

Deliverable

D1.4. Seismic Network Optimization

Report information	
Work package	WP1: Seismic Monitoring and Imaging
Lead	ETH Zurich
Authors	Arnaud Mignan, Marco Pilz
Reviewers	Verónica Antunes
Approval	Vanille Ritz
Status	Final
Dissemination level	Public
Will the data supporting this document be made open access?	Yes
If No Open Access, provide reasons	N/A
Delivery deadline	15.06.2026
Submission date	01.04.2026
Intranet path	GENERAL/FILES/C.DELIVERABLES/Deliverable_1_4

GEOTWINS Project Office | ETH Zurich, Dept. of Earth and Planetary Sciences
Sonneggstrasse 5, 8092 Zurich | geotwins-office@sed.ethz.ch | www.deepgeothermal.org



Co-funded by
the European Union

This Project is funded by CETPartnership, the Clean Energy Transition Partnership under the 2023 joint call for research proposals, co-funded by the European Commission (GA N°101069750) and with the national funding organizations detailed on <https://cetpartnership.eu/funding-agencies-and-call-modules>.

Table of Contents

Summary 3

1. General methodology.....4

1.1 Completeness magnitude prediction: BMC method.....4

1.2 Seismic network optimisation: Simulated annealing algorithm.....5

2. Seismic network Digital Twin Component (DTC)..... 5

3. Appendix A: Scientific publication 9

4. Appendix B: DTC codes.....9

5. Appendix C: Complementary approach based on stochastic modelling..... 10

References 14

Summary

According to the GEOTWINS project description, the seismic network 'DTC simulates a realistic distribution of seismicity by masking "undetected" events that fall below the completeness magnitude M_c , which is modelled from the spatial distribution of a seismic network virtual replica. The full-range magnitude distribution will follow the asymmetric Laplace model for a constant M_c (Mignan, 2012), with more complex distribution shapes to be retrieved for a spatially variable $M_c(x,y,z)$. The $M_c(x,y,z)$ distribution is itself a function of seismic station characteristics assessed with the Bayesian Magnitude of Completeness (BMC) method (Mignan et al., 2011) [...]. ML approaches, such as simulated annealing and genetic algorithms, will be employed to generate seismic network configurations [...]. In other words, the seismic network Digital Twin Component (DTC) primarily functions as a data filter, removing events from the input seismicity catalogue based on their magnitude (m) and location (x,y,z), with the resulting catalogue provided as output. The filtering process depends on the spatial configuration of the seismic network, which can either be supplied as input or optimized for the DTC by an add-on module. The filtering is governed by the BMC parameters, which are generated independently of the DTC. This workflow is illustrated in Figure 1a: The true seismicity distribution will be provided by physics-based models developed in GEOTWINS WP2. For demonstration purposes, however, a simplified statistical model calibrated to observations from the Utah FORGE Enhanced Geothermal System (EGS) was developed for D1.4. Within the DTC, events are filtered according to their probability of detection based on magnitude, represented by the exponential function $q(m, M_c(x,y,z))$. This function corresponds to the lower tail of the asymmetric Laplace model, with the upper tail following the Gutenberg-Richter law above M_c . This filtering rule thus depends on the $M_c(x,y,z)$, which is predicted by the BMC model given a specific seismic network configuration. The seismic network itself can be further optimised externally, using feedback from the DTC regarding how network design influences the output seismicity catalogue. For the FORGE 2024 stimulation test, the DTC (output shown in green) successfully reproduced the original magnitude distribution (in grey). Figure 1b demonstrates the DTC's applicability in a stress-test scenario, where the entire DAS network is removed (i.e., fails). Under this condition, and given the FORGE configuration and DTC models, more than 60% of seismic data would be lost (output in purple). These results both validate the DTC process and demonstrate its potential for network performance evaluation and resilience testing.

