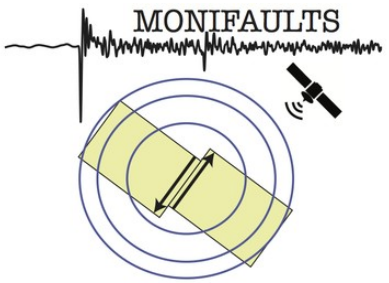
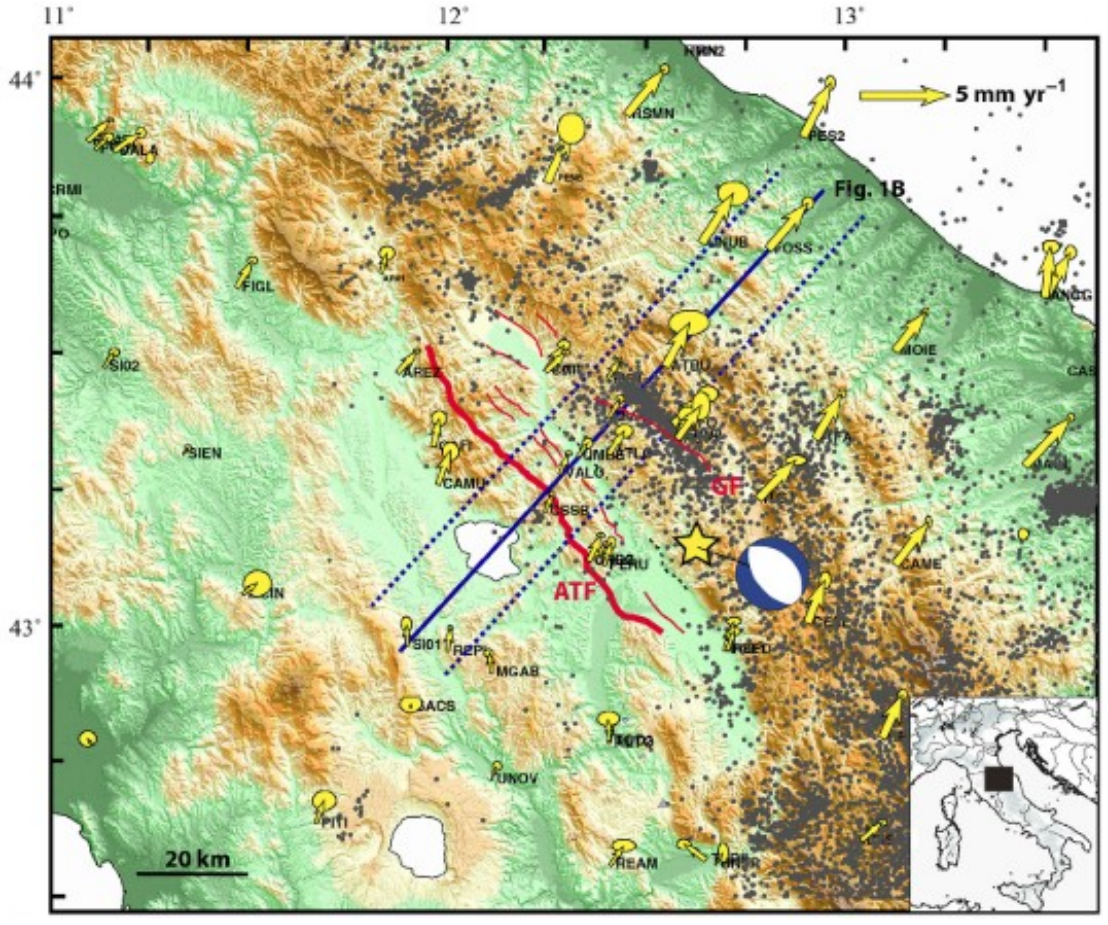
Seismicity at the Alto Tiberina Fault revealed by template matching using GPU

