

4 Event extraction by template matching

4.1 Introduction

To isolate individual events such as EPSCs or EPSPs from recorded data, Stimfit uses a template matching algorithm as described by Jonas et al. (1993), with some implementation details adopted from Clements and Bekkers (1997). The template consists of a waveform $p(t)$ with a length of n sampling points that represents the time course of a typical event. The template is slid over the trace of recorded values $r(t)$, and at each sampling point with index s , it is multiplied by a scaling factor m and an offset c is added or subtracted so that the sum of squared errors $\chi^2(t_s)$ between the trace and the template is minimised: *template matching*

$$\chi^2(t_s) = \sum_{k=0}^{n-1} [r(t_{s+k}) - (m \cdot p(t_k) + c)]^2$$

As can be seen from this equation, this amounts to the fairly simple operation of fitting a straight line that relates $p(t)$ and $r(t)$ at every sampling point.

Finally, some detection criterion has to be applied to decide whether an event has occurred at a sampling point. Two options are available in Stimfit: Jonas et al. (1993) suggest to use the linear correlation coefficient between the optimally scaled template and the data, whereas Clements and Bekkers (1997) compare the scaling factor with the noise standard deviation. *detection criterion*

4.2 A practical guide to event detection

In practice, the following steps need to be performed to extract events with Stimfit:

1. Create a preliminary template by fitting a function to a single, large and isolated event.
2. Use this preliminary template to extract some more exemplary large and isolated events using a high detection criterion threshold..
3. Create the final template by fitting a function to the average of the exemplary events.
4. Extract all events with the final template using a low detection criterion threshold.

4 Event extraction by template matching

5. Eliminate false-positive, add false-negative events.

This procedure will be explained in some more detail in the following sections.

4.2.1 Create a preliminary template

In general, the template waveform $p(t)$ can be of arbitrary shape. A typical way of creating such a template is to fit a function with a time course matching the event kinetics to some exemplary events. For example, EPSCs can typically be modelled with the sum or the product of two exponential functions¹. In practice, a robust estimate for a template can be obtained using an iterative approach, which will be illustrated here using a recording of miniature EPSCs that you can download here².

First, we fit a function to a single large and isolated event to create a preliminary “bait” template. In this case, we will use the EPSC that can be found roughly between

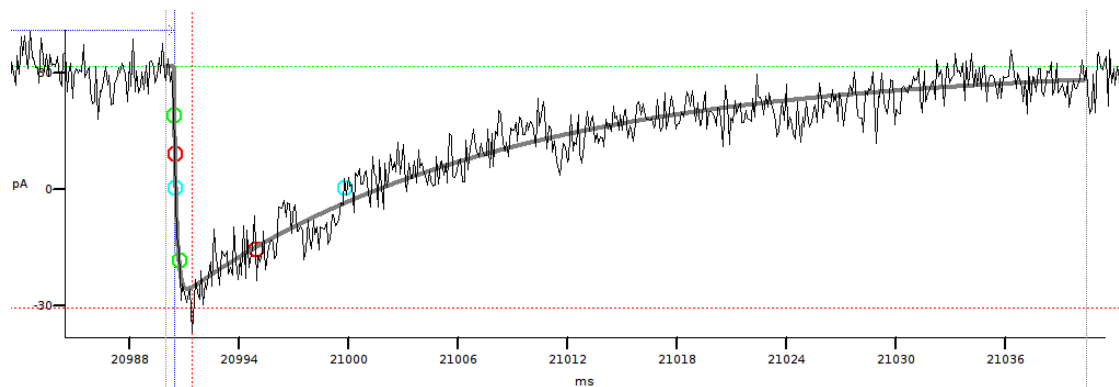


Fig. 21: Creation of a “bait” template.

$t = 20990$ ms and $t = 21050$ ms. Then, we fit the sum of two exponential functions with a delay to this EPSC. To obtain the same template as in the example, you can call the function `preliminary` from the `minidemo` module that comes bundled with `stimfit`:

```
>>> import minidemo
>>> minidemo.preliminary()
```

This will take care of the appropriate cursor positions and the biexponential fit (see p. 14 for setting the cursors from Python, and p. 9 for the `leastsq` function). If you prefer, you can use the fit settings dialog as described in chapter 1 (Fig. 16, p. 9).

