



# Biomedical Signals - Lab 3

Marta Maldonado & Marcos Oriol

November 15, 2025

Universitat Politècnica de Catalunya



# Spectral Estimation in Stationary Signals

## 1 Introduction

### 1.1 Stages of Sleep

Sleep scoring is based on characteristic patterns in the electroencephalogram (EEG), electro-oculogram (EOG), and electromyogram (EMG). The modern American Academy of Sleep Medicine (AASM) manual recognizes four sleep stages (N1, N2, N3, R) plus Wakefulness, updating the older Rechtschaffen and Kales (R and K) system.

1. **Wake (W)**: Defined by more than 50% scorable **alpha activity** (8 to 13 Hz) in the EEG. EOG shows blinking and rapid movements, and submental EMG tone is high.
2. **Stage N1** (Light/Transitional Sleep): A transitional state characterized by low-voltage, mixed-frequency **theta activity** (4 to 7 Hz). **Slow Rolling Eye Movements (SREMs)** are evident, but neither sleep spindles nor K complexes are present.
3. **Stage N2** (Intermediate Sleep): Accounts for up to 50% of total sleep time. Defined by the first appearance of **Sleep Spindles** (11 to 16 Hz) and **K Complexes**.
4. **Stage N3** (Slow Wave Sleep, SWS): Deepest sleep stage, combining R and K stages 3 and 4. Defined by **Slow Wave Activity (SWA)** or Delta activity (0.5 to 2 Hz) with an amplitude  $\geq 75\mu\text{V}$  occupying more than 20% of the epoch.
5. **Stage R** (REM/Paradoxical Sleep): Characterized by low-amplitude, mixed-frequency EEG activity, **Rapid Eye Movements (REMs)** in the EOG leads, and the **lowest muscle tone (atonia)** in the chin EMG. It typically occurs 90 to 120 minutes after sleep onset.

### 1.2 Defined Frequency Bands

For the quantitative analysis of the EEG signal across sleep stages, the Power Spectral Density (PSD) is segmented into the following frequency bands (selected from slide 18 of presentation of chapter 1). The power contained within these ranges correlates directly to distinct mental or sleep states.

- **Low Delta (0.5 – 2 Hz)**: A sub-band of Delta often associated with the deepest stages of non-REM sleep (N3) and slow wave activity.
- **Delta ( $\delta$ , 2 – 4 Hz)**: The hallmark of deep sleep (SWS/N3). Characterized by high amplitude and slow activity.
- **Theta ( $\theta$ , 4 – 7.5 Hz)**: Predominant activity during the transition into sleep and light sleep stages (N1 and N2).
- **Alpha ( $\alpha$ , 7.5 – 12 Hz)**: Associated with quiet, relaxed wakefulness, typically seen over the posterior scalp with eyes closed.
- **Sigma ( $\sigma$ , 12 – 15 Hz)**: This range encompasses the **Sleep Spindles**, which are a defining feature of Stage N2 sleep.
- **Beta ( $\beta$ , 15 – 35 Hz)**: High-frequency activity linked to active concentration, complex thought, and wakefulness.



### 1.3 Power Spectral Density (PSD) Methods

Power Spectral Density (PSD) estimation determines how the power of a signal is distributed over frequency. The methods fall into two main categories: Non-Parametric (based on the Fourier Transform) and Parametric (based on model fitting).

#### Non-Parametric Methods

These methods compute the PSD directly from the signal data without assuming an underlying generating model.

- **Periodogram:** The most basic PSD estimator. It is calculated as the squared magnitude of the Discrete Fourier Transform (DFT) of the signal. While simple, it typically suffers from high variance (noisy results) and poor spectral leakage due to implicit rectangular windowing.
- **Periodogram + Hamming Window (HW):** Applies a non-rectangular window (Hamming) to the time-domain signal before computing the DFT. This improves the estimate by smoothing the data boundaries, which significantly **reduces spectral leakage** (smearing of energy across frequency bins) but slightly **worsens frequency resolution**.
- **Welch Periodogram:** A technique to reduce the high variance of the raw Periodogram. The signal is segmented into overlapping sections, each segment is windowed, and the PSDs of all segments are calculated and then averaged together. This results in a much smoother, more stable (lower variance) PSD estimate.

#### Parametric (AR Model) Methods

These methods assume the signal is the output of an Autoregressive (AR) system driven by white noise. The PSD is calculated from the estimated coefficients (parameters) of the AR model, offering much higher frequency resolution, especially with short data records. All use the **Burg method** for coefficient estimation.