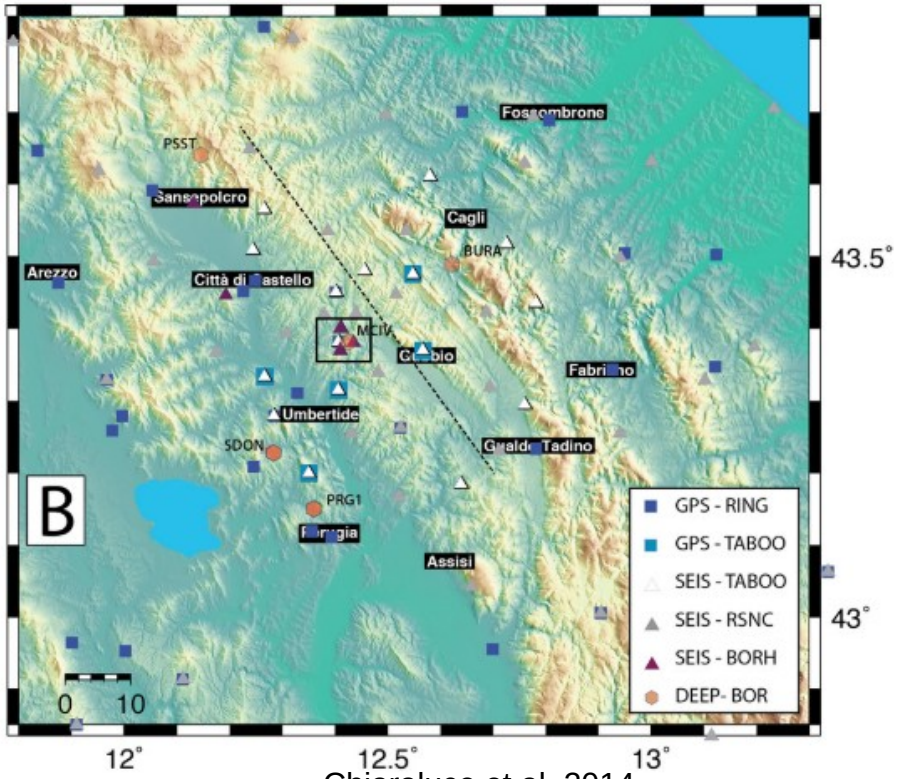
David Essing



The Alto Tiberina Fault



Vadacca et al. (2016)



Chiaraluce et al. 2014

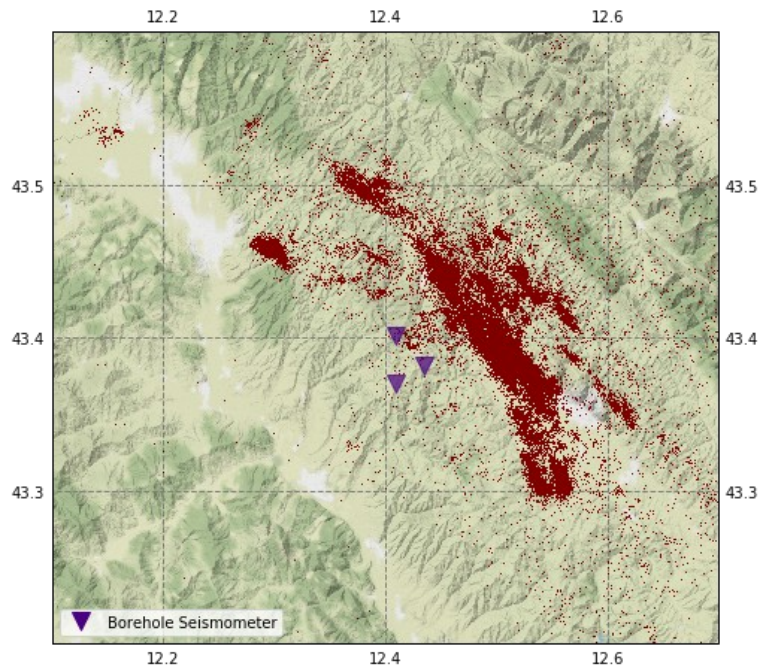
- shallow portion of the fault
  - capable of generating M6.5+ earthquakes
- attached faults
  - Magnitude < 5.2 in 1984 (Haessler et al. 1988)
- historical seismicity indicated
  - Magnitudes > 6