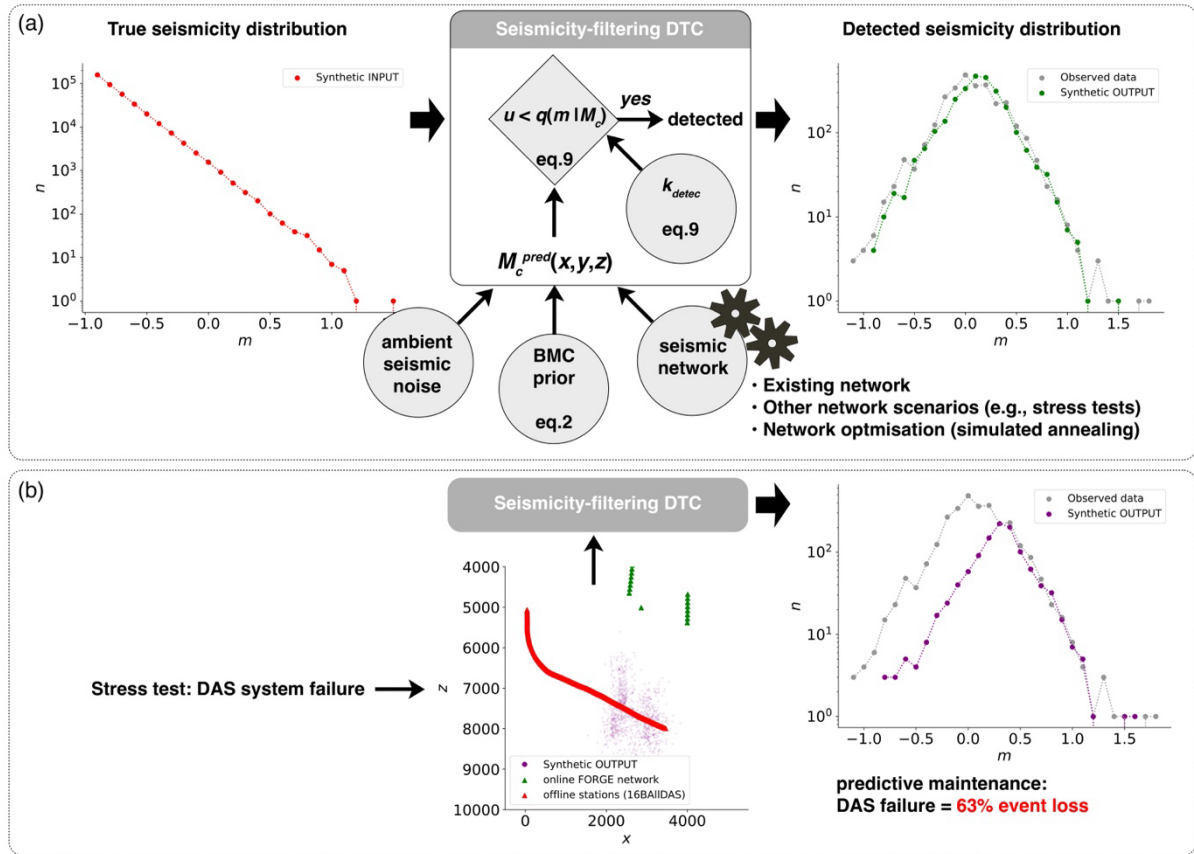


Fig. 1. The seismic network DTC. (a) DTC workflow. (b) Example of application at the FORGE EGS site (stress test of the existing network). See text for details (figure taken from the article linked in Appendix A).

1. General methodology

The full description of the methodology can be found in the article '*Bayesian Magnitude of Completeness Method with Machine Learning-based Ambient Seismic Noise Bias Correction for Network Optimisation in a Digital Twin*' (Appendix A). The main results are summarised below:

1.1 Completeness magnitude prediction: BMC method

The Bayesian Magnitude of Completeness (BMC) method is a straightforward and robust approach for mapping the completeness magnitude (M_c) using prior information on the distribution of seismic stations. It can be applied to assess the quality of national and regional seismicity catalogues with respect to earthquake detectability, and to forecast the lowest magnitudes future catalogues could capture for planned network configurations. Until now, its application to network design has been limited to general guidance rather than precise station siting, primarily due to the need to account for local ambient seismic noise. For GEOTWINS, we first improved the BMC method by taking into account the role of depth (pseudo-3D, 2D-layered BMC), which is crucial for the EGS context. Second, we corrected predicted M_c bias due to seismic noise using a Gradient Boosting model with two feature types for the k -th nearest station considered in the BMC mapping: (1) a land-use proxy for seismic noise, and (2) distance to roads. The method has been evaluated for the 2013–2024 Swiss seismic network and catalogue and shown to explain 54% of the BMC prior noise (down to 21–33% in spatial cross-validation of different resolutions). Correcting for ambient seismic noise is crucial in the present project to obtain realistic seismic network configurations during optimization (see section 1.2).

1.2 Seismic network optimisation: Simulated annealing algorithm

The applicability of the improved BMC approach was demonstrated for a local network configuration at an EGS site, using the Utah FORGE data as a reference setting. Station siting was optimised via a simulated annealing algorithm based on BMC-predicted M_c values that can now incorporate site-specific ambient noise estimates. Two types of stations were defined: surface stations (where ambient seismic noise may be present) and borehole stations (along a borehole path at depth). Multiple simulations have been realized to illustrate how optimal network configurations emerge given a number of stations, M_c threshold to reach, and local noise patterns. This approach, combining BMC M_c prediction and network optimisation, remains complementary to seismic waveform-based methods while providing a fast, catalogue-based tool for network evaluation and design. The potential of the method is illustrated in the context of the FORGE-EGS digital twin, where the impact of network changes on detected seismicity are dynamically explored (see section 2).

Learn more from Appendix A:

- **Scientific publication:** Full description of the BMC method, including its extension to the 3D case, its noise-bias correction by machine learning, the network optimization via simulated annealing, and the filtering strategy at the core of the proposed DTC.
- **Python codes and Jupyter notebooks:** Models and tutorial to reproduce the entire study.

Note: For completeness, a seismic waveform-based network optimization was also developed independently by GFZ, based on a genetic algorithm. The method and results are presented in Appendix C. Although independent of the main study presented here, the resulting optimal network can be also used as input in the DTC.

2. Seismic network Digital Twin Component (DTC)

The seismic network DTC to be handed over to WP5 of GEOTWINS comprises two main elements: (i) the filtering process, which incorporates the BMC method and the asymmetric Laplace function, and (ii) the seismic network optimisation add-on module. Enhancements to the BMC method, representing the core of the article referenced in Appendix A, are not included within the DTC itself but are instead treated as externally defined inputs.

