



## **Do czego służy pakiet epinet?**

Za jego pomocą można robić symulacje oraz analizy rozprzestrzeniania się epidemii w populacji.

















## Parametry epidemii $\beta$ , $\theta_E$ , $\theta_I$ , $k_E$ oraz $k_I$

- Czas oczekiwania na przekazanie zakażenia jest modelowany przy pomocy zmiennej losowej o rozkładzie wykładniczym ze średnią  $1/\beta$ ,
- Czas, gdy jednostka znalazła się w stanie zainfekowania bez możliwości zarażania i z możliwością przekazywania patogenu jest modelowany przez zmienne losowe o rozkładzie gamma z parametrami, odpowiednio,  $(\theta_E, k_E)$  oraz  $(\theta_I, k_I)$ , gdzie średnia dla tak zadanego rozkładu gamma jest postaci  $k\theta$ , a wariancja  $k\theta^2$ .

# Rozkłady a priori dla parametrów modelu

- Parametrom  $\eta$  regulującym prawdopodobieństwa istnienia kontaktu między osobnikami krawędzi przypisany jest rozkład normalny.
- Rozkład a priori dla parametrów epidemii jest wybierany tak, żeby uwzględniał każdą ważną biologiczną informację posiadaną na temat patogenu (np. parametry epidemii  $\theta_E$  oraz  $\theta_I$  mogą być opisane odwróconym rozkładem gamma, wtedy reszta parametrów epidemii opisana jest rozkładem gamma).

# Symulacja epidemii

Pakiet `epinet` może być użyty do symulacji rozprzestrzeniania się epidemii w populacji.

- 1 Budujemy macierz danych,
- 2 Tworzymy macierz zmiennych (`BuildX()`) dla par osobników,
- 3 Tworzymy sieć interakcji/kontaktów między jednostkami, (`SimulateDyadicLinearERGM()`),
- 4 Tworzymy symulację epidemii (`SEIR.simulation()`).

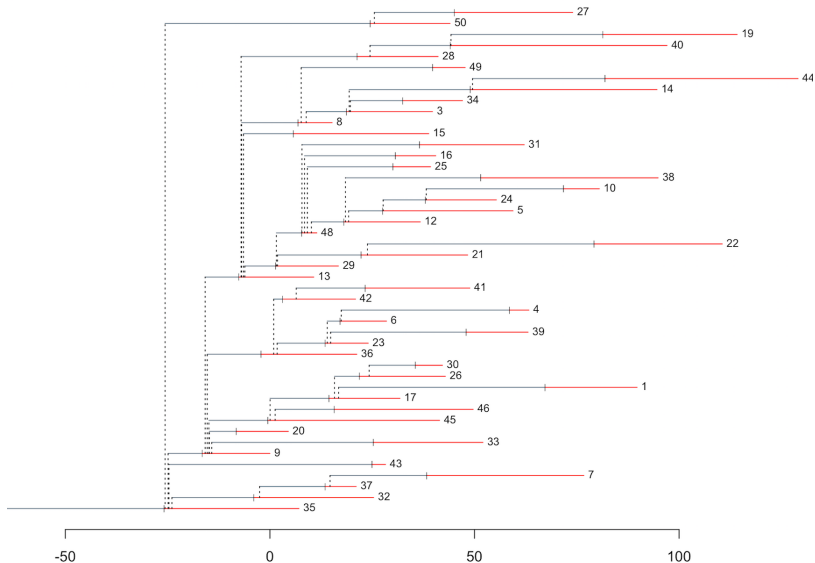
Przyjrzyjmy się ostatniej funkcji:

```
SEIR.simulator(M, N, beta, ki, thetai, ke = ki, thetai = thetai,
               latencydist = "fixed", latencyperiod = 0)
```

- M - macierz kontaktów (obiekt wygenerowany przez funkcję `SimulateDyadicLinearERGM()`),
- beta, thetai, ke, thetai, ki odnoszą się do parametrów epidemii:  $\beta, \theta_E, k_E, \theta_I, k_I$ ,
- latencydist="fixed" (latencydist="gamma") odnosi się do czasu spędzonego w stanie "exposed" dla każdego osobnika i może być modelowany jako stały lub przy pomocy rozkładu gamma .