- **Burg Method (Low/High Order):** The AR model order is chosen arbitrarily low (e.g.,  $P = 2$ ) or high (e.g.,  $P = 50$ ). A low order yields a very smooth, low-resolution PSD, while a very high order can capture noise components and become unstable.
- **Burg Method (AIC):** Selects the model order ( $P$ ) using the **Akaike Information Criterion (AIC)**. AIC favors models that minimize estimation error while penalizing complexity, but it tends to slightly overestimate the true order.
- **Burg Method (MDL):** Selects the model order ( $P$ ) using the **Minimum Description Length (MDL)** criterion. MDL imposes a stricter penalty on model complexity than AIC, resulting in a more parsimonious (lower) order, which is often considered the most stable and reliable estimate for short data sets.



## 2 Feasibility Assessment

In this section, we will be working with S1.mat recording.

### Data Loading

In the "S1.mat" file there are 21 biological channels recorded during 30 seconds with a sampling frequency of 100 Hz. In the document it is mentioned that: *It is usually used to consider EEG signals stationary during 5-second epochs. Therefore, from here we can determine that there are 5 epochs of 5 seconds each.*

#### 2.1 a) Compare the PSD functions for the channels Fz, Cz and Pz obtained in all 7 spectral methods in a figure (consider only the epoch 1).

- Epoch 1: corresponds to the first 5 seconds.
- First, we compute the PSD for the last 19 EEG channels, for the first 5 seconds (epoch 1).
- Then we selected from the third channel (this included) until the last one, in total 29 channels. This is because the first two correspond to the two ocular channels.

```
1 % load data
2 load("data/EEG_signals-20251023/S1.mat")
3
4 fs = 100; % Sampling frequency (Hz)
5 N = 5*fs; % samples in one 5-seconds epoch
6
7 % Compute all PSD from the 19 EEG channels (epoch 1)
8 % We select the first 5 seconds epoch and the last 19 EEG channels:
9 eegi = signals(1:N,3:21); % From 1 to 500 samples (5 seconds)
```

Then, it is important to filter the signal. As frequency bands come from 0.5 Hz to 35 Hz, it is convenient to remove the very low frequency components associated mainly with the movement and other brain activities different to delta rhythms. By doing this it will permit to estimate better the PSD.

```
1 % Apply high-pass filter
2 [b,a] = ellip(6,0.5,40,.4/(fs/2), 'high');
3 eeg = filtfilt(b,a,eegi);
```

With the data filtered, we will proceed to compute the PSD using the 7 spectral methods, defined in 1.3.

```
1 % Method 1 - Periodogram, considering a FFT with 1000 points:
2 [Pxx,f]=periodogram(eeg,[],1000,fs,'onesided');
3
4 % Method 2 - Periodogram using a Hamming window
5 [PxxH,f]=periodogram(eeg,hamming(500),1000,fs,'onesided');
```



```
6 % Method 3 - averaged Welch periodogram
7 number_segments = 4;
8 overlap = 0.5;
9 samples_segment =
10     floor(length(eeg)/(number_segments-(number_segments-1)*overlap));
11 [PxxW,f] = pwelch(eeg,samples_segment,floor(samples_segment*overlap),
12 1000, fs, 'onesided');
13 % Method 4 - Burg method with low order of 2
14 [Parb2,f]= pburg(eeg,2,1000,fs,'onesided');
15 % Method 5 - Burg method with high order of 50
16 [Parb50,f]= pburg(eeg,50,1000,fs,'onesided');
17 % Method 6 and 7 - AIC and MDL, respectively
18 nch = size(eeg, 2);
19 nnaic = zeros(nch, 1);
20 nnmdl = zeros(nch, 1);
21
22 for n = 1:nch
23     data1= iddata(eeg(1:N/2, n),[],fs);
24     data2= iddata(eeg(N/2 + 1: N,n),[], fs);
25     V1=arxstruc(data1,data2, (1:60));
26     nnaic(n)=selstruc(V1, 'aic');
27     nnmdl(n)=selstruc(V1, 'mdl');
28 end
29
30 % Method 6 (AIC)
31 naic = round(mean(nnaic));
32 [Parbaic,f] = pburg(eeg,naic,1000,fs,'onesided');
33
34 % Method 7 (MDL)
35 nmdl=round(mean(nnmdl));
36 [Parbmdl,f]= pburg(eeg,nmdl,1000,fs,'onesided');
```

Listing 1: Code used for running the different Power Spectral Methods.

