A set of R packages to compute population count from mobile phone data

- Bogdan OANCEA, Statistics Romania and University of Bucharest, Romania
- David SALGADO, Statistics Spain (INE),
- Luis SANGUIAO-SANDE, Statistics Spain (INE),
- Sandra BARRAGAN, Statistics Spain (INE)

The context

- The work was done within the ESSnet Big Data II project – Work Package I Mobile Phone Data
- The full details about the methodological framework beyond the R packages are presented on the website of the project:

https://webgate.ec.europa.eu/fpfis/mwikis/essnetbigdata/index.php/WPI_Milestones_and_deliverables

- No access to mobile phone data during the project
- All our R packages were tested using simulated data
  - The simulation software used is available here: https://github.com/MobilePhoneESSnetBigData/simulator

The R packages

- A set of R packages to compute population count starting from mobile phone data (network events data):
  - *destim* – computes the location probabilities for mobile devices;
  - *deduplication* – solves the device multiplicity issue;
  - *aggregation* – computes the probability distribution for the number of individuals;
  - *inference* – combines auxiliary information with number of individuals to compute the probability distribution for total number individuals and OD matrices;

- All these software tools are freely available:
- https://github.com/MobilePhoneESSnetBigData

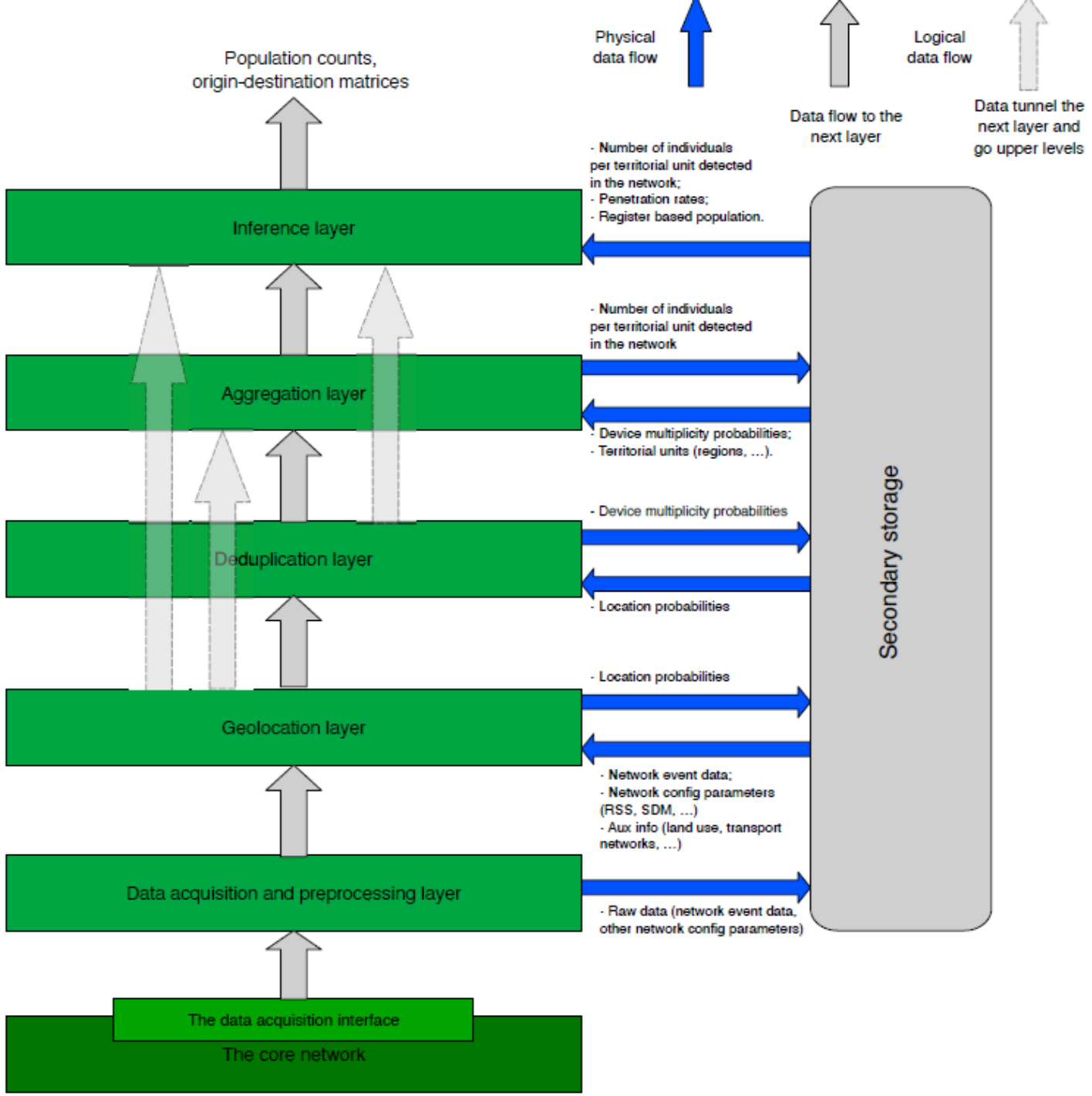
The R packages

- They closely follow the **architecture of the methodological framework** (https://webgate.ec.europa.eu/fpfis/mwikis/essnetbigdata/images/f/fb/WPI_Deliverable_I3_A_proposed_production_framework_with_mobile_network_data_2020_05_31_draft.pdf)

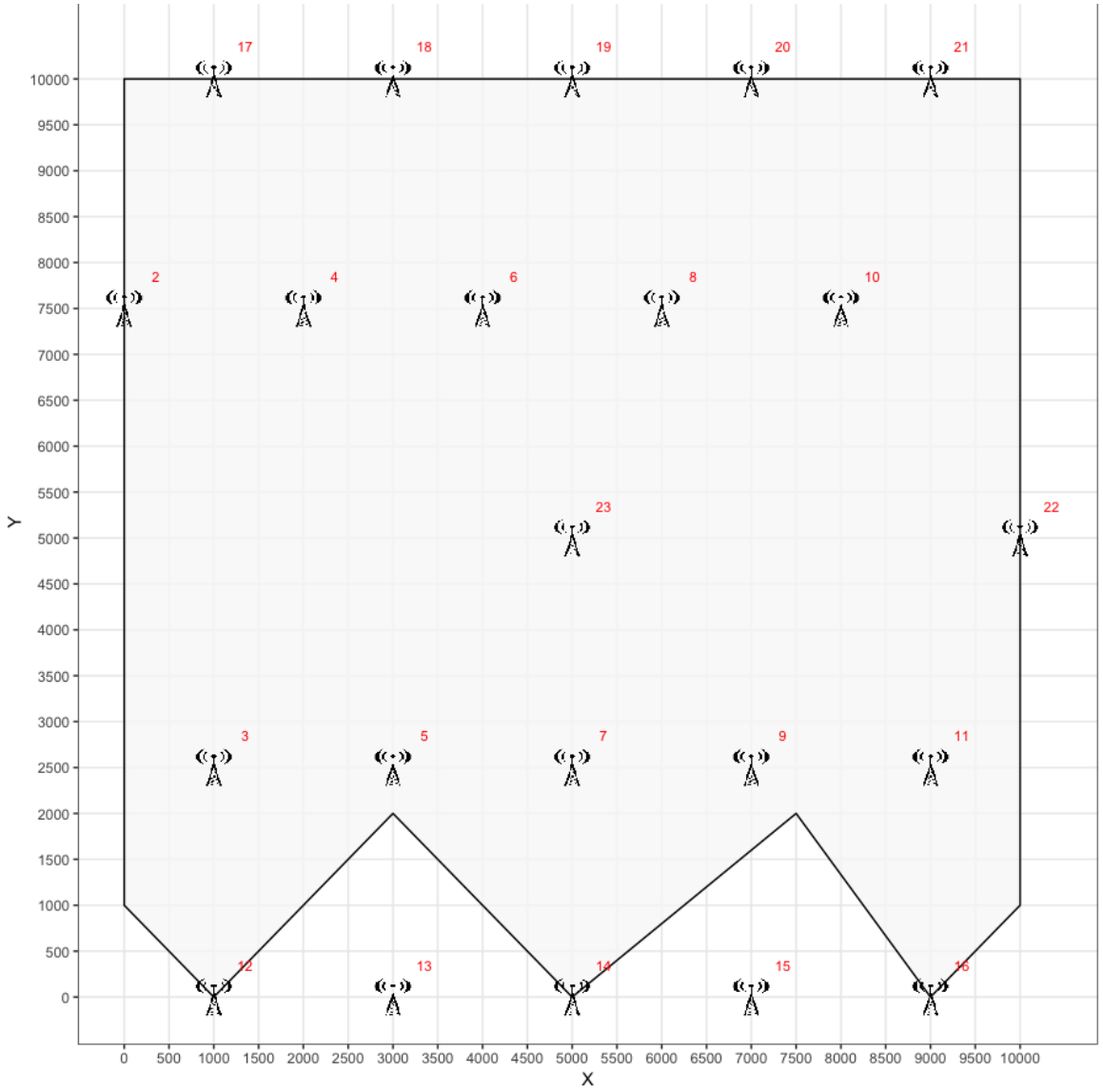
- Development **principles**:
  - **modularity** - the division of the production process into separate modules
  - **abstraction** - interaction through interfaces making the internals of each module independent of the rest of modules