The DTC is fully encapsulated in the Python code package and accompanying tutorial provided in Appendix B (see code excerpt therein).

The overall structure of the DTC is as follows:

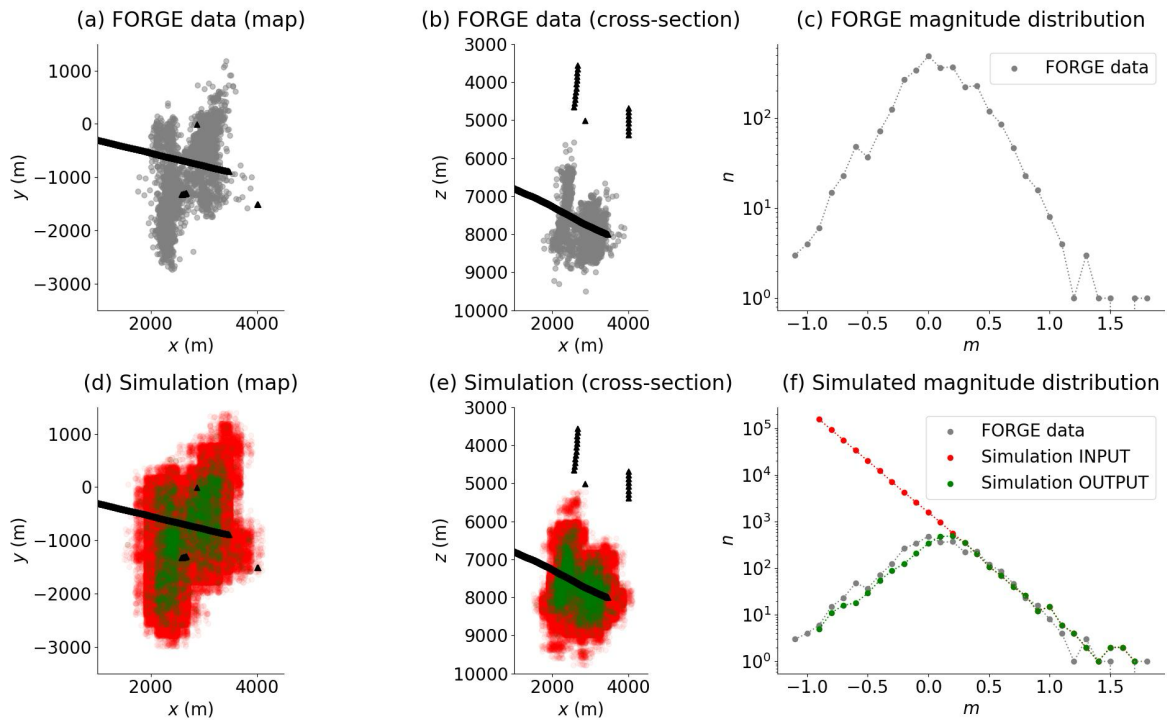
- **INPUTS:**
 - Seismicity catalogue (`seism_IN`)
 - Seismic network (`net_IN`)
 - BMC model (`BMC_par`): BMC prior parameters, Gradient boosting regressor model (for noise-bias correction), and detectability parameter (for function q)
- **Process:** Filter `seism_IN` according to `net_IN` and `BMC_par`
- **OUTPUTS:**
 - Filtered seismicity catalogue (`seism_OUT`)

The structure of the network optimization add-on module is as follows:

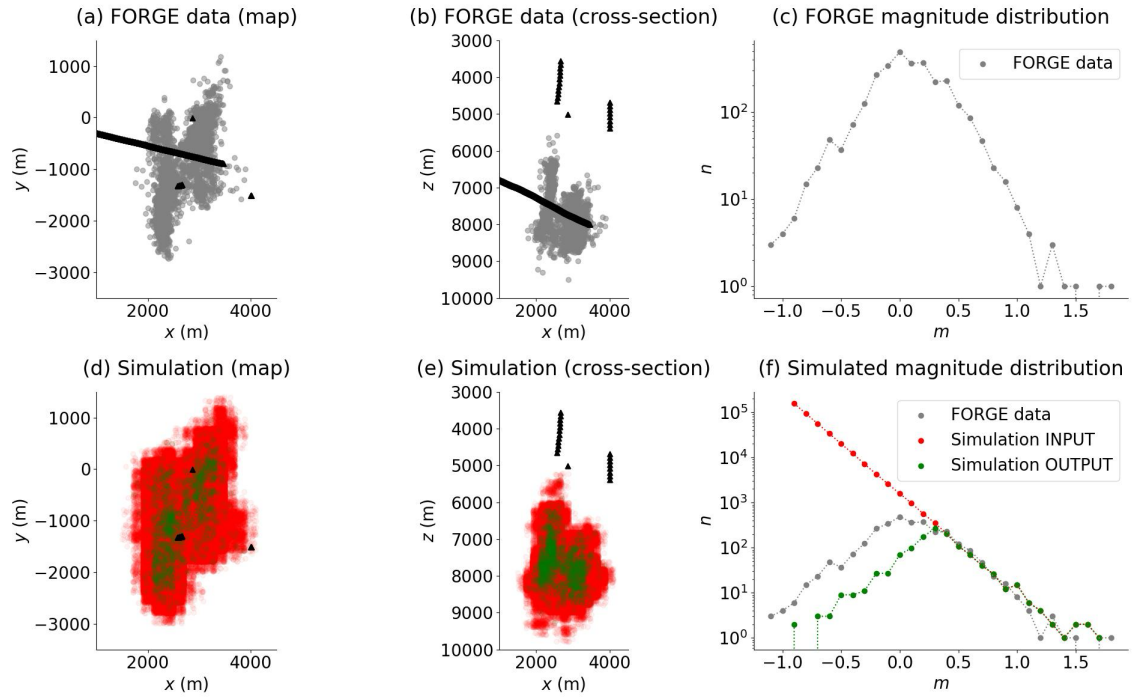
- **INPUTS:**
 - Seismic network (`net_IN`): random network for initialization (or existing network)
 - Seismicity catalogue (`seism_IN`): for back-azimuth cost function in simulated annealing
 - Optimisation parameters (`netOpt_par`): Pre-selected simulated annealing parameters
 - Ambient seismic noise (Optional, `landuse_noise`, `roads`): Land-use noise proxy and road network
- **Process:** Create `net_OPT` given `net_IN` via simulated annealing algorithm
- **OUTPUTS:**
 - Optimised seismic network (`net_OPT`): which can become `net_IN` in the DTC

