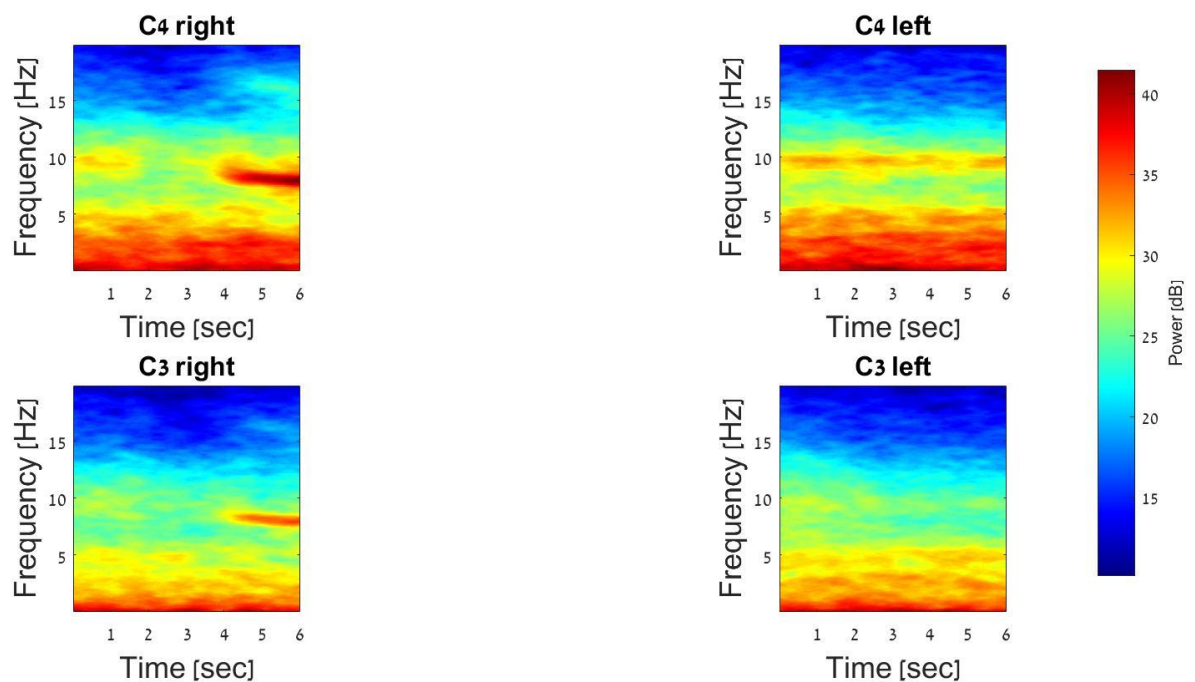
בגרפים אלו מוצגים 20 ניסויים רנדומליים מתוך כלל הניסויים המתויגים ליד שמאל. המרובעים השחורים מדגישים את האזור בו נראית פעולה שונה בין התנאים השונים. ציר ה-X הוא זמן הניסוי בשניות, ציר ה-Y.



#### תמונה 4:

בגרפים אלו ניתן לראות את העוצמה הממוצעת (בדציבלים) המיוחסים לכל תדר (ציר ה-y) בכל נקודת זמן (ציר ה-x). מצד שמאל עליון ועם כיוון השעון: ניתן לראות את העוצמה הממוצעת של אלקטרודה C4 בתנאי מחשבה על ימין, העוצמה הממוצעת של אלקטרודה C4 בתנאי מחשבה על שמאל, העוצמה הממוצעת של אלקטרודה C3 בתנאי מחשבה על שמאל והעוצמה הממוצעת של אלקטרודה C3 בתנאי מחשבה על ימין. סקאלת הצבעים נעה בין מספרים שליליים וחיוביים. ככל שהעוצמה גדולה יותר, כך הצבע מקבל גוון חם יותר (לכיוון האדום).

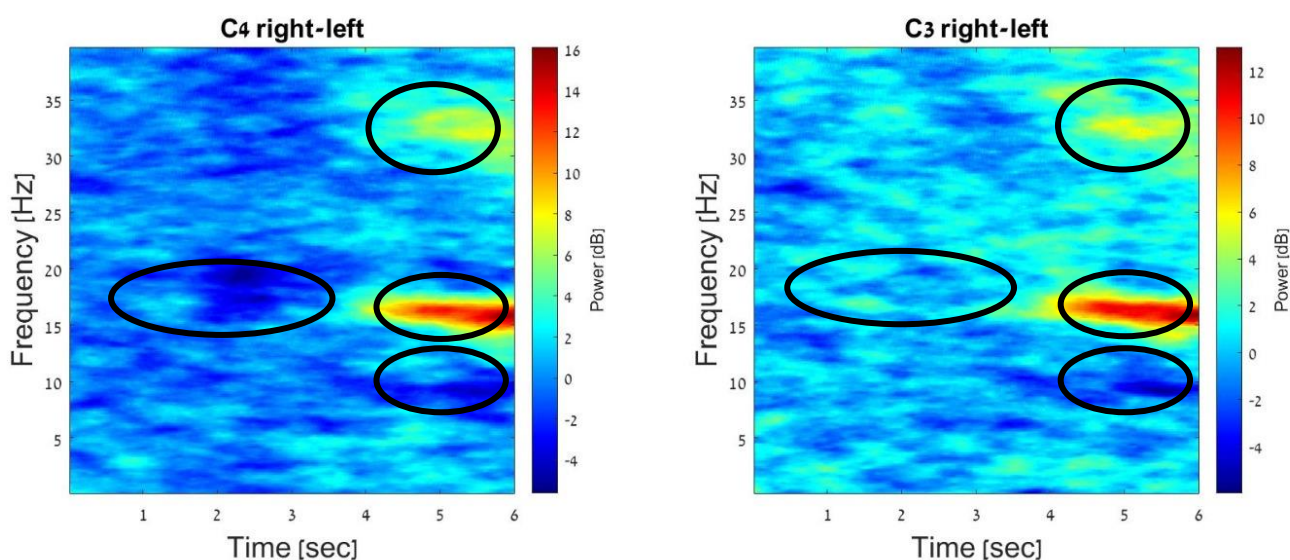
Spectrograms



## תמונה 5:

בגרפים אלו ניתן לראות את הפרשי העוצמה הממוצעת (בדציבלים) המיוחסים לכל תדר (ציר ה-y) בכל נקודת זמן (ציר ה-x). בגרף השמאלי ניתן לראות את הפרש העוצמה הממוצעת בין ימין ושמאל באלקטרודה C4, ובגרף הימני את אותו ההפרש באלקטרודה C3. סקאלת הצבעים נעה בין מספרים שליליים וחיוביים. מאחר ומדובר בהפרשים, ככל שהצבע חם יותר (לכיוון הצבע האדום), כך העוצמה הממוצעת עבור תנאי ימין גדולה יותר מהעוצמה הממוצעת עבור תנאי שמאל.

Spectrograms



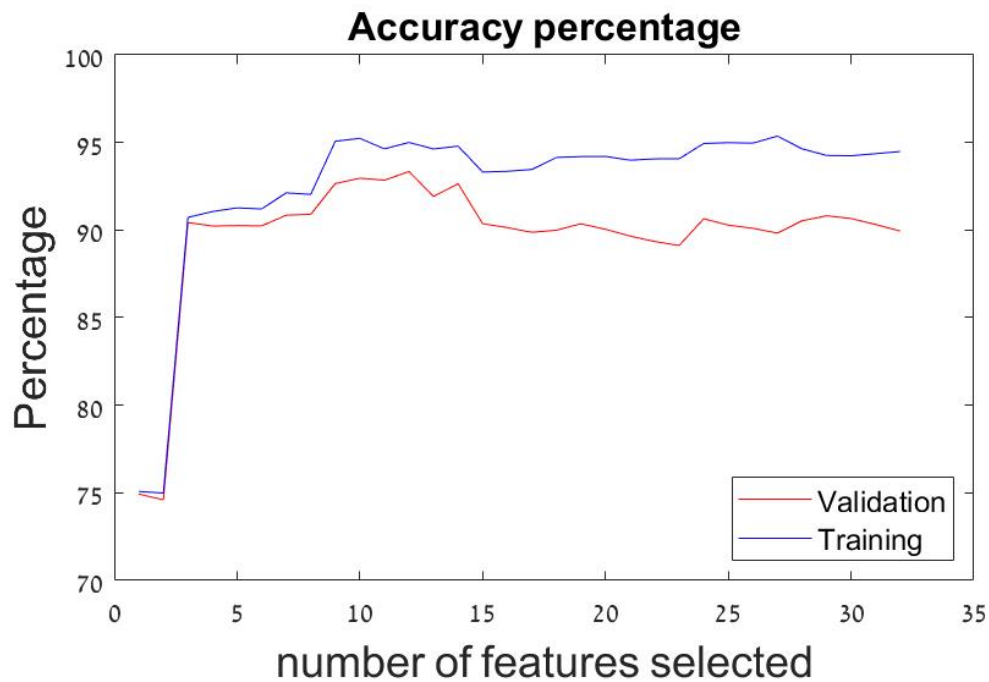
## טבלה 2:

תרגום מספרי של טווחי התדרים והזמנים המזוהים בספקטוגרמות בתמונה 5.

Time [sec]	Bands [Hz]
4-6	14-19
4-6	8-13
4-6	31-36
0.5-4	16-21

### תמונה 6:

בגרף זה ניתן לראות כי כשמספר המאפיינים מועט, אחוז הדיוק הן באימון והן בתשריר נמוכים. ככל שמוסיפים יותר מאפיינים (ציר ה-x), אחוזי הדיוק הממוצעים גדלים (ציר ה-y). עם זאת, כאשר אנו עוברים את המאפיין ה-12, אחוז דיוק התשריר קטן בעוד שאחוז דיוק האימון נשאר קבוע בערך.