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ≡ ↺ 🔍 ↻

# Transmission Tree



# Funkcje

Function	Input(s)	Output
<code>BuildX</code>	Types of covariate(s) desired, e.g., matching, distance, etc.	A matrix of dyadic covariates of the form required by <code>SEIR.simulator()</code> and <code>epinet()</code> .
<code>SimulateDyadicLinearERGM</code>	Network ( $\eta$ ) parameters for desired network.	A simulated ERGM network in edge list matrix format.
<code>plot.epidemic</code>	An epidemic in the form produced by <code>SEIR.simulator()</code> .	A graph visually depicting the epidemic.
<code>plot.epinet</code>	MCMC output in the form produced by <code>epinet()</code> .	A graph visually depicting the state of the epidemic at a particular point in the MCMC chain.
<code>epi2newick</code>	An epidemic in the form produced by <code>SEIR.simulator()</code> .	The transmission tree corresponding to the epidemic as a string in Newick format.
<code>epi2newickmcmc</code>	MCMC output in the form produced by <code>epinet()</code> .	The transmission tree corresponding to an inferred epidemic as a string in Newick format.
<code>write.epinet</code>	MCMC output in the form produced by <code>epinet()</code> .	Written output files consisting of the posterior samples of the epidemic parameters and transmission trees.
<code>ess</code>	MCMC output for a single variable, i.e., a sample from the posterior distribution of a variable.	An estimate of the ESS (effective sample size) of the sample.

Table 1: Description of some of the functions in package **epinet**.



# Przykład 1

**Dane:** Mamy populację składającą się z 50 osób o identyfikatorach od 1 do 50.

Będą rozważane dwie zmienne - stała oraz odległość euklidesowa między każdą parą osób. Dla każdej osoby przypisujemy losowo dwuwymiarowe dane przestrzenne.

```
library("epinet")
set.seed(1)
N <- 50
mycov <- data.frame(id = 1:N, xpos = runif(N), ypos = runif(N))
dyadCov <- BuildX(mycov, binaryCol = list(c(2, 3)), binaryFunc = "euclidean")
```

Pierwsze wiersze wynikowe prezentują się w następujący sposób:

	node.1	node.2	(Intercept)	xpos.ypos.L2Dist
[1,]	1	2	1	0.3981306
[2,]	1	3	1	0.3098754
[3,]	1	4	1	0.6835703
[4,]	1	5	1	0.4119156
[5,]	1	6	1	0.7372506
[6,]	1	7	1	0.6980691

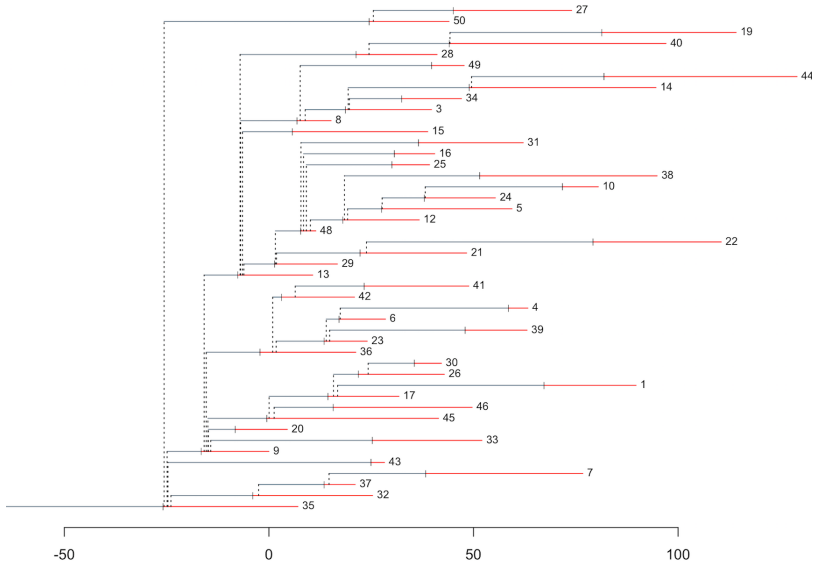
Mając powyższą macierz par węzłów i zmiennych możemy zasymulować sieć interakcji w populacji. Do tego posłuży nam funkcja `SimulateDyadicLinearERGM()`.

```
eta <- c(0, -7)
net <- SimulateDyadicLinearERGM(N = N, dyadiccovmat = dyadCov, eta = eta)
```