For guidance on the process and available options, five built-in test cases have been implemented, with their results presented below (see Appendix A and Appendix B for the full description):

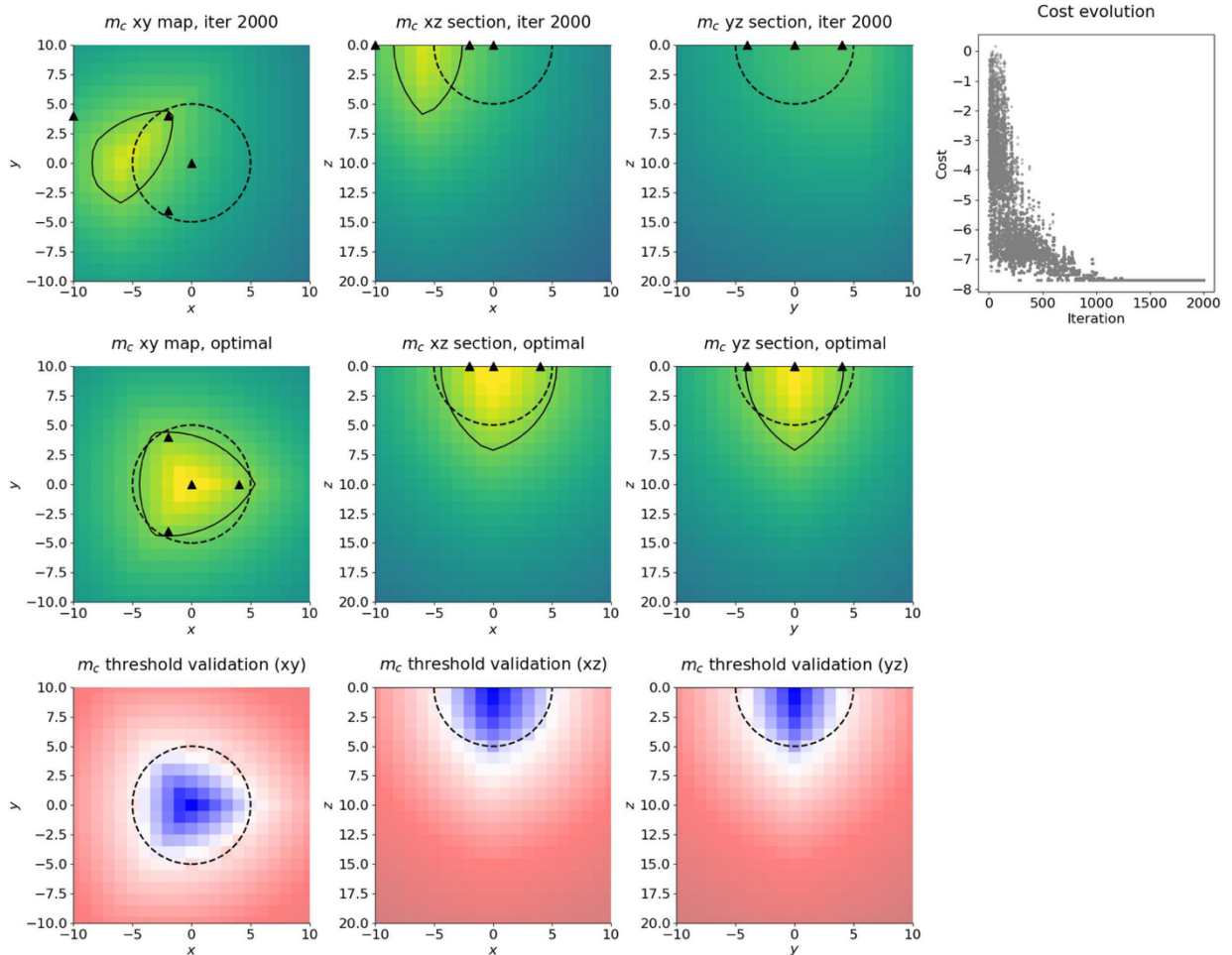
- **'FORGE2024_reproduced'**: Simulate the FORGE seismicity filtering.