We now select the channels indicated (Fz, Cz and Pz). Since the data we are working with ("eeg") has already 19 channels (the two first, corresponding to the ocular channels, have been removed previously, variable "eegi"), we can assign the following correspondences between channels and rows from "eegi":

- Fz: 5th channel
- Cz: 10th channel
- Pz: 15th channel



## Plotting code

To perform the following code, we have used ChatGPT, as the code for plotting the PSD from the 7 methods is very similar, just changing the data used (Pxx, PxxH, PxxW, etc.) and the title with the corresponding method used.

```
1 methods = {'Periodogram', 'Periodogram + Hamming', 'Welch', ...
2             'Burg(2)', 'Burg(50)', 'Burg(AIC)', 'Burg(MDL)'};
3
4 idxFz = 5;
5 idxCz = 10;
6 idxPz = 15;
7
8 nexttile;
9 plot(f, Pxx(:,idxFz), 'LineWidth',1); hold on;
10 plot(f, Pxx(:,idxCz), 'LineWidth',1);
11 plot(f, Pxx(:,idxPz), 'LineWidth',1); hold off;
12 title('Periodogram');
13 xlabel('Frequency (Hz)');
14 ylabel('Power Spectral Density (Power/Hz)');
15 xlim([0 50]);
16 grid on;
17 legend({'Fz', 'Cz', 'Pz'}, 'Location', 'northeast');
```

Listing 2: Matlab code for plotting PSD.

## Plotting Result

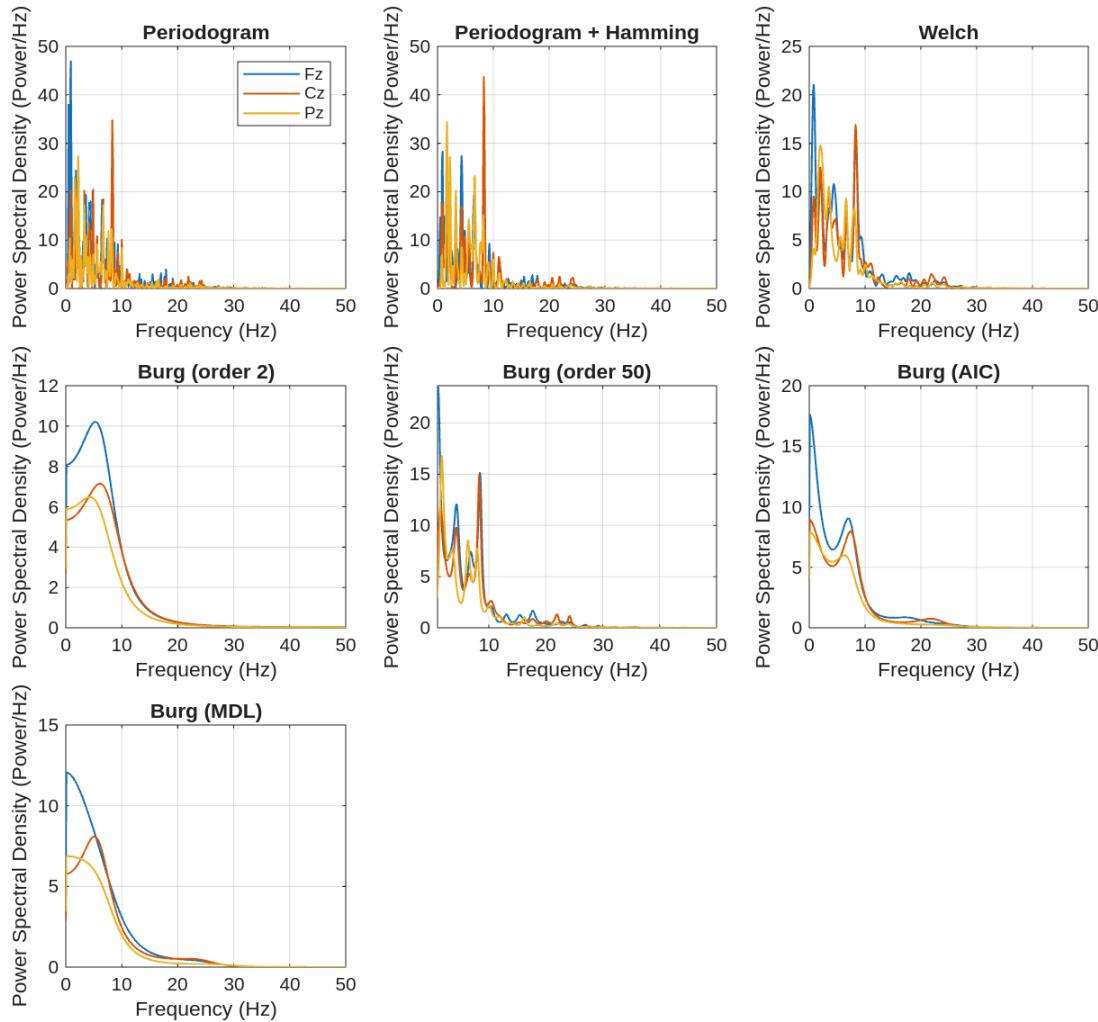


Figure 1: 7 PSD methods applied to the epoch 1 of S1 data.

**2.2 b)** Compare the topograms of all 7 spectral methods in a figure for each relative power (indicate the max value considered in each of both figures).

