# MatchLocate2.0

https://github.com/Dal-mzhang/MatchLocate2

Aug. 29, 2019

## 0. Author

Miao Zhang, Dalhousie University (miao.zhang@dal.ca)

## 1. References

Zhang M. and Wen L. An effective method for small event detection: match and locate (M&L). Geophysical Journal International, 200 (3), 1523-1537, 2015. [method introduction]

Zhang M. and Wen L. Earthquake characteristics before eruptions of Japan's Ontake volcano in 2007 and 2014. Geophysical Research Letters, 42 (17), 6982–6988, 2015. [application to volcanic earthquake detection and location]

Zhang M. and Wen L. Seismological Evidence for a Low-Yield Nuclear Test on 12 May 2010 in North Korea. Seismological Research Letters, 86 (1), 138-145, 2015. [application to low-yield nuclear test detection and location]

## 2. Introduction

Compared to the current methods of small event detection (template matching/matched filter), the M&L method places event detection to a lower magnitude level and extends the capability of detecting small events that have large distance separations from the template (Fig. 1). The method has little dependence on the accuracy of the velocity models used, and, at the same time, provides high-precision location information of the detected small-magnitude events (Fig. 2).

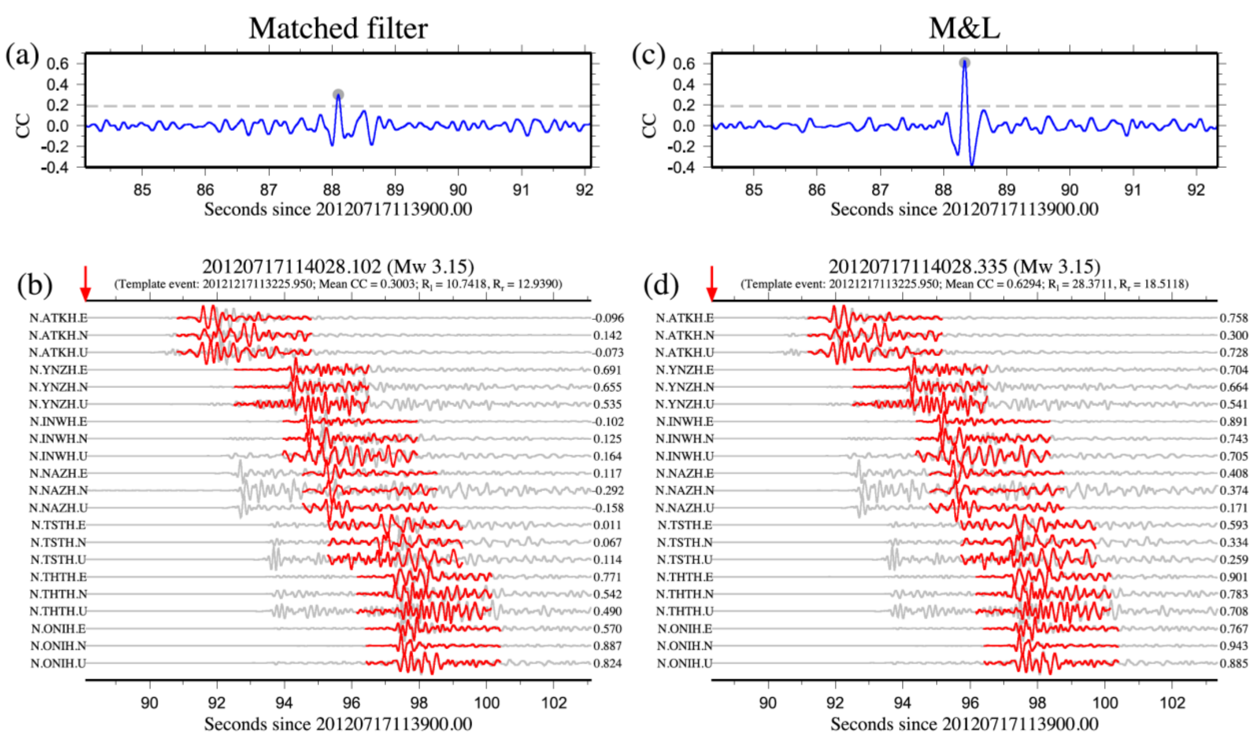

**Figure 1.** (a, c) Stacked cross-correlograms and (b, d) comparison of the template seismograms (red traces) with portions of the signals detected for slave earthquake in the continuous waveforms (grey traces), between the matched filter (a, b) and the M&L method (c, d). Grey points in (a, c) and red arrows in (b, d) mark the determined origin time of the detected earthquake.