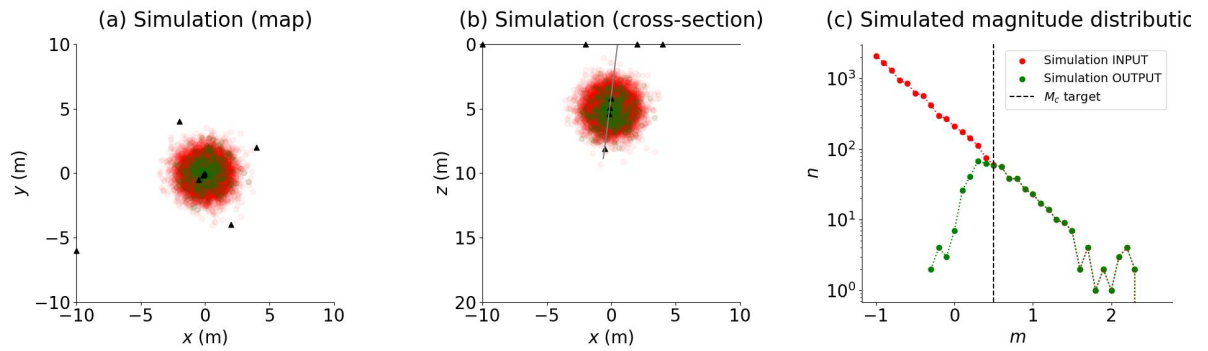
- **'FORGE2024_stresstest'**: Explore the impact of a failed DAS.



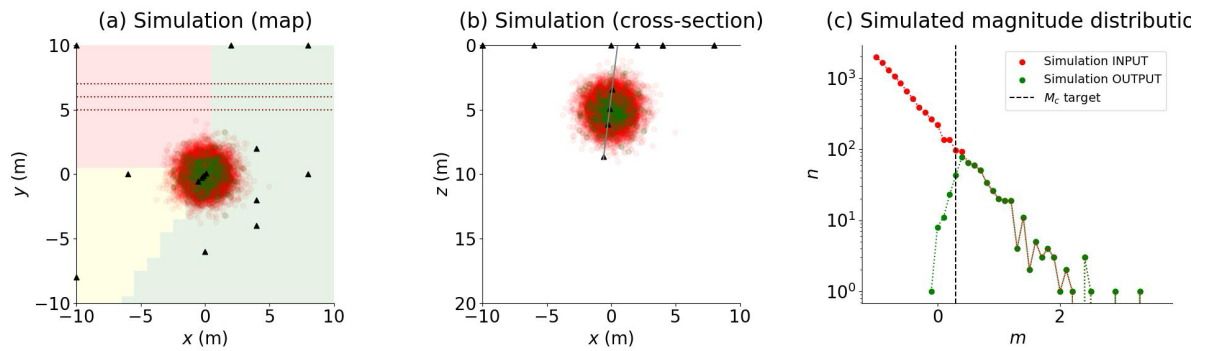
- **'generic_benchmark'**: Basic network optimisation validation test. Intermediary result shown below, representing the last iteration of the simulated annealing process.



- **'generic_cloud'**: Basic network optimisation for simulated EGS seismicity cloud.



- **'generic_cloud_wSeismicAmbientNoise'**: As above but including some simulated ambient seismic noise as input.



3. Appendix A: Scientific publication

This deliverable reports on work that is being prepared as a manuscript to be submitted for peer-reviewed publication. The associated Python codes and tutorial will be made available on the ETH research collection once the article is published.

