**Supplement 1. Minimization method**

The computational direct minimization algorithm used in this work is slightly adapted version of the method presented in Annex B of Kovalets et al. (2018). There are some variations in the algorithm, used in this work to account for differences in statement of the problem. The first loop over index  spans all measurements; the second loop over index  spans all possible source locations (grid nodes); the third loop over index **  spans possible the start times of the release; the last loop over index **  spans all release durations. The arrays *A, B, C* store values of correspondently, for each source of unit intensity, located in grid node *i*, started at start time  and having duration . Therefore arrays *A, B, C* have size . Those arrays are then used for evaluation of the cost function (1).

The arrays *A, B, C* are filled following the computational sequence presented in Fig. S.1 and then values of cost function  are evaluated for all values of indices  using obvious relationships:

 (S.1)

The values of indices  which yield minimum of cost function *J* are then easily obtained and thus the most probable release location, start time and duration are defined.

Note, that, as shown in Fig. S.1, as intermediate result we store minimum values of cost function  , minimized with respect to release start time and duration at all grid points  The spatial distribution of this cost function is plotted in Fig. 2 of the paper.

 **Figure S.1.** Pseudo-code of the direct minimization algorithm. Variables are defined in text of the paper.