- All packages have **vignettes**, **reference manuals** and **end-to-end examples**
- All packages are **computationally intensive:**
  - They use **parallel computing** techniques
  - One of them use C++ code to improve the execution speed

The layered structure of the packages



- map + network + individuals + mobile devices
- rectangular grid of tiles
- we start from computing the location probability of each device for each tile and time instant and end to population counts



destim

- **destim** (https://github.com/MobilePhoneESSnetBigData/destim)
  - It exploits the network events data and derive **localization probability** for each device at the level of geographical units using a **Hidden Markov Model** (HMM);
  - ***Inputs***:
    - Network event data $\boldsymbol{E} = \{\boldsymbol{E_{dt}}\}_{\boldsymbol{t=0,1,\ldots,T}}^{\boldsymbol{d=1,\ldots,D}}$
    - Network configuration parameters such as emission power P, path loss exponent γ, BTS orientation,...
    - Signal strength (needed to compute the emission probabilities for HMM);
    - Some general parameters: grid/tile size, the sequence of time instants;
  - ***Outputs*** *(as csv files)*:
    - Location probabilities for each device, each tile and time instant;
    - Joint location probabilities for each device, each tile and consecutive time instants.

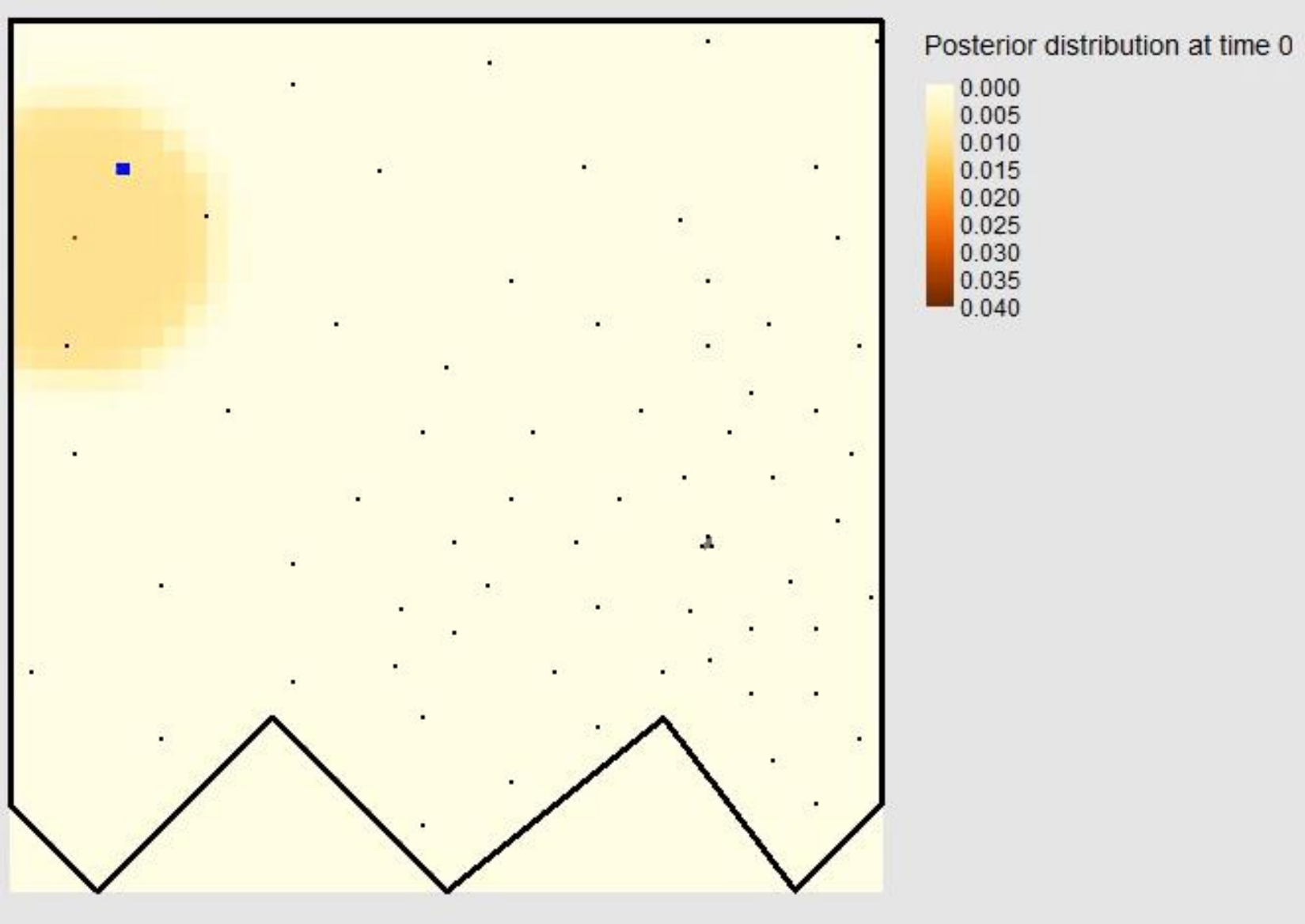
destim

location probabilities

```
tile, time, probL
257,0,3.68545268791995e-16
258,0,2.13778597819168e-13
259,0,2.23977953132008e-11
260,0,1.12347736274898e-08
261,0,7.25571332062322e-11
```

joint location probabilities

```
time_from, time_to, tile_from, tile_to, probL
 880,890,1361,1400,0.00144324860111083
 880,890,1361,1401,3.98470051712904e-05
 880,890,1362,1401,0.0031705523664934 3
 880,890,1361,1402,0.00144324860111083
 880,890,1362,1402,8.75365591529227e-05
```


Posterior distribution at time 0
0.000
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0.040