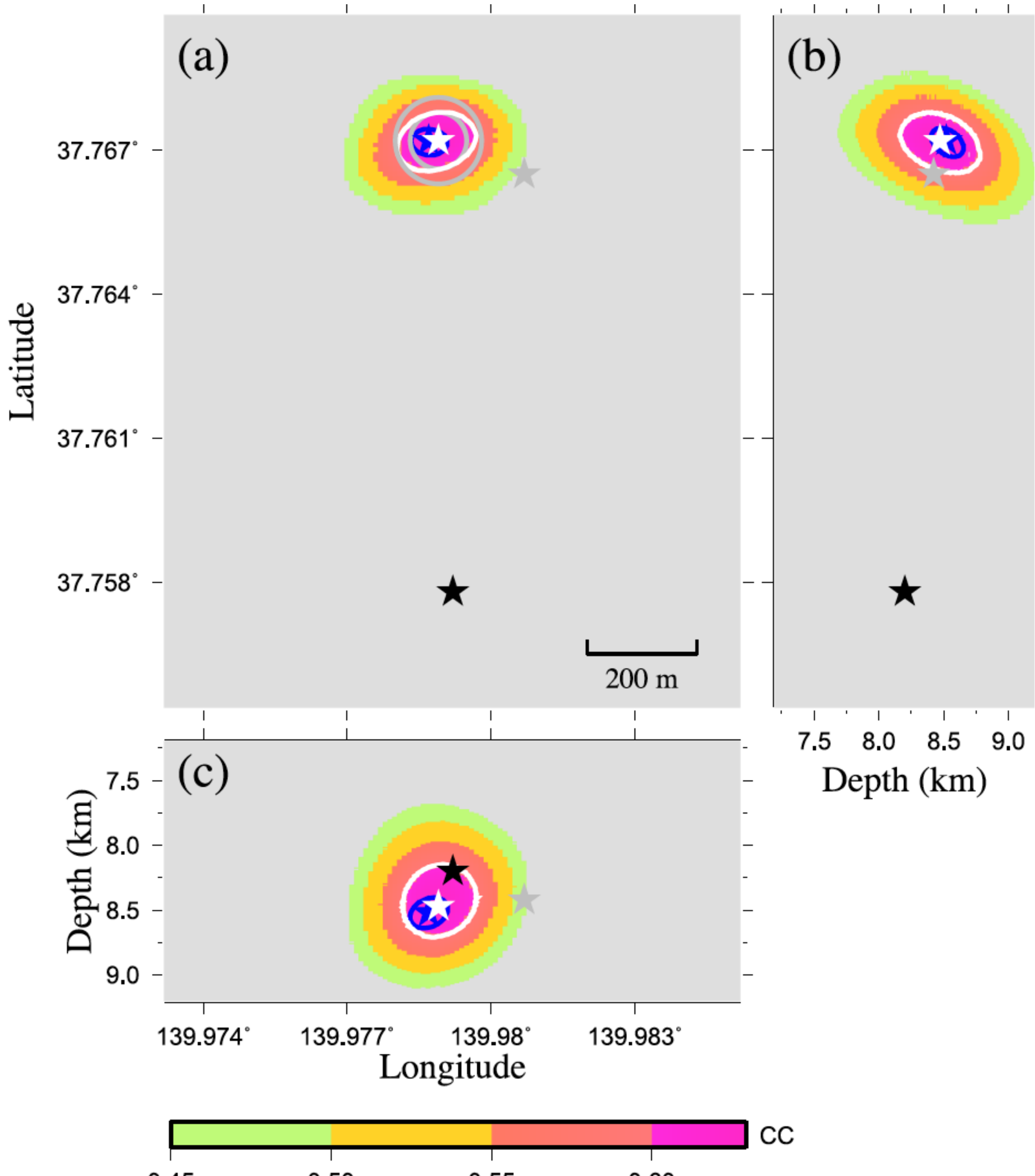


**Figure 2.** Mean CC value of the stacked correlogram as a function of assumed location of slave earthquake (only the regions with CC > 0.45 are plotted) in three plane views: (a) longitude–latitude plane, (b) latitude–depth plane and (c) longitude–depth plane, along with the template location (black stars), M&L determined location (with the maximal CC value, white stars), the reference location (blue stars; determined using a cross-correlation based master event location method) and the JMA catalogue location (grey stars) of the slave event, the 95 per cent confidence

ellipse of the reference location based on the chi-square distribution (blue ellipses), and the confidence level of slave earthquake location (white ellipses) (detailed location analysis can be found in Zhang and Wen, GJI, 2015).

**3. Requirements**

SAC (http://ds.iris.edu/ds/nodes/dmc/software/downloads/sac, for data processing)
TauP (https://www.seis.sc.edu/taup/, for traveltime prediction and slowness calculation)
GMT4 (https://www.soest.hawaii.edu/gmt/, for figure plotting)
PSSAC (http://www.eas.slu.edu/People/LZhu/home.html, for figure plotting)

**4. Usage (Type "MatchLocate2")**

Usage: MatchLocate2 -F(refevla/refevlo/refevdp) -R(maxlat/maxlon/maxh) -I(dlat/dlon/dh)
    -T(window/before/after) -H(N*MAD) -D(INTD) -O(ouput) INPUT.in
-F: searching center (e.g., 37.799/139.998/7.8).
-R: searching area (e.g., 0.05/0.05/5.0).
-I: searching interval (e.g., 0.01/0.01/1.0).
-T: time length of the reference phase (e.g., 4.0/1.0/3.0).
-H: cross-correlation thresholds CC && N(*MAD) (e.g., 0.3/10.0, or 0.3/0.0, or 0.0/10.0).
-D: keep one event within INTD sec (e.g., 6.0).
-O: output (1,2,3) or don't output (0) the cross-correlogram or CC coefficient.
INPUT.in: directories of templates and continuous data, horizontal and vertical slowness, etc.