¹Note that the product of two exponentials $f(t) = a \left(1 - e^{-\frac{t}{\tau_1}}\right) e^{-\frac{t}{\tau_2}}$ can equivalently be expressed as the sum of two exponentials: $f(t) = a \left(e^{-\frac{t}{\tau_2}} - e^{-\frac{t}{\tau_3}}\right)$, with $\tau_3 = \frac{\tau_1 \tau_2}{\tau_2 - \tau_1}$.

²<http://stimfit.org/tutorial/minis.dat>

4.2.2 Extract exemplary events

We now use the bait template to fish some more large and isolated events. Choose “Analysis”→“Event detection” →“Template matching...” from the menu. In the dialog

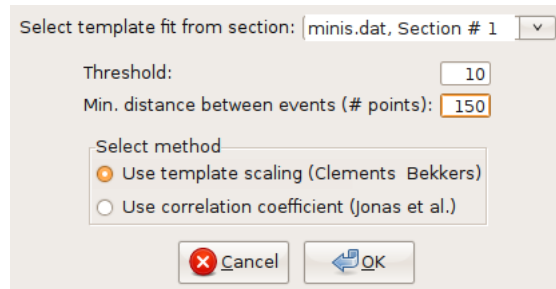


Fig. 22: Event detection settings.

that will pop up (Fig. 22), you can set the threshold for the detection criterion. Since we want to extract some large and isolated events during this first pass, we set this to a high number, say 10, using the template scaling factor (Clements and Bekkers, 1997). Click “OK” to start the event detection. When finished, press **[F]** to fit the whole trace to the window. The detected events will be marked by blue arrows in the upper part of the window, and blue circles will indicate the peak values of the detected events (Fig. 23). To view the isolated events in a new window, you have to switch to the event editing

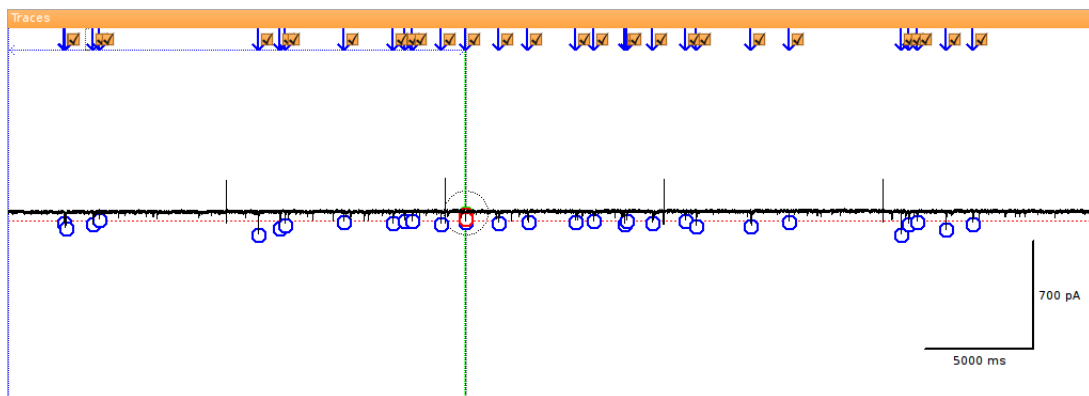


Fig. 23: Detected events.

mode, either by pressing **[E]** or by activating the corresponding button in the toolbar (Fig. 24). When you now click on the trace with the right mouse button, a menu will show up. Select “Extract selected events” from this menu. This will put the exemplary

4 Event extraction by template matching

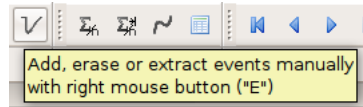


Fig. 24: Switching to event editing mode.

EPSCs into a new window.

4.2.3 Create the final template

We now create the average of all extracted events, as explained in chapter 1 (p. 7). Then, we fit a biexponential function to the average, as explained above for the single EPSC (see p. 26). Remember to set the baseline, peak and fit window cursors appropriately before performing the fit, and to update all calculations. Again, you can make use of a function from the `minidemo` module to set the cursors and perform the fit:

```
>>> import minidemo # if you haven't imported it already
>>> minidemo.final()
```

The final template should look similar as shown in Fig. 25.