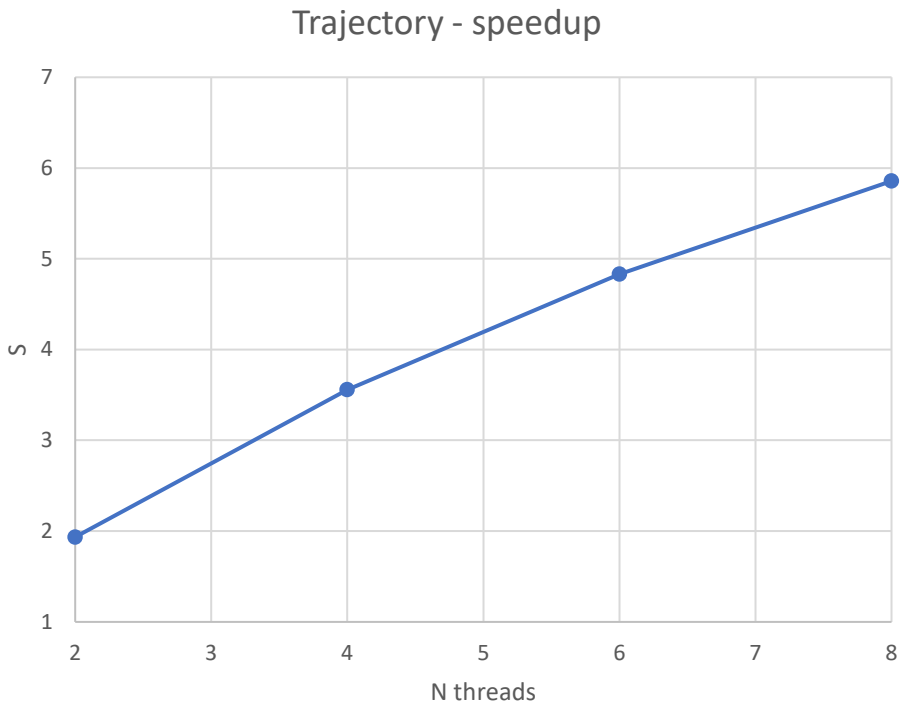
deduplication

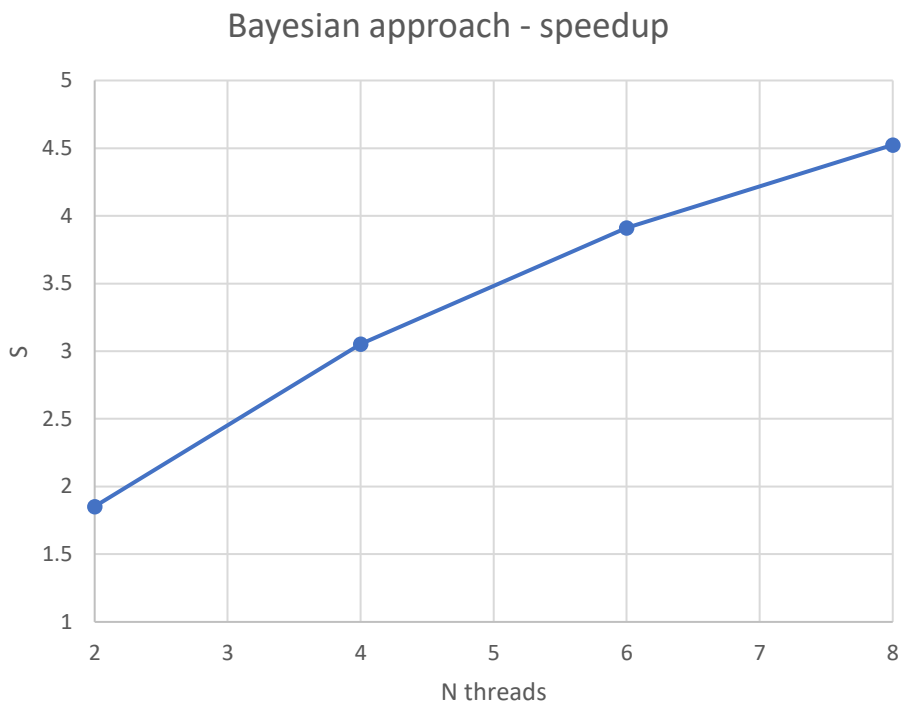
- **deduplication** (https://github.com/MobilePhoneESSnetBigData/deduplication)
  - It **classifies the devices** as being in 1:1 or 2:1 correspondence with its owner;
  - It uses a **probabilistic** classification assigning each device a **probability** to belong to one of the two classes;
  - Two approaches:
    - A **Bayesian approach** based on network events (with 2 variants);
    - A **trajectory similarity approach.**
  - ***Inputs***:
    - Network event data $\boldsymbol{E} = \{\boldsymbol{E_{dt}}\}_{\boldsymbol{t=0,1,\ldots,T}}^{\boldsymbol{d=1,\ldots,D}}$;
    - Location probabilities (coming from destim package);
    - Network configuration parameters such as emission power P, path loss exponent γ, BTS orientation, cell boundaries, etc.
  - ***Outputs***:
    - A csv file with the duplicity probability for each device;

deduplication

```
# set the folder where the necessary input files are stored
path_root      <- 'extdata'

# set the grid file name, i.e. the file where the grid parameters are found
gridfile <-system.file(path_root, 'grid.csv', package = 'deduplication')

# set the events file name, i.e. the file with network events registered
eventsfile<-system.file(path_root, 'AntennaInfo_MNO_MNO1.csv', package = 'deduplication')

# set the signal strength file name,
signalfile<-system.file(path_root, 'SignalMeasure_MNO1.csv', package = 'deduplication')

# set the antenna cells file name,
antennacellsfile<-system.file(path_root, 'AntennaCells_MNO1.csv', package = 'deduplication')

# set the simulation file name, i.e. the file with the simulation parameters used to produce
the data set
simulationfile<-system.file(path_root, 'simulation.xml', package = 'deduplication')

# compute the duplicity probabilities using the "pairs" method
out1<-computeDuplicity("pairs", gridFileName = gridfile, eventsFileName = eventsfile,
signalFileName = signalfile, antennaCellsFileName = antennacellsfile, simulationFileName =
simulationfile)

# compute the duplicity probabilities using the "trajectory" method
prefix <- 'postLocDevice'
out2<-computeDuplicity("trajectory", gridFileName = gridfile, eventsFileName = eventsfile,
signalFileName = signalfile, antennaCellsFileName = antennacellsfile, simulationFileName =
simulationfile, path= system.file(path_root, package='deduplication'), prefix = prefix)
```