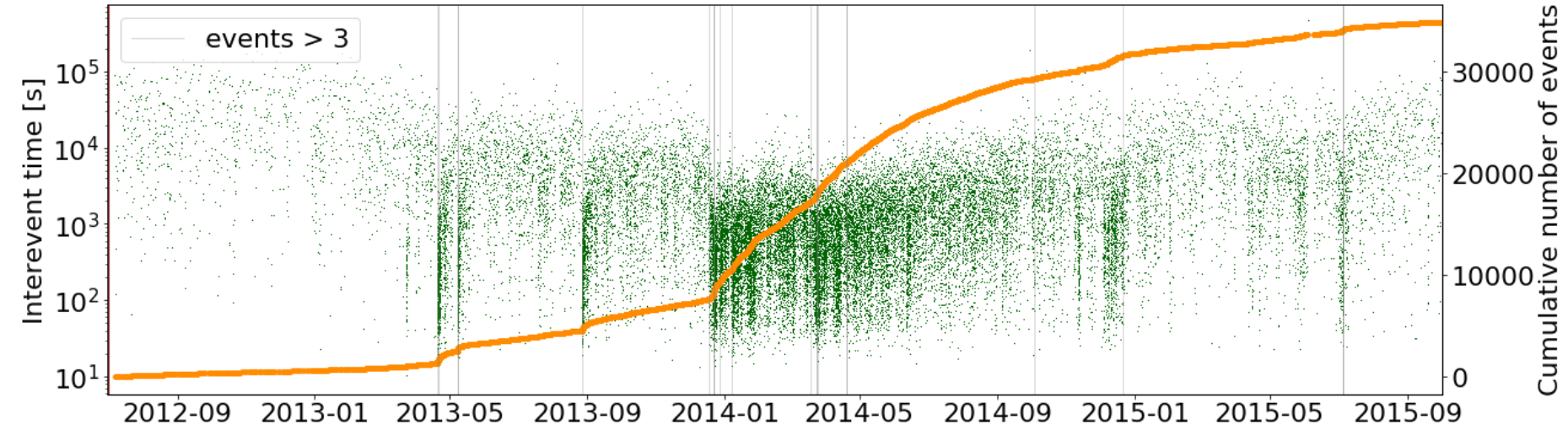
Questions to investigate

- Studying the short- and long-term deformation
- Interaction between the different faults
- Is this type of fault capable of generating earthquakes (L. Anderlini et al. 2016)