	node.1	node.2
[1,]	1	17
[2,]	1	25
[3,]	1	33
[4,]	1	34
[5,]	2	11
[6,]	3	8
[7,]	3	14
[8,]	3	28
[9,]	3	33



# Transmission Tree



Zaprezentujemy procedurę użycia funkcji `epinet()`, w celu przedstawienia wnioskowania na temat parametrów modelu, gdy ich nie znamy.

```
epinet(formula, epidata, dyadiccovmat, mcmcinput = MCMCcontrol()  
       priors = priorcontrol(), verbose = TRUE)
```

Kod w skrypcie.

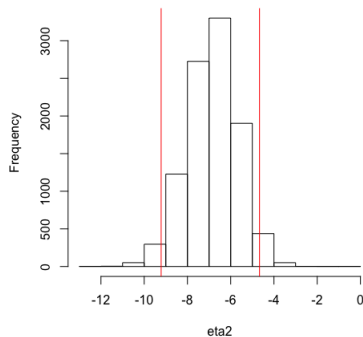
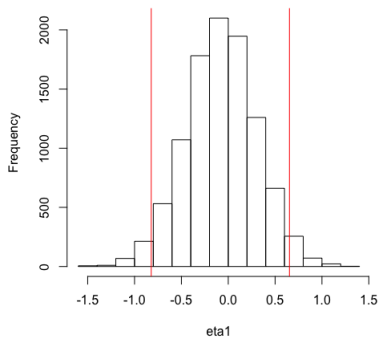


<b>Epidemic parameter</b>	<b>Mean</b>	<b>Orginal</b>
Beta	0.7210	1
Theta_e	10.065	7
Theta_i	8.468	7
k_e	2.257	3
k_i	2.736	3

<b>Network parameter</b>	<b>Mean</b>	<b>Orginal</b>
Intercept	-0.07679	0
xpos.ypos.L2Dist	-6.835	-7



Możemy zobaczyć, czy algorytm był w stanie parametry sieci  $\eta$ .



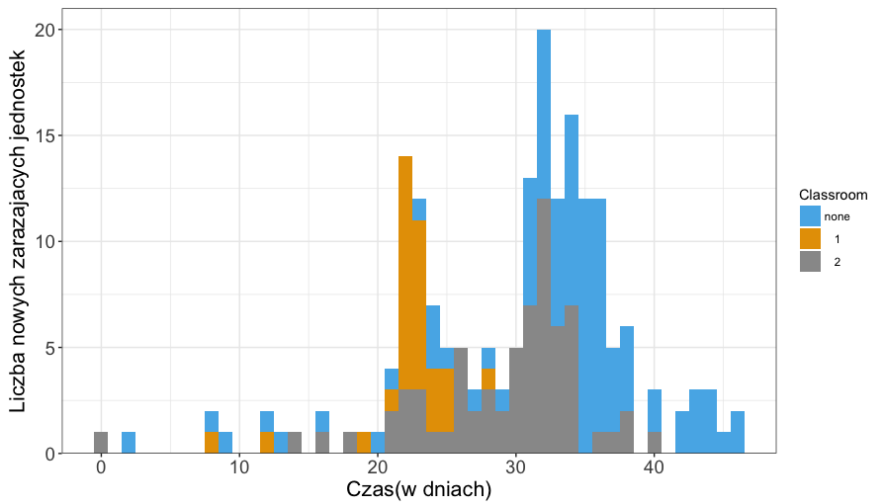
## Przykład 2: Analiza danych dotyczących szerzenia się odry, Hegelloch, Niemcy, zima 1861