deduplication





aggregation

- **aggregation** (https://github.com/MobilePhoneESSnetBigData/aggregation)
- It provides the **probability distribution of the number of individuals**
- We define the vectors $e_i^{(1)} = e_i$ and $e_i^{(2)} = \frac{1}{2} e_i$, where e_i is the canonical unit vector in R^{NT} (with NT the number of tiles in the reference grid).
- We define the random variable $T_{dt} \in \{e_i^{(1)}, e_i^{(2)}\}_{i=1,\dots NT}$ with probability mass function given by:

$$\begin{cases} P\left(T_{dt} = e_i^{(1)} \middle| E_{1:D}\right) = \gamma_{dti}(1 - p_d) \\ P\left(T_{dt} = e_i^{(2)} \middle| E_{1:D}\right) = \gamma_{dti} p_d \end{cases}$$

- It can be easily observed that this is a this is a categorical or multinoulli random variable.
- Finally, we define the multivariate random variable N_t^{net} providing the number of individuals detected by the network at each tile and time instant t: $N_t^{net} = \sum_{d=1}^{D} T_{dt.}$
- This is a **Poisson multinomial** distributed random variable, and we used a Monte Carlo simulation method to generate random variates from this distribution

aggregation

- ***Inputs***:
  - Duplicity probabilities;
  - General parameters: grid/tile size, geographical regions, the sequence of time instants;
- ***Outputs***:
  - Two csv files - random variates to compute a point estimation of the number of individuals in a region and the number individuals moving from one region to another;

