

Procedure for creating a “blink template” to use in identification of blink components after EEG data decomposition

Introduction

ICA (Independent Components Analysis) is a mathematical procedure that can be used to identify and remove blinks (and other noise components, or *artifacts*) from EEG data. ICA decomposes the data into spatial factors that are statistically independent in time (the *independent components*). The independent components can then be matched (e.g., cross-correlated) with a *blink template* — a spatial vector that is defined a priori to characterize the spatial topography of a typical blink. Activations that closely match the template are then zeroed out, or removed from the data. This process is referred to as *data cleaning*, and can be implemented with a variety of ICA algorithms, as described in NIC Technical Report #003. The present document describes how to create a “blink template,” beginning with raw (un-processed) EEG data. Note this process is currently implemented within Net Station Tools.

An important issue, which remains to be explored, is whether there is any advantage in using a blink template that is optimized for an individual subject, versus one that is derived from multiple subjects. The advantages of an averaged template are: (1) it only needs to be created once and (2) it is likely to be “cleaner,” assuming that it is derived from a greater number of blinks than a template that is constructed for an individual subject. On the other hand, a template that is optimized for a given subject is more likely to match blink components that deviate from the norm, e.g., due to individual differences in the blinks themselves (i.e., blink physiology) or to variations in electrode placement. In either case, the following procedure can be used to construct a blink template from single or multiple subjects. All or part of the procedure may eventually be automated; currently, several steps are implemented manually.

Step 1: Marking eyeblinks

The first step is to identify and mark a set of representative blinks in the raw (continuous) EEG file. Open the data file and create a new User Track called “artifacts”: this is where you will insert the eyeblink markers for “events,” which will be used for segmentation in Step 2.

Next scroll through the file, searching for representative blinks. There is no rule of thumb, but I have been marking between 15 and 20 blinks per session: this may be the minimum to obtain a usable blink template (see below). For each blink, insert *eyeb* (“eye blink”) events at the peak of each blink, that is, at a single time sample. This ensures that the blink events are lined up when averaging. Note that when selecting blink events, the montage can be set to *Eyes* to facilitate manual blink detection. On the other hand, it is important to mark only those blinks that occur during otherwise (apparently) artifact-free intervals, as this will result in a cleaner template. Therefore, the user should verify for each blink event that the non-blink channels are as noise-free as possible.

Step 2: Segment continuous data into blink segments

Step 2 is to segment the continuous data, time-locking each segment to the peak of each blink “event.” Begin by creating a new Segmentation tool in Net Station, called Blink Segmentation. Set event segmentation code equal to *eyeb*. Define an epoch from ~500ms prior to the *eyeb* event, until ~500ms after the event. Apply to your continuous data file. It is recommended that the data be low-pass filtered to remove high-frequency “noise” (e.g., 60hz activity) prior to segmentation.

The result should be a segmented file, consisting of 15-20 “blink segments,” as shown in Figure 1 below.



Figure 1. Some exemplary blinks, marked as events for segmentation of a session file. The *artifacts* event track was created for storing the manually marked blinks. The event marker *eyeb* is one of the defaults in Net Station. Note that user montage is set to *Eyes* (also a default) for easy detection of blinks.

Figure 2 shows 4 1-second blink segments that are overlaid. The peaks line up well, because the blink event markers were inserted at the peak of each blink activation.