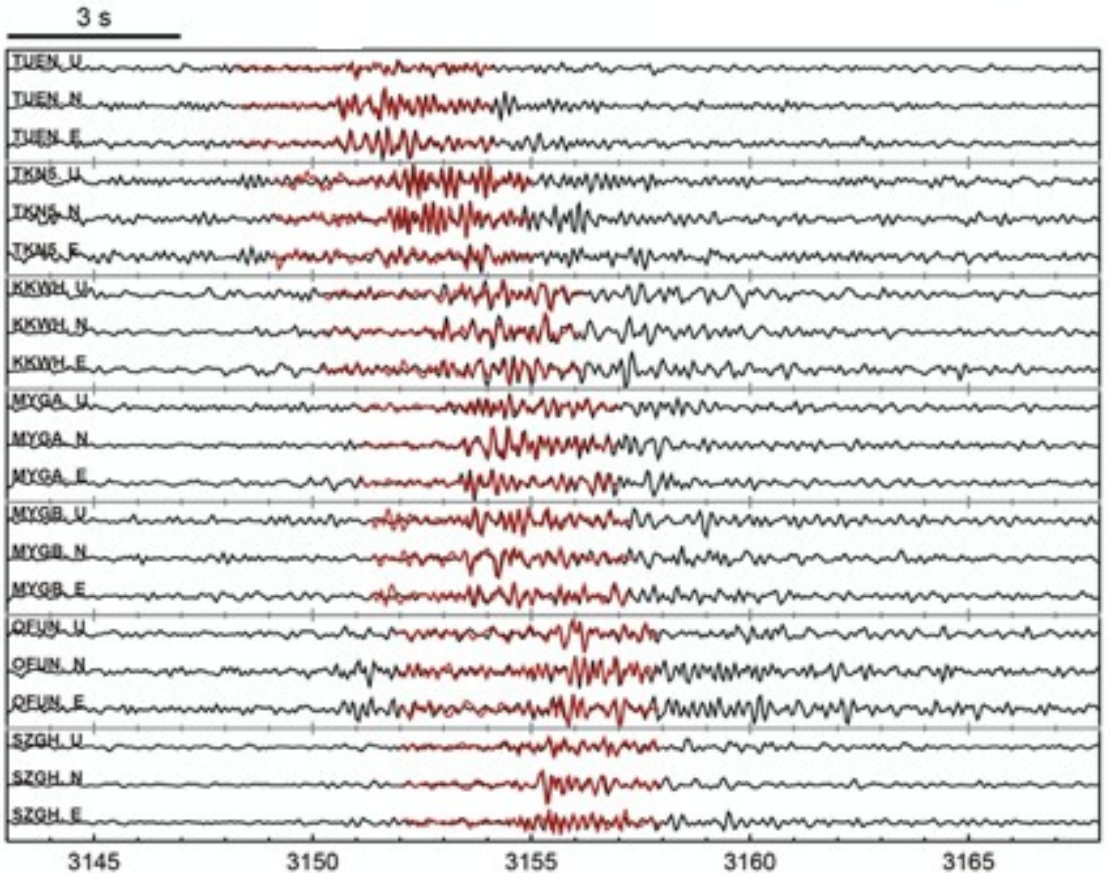
A high resolution catalog is needed

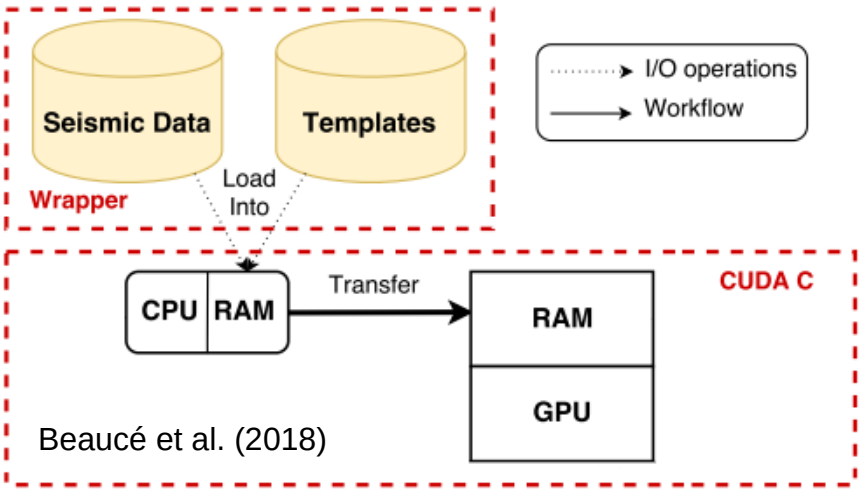
Existing catalog (Chiaraluce, 2019)