### טבלה 3:

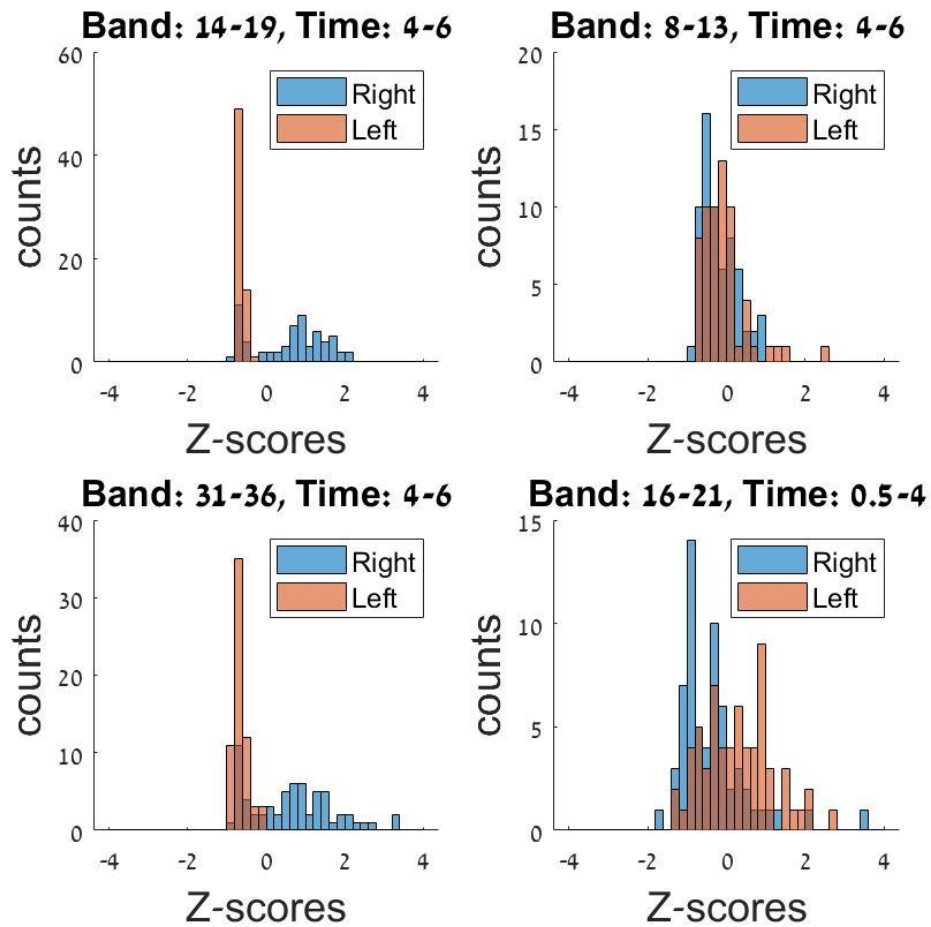
סטטיסטיקה תיאורית, הכוללת ממוצעים וסטיות תקן, של אחוז דיוק התשריר והאימון בבחירת 12 מאפיינים.

Variable Name	Percentage
Mean Validation Accuracy	93.3402%
Validation Accuracy Standard-Deviation	6.9411%
Mean Training Error	4.9991%
Training Error Standard-Deviation	0.74503%

תמונה 7:

בגרפים הבאים מוצגים 4 מהמאפיינים המחולצים באלגוריתם. המאפיינים מוצגים בהיסטוגרמות בצורה כזו שכל צבע מגדיר קטגוריה שונה (ימין ושמאל). בציר ה-X מוצגים ערכי ציוני התקן של נתוני המאפיין, בציר ה-Y מוצגת הספירה של כמות הערכים הללו. ארבעת הגרפים הם של טווחי הזמן והתדר השונים שחולצו באמצעות bandpower מאלקטרודה C4.

bandpower C4

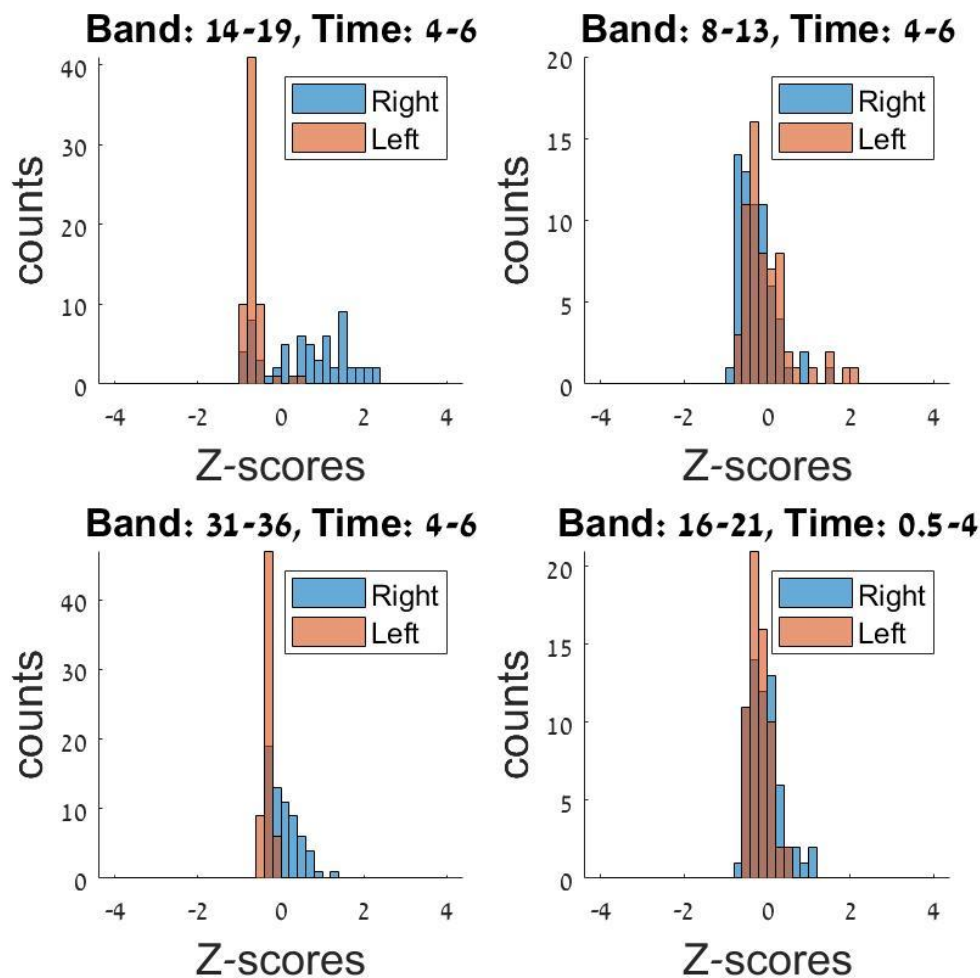




## תמונה 8:

בגרפים הבאים מוצגים 4 מהמאפיינים המחולצים באלגוריתם. המאפיינים מוצגים בהיסטוגרמות בצורה כזו שכל צבע מגדיר קטגוריה שונה (ימין ושמאל). בציר ה-X מוצגים ערכי ציוני התקן של נתוני המאפיין, בציר ה-Y מוצגת הספירה של כמות הערכים הללו. ארבעת הגרפים הם של טווחי הזמן והתדר השונים שחולצו באמצעות bandpower מאלקטרודה C3.

### bandpower C3

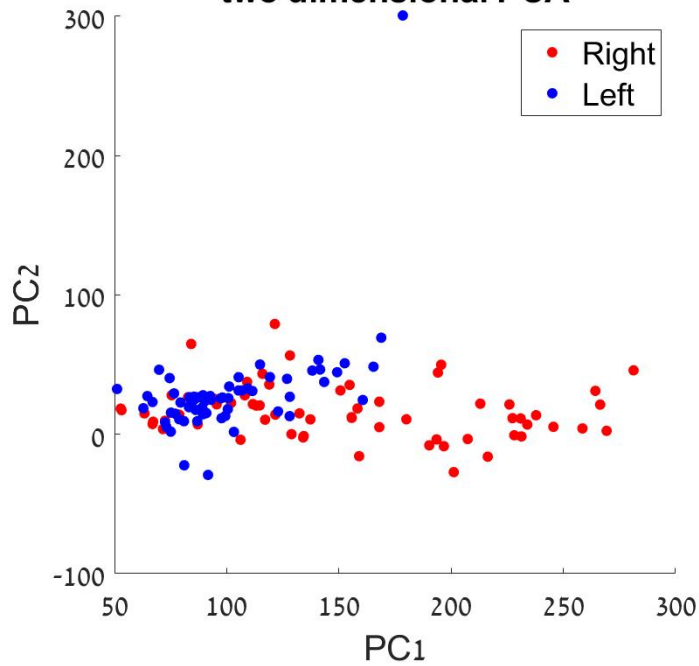


**תמונה 9:**

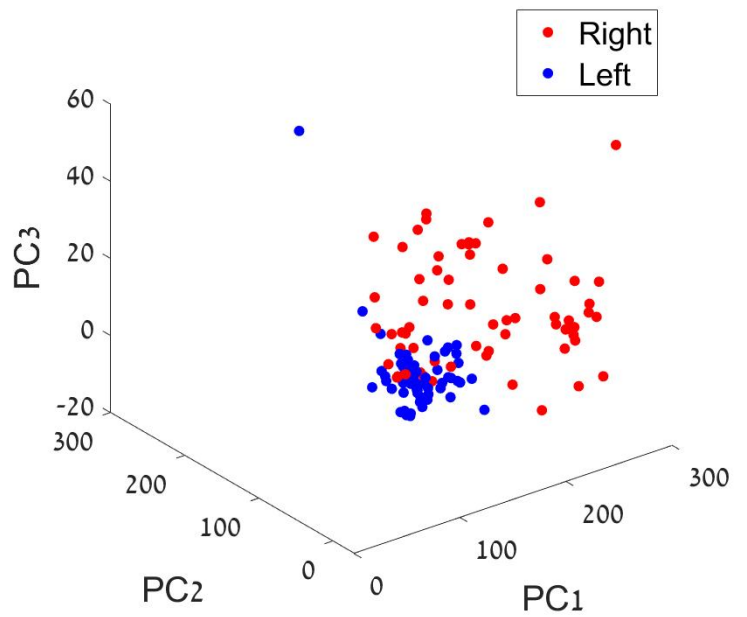
בגרפים אלו ניתן לראות את ההיטל הדו-ממדי והתלת-ממדי של שיטת הפחתת הממדים, PCA. הצבעים השונים מעידים על הקטגוריות השונות. PC1 הינו הציר החד-ממדי המכיל את מקסימום השונות המוסברת, אחריו PC2 ולבסוף PC3.