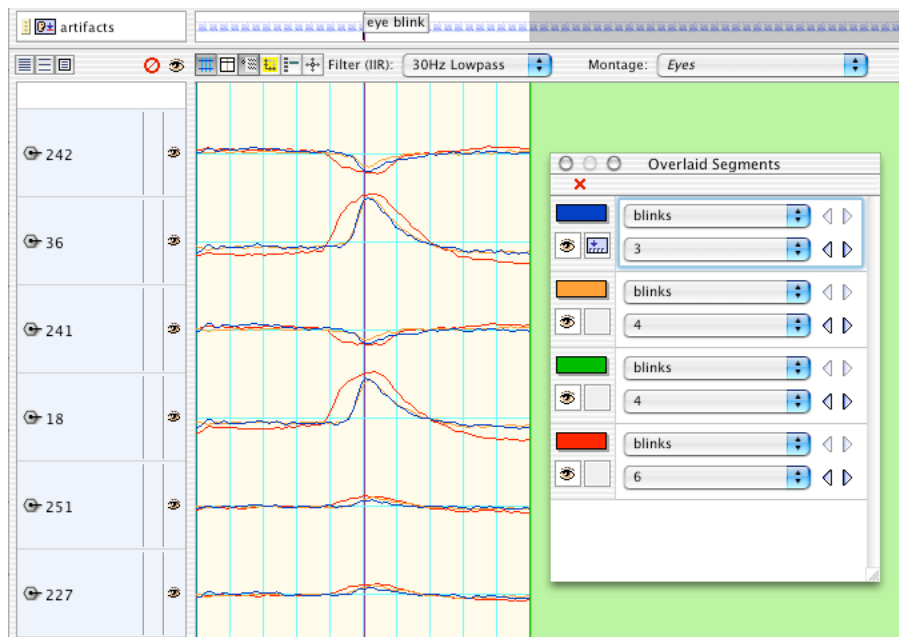


Figure 2. Four blink segments for session ON101. Because events were inserted at the peak of the blink activation, they line up cleanly when overlaid.

Step 3: Create an averaged blink

In Step 3, each blink segment is preprocessed to remove non-blink artifacts, prior to averaging. First, manually mark the bad channels for each subject. Figures 3 and 4 illustrate how the blink template may be degraded if bad channels are not removed from the blink segments prior to averaging.

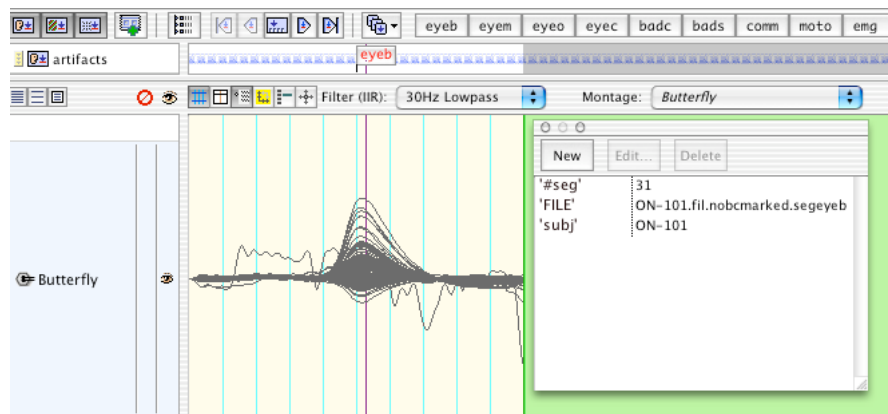


Figure 3. Butterfly plot of the averaged blink (n=31 blinks) for session ON101 when no bad channels were removed prior to averaging. Channel 232 (right side, under ear) is clearly bad.