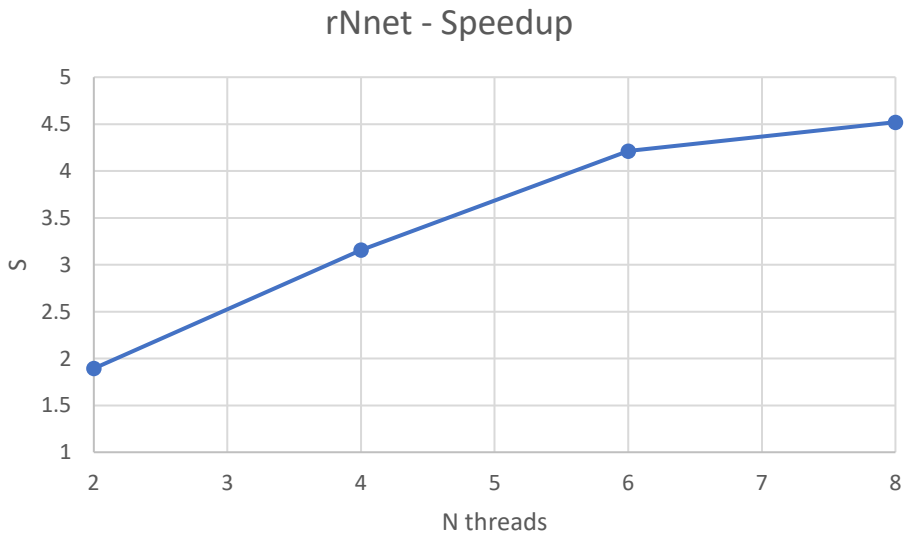
aggregation

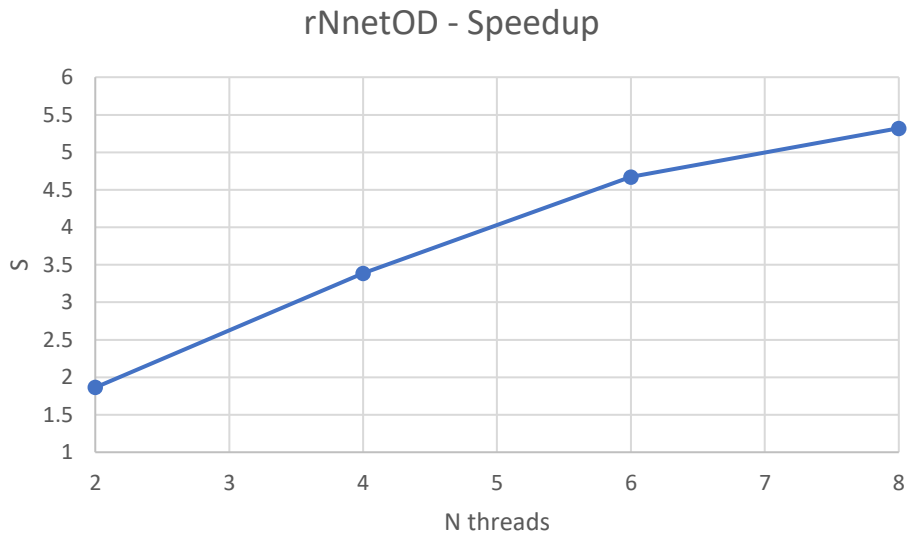
```
# set the folder where the necessary input files are stored
path      <- 'extdata'
prefix = 'postLocDevice’
prefix = 'postLocJointProbDevice'
# set the duplicity probabilities file name, i.e. the file with duplicity probability for each device
dpFile <- system.file(path, 'duplicity.csv', package = 'aggregation')
# set the regions file name, i.e. the file defining the regions for which we need the estimation of
the number
# of individuals detected by network.
rgFile <- system.file(path, 'regions.csv', package = 'aggregation')
# set the path to the posterior location probabilities file
pathLoc <- system.file(path, package = 'aggregation')
# set the number of random values to be generated
n <- 1e3
# call rNnetEvent
nNet <- rNnetEvent(n, dpFile, rgFile, pathLoc, prefix)
nnetOD <- rNnetEventOD(n, dpFile, rgFile, system.file(path, package = 'aggregation'), prefixJ)
```

aggregation

time	region	N	iter
1	1	11	1
1	1	11	2
1	1	10	3
1	1	13	4
1	1	10	5

time_from	time_to	region_from	region_to	Nnet	iter
0	10	1	1	18.0	1
0	10	1	1	18.5	2
0	10	1	1	19.0	3
0	10	1	1	18.0	4
0	10	1	1	19.0	5
0	10	1	1	18.0	6





inference

- **inference** (https://github.com/MobilePhoneESSnetBigData/inference)
- Combines **auxiliary information** (population registers, MNO penetration rates) with the **detected** number of individuals to provide the **distribution of the target population count**;
- The observed data, the target process, and its underlying parameters must be given as a joint probability distribution P(data, process, parameters):

P(data, process, parameters) = P(data | process, parameters) P(process | parameters) P(parameters)

  - an observation process (data given the underlying dynamical process driving the target variable N);