## PCA

**two dimensional PCA**



**three dimensional PCA**





## דיון

### סימולציה

בתמונה 5 ניתן לראות את אחוזי הדיוק הממוצעים של חלוקת קבוצת האימון וניבוי קבוצת התשריר בעבור מספר מאפיינים שונה. בסימולציה שערכנו, לאחר בחירת סדר המאפיינים בעזרת פונקציית fscnca, הרצנו 500 פעמים תיקוף-צולב מסוג k-fold עבור כל קבוצות המאפיינים – ממאפיין בודד ועד כלל המאפיינים יחדיו (הוספת המאפיינים נקבעה על פי הסדר שניתן לנו מהפונקציה של fscnca). מאחר וכל מאפיין מסביר חלק אחר בשונות (גם אם קיימת חפיפה), שילוב המאפיינים תורם להעלאת יכולת ההבחנה. כפי שניתן לראות בנתונים, שימוש במאפיין אחד בלבד (גם הטוב ביותר) מניב תוצאות פחות טובות מאשר שימוש במספר מאפיינים יחדיו. מספר המאפיינים המיטבי בו מצאנו לנכון להשתמש הינו 12, הוא הנותן את אחוז דיוק התשריר הגבוה ביותר.

### המאפיינים שחולצו

כל המאפיינים שחולצו, הוצגו בהיסטוגרמות הדומות להיסטוגרמות המוצגות בפרק התוצאות (לדוגמה, ראה תמונות 7-8. לכלל היסטוגרמות המאפיינים, ראה נספח ב). אם בחירת המאפיינים הייתה נעשית בצורה ידנית, היינו משתמשים בהפרדה הוויזואלית בין ערכי ציוני התקן של המאפיינים בכל קטגוריה על מנת לבחור את המאפיינים שבהם ההפרדה אופטימלית. אנו עדים לכך שלאחר משקול המאפיינים שלנו בשיטת NCA, לא בהכרח אותם מאפיינים שהיינו בוחרים וויזואלית, יוסיפו לאחוז דיוק הניבוי באבחנה בין הקטגוריות.

### בחירת המאפיינים

כפי שניתן לראות מהגרף, ולהסיק מהנתונים (על ידי שימוש בפקודה max), אחוז הדיוק הממוצע בתשריר הוא הגבוה ביותר, בבחירת 12 מאפיינים. זו הסיבה שמספר זה נבחר להיות מספר המאפיינים שילקחו ממטריצת המאפיינים, גם בהרצת האלגוריתם שלא בסימולציה (אחד מהפרמטרים הוא FeatNum שערכו 12).

### השוואה ל-PCA

להבדיל משיטת הסיווג LDA בה עבדנו בעבודה זו, ניתן לראות שבשימוש בשיטת ה-PCA לא ניתן לבצע הפרדה ליניארית בין הקטגוריות השונות. הסיבה לכך היא החישוב השונה של שיטות אלו. ה-PCA לא מתחשב בסיווגים השונים כאשר הוא מחפש את הצירים המכילים את השונות המוסברת הגדולה ביותר ואילו ה-LDA מתייחס לתיוג הקטגוריות בכך שהוא מחפש את השונות התוך-קבוצתית הקטנה ביותר ואת השונות הבין-קבוצתית הגדולה ביותר.

**תוצאות מבחן התוצאה**

96.88

הסתיים

הוגש 13:50, 29/07/2020

בבדיקת נתוני המבחן על אלגוריתם ההפרדה שאימנו, הגענו לאחוזי דיוק גבוהים במיוחד. תוצאות אלו מעידות על כך שהאלגוריתם אכן למד מאפיינים רלוונטים לביצוע הפרדה בין הקטגוריות השונות.

- Conceição, R. C., O'halloran, M., Glavin, M., & Jones, E. (2011). Evaluation of features and classifiers for classification of early-stage breast cancer. *Journal of Electromagnetic Waves and Applications*, 25(1), 1-14.
- Goldberger, J., Hinton, G. E., Roweis, S. T., & Salakhutdinov, R. R. (2005). Neighbourhood components analysis. In *Advances in neural information processing systems* (pp. 513-520).
- Hossain, A. B. M. A., Rahman, M. W., & Riheen, M. A. (2015). Left and right hand movements EEG signals classification using wavelet transform and probabilistic neural network. *International Journal of Electrical and Computer Engineering (IJECE)*, 5(1), 92-101.
- Jeannerod, M. (1994). The representing brain: Neural correlates of motor intention and imagery. *Behavioral and Brain sciences*, 17(2), 187-202.
- Jeannerod, M. (1995). Mental imagery in the motor context. *Neuropsychologia*, 33(11), 1419-1432
- Jin, J., Miao, Y., Daly, I., Zuo, C., Hu, D., & Cichocki, A. (2019). Correlation-based channel selection and regularized feature optimization for MI-based BCI. *Neural Networks*, 118, 262-270.
- Zhao, W., Krishnaswamy, A., Chellappa, R., Swets, D. L., & Weng, J. (1998). Discriminant analysis of principal components for face recognition. In *Face Recognition* (pp. 73-85). Springer, Berlin, Heidelberg.

**נספחים**

**נספח א**

Feature Selecti on order	Serial numb er of featur e	Name of feature	Electr ode	Feature Selecti on order	Serial numb er of featur e	Name of feature	Electr ode
1	24	Spectral moment of Partial normalized power	C4	17	1	Mean power in range 14- 18 Hz	C3
2	10	Bandpower. 8-13 Hz (4- 6 sec)	C4	18	3	Std of power in range 14- 18 Hz	C3
3	8	Amplitude Variance in experiment time (4.5-6 sec)	C4	19	27	1 <sup>st</sup> order linear coefficient of full power	C4
4	20	Spectral edge partial normalized power	C4	20	21	Spectral moment of full normalized power	C3
5	18	Spectral edge of full normalized power	C4	21	12	Bandpower. 16-21 Hz (0.5-4 sec)	C4
6	6	Amplitude Variance in experiment time (4.5-6 sec)	C3	22	31	1 <sup>st</sup> order linear coefficient of partial power	C4

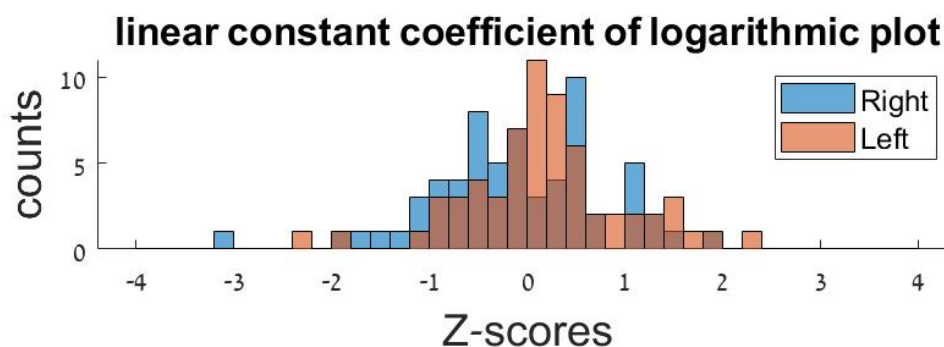
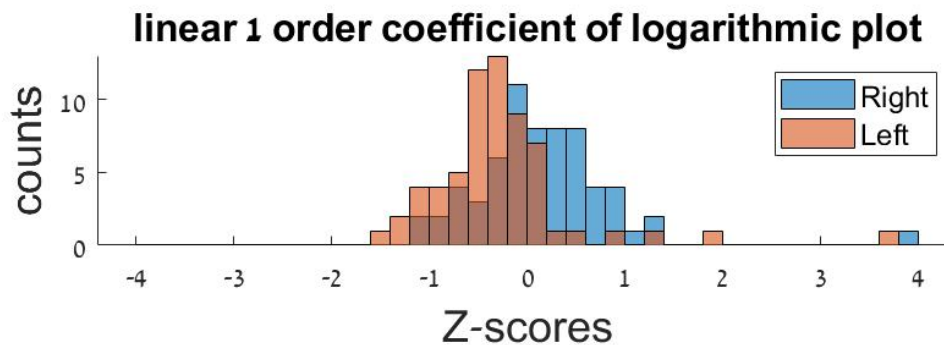
7	7	Amplitude Variance in experiment time (3-4.5 sec)	C4	23	25	1 <sup>st</sup> order linear coefficient of full power	C3
8	14	Bandpower. 8-13 Hz (4- 6 sec)	C3	24	30	Free constant linear coefficient of partial power	C3
9	17	Spectral edge of full normalized power	C3	25	16	Bandpower. 16-21 Hz (0.5-4 sec)	C3
10	5	Amplitude Variance in experiment time (3-4.5 sec)	C3	26	28	Free constant linear coefficient of full power	C4
11	19	Spectral edge partial normalized power	C3	27	13	Bandpower. 14-19 Hz (4- 6 sec)	C3
12	22	Spectral moment of full normalized power	C4	28	29	1 <sup>st</sup> order linear coefficient of partial power	C3
13	4	Std of power in range 14-18 Hz	C4	29	11	Bandpower. 31-36 Hz (4- 6 sec)	C4
14	9	Bandpower. 14-19 Hz (4-6 sec)	C4	30	26	Free constant linear	C3

						coefficient of full power	
15	2	Mean power in range 14-18 Hz	C4	31	32	Free constant linear coefficient of partial power	C4
16	23	Spectral moment of partial normalized power	C3	32	15	Bandpower. 31-36 Hz (4-6 sec)	C3