The frequency bands and their corresponding limits are the ones indicated in 1.2. Next, we present the code used for calculating the PSD methods for the defined bands.

```

1 % Define the 7 spectral methods and their names
2 PSD_methods = {Pxx, PxxH, PxxW, Parb2, Parb50, Parbaic, Parbmdl};
3 method_names = {'Periodogram', 'Hamming', 'Welch', 'Burg (p=2)', 'Burg
4 (p=50)', 'Burg (AIC)', 'Burg (MDL)'};
5 num_methods = length(PSD_methods);
6 num_channels = size(eeg, 2);

7 % Define all 6 frequency bands and their limits
8 Band_Limits = {
9     'Low Delta (0.5 - 2 Hz)', [0.5, 2.0];
10    'Delta (2 - 4 Hz)', [2.0, 4.0];

```



```
11     'Theta (4 - 7.5 Hz)',      [4.0, 7.5];
12     'Alpha (7.5 - 12 Hz)',    [7.5, 12.0];
13     'Sigma (12 - 15 Hz)',    [12.0, 15.0];
14     'Beta (15 - 35 Hz)',     [15.0, 35.0];
15 };
16
17 num_bands = size(Band_Limits, 1);
18
19 % Total frequency range for normalization
20 f_min_total = 0.5; % Hz
21 f_max_total = 30.0; % Hz
22 idx_total = find(f >= f_min_total & f <= f_max_total); % f: frequency
23   vector
24
25 for band_idx = 1:num_bands
26
27   % --- Get Current Band Info ---
28   band_name = Band_Limits{band_idx, 1};
29   f_min_band = Band_Limits{band_idx, 2}(1);
30   f_max_band = Band_Limits{band_idx, 2}(2);
31
32   disp(['Processing band: ', band_name]);
33
34   % Find frequency indices for the current band
35   idx_band = find(f >= f_min_band & f <= f_max_band);
36
37   % Initialize matrix to store relative power for all methods
38   % (Rows=Methods, Cols=Channels)
39   % Relative power calculated as before
40   Relative_Power_Matrix = zeros(num_methods, num_channels);
41
42   for m = 1:num_methods
43     current_Pxx = PSD_methods{m};
44
45     for ch = 1:num_channels
46       % Absolute power
47       absolute_power = sum(current_Pxx(idx_band, ch));
48
49       % Total power
50       total_power = sum(current_Pxx(idx_total, ch));
51
52       % Relative power
53       Relative_Power_Matrix(m, ch) = absolute_power / total_power;
54     end
55   end
56
57   % --- Determine Common Limits for plotting ---
58   % Find the overall maximum relative power for consistent color scaling
59   max_relative_power = max(Relative_Power_Matrix(:));
   common_limits = [0, max_relative_power];
```