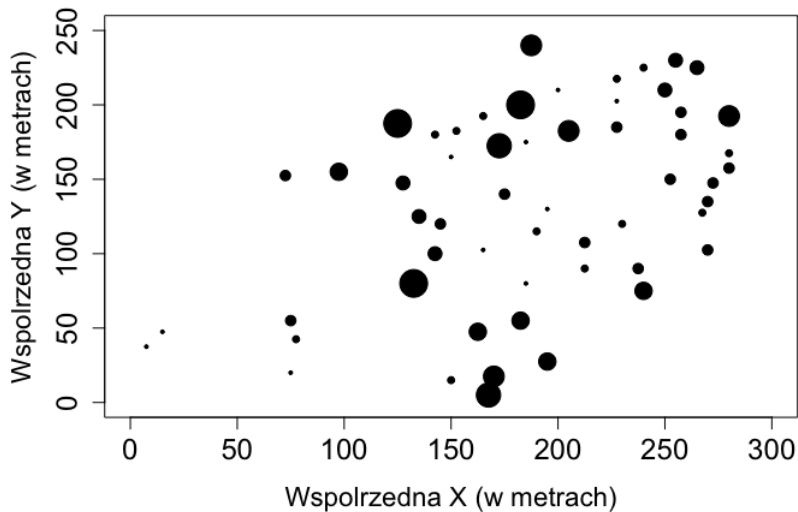
Dane składają się z:

- Macierzy `HagellochTimes`, która zawiera informacje o jednostkach w populacji.

	NodeID	Parent	Etime	Itime	Rtime
[1,]	1	179	NA	51	59
[2,]	2	5	NA	53	61
[3,]	3	73	NA	58	66
[4,]	4	23	NA	57	62
[5,]	5	10	NA	52	61
[6,]	6	21	NA	56	63







```
mcmcinput <- MCMCcontrol(nsamp = 20000000, thinning = 1000, extrathinning = 10,  
                        burnin = 1000000, seed = 1, etapropsd = c(rep(0.05, times = 3), 0.005))  
priors <- priorcontrol(etaprior = c(0, 3, 0, 3, 0, 3, 0, 3/40),  
                      bprior = c(0, 4), tiprior = c(0.25, 0.75),  
                      teprior = c(0.25, 1), keprior = c(8, 20),  
                      kiprior = c(15, 25), priordists = "uniform")  
out <- epinet(~ `Classroom 1` + `Classroom 2` + `House Distance`,  
             epidata = HagellochTimes, dyadiccovmat = HagellochDyadCov,  
             mcmcinput = mcmcinput, priors = priors)
```



## Podsumowanie

Pakiet **epinet** pozwala nam na podstawie danych na temat epidemii otrzymać jej parametry struktury tej epidemii, a w szczególności prawdopodobieństwo, że pomiędzy dwiema jednostkami istniał kontakt.

### Ograniczenia:

- założenie niezależności w istnieniu krawędzi między jednostkami,
- zdobycie dokładnych danych na temat przechodzenia przez różne stadia epidemii jest często niemożliwe, dlatego wymaga dodatkowych założeń,
- wielkość epidemii - czas procedury MCMC zachowuje się jak kwadrat wielkości populacji,
- MCMC wymaga dużej ilości iteracji.