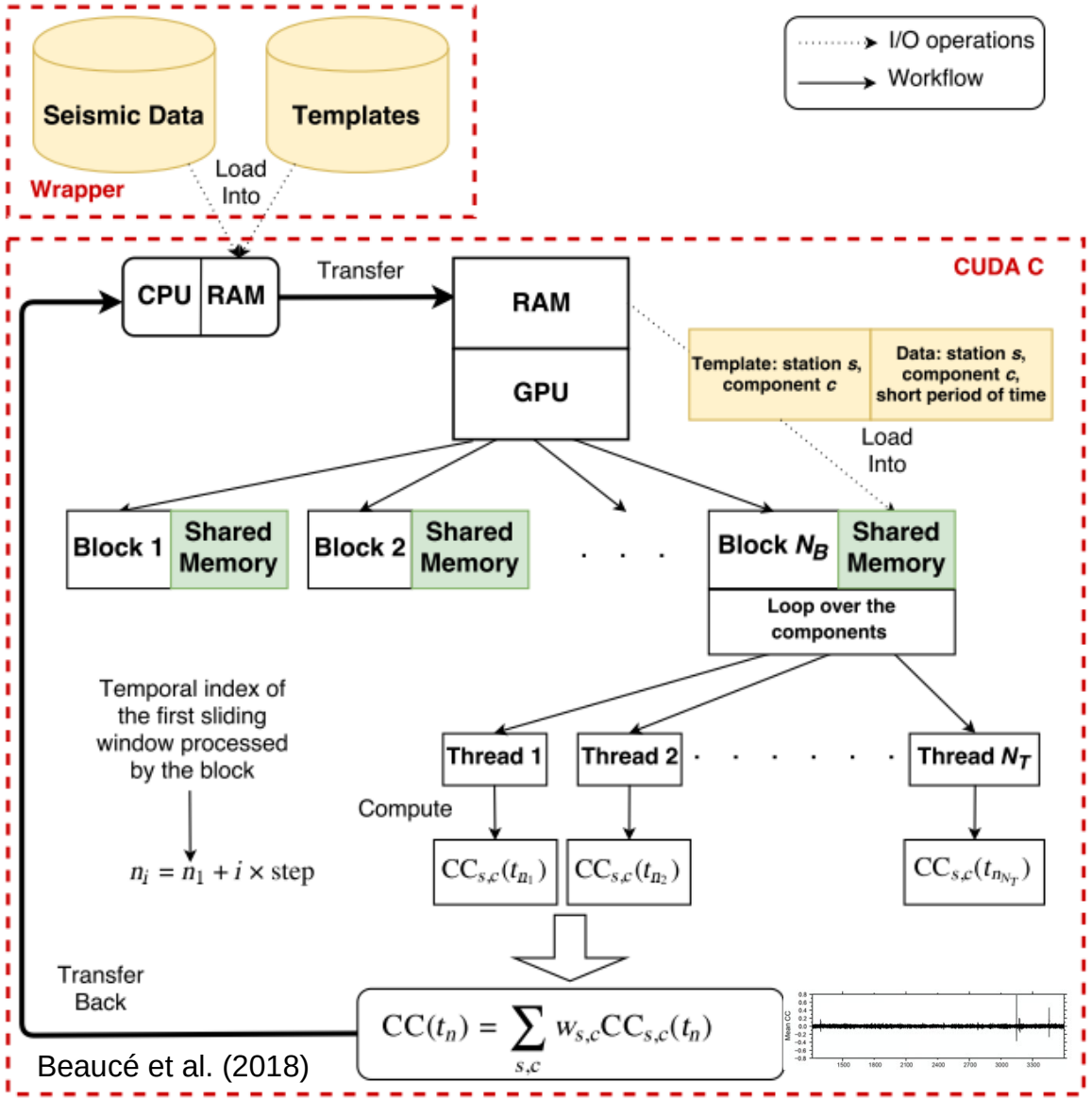




Template matching on GPU

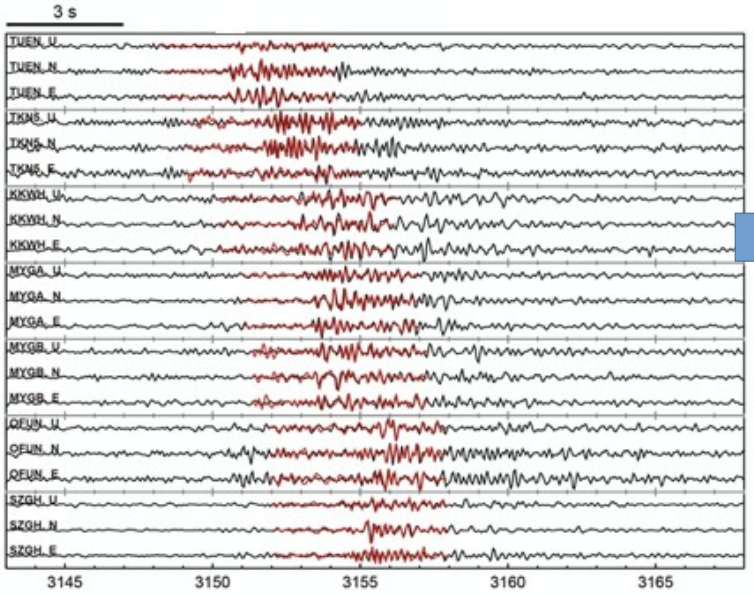


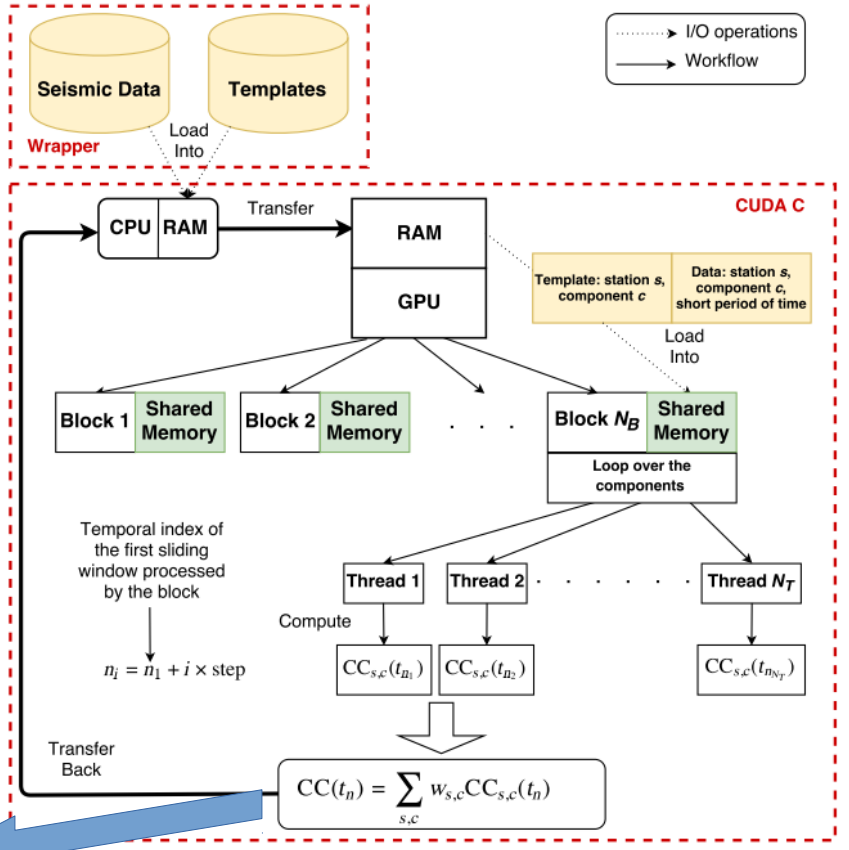


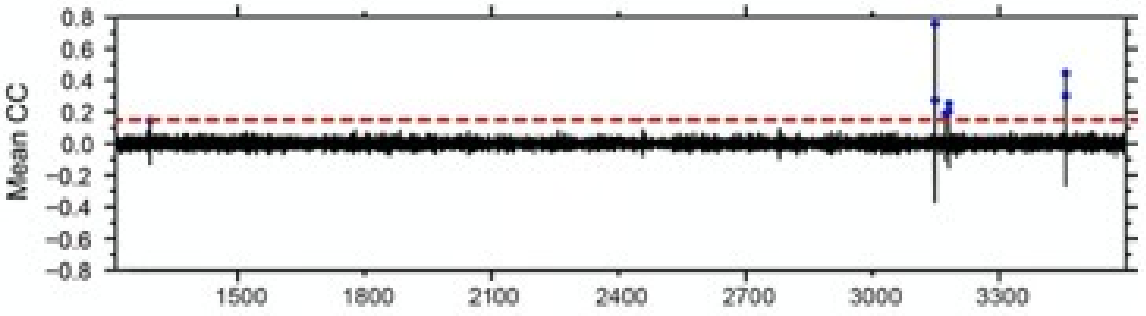


Beaucé et al. (2018)

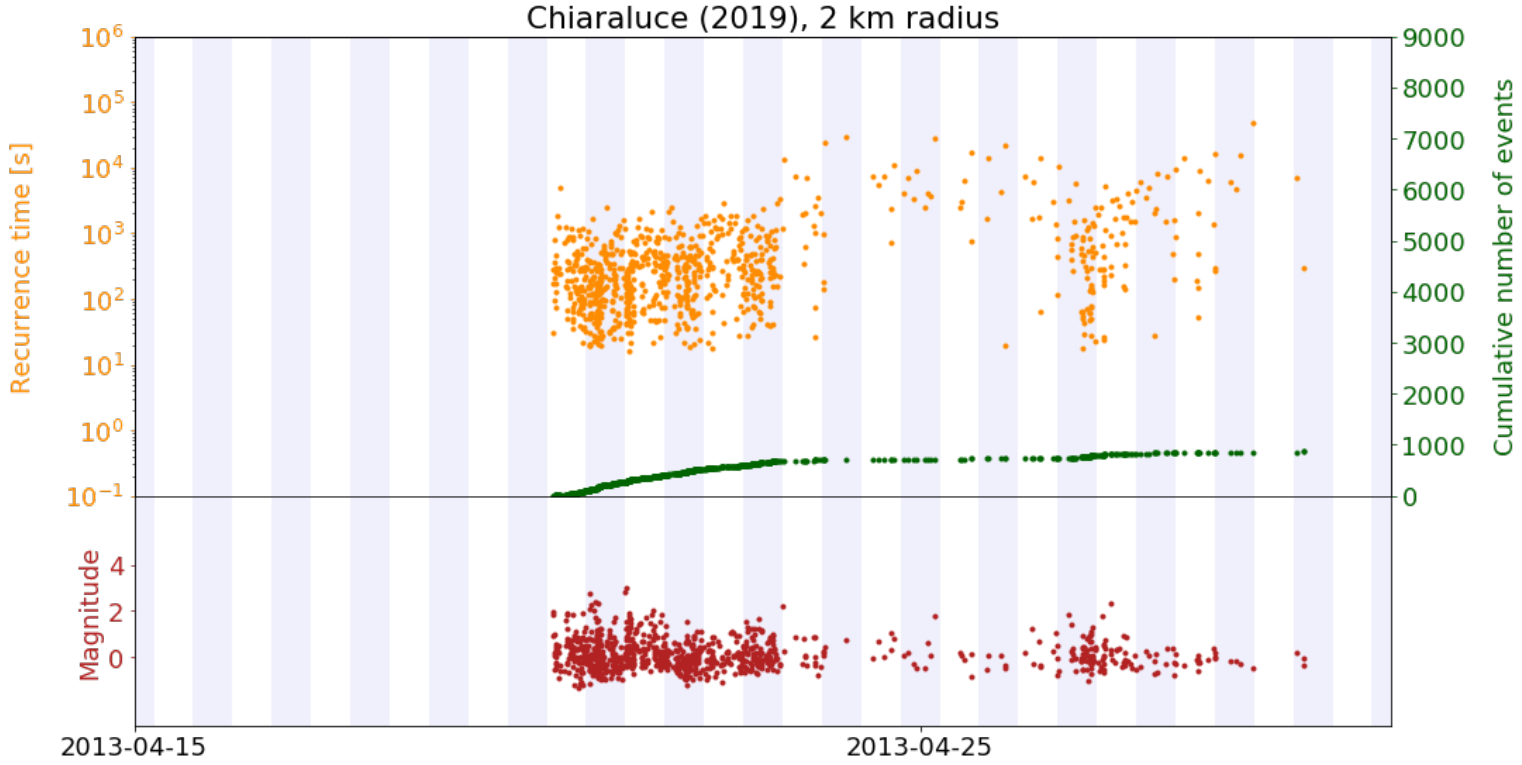
- organization of the computation into blocks and threads
- cc between template and one sliding window of continuous seismic data is computed in a single thread
- throughput of shared memory is 10x larger than the one of global memory