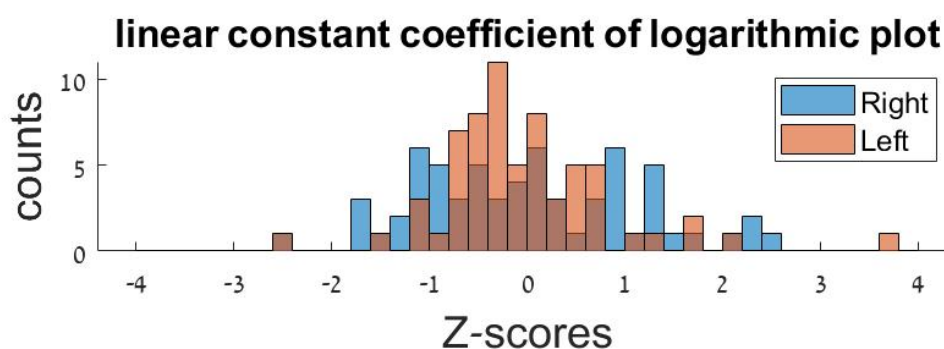
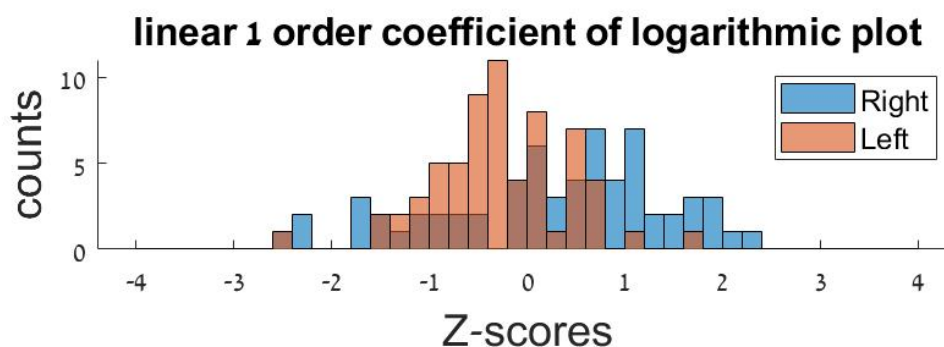
## נספח ב

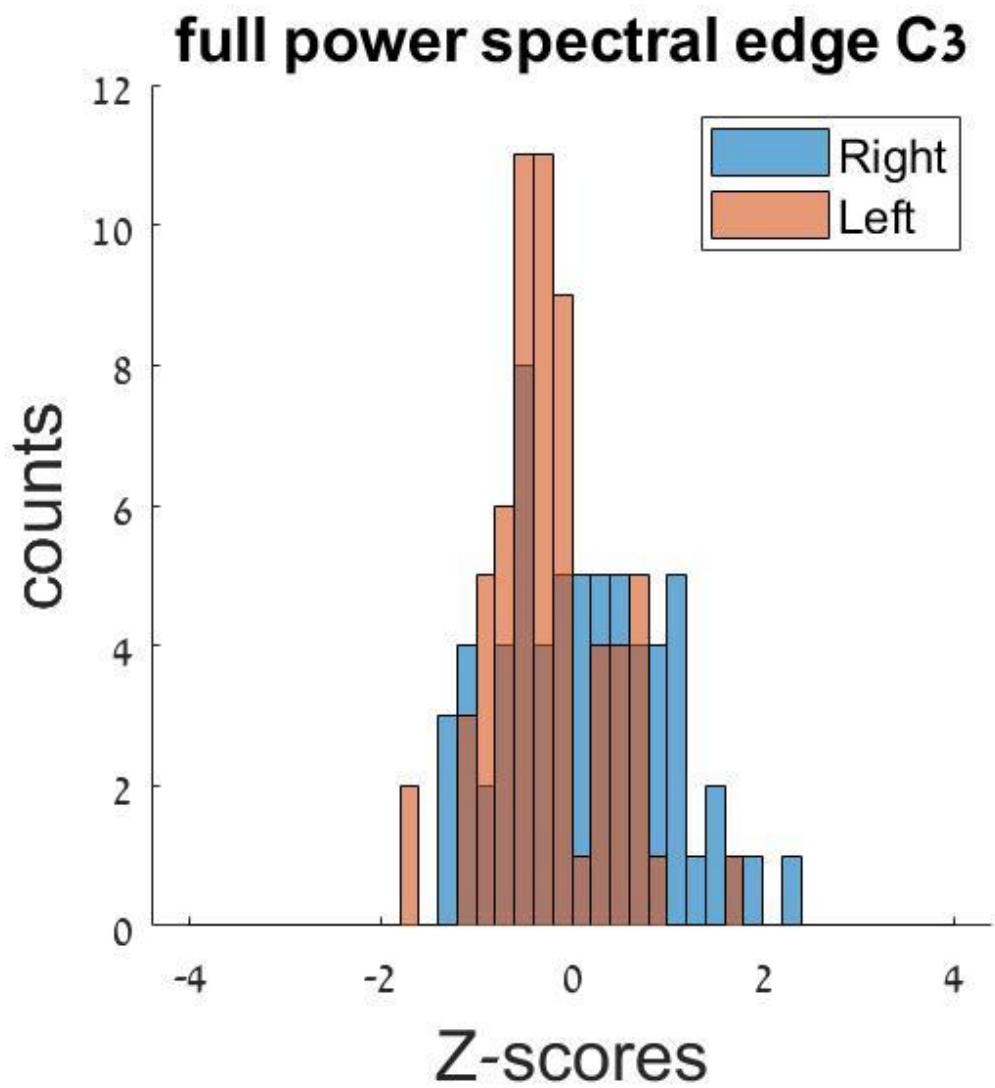
בגרפים הבאים ניתן לראות את ציוני התקן של כל אחד מהמאפיינים בחלוקה לשמאל (עמודות אדומות) וימין (עמודות כחולות).

