

Taudin leviämissimulaattori

1 Johdanto

Tautien leviämisen simulointi on nykymaailmassa erittäin ajankohtaista: Covid-19 on muuttanut maailmaamme merkittävästi jo useamman vuoden ajan. Tautien leviämissimulaatioiden ohjelmointi ja niillä tehtävät mallinnukset ovat merkittävä osa kulkutautien ennustamista ja torjumista. Leviämissimulaatioilla on muun muassa mahdollista tutkia rokotteiden ja immunitetin vaikutusta siihen, miten eri tekijät vaikuttavat taudin tarttuvuuteen.

Tämä raportti käsittelee ensiksi taudin leviämissimulaattorin eri menetöt lyhyesti, minkä jälkeen raportti esittelee metodien implementointia. Tämän jälkeen raportti esittelee taudin leviämissimulaattorilla saatuja tuloksia, minkä jälkeen siirrytään raportin yhteenvetoon.

2 Metodit

Ohjelman toiminta on varsin yksinkertainen, sillä tämän ohjelman tavoitteena ei ole toimia vaativana ja realistisena simulaationa, vaan harjoituksena.

Ohjelma generoi satunnaisiin koordinaatteihin $N \times N$ -kokoisessa kaksiulotteisessa maailmassa ihmisiä (mitä ohjelma kutsuu termillä *person*). Ihmisistä osa on simulaation käynnistyshetkellä sairaita, osa on valmiiksi immuuneita, ja muut ovat neutraaleita.

Simulaation edetessä neutraaleilla ihmisillä on jokin todennäköisyys, joilla he saattavat saada tartunnan, mikäli heidän kanssa samassa koordinaatissa on vähintään yksi sairas ihminen. Todennäköisyys saadaan simulaatiota ennen määritellyistä parametreistä, kuten kaikki muutkin simulaation käyttämät arvot.

Simulaatiota jatketaan jokin määrä aika-askelia eteenpäin ja simulaation tuloksista jokaisesta aika-askeleesta tehdään merkintä output.csv -tiedostoon, mistä voidaan piirtää tilastoja. CSV-tiedosto on valittu tiedostomuodoksi sen takia, että kyseisestä tiedostomuodosta on helppo tehdä kuvaajia esimerkiksi Excelillä, kuten Tulokset-osiossa on näytetty. Aika-askeleella tämän ohjelman kontekstissa tarkoitetaan yhtä silmukan iteraatiota, missä ihmiset liikkuvat, paranevat ja tartuttavat tautia eteenpäin. Tulokset-osiossa ollaan käytetty aina aika-askelien määränä sataa, sillä tämän nähtiin testeissä olevan hyvä määrä nähdä leviämistä.

3 Implementointi

Ohjelma on jaettu kolmeen eri tiedostoon: disease.f90, population.f90 ja utils.f90 ohjelman rakenteen selkeyttämiseksi.

3.1 disease.f90

Disease.f90 -tiedosto toimii ohjelman päätiedostona, josta löytyy simulaation kannalta tärkeimmät ominaisuudet. Kyseinen tiedosto hyödyntää population- ja utils-tiedostoja sen alussa, joista etenkin population-tiedoston tuoma person-tyyppi on olennainen.

Person-tyyppi on ohjelman olennaisin tietorakenne, sillä jokainen person sisältää neljä muuttujaa: ID, x-koordinaatti, y-koordinaatti ja terveyden. Terveys-muuttuja on jotakin 0-2 väliltä, jossa:

0 = neutraali, eli ei vielä toistaiseksi sairastunut tai immuuni.

1 = sairastunut, odottaa parantumista.

2 = immuuni, eli jo aikaisemmin taudin sairastanut, ei voi enää sairastua.

Disease.f90 -tiedosto alustaa ohjelman parametrit (maailman koko, maksimi aika-askeleet, populaation määrä, sairaiden määrä alussa, immuunien määrä alussa, parantumisen todennäköisyys per aika-askele, sairastumisen todennäköisyys per aika-askele). Nämä parametrit ladataan tiedoston alussa olevista kovakoodatuista riveistä 17-30. Ohjelma on myös mahdollista suorittaa antamalla sille suorituksessa seitsemän komentoriviargumenttia, milloin ohjelma suoritetaan niiden perusteella. Ohjeet tähän löytyy /src ja /run kansioden README-tiedostoista.

Ohjelma kertoo käynnistymisestään, lataamista parametreistaan ja tämän jälkeen alustaa popArray-tietorakenteen, joka koostuu person-tyypeistä. Tämän jälkeen ohjelma alkaa suorittamaan pääsilmuttaansa i...maksimi aika-askeleet -välillä, joissa jokaisessa silmukan iteraatiossa liikutetaan person-tyyppisiä satunnaisesti koordinaatistossa ja tehdään tulostuksia sekä konsoliin että output.csv -tiedostoon.