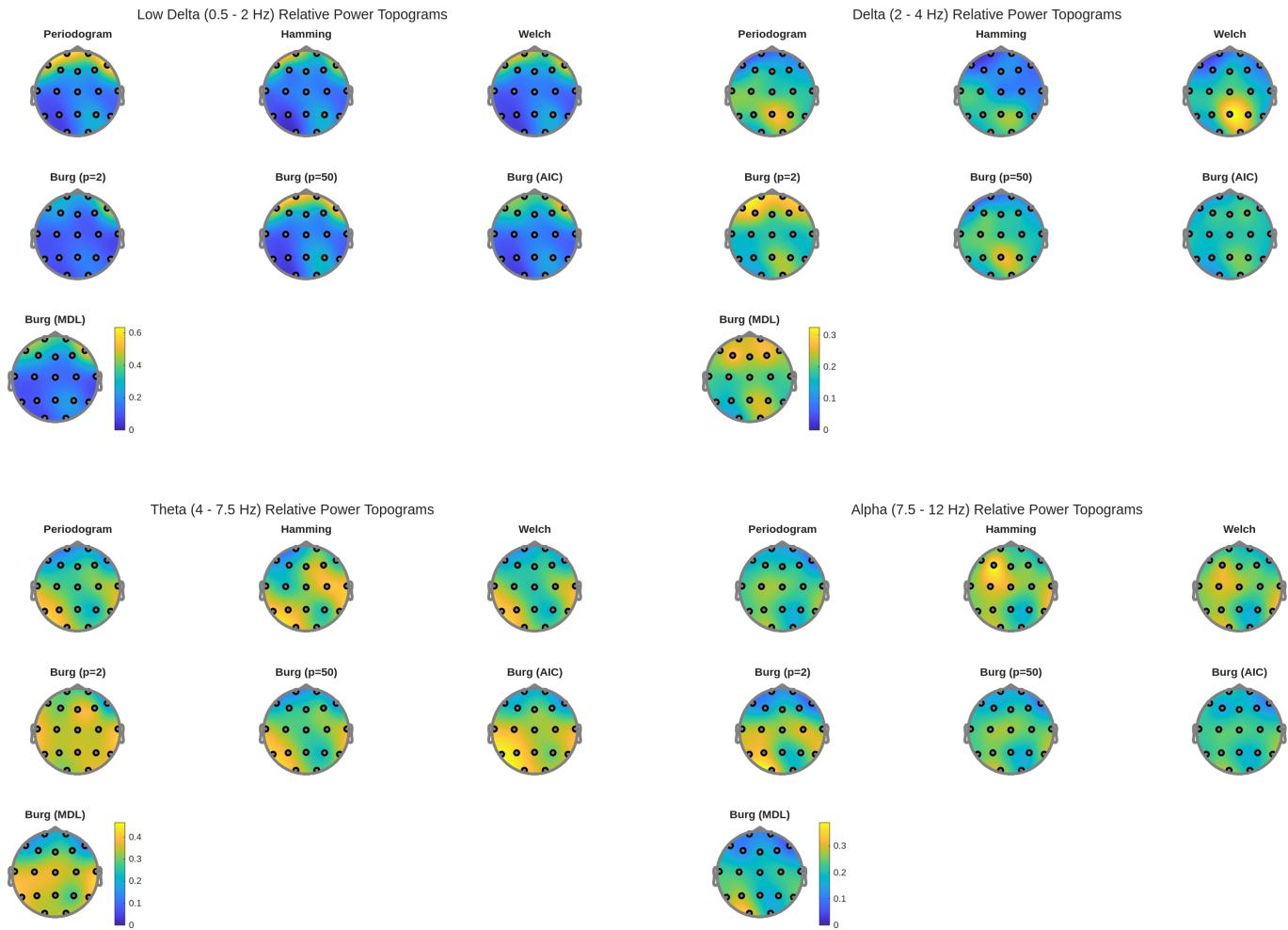


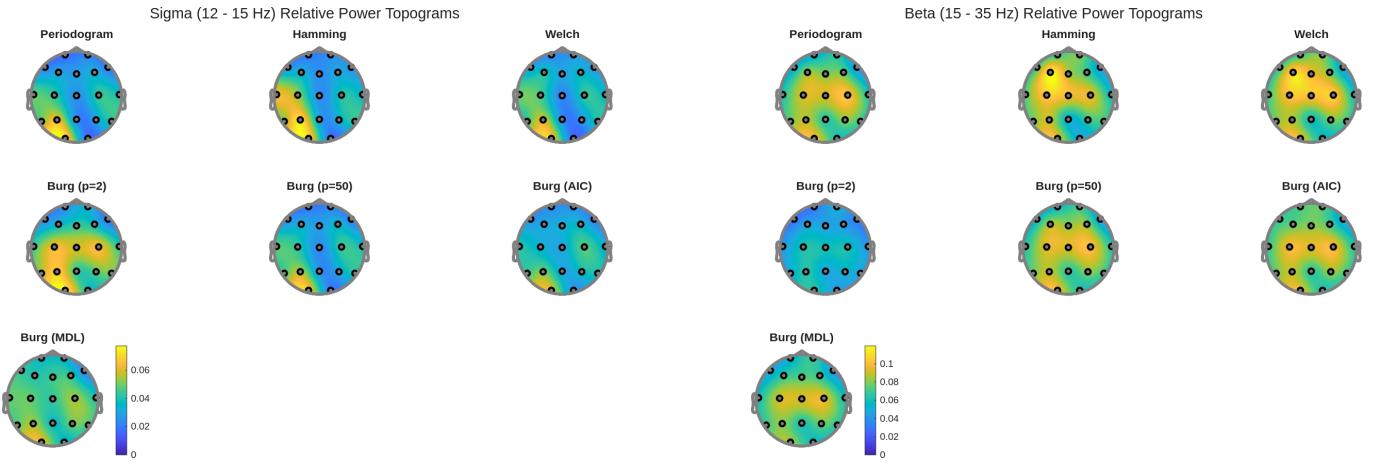
```
60 figure;
61 sgttitle([band_name, ' Relative Power Topograms'], 'FontSize', 14);
62
63 for m = 1:num_methods
64 subplot(3, 3, m);
65 draw_topogram(Relative_Power_Matrix(m, :), common_limits);
66
67 title(method_names{m});
68
69 % Add a colorbar only to the last plot (Method 7) for reference
70 if m == num_methods
71 colorbar('Location', 'EastOutside');
72 end
73 end
74
75 end
```

Listing 3: Matlab code used for running the different PSD for each frequency band.

## 2.3 Results

Below we are presenting the different relative power topograms obtained for the different frequency bands.





It can be observed that we have satisfied the requirement to use the same limits [min, max] across all seven topograms in a single figure, where min is 0 and max corresponds to the maximum value for all channels and all methods in the figure.

## 2.4 Discussion

From Figure 1, we can see that depending on the estimator used, the PSD functions obtained are different.