### full power coefficients C3

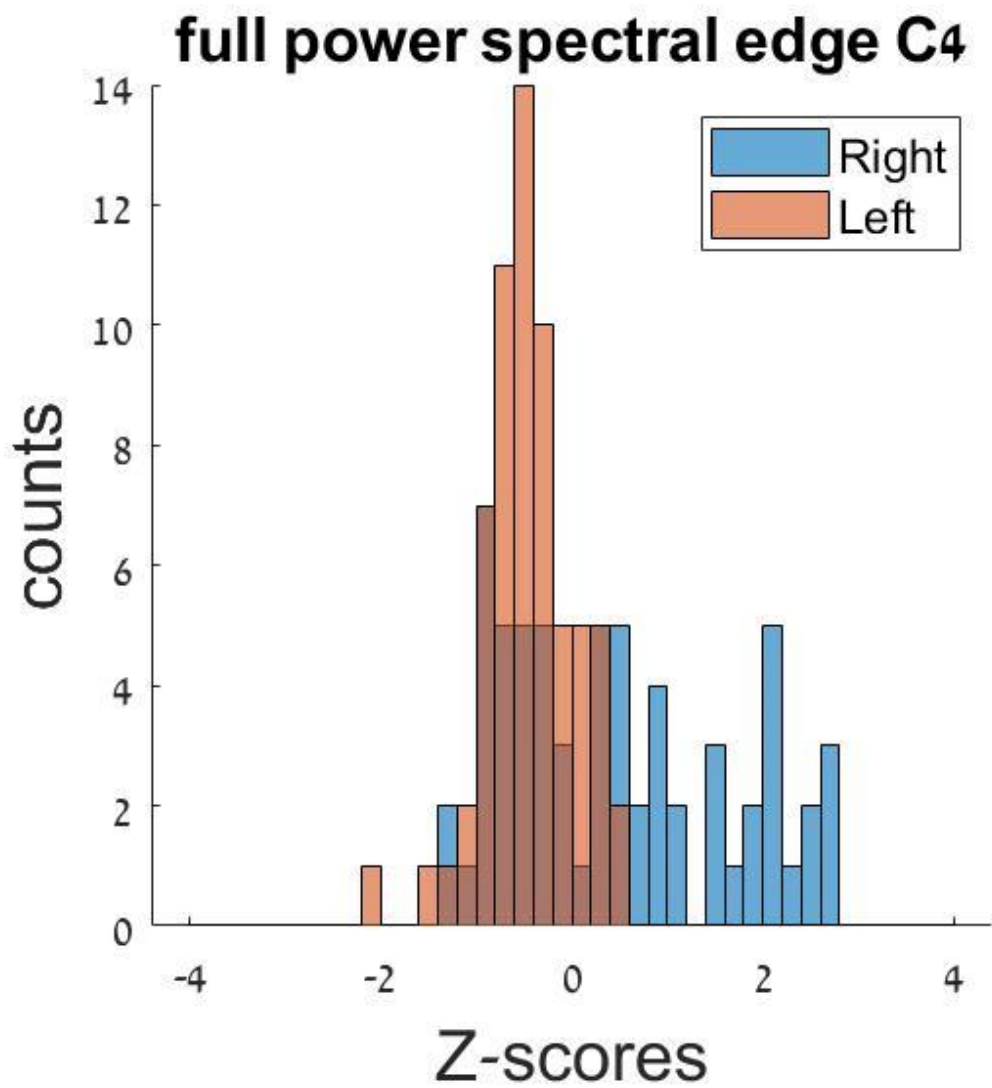


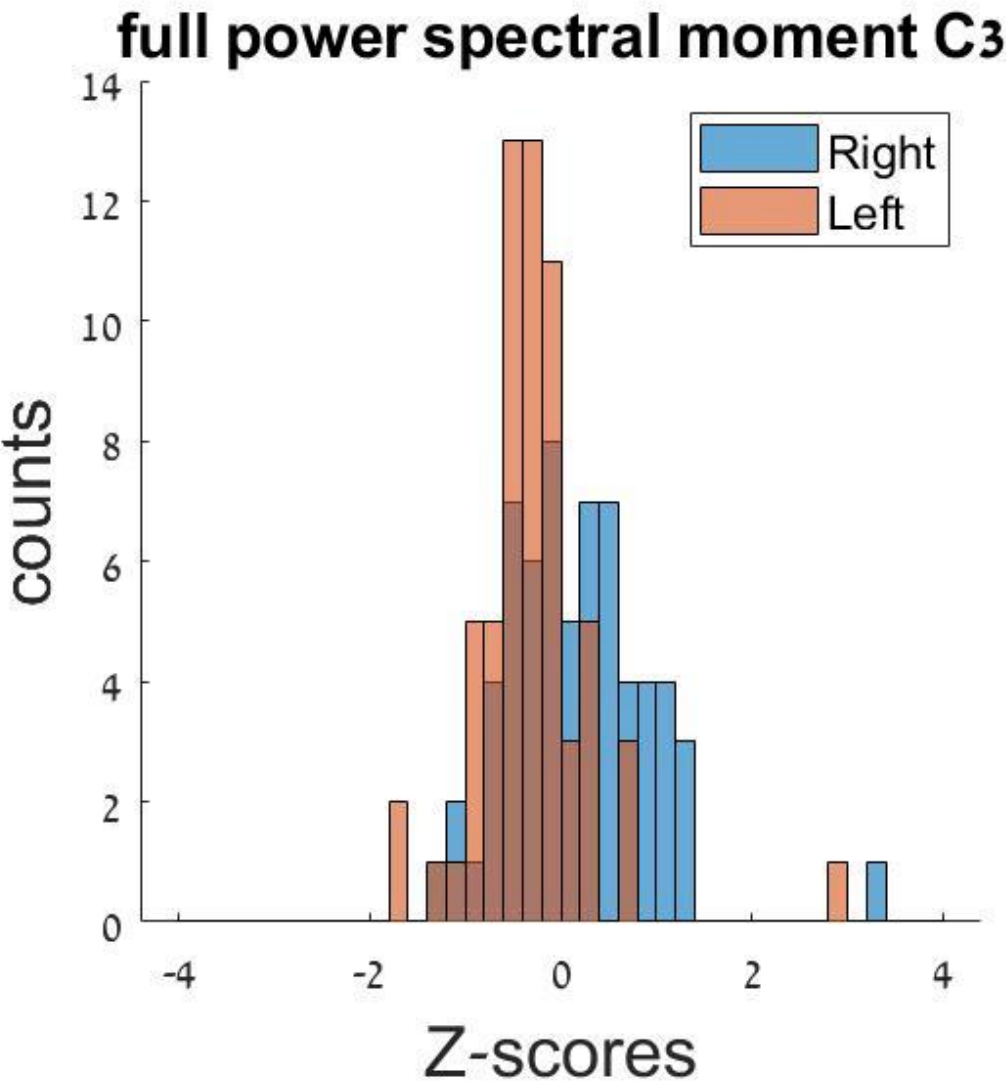
### full power coefficients C4

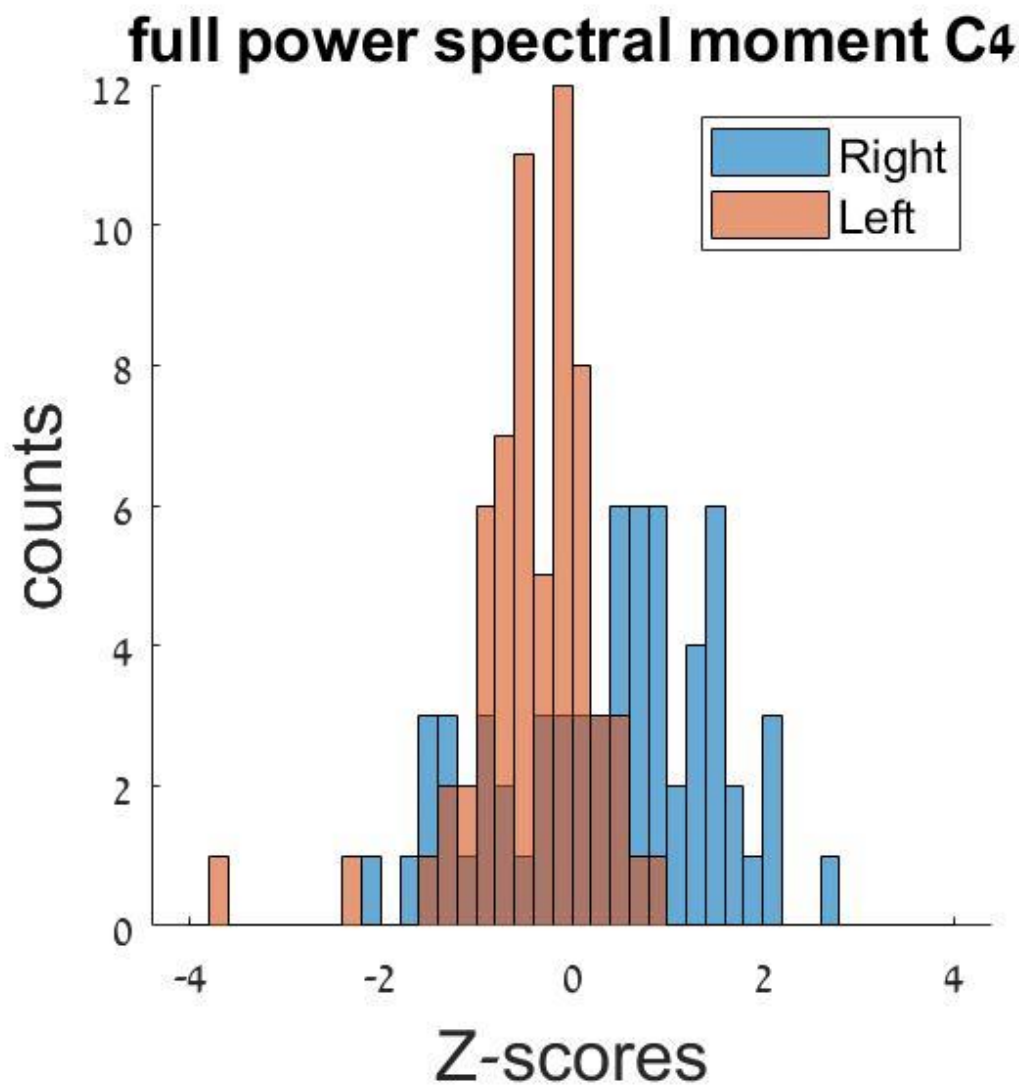


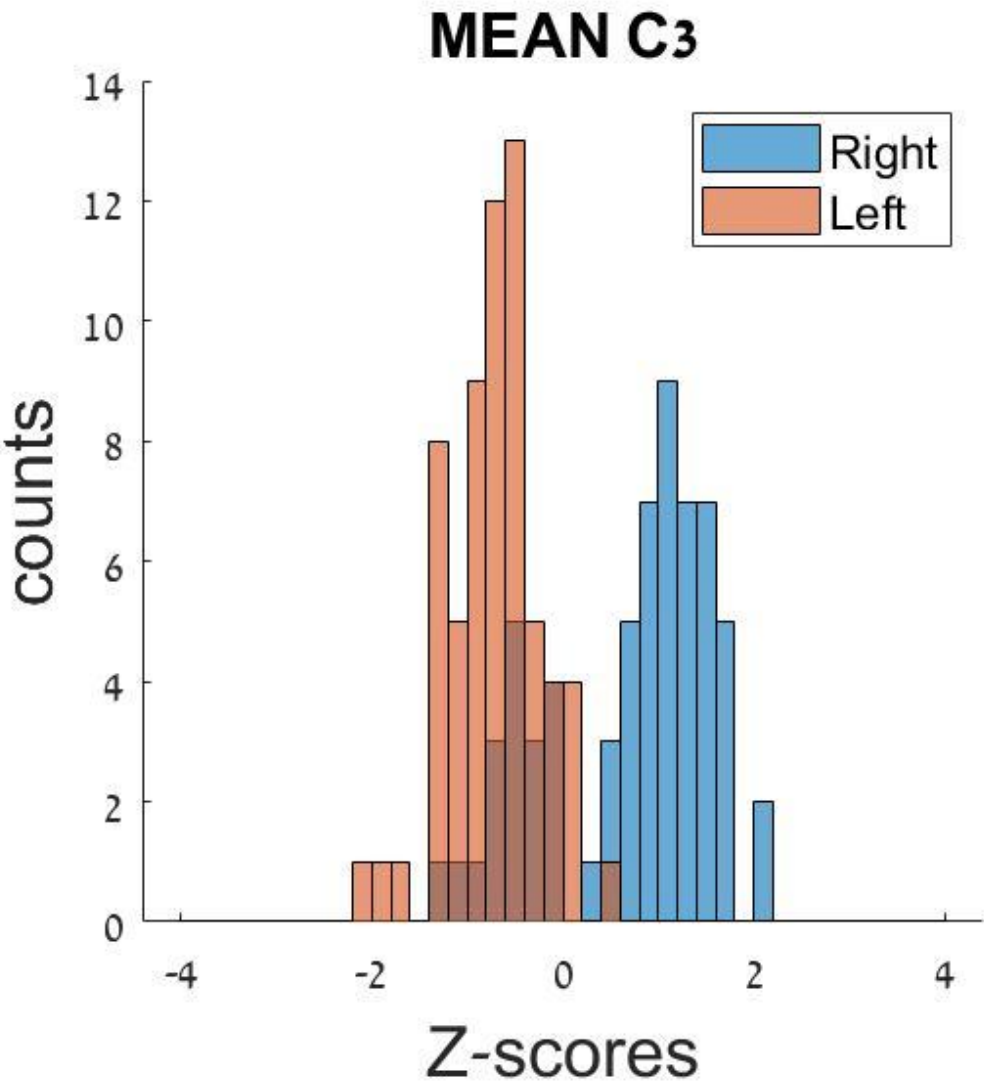