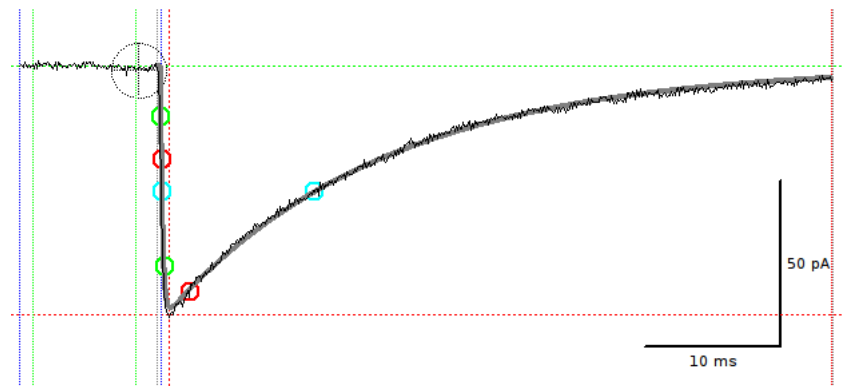


Fig. 25: Creating a final template.

4.2.4 Extract all events

Go back to the original file (`minis.dat`). Extracting all events with the final template is done in nearly the same way as described above for the preliminary template. However, you have to choose the correct template in the event dialog: The final template in this

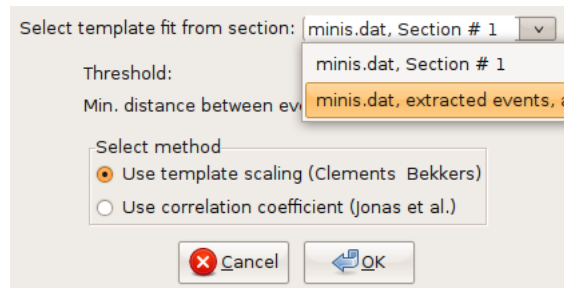


Fig. 26: Selecting the final template.

case is the second on the list (Fig. 26). For this final run, we will lower the detection threshold to a value of 3, as suggested by Clements and Bekkers (1997).

4.2.5 Edit detected events

Usually, the detected events have to be screened visually to remove false-positives and add false-negatives. Removing false positives is done by unselecting the checkbox next to the arrow indicating an event (Fig. 23). To add false-negatives, you have to switch to the event-editing mode (Fig. 24) and then right-click on the trace at the position where the event starts. From the context menu that will pop up, select “Add an event that starts here” (Fig. 27). To efficiently screen the whole trace, it is convenient to use **Shift** and **←** at the same time. This will move the trace left by the width of one window. Once you are done with editing, choose “Extract selected events” from the context menu.

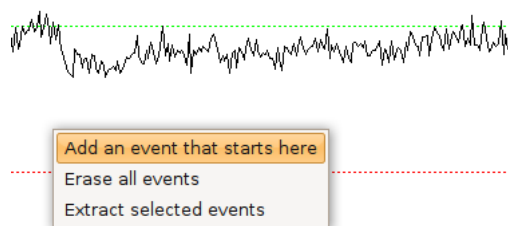


Fig. 27: Adding a false-negative event.

4 Event extraction by template matching

4.2.6 Analyse extracted events

If you used the same settings as suggested above, 97 events will be extracted. You will find a table on the left of the traces: This will show you the time of onset of the events and the inter-event intervals. Usually, you will want to apply some further analysis to the extracted events. To do so, you first have to adjust the baseline, peak and fit cursors. Again, there is a function in the `minidemo` module taking care of that:

```
>>> minidemo.batch_cursors()
```

batch analysis To analyse all traces efficiently, you can now perform a “batch analysis” on all traces at once: First, select all traces, either using `select_all()` from the shell (see p. 15) or “Edit”→“Select all traces” from the menu or pressing **Ctrl** and **A** at the same time. Then, choose “Analysis”→“Batch analysis” from the menu. From the dialog (Fig. 28),

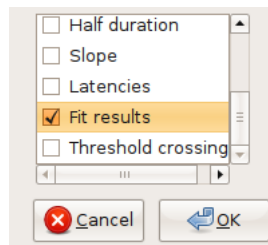


Fig. 28: Batch analysis settings.

choose the analysis functions that you want to apply to your data. Click “OK” once you are done. A new table will appear to the left of the traces. You can copy and paste values from the tables to spreadsheet programs for further analysis.

4.2.7 Adjusting event detection settings

<i>Problem</i>	<i>Solution</i>
Too many false-positive events have been detected	Increase the detection threshold
Too many events have been missed (false-negatives)	Decrease the detection threshold
One and the same event is detected multiple times at short time intervals	Increase the number of sampling points between events
Closely spaced events are not detected separately	Decrease the number of sampling points between events