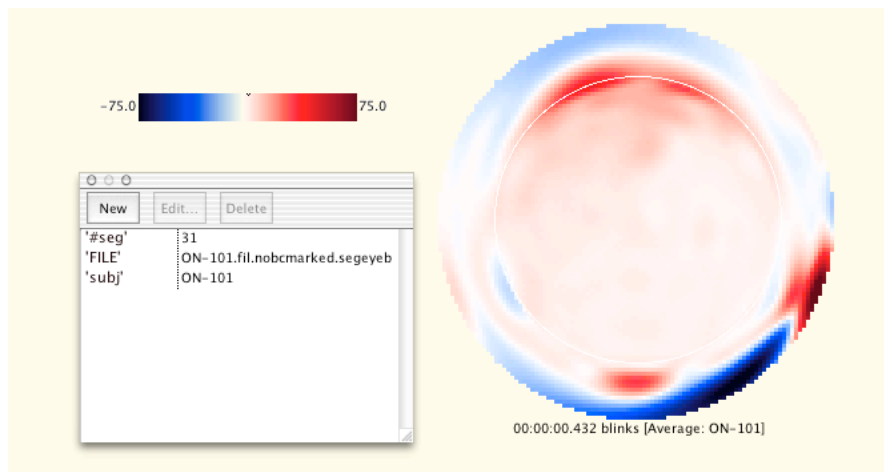


Figure 4. Topography of the averaged blink (n=31 blinks) for session ON101 when bad channels are not interpolated prior to averaging. This would probably not be an effective blink template.

Figure 5 illustrates how to mark up bad channel in each blink segment. See [Net Station Tutorial.pdf](#) for guidelines on detecting bad channels.

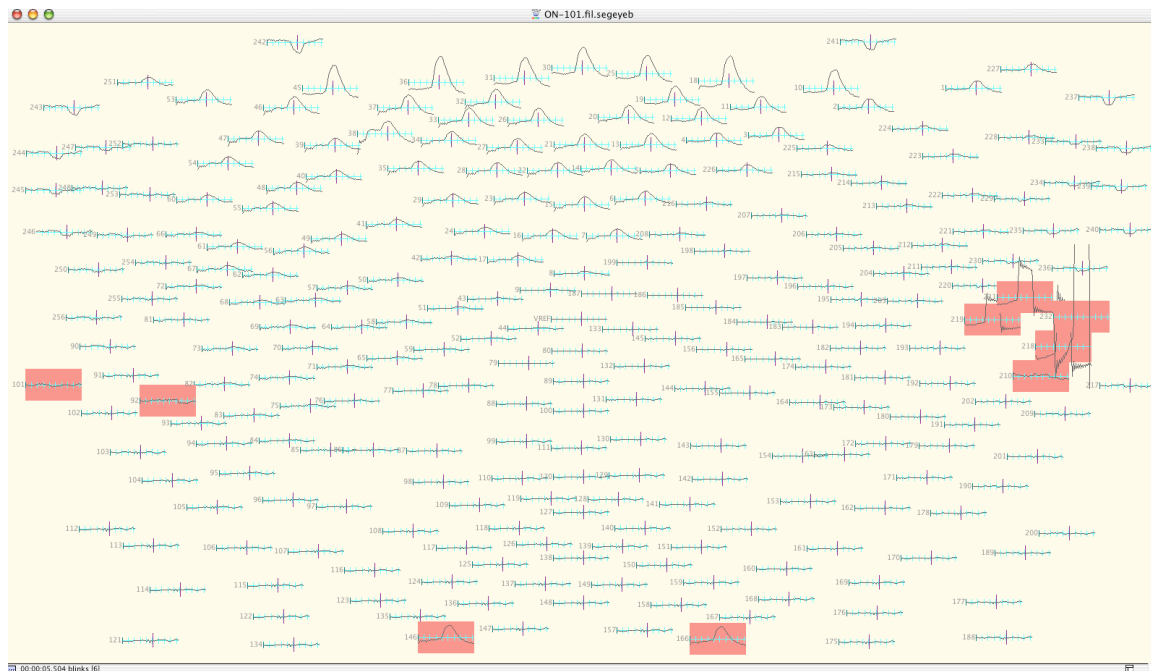


Figure 5. Bad channels are marked for every blink segment. In this subject (ON101), there were a cluster of channels below the right ear that were not seated well. Therefore, these channels were marked bad throughout. Channels 146 and 166 (bottom) are configured for EKG acquisition, and were therefore marked bad, as well.

After marking bad channels in each segment, perform the following data transformations:

- Replace bad channels and average across the blink segments. Uncheck both boxes under “Exclude Subjects and Channels.”
- It may or may not be desirable to baseline-correct the average blink data prior to creating the blink template. I have been using a baseline that begins 200ms after segment start (i.e., 300ms prior to peak of blink) and is 200ms long.