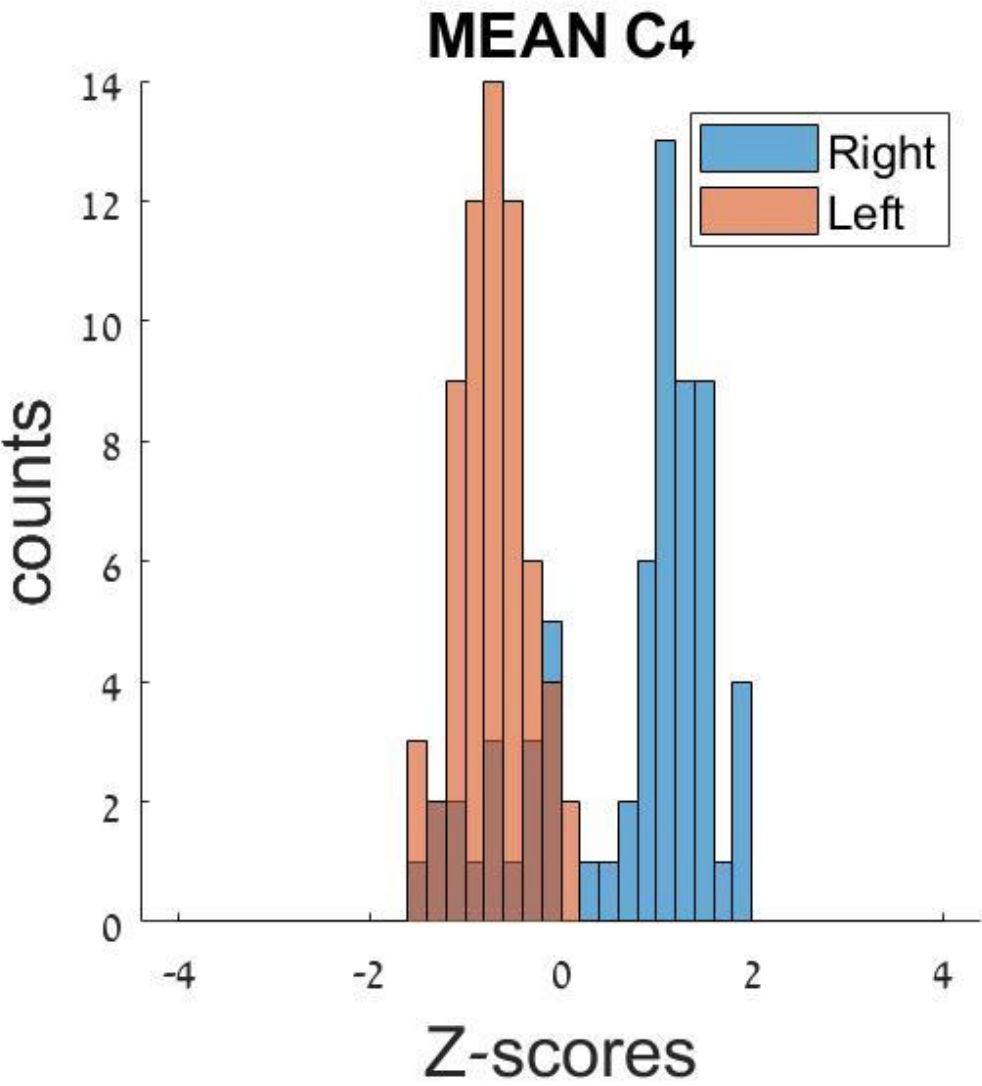




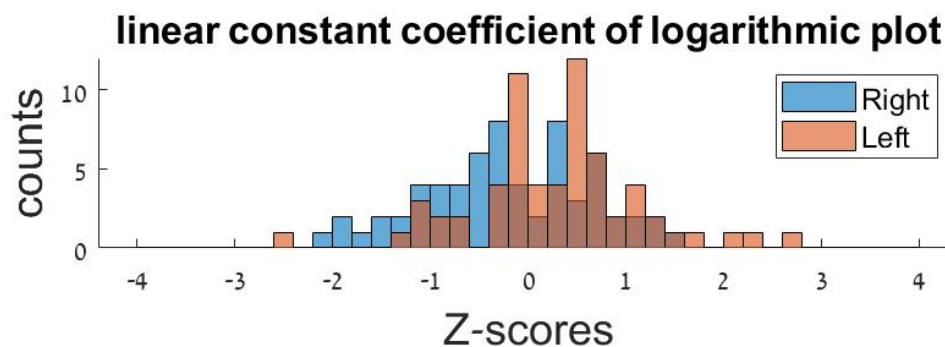
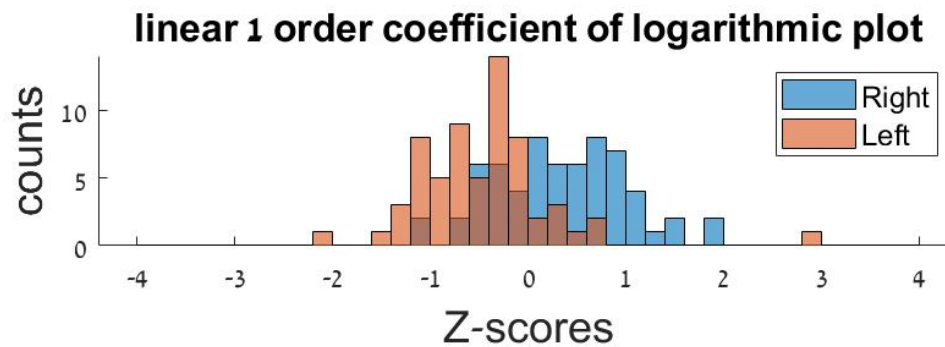




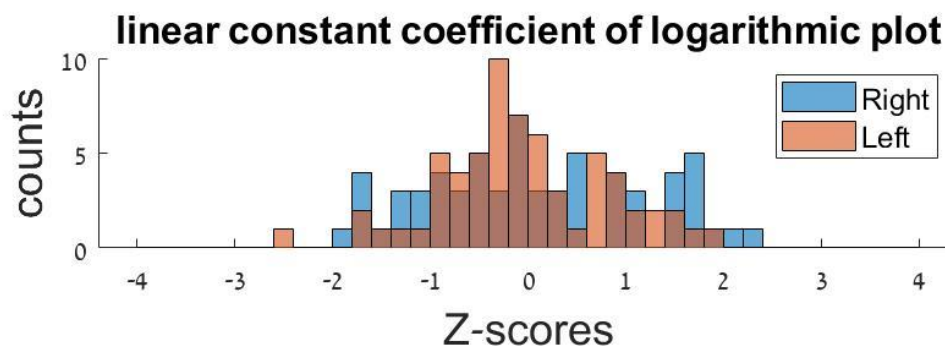
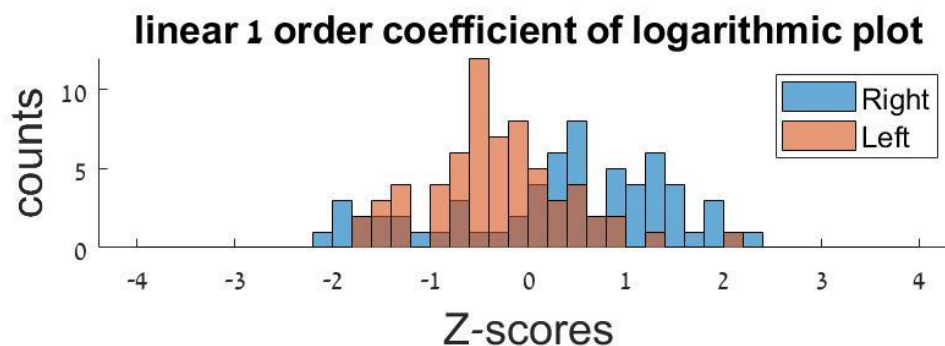