4. Appendix B: DTC codes

See the files ‘DTC1_4.py’ and ‘DTC1_4.ipynb’ for the DTC codes and tutorial, respectively. Below is an excerpt of the code where the core filtering process is done:

```
#####
## DTC PROCESS: FILTERING ##
#####

def calc_mcmesh(net_IN, mesh_par, BMC_par):
    """
    Calculate the completeness-magnitude 3D mesh given a seismic network and the BMC
    prior.
    """

    xi = np.arange(mesh_par['xmin'], mesh_par['xmax'] + mesh_par['dxyz'],
mesh_par['dxyz']) + mesh_par['dxyz']/2
    yi = np.arange(mesh_par['ymin'], mesh_par['ymax'] + mesh_par['dxyz'],
mesh_par['dxyz']) + mesh_par['dxyz']/2
    zi = np.arange(mesh_par['zmin'], mesh_par['zmax'] + mesh_par['dxyz'],
mesh_par['dxyz']) + mesh_par['dxyz']/2
    xii, yii, zii = np.meshgrid(xi, yi, zi, indexing = 'ij')
    nx, ny, nz = xii.shape

    N = len(net_IN)
    stations = net_IN[['x', 'y', 'z']].values
    d2sta_flat = np.array([np.sqrt((stations[i,0] - xii.flatten())**2 + \
                                (stations[i,1] - yii.flatten())**2 + \
                                (stations[i,2] - zii.flatten())**2) for i in
range(N)])
    d_k_flat = np.array([np.sort(d2sta_flat[:,i])[BMC_par['prior']['kth'] - 1] for i
in range(nx*ny*nz)])
    d_k = d_k_flat.reshape(nx,ny,nz)

    if mesh_par['spatial_unit'] == 'm':
        d_k = d_k * 1e-3      # km

    mc_pred = BMC_par['prior']['c1'] * np.log(d_k) + BMC_par['prior']['c2']      # d_k
in km in BMC prior

    return mc_pred, xii, yii, zii

def filter_m(m, mc, kappa):
    """
```

```

    Simulate earthquake detectability by returning a boolean array indicating which
    magnitudes are detected based on completeness
    threshold and detection probability.
    '''
    detectability = np.exp(kappa * (m - mc))
    u = np.random.rand(len(m))
    detected = (m >= mc) | (u <= detectability)
    return detected

def filter_seism(seism_IN, mc_mesh, xii, yii, zii, mesh_par, BMC_par):
    '''
    Filter a seismic catalogue by applying spatially varying magnitude-of-
    completeness thresholds, returning only the detected events.
    '''
    x = np.asarray(seism_IN['x'])
    y = np.asarray(seism_IN['y'])
    z = np.asarray(seism_IN['z'])
    m = np.asarray(seism_IN['m'])

    detected = np.zeros(len(m), dtype=bool)
    nx, ny, nz = xii.shape

    print('Filter seismicity in mesh (x,y,z)...')
    for i in range(nx):
        print(f'{i+1}/{nx}', end='\r')
        for j in range(ny):
            for k in range(nz):
                indcell = (
                    (x > xii[i,j,k] - mesh_par['dxyz']/2.) & (x <= xii[i,j,k] +
                    mesh_par['dxyz']/2.) &
                    (y > yii[i,j,k] - mesh_par['dxyz']/2.) & (y <= yii[i,j,k] +
                    mesh_par['dxyz']/2.) &
                    (z > zii[i,j,k] - mesh_par['dxyz']/2.) & (z <= zii[i,j,k] +
                    mesh_par['dxyz']/2.)
                )

                Nev_all = np.sum(indcell)
                if Nev_all > 0:
                    indcell_detected = filter_m(m[indcell], mc_mesh[i,j,k],
                    BMC_par['kappa'])
                    abs_indices = np.where(indcell)[0][indcell_detected]
                    detected[abs_indices] = True

    return seism_IN[detected]

```

5. Appendix C: Complementary approach based on stochastic modelling

Database

A complementary approach is based on a representative sample of relevant seismic recordings derived from stochastic simulations of scenario earthquakes (Boore 2009). The simulations are based on a theoretical catalogue of seismicity (or physics-based catalogue from WP2). Overall, 300 randomly chosen events were considered. The events are located at depths between 4 km and 6 km and the magnitudes are varying between -2 and 3 . The range of magnitude and depth was selected considering that the monitoring network is aimed at the accurate characterization of both small events in vicinity of the reservoir and of deeper, likely slightly larger ones. Since the main goal is to detect small magnitudes, we set the stress drop relatively low (1 MPa). We considered a simplified 1D layered velocity and attenuation model for the Utah FORGE test site proposed by Wang and Stewart (2022). Finally, the effect of noise is considered by adding two different noise levels (10^{-8} m/s and 10^{-5} m/s). For each event, synthetic seismic traces are computed on a 50 m grid for a total area having an extension of 2.0×2.5 km². The final database consists of 288,000 synthetic traces representing ground velocity.

Network optimization

The optimal network is the one that can provide give the maximum number of detectable and localizable events for the highest number of scenario events. To quantify the network quality, we modify the cost function developed by Oth et al. (2010):

$$\text{cost} = \sum_{i=1}^N W_i H_{ij} \quad (\text{C1})$$

where N corresponds to the number of events (300 in our case). H is 1 if event i has been identified correctly at site j and 0 otherwise, and W is the weight associated with each event. By minimizing the cost function in equation (C1), we maximize at the same time the number of correct identifications. The final cost for the network system is thus given by the weighted sum (normalized to unity) of these individual network costs, with weights chosen as $W = (M+3)^{-1}$ with M being the magnitude of the simulated scenario event. Since we only simulate magnitudes greater than -2 , the weights will always be positive and the smaller the magnitude, the greater the weight (i.e., correct identification of small magnitude events is weighted more strongly). The weighting of events in equation (C1) is necessary because the number of small magnitude events is much larger than the number of larger magnitude events; this is also to be expected in reality. If all earthquakes are equally weighted, the events of larger magnitude, which will be detected everywhere, would clearly dominate the evaluation scheme.

To minimize the cost function and thus determine the optimal station locations, a microgenetic algorithm (MGA, Krishnakumar 1990) has been applied. MGAs are a specific subset of genetic algorithms (GAs) which use evolutionary principles to guide the search for optimal models with respect to a given objective function (i.e. the cost function). Typically, a random set of initial models (i.e. network sites), known as the population, is generated at the beginning of the algorithm. Each model, known as a chromosome, consists of parameters known as genes. In our case, the genes are the positions of the stations within the grid. The cost function is then computed for each chromosome, and the fittest chromosomes (i.e. network with the lowest cost function) are permitted to mate and exchange genetic information via a predefined crossover operator.