Figure 6 shows the resulting waveforms (activations) for the averaged blink for a single session (ON101).

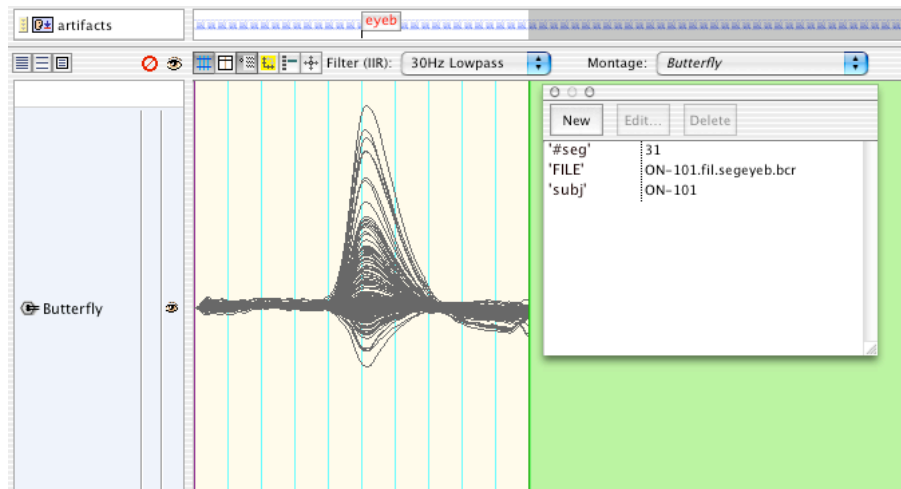


Figure 6. Butterfly plot of the averaged blink (n=31 blinks) for session ON101. Bad channels were manually marked and replaced prior to averaging. Compare with Figure 3.

Step 4: Export blink topography (single timepoint)

The final step is to export the averaged blink data to text and select the peak (central timepoint) for the epoch, which represents the maximum of the averaged blink activity for your dataset. This spatial vector constitutes the final blink template that will be used to identify blink “components” after data decomposition using ICA, PCA or similar factor analysis procedures.

Figure 7 shows the topography of the averaged blink for our example dataset at the peak of the blink activation.

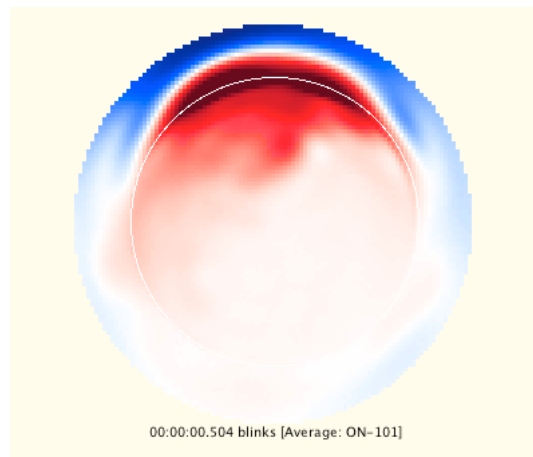


Figure 7. Topography of the averaged blink (n=31 blinks) for session ON101. Compare with Figure 6.

Summary

The steps for creating a blink template can be summarized as follows:

1. Identify and mark blink “events” in raw data
2. Segment continuous data into blink events
3. Preprocess and average the blink segments
4. Export averaged blink data (a single time series) to text
5. Select peak (central timepoint) to use as spatial vector or “blink template”

The blink template can then be used, subsequent to factor analyses of the EEG, for identification of ocular artifacts: the template is cross-correlated with each spatial “component” (or “projection” in ICA), and components that are correlated with the template above a predetermined threshold (typically .80 or greater) are removed from the data. The procedure for cross-correlation and removal of blink-related ICA components is detailed in NIC Technical Report #004.