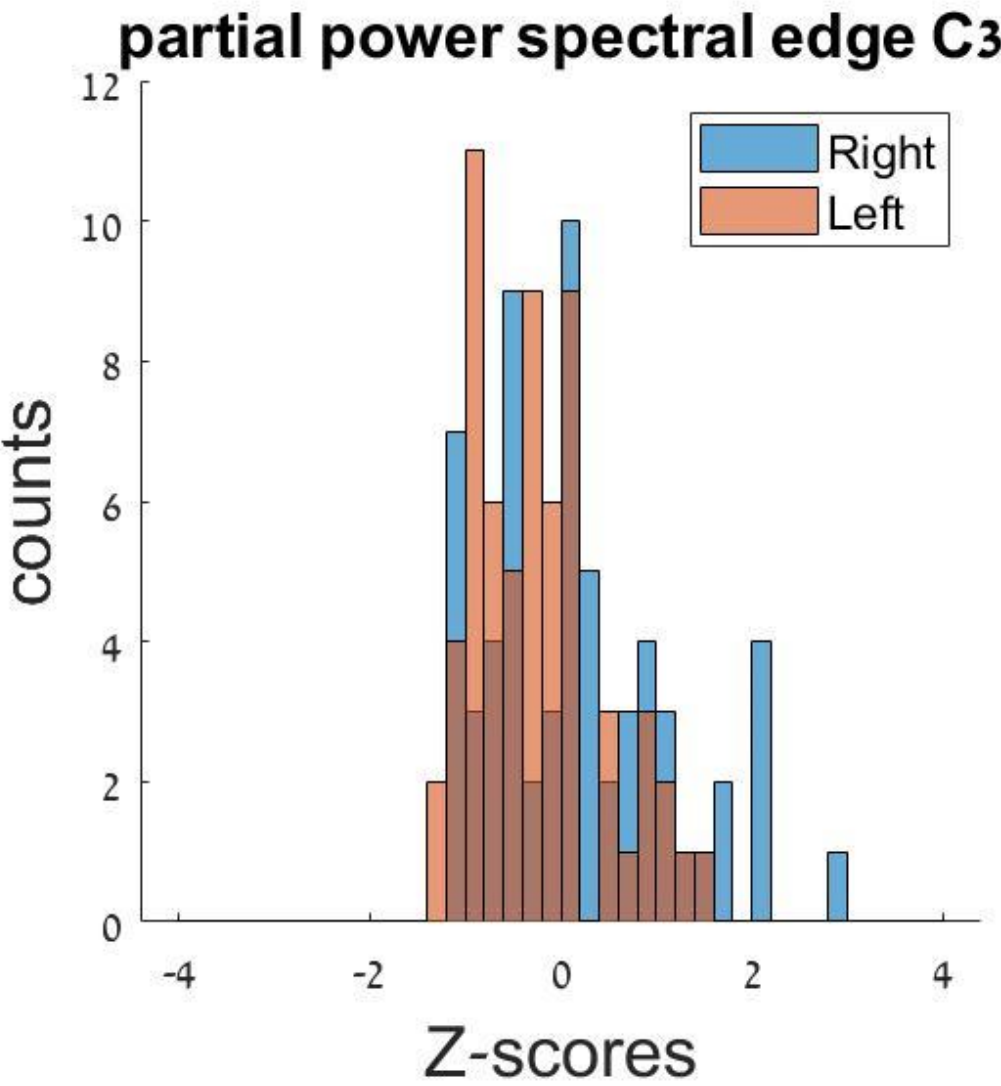


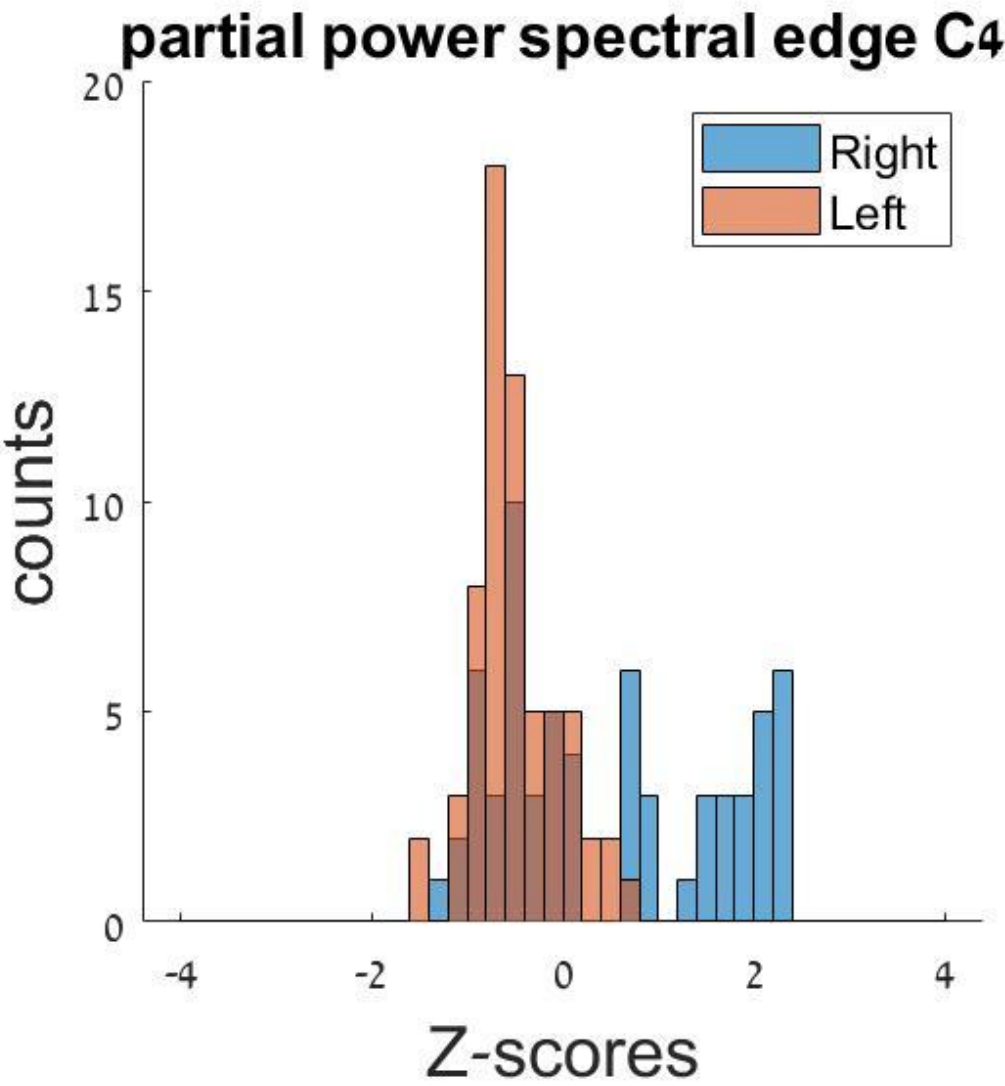
### partial power coefficients C3



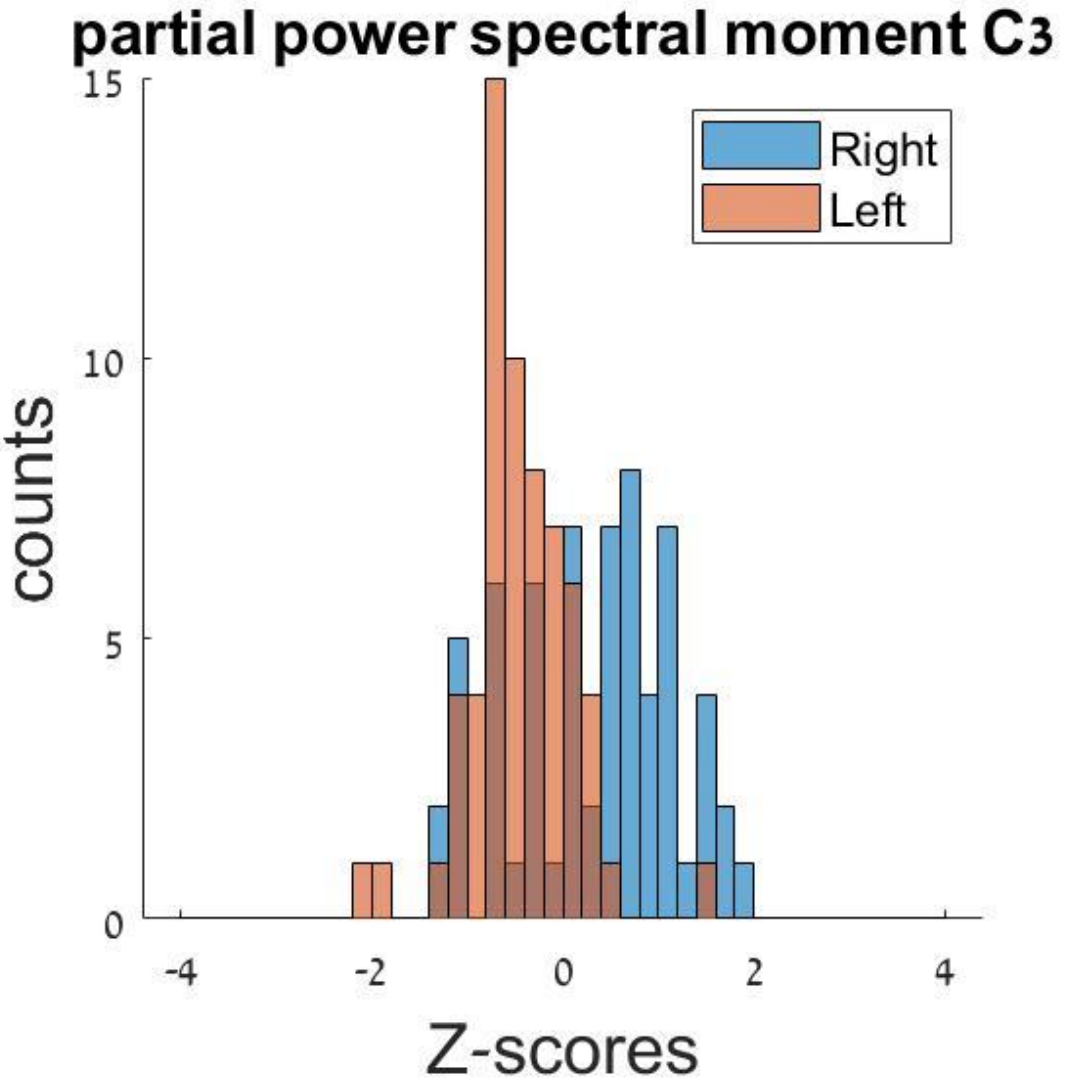
### partial power coefficients C4

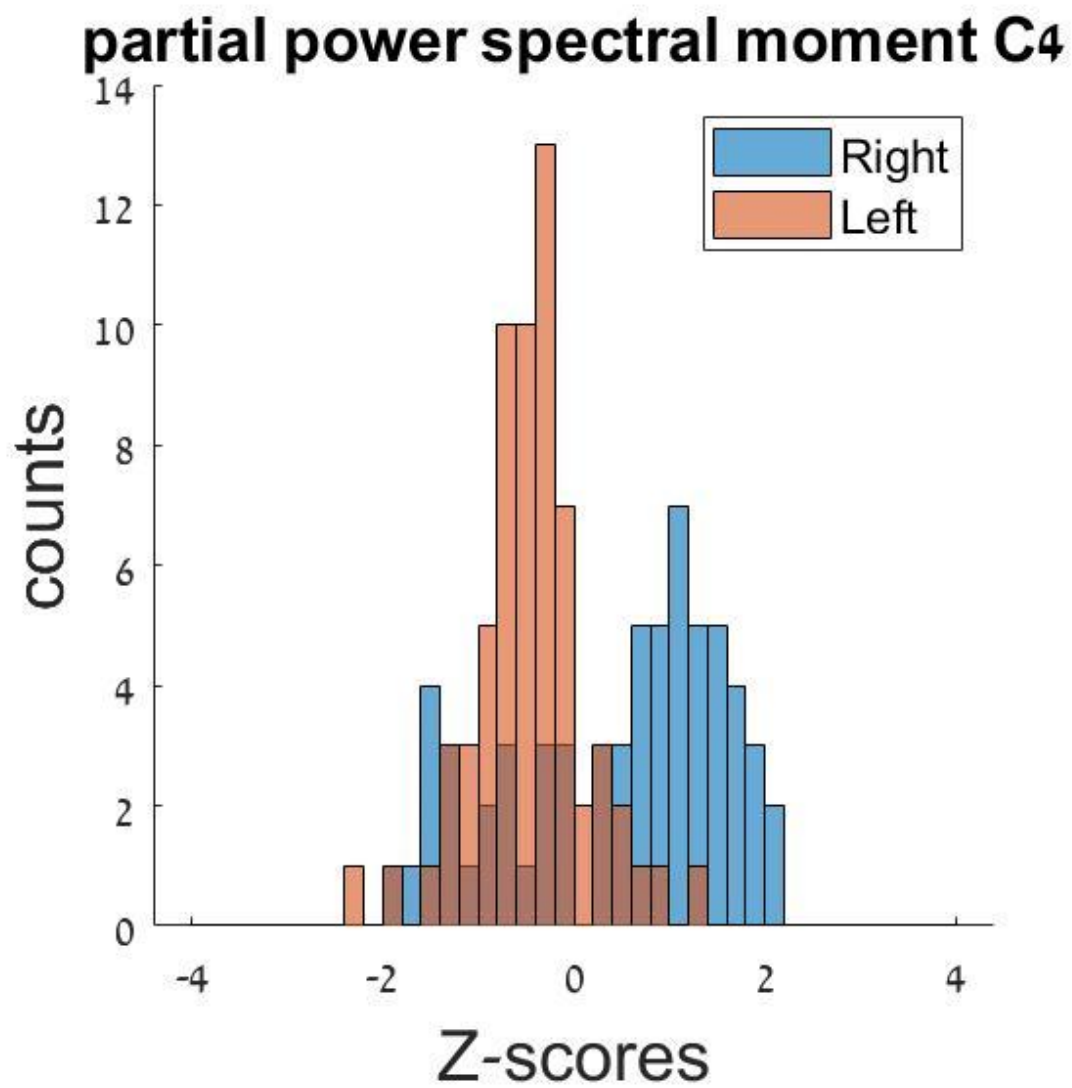