They may also undergo random mutations. After a large number of iterations (known as generations), the algorithm converges on a minimum cost value representing the optimal network. Due to the random initial conditions and the non-uniqueness of the optimisation problem, the solution is likely to be local rather than absolute minimum. Therefore, it is important to perform a number of independent runs to obtain an overview of the range of optimal solutions. MGAs differ from classic GAs in that they use small population sizes. Each time the algorithm has converged (in this study, this is considered to be the case if fewer than 5% of genes differ between chromosomes in the population), the population size is reset at random, except for the fittest chromosome, which remains in the population. Oth et al. (2010) found that MGAs were appropriate for computing optimal networks for Istanbul, and we apply a similar approach here. The population size was set to 100 for all simulations. The crossover operator was set to 0.95 and no mutations were used. Each algorithm was permitted to run for 1,000 generations. These values were deemed appropriate following trial-and-error simulations with different parameter values. Networks

with between six and 14 stations were considered. For each combination of number of stations and sigmoid, three independent runs of the MGA, each using random input, were performed. The output of each MGA run is not just the network with the lowest cost function, but rather a number of networks with slightly higher cost functions.

Results and discussion

An important question in designing a network is how many stations are necessary. To establish this, we look at the lowest cost function found by three independent MGA runs for networks between six and 14 stations. Figure C1 indicates that for the sample case with a smaller number of stations, adding one station can make a significant improvement to the cost function. While the cost function always decreases with an increasing number of stations, this decrease becomes gradual once more than seven stations exist in a network.

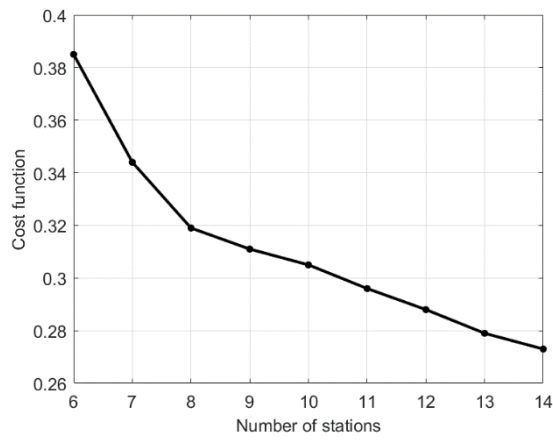
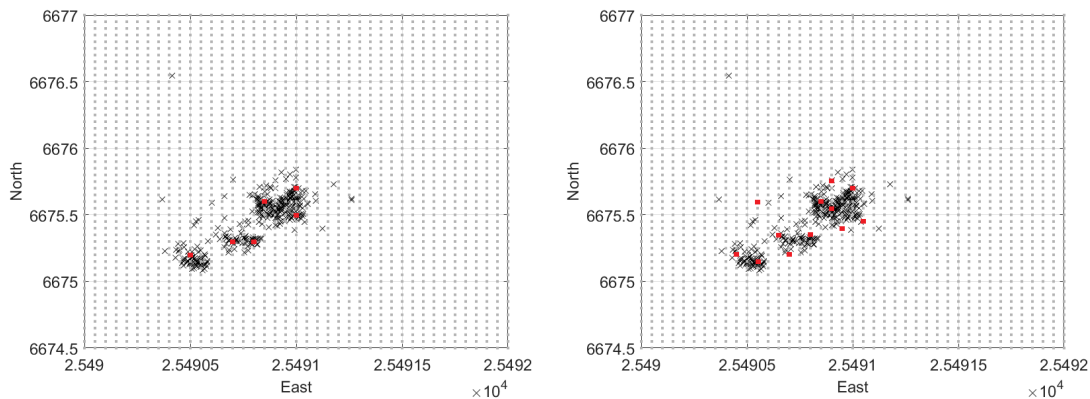


Figure C1: Lowest values for the cost function obtained for different numbers of stations used.

Using the MGA described above, we obtained tens of thousands network layouts. As these are impossible to present in their entirety, we attempt to identify patterns in the best stations. For this, Figure C2 shows the locations of the stations appearing in at least 10 % of all solutions.