- When we use the non-parametric methods (periodogram and Welch methods) it can be seen that, for the periodogram method, the PSD functions obtained are quite noisy (have several peaks), whereas when using the Welch's method, the PSD functions are smoother, which means that in this case there it is not capturing that much details.
- When using the parametric methods, we can see that the PSD obtained are much smoother in comparison with the ones obtained with the non-parametric methods. However, when using the Burg method with order 50, the spectrum obtained shows sharp peaks, therefore, it is trying to include more details, but consequently, it is introducing more noise.

Regarding the topograms, these show the spatial distribution of relative power across all six EEG frequency bands, and how that distribution is affected by the seven methods we have used. By comparing the topograms generated by the seven methods for the same frequency band, we want to assess the feasibility of the results. Therefore, looking at the topograms, we see that most of the methods produce highly similar topographic patterns. In addition, we also see that the Burg method with order 2 can be considered infeasible since for all the frequency bands, this method produces a spatial distribution different from the other methods. In conclusion, six out of the seven methods used agree on the power distribution (have similar results), therefore, we can say that the results obtained are feasible.



### 3 Changes in Relative Power during Sleep

To observe changes of each relative power in frequency band during sleep (comparing the five sleep stages in a figure with the same max value for the five topograms which must be indicated). You can select only two spectral methods: selecting one of them with similar performance in most methods and another different to observe the influence of the estimator in checking these changes (each method in separated figures). Select the Welch periodogram and AR with order selected by MDL criteria.

#### Welch Periodogram vs AR MDL

In this exercise we will:

- Compare the power changes across all stages of sleep
- Both for Welch Periodogram and AR MDL