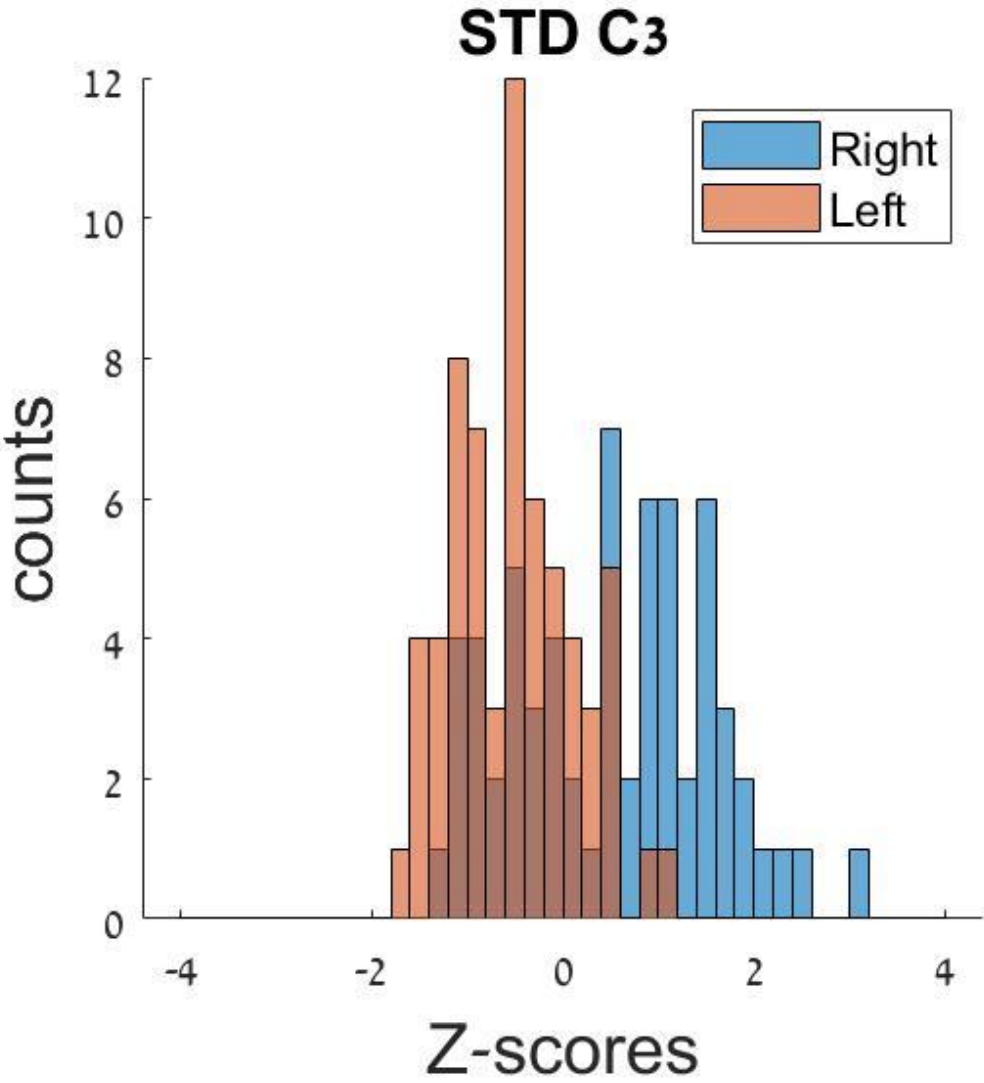


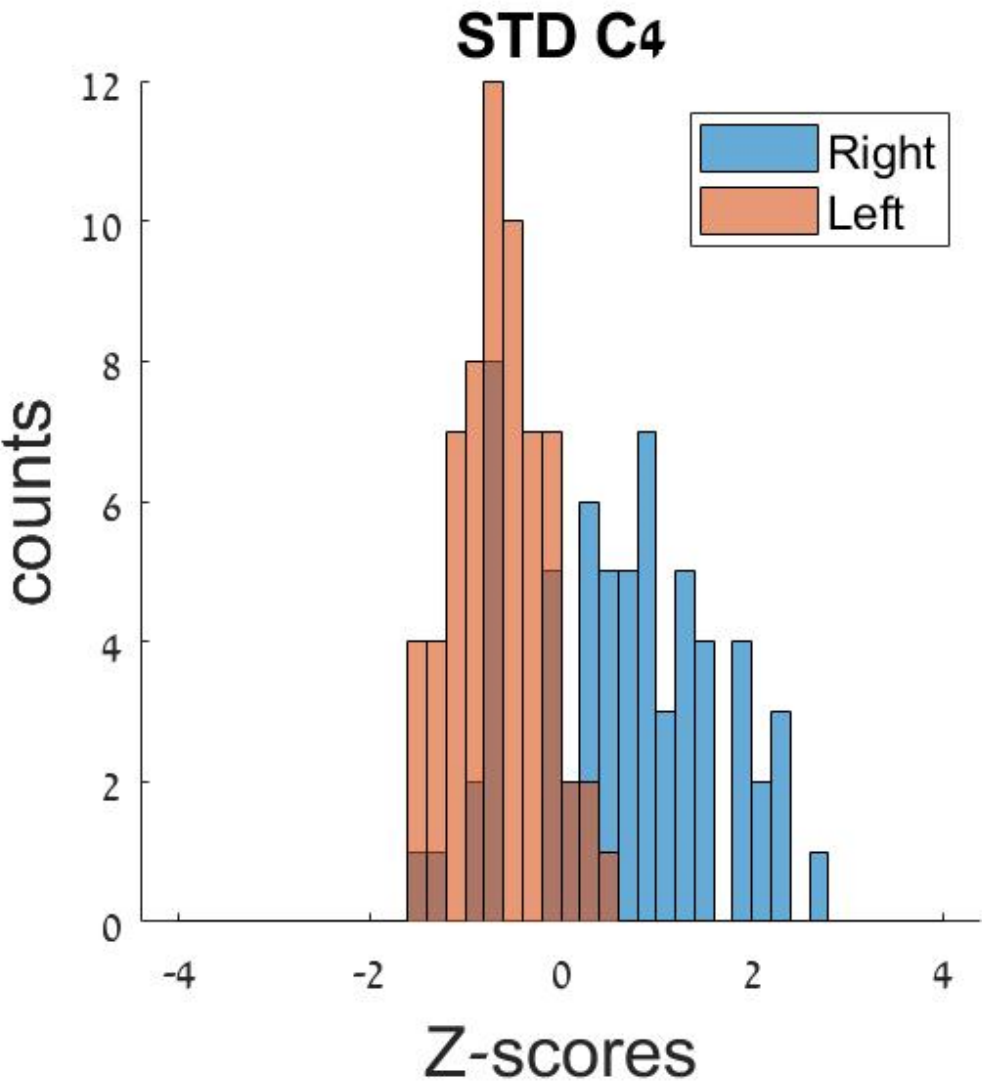




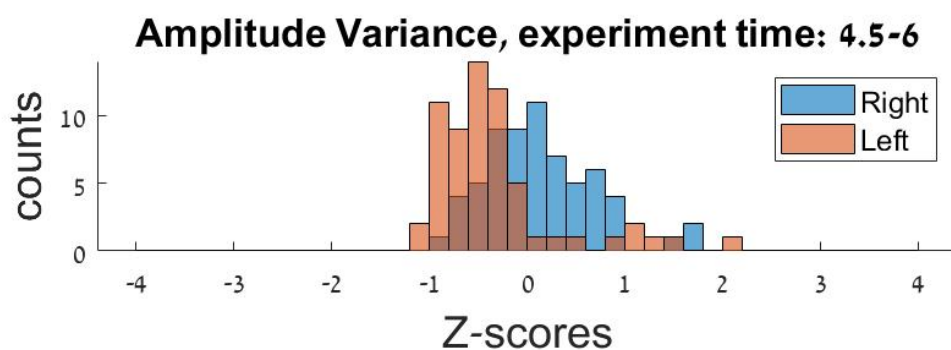
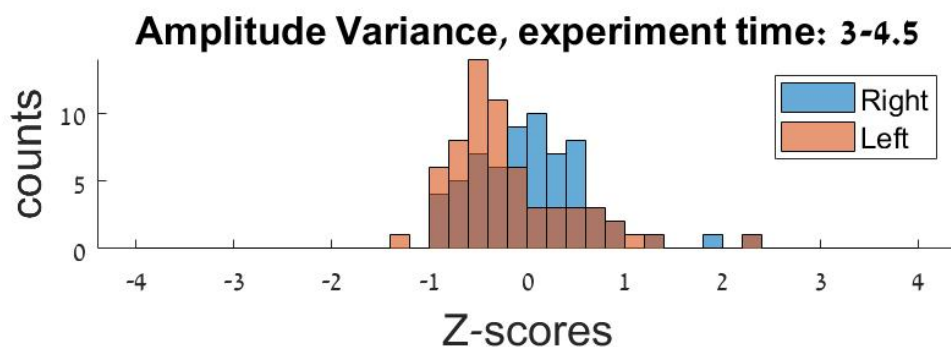








### VAR C<sub>3</sub>



### VAR C<sub>4</sub>