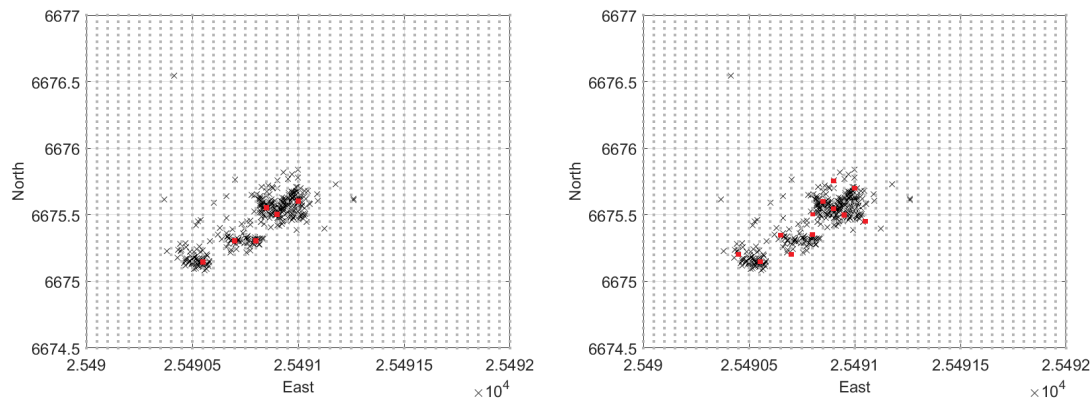


Figure C2: Analysis and comparison of variability in terms of station configuration for three independent runs of the MGA for different numbers of stations (left: six stations, right: 12 stations) for low (top row) and high (bottom row) levels of noise. Each gray grid point (possible stations location) appearing in more than 10% of the best configurations is depicted with a red circle. The sample events are indicated by black crosses.

The results reveal a partially modified sensor configuration in the case of 12 with respect to the configuration of six stations. In the case of 12 stations, the outer sites in particular are better sampled. Results further show that for low levels of noise (10^{-8} m/s), all magnitudes can be detected. A high level of noise (10^{-5} m/s), on the other hand, leads to stations being closer together and events further away are no longer detected reliably.

Summary

We examined the theoretical performance of various station layouts in terms of optimal network design and maximum detection capability. While we have focused on a sample application with a limited depth/distance range with a fixed stress drop and two different levels of noise, it is worth noting that key parameters (i.e. station-specific noise level and event-specific stress drop) can also be adjusted accordingly. We simulated different station layouts and use a micro-genetic algorithm to identify best-possible station locations for a variable number of stations at the surface. While we have presented examples of highly efficient networks, it is necessary to stress that these are not the only solutions possible. With several different network configurations of similar quality available, it is possible to introduce further constraints on the optimal network. This can be done by excluding certain sites (i.e., by introducing a penalty into the cost function and re-running the MGA). For example, if prior to the installation, a certain part of the study area is inaccessible, networks containing stations there can be removed from the list of solutions and the network with the lowest cost among those remaining can be installed.

References

- Boore, D. M. (2009). Comparing stochastic point-source and finite-source ground-motion simulations: SMSIM and EXSIM. *Bulletin of the Seismological Society of America*, 99(6), 3202-3216.
- Davies, R., Foulger, G., Bindley, A., & Styles, P. (2013). Induced seismicity and hydraulic fracturing for the recovery of hydrocarbons. *Marine and petroleum geology*, 45, 171-185.
- Ellsworth, W. L. (2013). Injection-induced earthquakes. *Science*, 341(6142), 1225942.
- Kijko, A. (1977a). An algorithm for the optimum distribution of a regional seismic network—I. Pure and applied geophysics, 115, 999-1009.
- Kijko, A. (1977b). An algorithm for the optimum distribution of a regional seismic network—II. An analysis of the accuracy of location of local earthquakes depending on the number of seismic stations. *Pure and Applied Geophysics*, 115(4), 1011-1021.
- Krishnakumar, K. (1990). Micro-genetic algorithms for stationary and non-stationary function optimization. In *Intelligent control and adaptive systems* (Vol. 1196, pp. 289-296). SPIE.
- Mignan, A., Werner, M.J., Wiemer, S., Chen, C.-C., Wu, Y.-M. (2011). Bayesian Estimation of the Spatially Varying Completeness Magnitude of Earthquake Catalogs. *Bull. Seismol. Soc. Am.* 101, 1371-1385.
- Mignan, A. (2012). Functional shape of the earthquake frequency-magnitude distribution and completeness magnitude. *J. Geophys. Res.* 117, B08302.
- Oth, A., Böse, M., Wenzel, F., Köhler, N., & Erdik, M. (2010). Evaluation and optimization of seismic networks and algorithms for earthquake early warning—the case of Istanbul (Turkey). *Journal of Geophysical Research: Solid Earth*, 115(B10).
- Rabinowitz, N., & Steinberg, D. M. (1990). Optimal configuration of a seismographic network: a statistical approach. *Bulletin of the Seismological Society of America*, 80(1), 187-196.
- Stankiewicz, J., Bindi, D., Oth, A., & Parolai, S. (2013). Designing efficient earthquake early warning systems: case study of Almaty, Kazakhstan. *Journal of seismology*, 17, 1125-1137.
- Uhrhammer, R. A. (1980). Analysis of small seismographic station networks. *Bulletin of the Seismological Society of America*, 70(4), 1369-1379.
- Wang, Y. K., & Stewart, R. R. (2024). Subsurface Imaging by a Post-Stimulation Walkaway Vertical Seismic Profile Using Distributed Acoustic Sensing at the Utah FORGE Enhanced Geothermal System Site. *Energies*, 17(13), 3119.
- Zollo, A. L. D. O., Iannaccone, G. I. O. V. A. N. N. I., Lancieri, M., Cantore, L., Convertito, V., Emolo, A., ... & Gasparini, P. (2009). Earthquake early warning system in southern Italy: Methodologies and performance evaluation. *Geophysical research letters*, 36(5).