  - a state process (the underlying process for the target variable N modelled in terms of its parameters);
  - assumptions about the parameters driving not only the state process but possibly also the observation process;
- 3 approaches:
  - negative binomial
  - displaced beta negative binomial
  - state process - Poisson limit of the negative binomial distributions

Inference

- ***Inputs***:
    - **Posterior location probabilities** for each device (from *destim*);
    - **Duplicity probabilities** for each device (from *deduplication*);
    - The **random variates** provided by *aggregation*;
    - **Auxiliary information**:
        - register-based population counts for each territorial unit;
        - MNO penetration rates for each territorial unit.
- ***Outputs***:
    - The probability distribution of the target population count (as random variates) at each time instant;
    - The Origin – Destination matrices (again as random variates);

inference

- The results looks like:

```
 region Mean Mode Median Min Max Q1 Q3 IQR    SD      CV    CI_LOW CI_HIGH
[1,]      1   43     41     39  13 136 31 50  19 17.74 41.55  22.00   77.00
[2,]      2   60     57     58  24 142 49 69  20 16.19 26.98  38.00   87.53
[3,]      3   80     68     77  42 176 66 92  25 19.32 24.07  54.97  113.50
[4,]      4   38     33     37  14 102 30 44  14 11.41 29.86  23.50   58.00
[5,]      5   77     64     73  26 182 62 89  28 23.19 30.02  47.97  120.06
[6,]      6   49     36     46  14 140 36 58  22 18.41 37.59  26.00   82.03

region    N NPop
     1 11.0 46.0
     1  9.0 42.0
     1 13.0 52.0
     1 12.0 33.0
     1 12.0 35.0
     1 12.5 30.5
```

inference

- The *inference* package exposes its main functions as an http REST API;
- This feature is implemented using the *plumber* package.