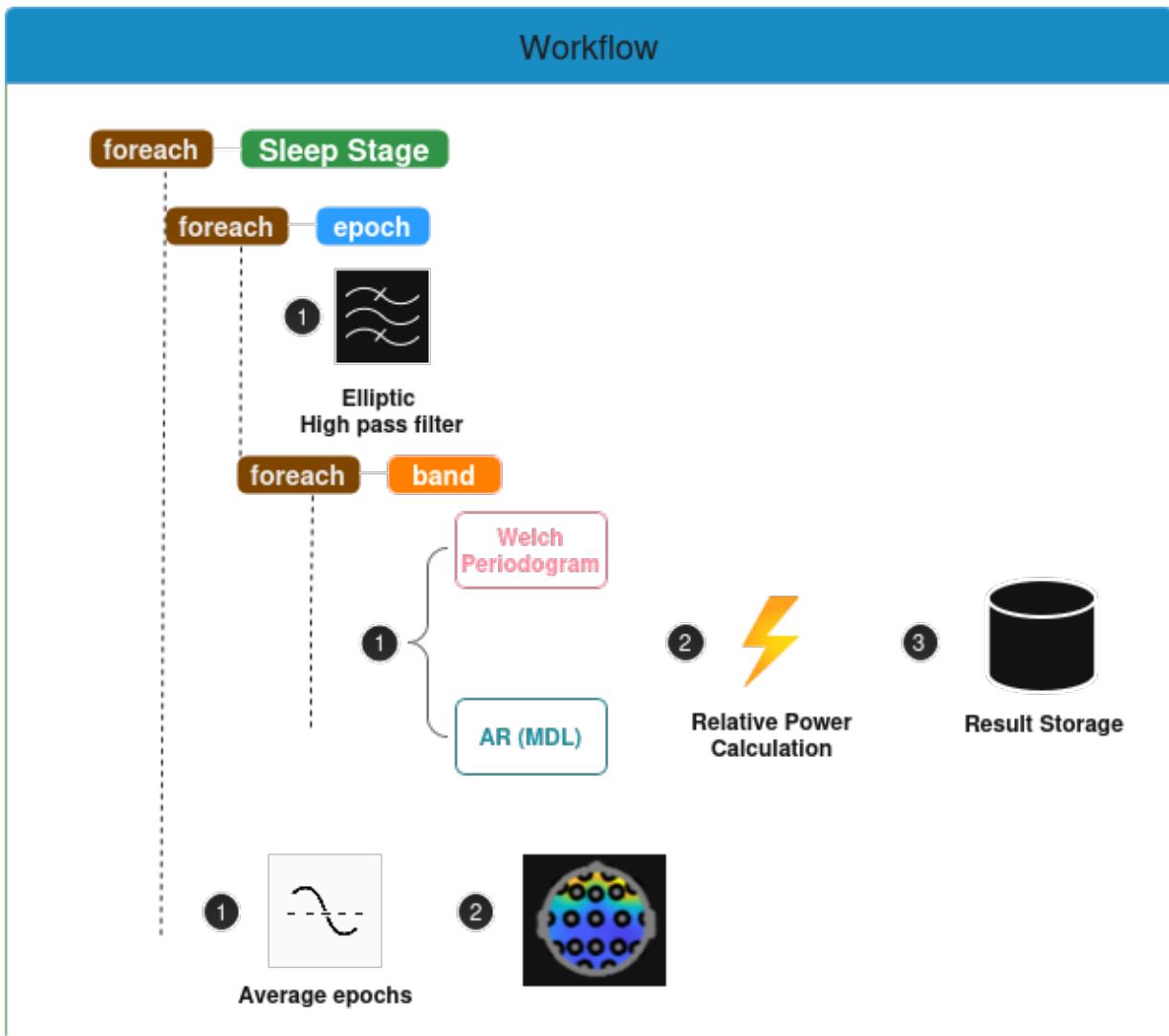


Figure 2: Workflow done to achieve the exercise.



With that logic, we implemented the following code.

```
1 for i = 1:num_stages
2     stage_name = stages{i};
3     fprintf(' Processing Stage: %s\n', stage_name);
4
5 % Load Data
6 file_name = sprintf('%s%s.mat', data_path, stage_name);
7 data = load(file_name);
8 signals = data.signals;
9
10 % Temporary storage for all epochs within this stage (Channels x
11 % Epochs)
12 stage_power_welch = struct();
13 stage_power_ar = struct();
14 for b = 1:num_bands
15     stage_power_welch.(band_names{b}) = zeros(num_channels,
16         num_epochs);
17     stage_power_ar.(band_names{b}) = zeros(num_channels, num_epochs);
18 end
19
20 % Loop through Epochs
21 for j = 1:num_epochs
22
23     % Extract Epoch
24     start_idx = (j-1) * nsamples_epoch + 1;
25     end_idx = j * nsamples_epoch;
26     eeg_epoch_raw = signals(start_idx:end_idx, eeg_channel_indices);
27
28     % Filter Epoch
29     eeg_epoch_filtered = filtfilt(b_filter, a_filter, eeg_epoch_raw);
30
31     % Method 1: Welch
32     [PxxW, f_welch] = pwelch(eeg_epoch_filtered, samples_segment,
33         overlap_samples, nfft, fs, 'onesided');
34     rel_power_w = calculate_relative_powers(PxxW, f_welch, bands);
35
36     % Method 2: AR (MDL)
37     avg_nmdl = get_ar_mdl_order(eeg_epoch_filtered, fs, max_ar_order);
38     [PxxAR, f_ar] = pburg(eeg_epoch_filtered, avg_nmdl, nfft, fs,
39         'onesided');
40     rel_power_ar = calculate_relative_powers(PxxAR, f_ar, bands);
41
42     % Store results for this epoch
43     for b = 1:num_bands
44         stage_power_welch.(band_names{b})(:, j) =
45             rel_power_w.(band_names{b});
46         stage_power_ar.(band_names{b})(:, j) =
47             rel_power_ar.(band_names{b});
48     end
49 end
```



```
44
45 % Average Epochs
46 for b = 1:num_bands
47     results_welch.(band_names{b})(:, i) =
48         mean(stage_power_welch.(band_names{b}), 2);
49     results_ar_mdl.(band_names{b})(:, i) =
50         mean(stage_power_ar.(band_names{b}), 2);
51 end
```

Listing 4: Code used for batch processing both PSD methods on different stages of sleep for different frequency bands.

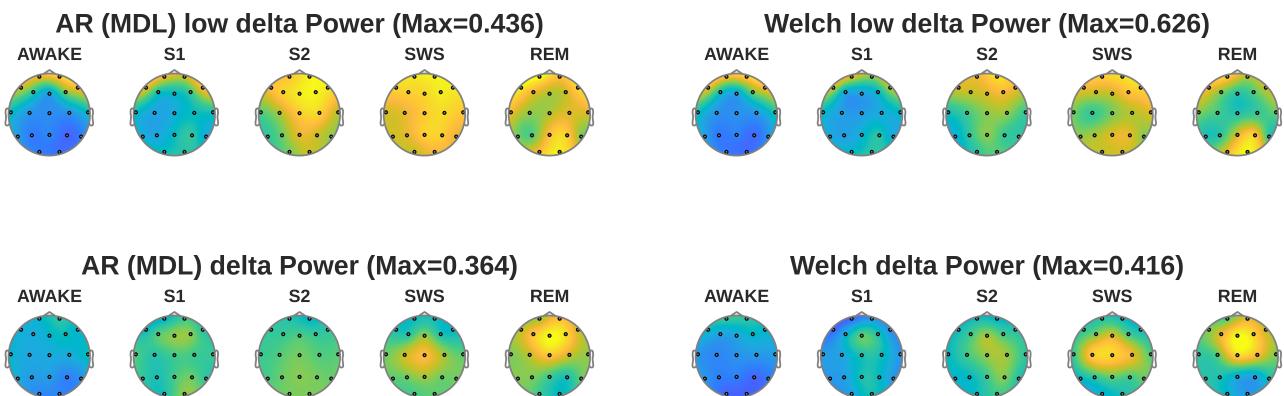
Once processed all the data, we plotted the results with the following code:

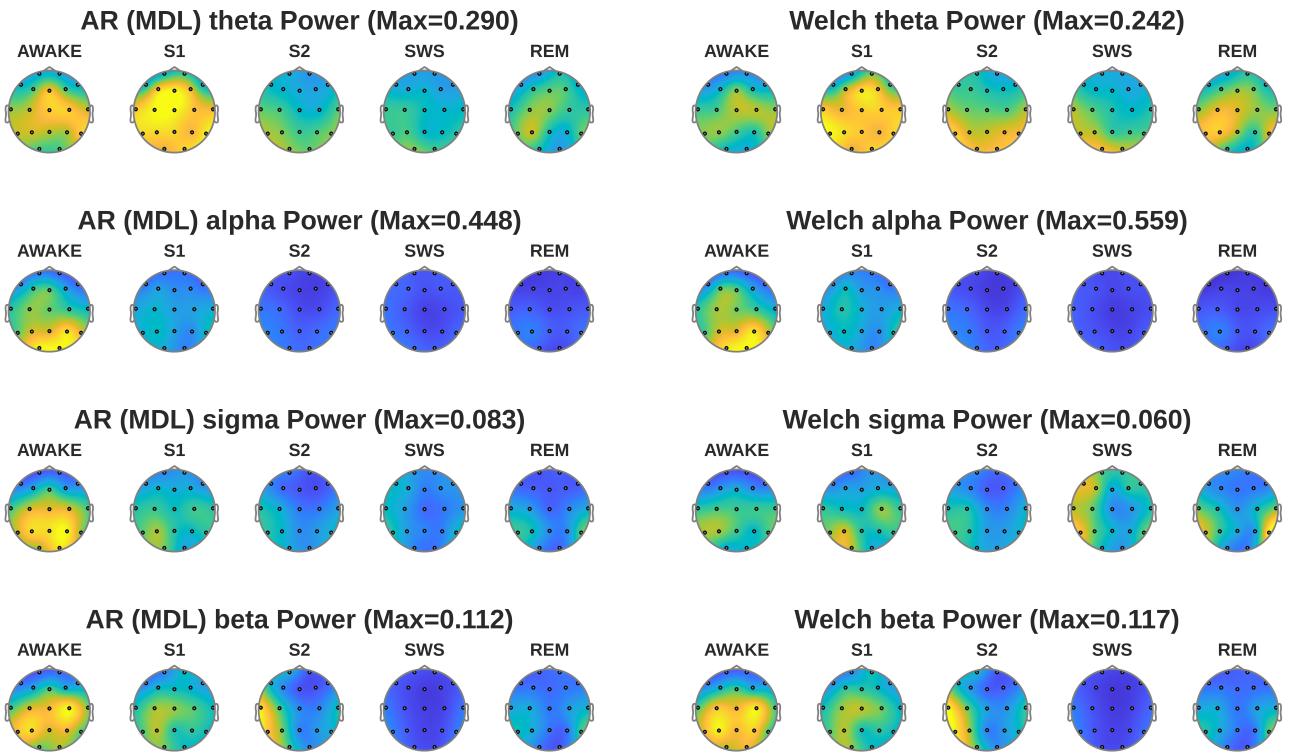
```
1 for b = 1:num_bands
2     band_name = band_names{b};
3
4     % Welch results
5     plot_comparison_topograms(results_welch.(band_name), stages,
6         band_name, 'Welch', ['report/img/topogram_welch_', band_name,
7             '.png']);
8
9     % AR (MDL) results
10    plot_comparison_topograms(results_ar_mdl.(band_name), stages,
11        band_name, 'AR (MDL)', ['report/img/topogram_ar_', band_name,
12            '.png']);
13 end
```

Listing 5: Customized plotting function for laying topogram plots horizontally.

### 3.1 Results

After applying the defined workflow to the 5 stages of sleep, we obtained the following results for the previously defined bands.





### 3.2 Discussion

The obtained results match very well with what is described in the literature. Showing a clear correlation with Power Spectral Density and activity in each specific case. In our opinion, the stage S1 (N1) is very interesting because appears to have oscillations from both stages (sleep/awake) and has activity of all the bands except from alpha band.

Both Welch Periodogram and AR (MDL) PSD methods perform very similarly. However, by observing topogram representations of PSD power/electrode, we think that the AR method is more prone to saturate (giving less visual evidence of differences across electrodes), whereas the Welch method seems more balanced.

No specific band based on the topograms from 3.1. results shows difference in deep sleep stages. All the results in the AR (MDL) and Welch methods appear to have very similar outputs for each band.