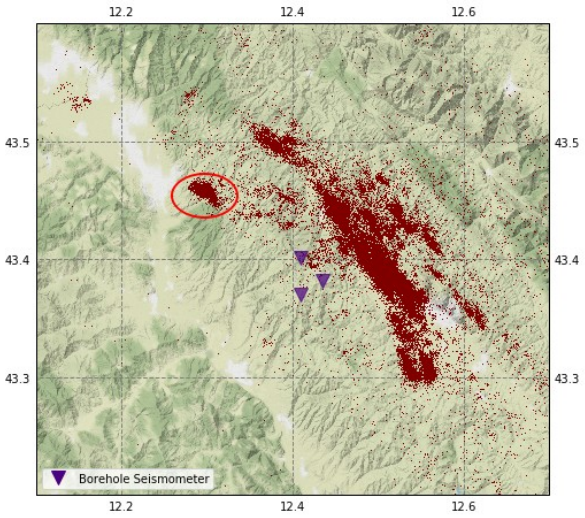

3 s
TUEN_U
TUEN_N
TUEN_E
TKNS_U
TKNS_N
TKNS_E
KKWH_U
KKWH_N
KKWH_E
MYGA_U
MYGA_N
MYGA_E
MYGB_U
MYGB_N
MYGB_E
QFUN_U
QFUN_N
QFUN_E
SZGH_U
SZGH_N
SZGH_E
3145
3150
3155
3160
3165


I/O operations
Workflow
Seismic Data
Templates
Wrapper
Load Into
CUDA C
CPU
RAM
Transfer
RAM
GPU
Template: station s, component c
Data: station s, component c, short period of time
Load Into
Block 1
Shared Memory
Block 2
Shared Memory
Block N_B
Shared Memory
Loop over the components
Temporal index of the first sliding window processed by the block
$n_i = n_1 + i \times \text{step}$
Thread 1
Thread 2
Thread N_T
Compute
$CC_{s,c}(t_{n_1})$
$CC_{s,c}(t_{n_2})$
$CC_{s,c}(t_{n_{N_T}})$
Transfer Back
$CC(t_n) = \sum_{s,c} w_{s,c} CC_{s,c}(t_n)$