**5. Examples**

**EX1**. example: an example to show how to use MathLocate2.0 and plot waveform comparison between template and continuous data [see step by step tutorial in README]

**EX2.** example_CCdistribution: an example to show how to plot location resolution based on CC distribution [see step by step tutorial in README]

**6. What's new in Match&Locate2.0?**

See Modified_History. Compared to Match&Locate1.0, main updates include 1) multiple phases can be used together; 2) multiple templates can be used together (useful for improving location, see application in Zhang and Wen, SRL, 2015); 3) location resolution analysis based on CC distribution; 4) adopt CC and MAD thresholds (either or both) and remove SNR threshold.

**7. Q&A**

**Q1:** How to determine thresholds?
**A1:** Thresholds vary with data quality, number of stations, sampling rate, frequency etc. I would suggest you give low thresholds first (reasonable and not too low) and sort your detected events reversely by their CC (from worst detections to best detections). Sequentially plot and eye-check waveforms of those detections to determine reliable thresholds.

**Q2:** Seems it is slow. How to speed up?
**A2:** Compared to matched-filter, Match&Locate would be slower because we search locations in space without the assumption that slave events are co-located with templates. We compute CC function only once and do shift-and-stack (many times) after relative traveltime correction. We employ OpenMP technique in the code. Thus, try to run it on a computer/cluster with multiple cores. Some other suggestions: 1) search a small region and/or use a large search grid; 2) down-

sample both templates and continuous data (e.g., from 100 Hz to 50 Hz); 3) try our GPU-Match&Locate: https://github.com/MinLiu19/GPU-MatchLocate1.0

**Q3:** Why my CCs of self-detections are not exact one (1.0000)?
**A3:** Make sure templates and continuous data have the same sampling rate and frequency filtering. Tips: process and filter the continuous data first and cut your templates from the processed continuous data.

**Q4:** How to run it as matched-filter case?
**A4:** Super easy. Do not search locations (i.e., -R0.0/0.0/0.0).