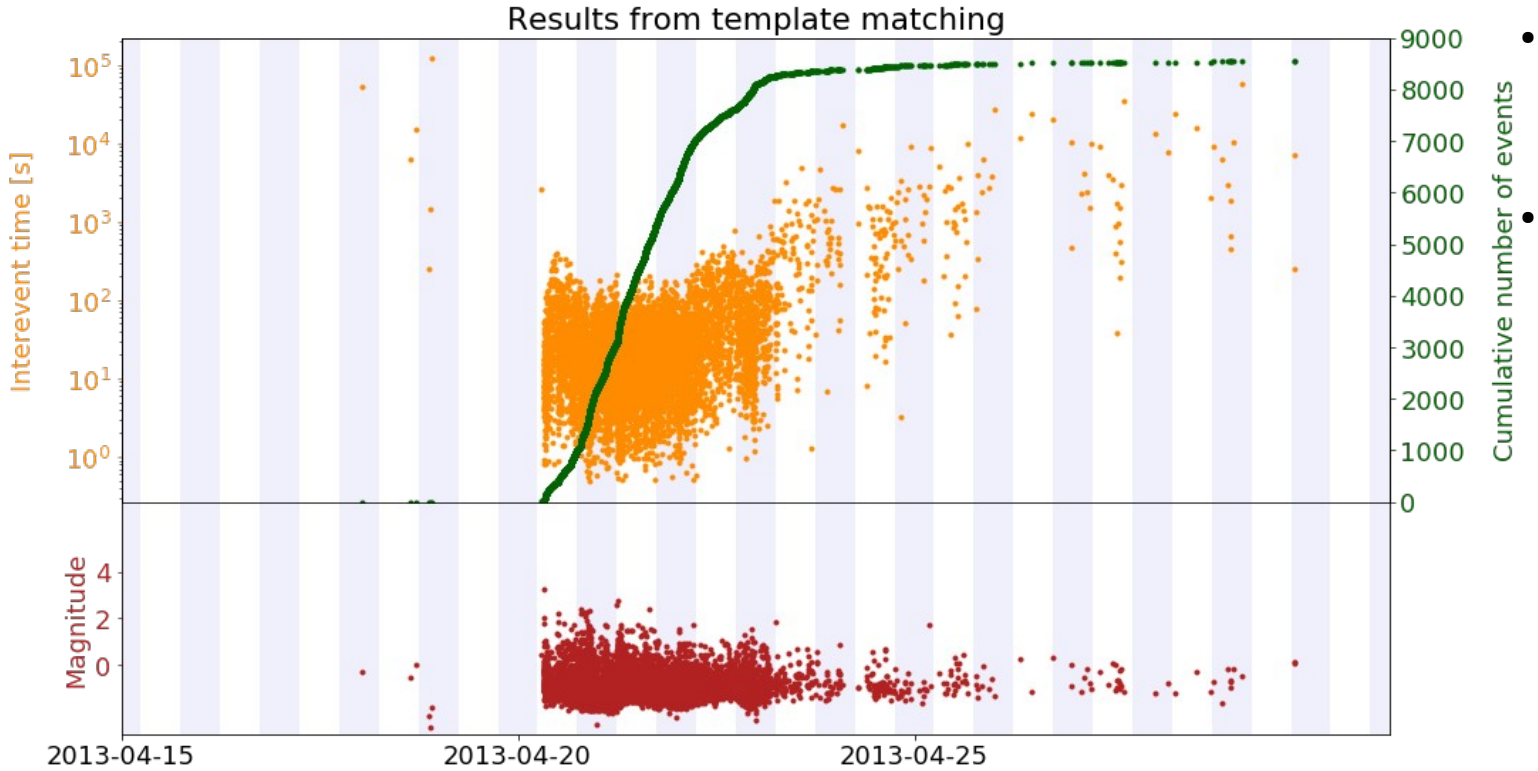

Mean CC
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
1500
1800
2100
2400
2700
3000
3300

Existing catalog





Extended catalog – New detection’s



- with ~450 template’s we are able to find ~9000 events
- we are able to find 9 times more events as listed in the existing catalog

Tests on dahu

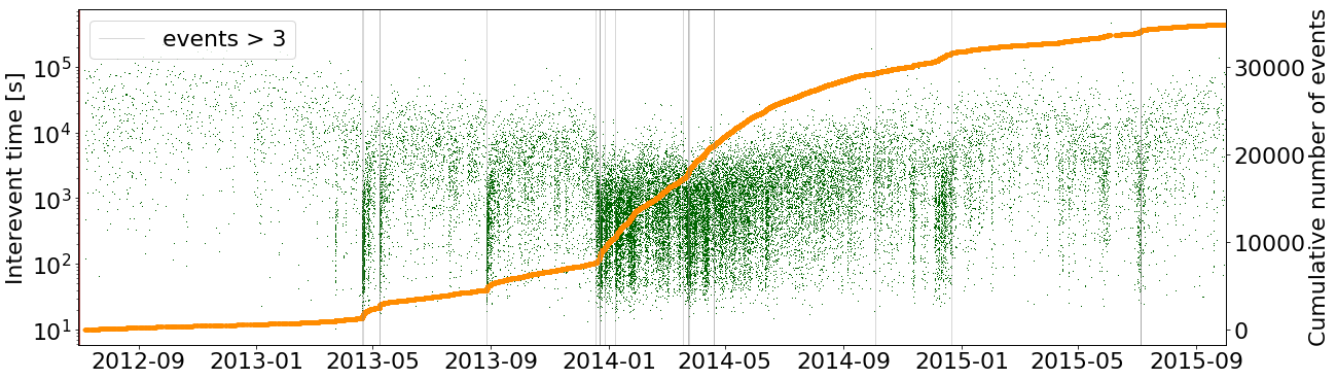
- for computation we used 1 GPU device with 8 cores

	test #1	test #2	test #3	test #4
Input				
days of continuous seismic data [d]	1	860	30	30
templates	200	200	231	218
computational time [s]	60	40080	1800	1680
computational time per template and day [s]	0.30	0.23	0.26	0.26

Length of the whole experiment

- 1,200 days of continuous seismic data
- 24,000 templates (we will only use high quality templates)
- 0.26 seconds of computational time per day and template
- ~ 90 days of computational time (1 GPU device, 8 cores on dahu)

How can we do that in a reasonable amount of time?



Thank you

- for attention
- for providing the CIMENT wiki
- for any help and suggestions

Additional things