Uusia ihmisiä lisätessä (addNewPerson -alarutiini) ohjelma myös tarkistaa, kuinka mones lisätty ihminen on. Tätä käytetään hyödyksi jo valmiiksi sairaiden ja immuunien luonnissa: ensimmäisenä luoduista ihmisistä tehdään sairaita siihen asti kun ohjelman parametreissa on asetettu (eli vaikka ensimmäiset viisi, mikäli halutaan viisi sairasta), kun taas viimeisenä olevista ihmisistä tehdään immuuneita, mikäli näin on parametreissa asetettu.

Disease.f90-tiedoston moveAllPops -alarutiini kutsuu population.f90 -aputiedostossa olevia person-tyyppiin liittyviä apufunktioita, kuten sairauden levittämistä, ihmisten liikuttamista ja heidän parantamistaan.

3.2 population.f90

Population-tiedosto sisältää kolme person-tyypin manipulointiin tarkoitettua funktiota, jotka ovat eriytetty päätiedostosta helpomman hallittavuuden vuoksi. Funktiot ovat healPerson, movePerson ja spreadDisease.

healPerson-funktio palauttaa arvotaan person-tyypin, millä voidaan korvata popArray:ssa aikaisemmin ollut person-tyyppi. Funktio arpoo luvun 0-100 väliltä ja tarkistaa kokonaislukuna esitettyyn prosenttiin verrattuna, parantuuko ihminen.

movePerson-funktio toimii hyvin samalla tavalla healPerson-funktioon verrattuna, mutta arpoo kokonaisluvun, mitä se käyttää liikkumissuuntansa arvontaan.

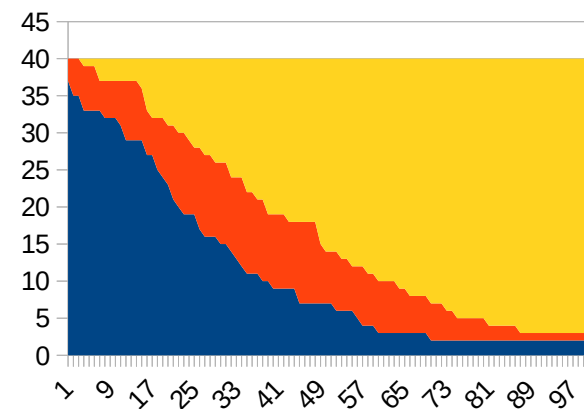
spreadDisease-funktio puolestaan käyttää jälleen samaa healPersonin tyylistä arvontaa 0-100 väliltä, mutta tekee myös tarkistuksia käymällä kaikki muut popArray:n person-tyypit läpi, että onko samassa koordinaatissa muita ihmisiä. Mikäli muita ihmisiä on samassa koordinaatissa ja joku heistä on sairas, voidaan silloin sairastua, jolloin vielä tarkistetaan arvonnän tulos.

3.3 utils.f90

Utils-tiedosto sisältää tulostamiseen käytetyn printAllPops-funktion, joka tekee konsolitulostuksia. Tämän lisäksi löytyy getArg-funktio, joka lataa komentoriviparametrin muuttuun. Luokassa oli aikaisemmin useampia tulostusfunktioita, mutta ohjelman refaktoroinnin jälkeen ne sai yhdistettyä yhdeksi funktioksi. Ohjelman selkeyttämiseksi funktio pidetään eri tiedostossa, sillä funktiolla ei ole ohjelman toiminnan kannalta mitään olennainen.

4 Tulokset

Ohjelman tuloksista saatiin seuraavanlaisia output.csv -tiedostoista muodostettuja kuvaajia eri parametreillä, mitkä ovat lueteltu kuvaajan oikealla puolella:



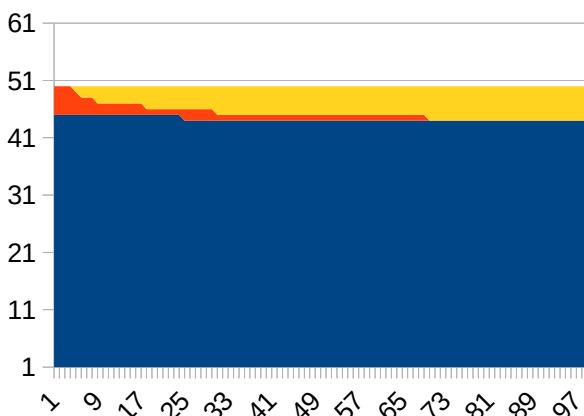
#1

N = 10, Aika-askeleet = 100,

Väkiluku = 40, Alkup. Sairaat = 2, Alkup. Immuunit = 0

Parantumisen % = 5,

Sairastumisen % = 50



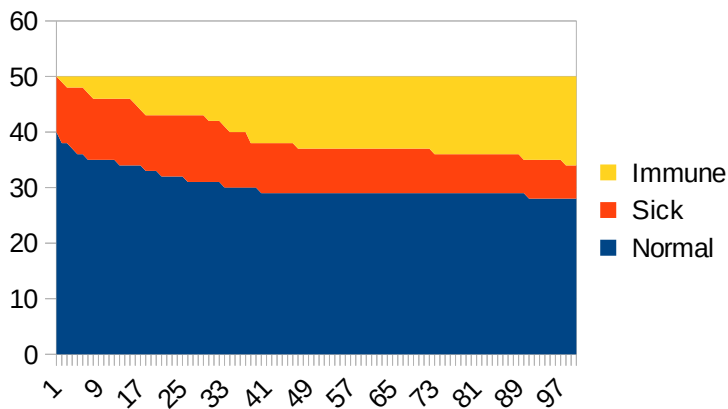
#2

N = 40, Aika-askeleet = 100,

Väkiluku = 50, Alkup. Sairaat = 5, Alkup. Immuunit = 0

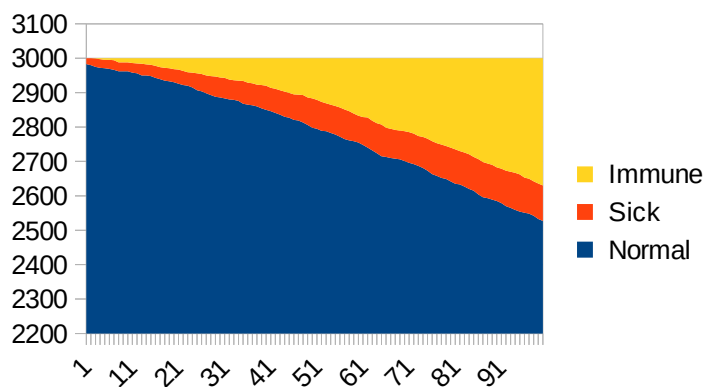
Parantumisen % = 5,

Sairastumisen % = 50



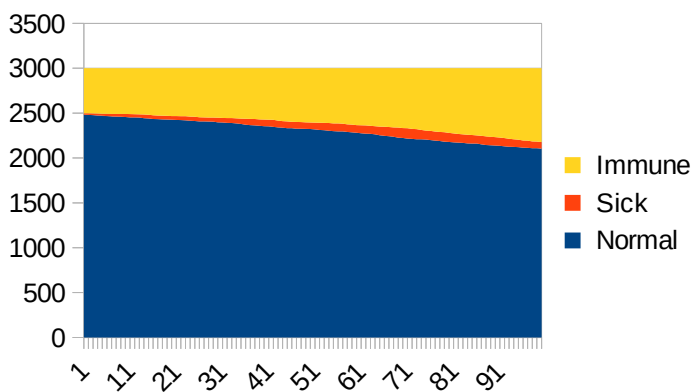
#3

N = 40, Aika-askeleet = 100,
 Väkiluku = 50, Alkup. Sairaat = 10,
 Alkup. Immuunit = 0
 Parantumisen % = 2,
 Sairastumisen % = 80



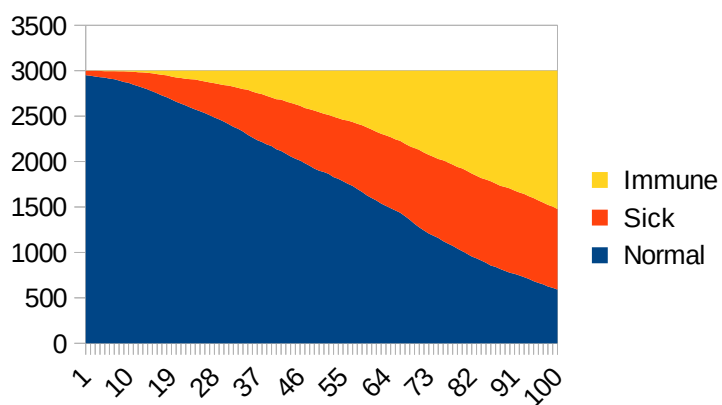
#4

N = 100, Aika-askeleet = 100,
 Väkiluku = 3000, Alkup. Sairaat = 15,
 Alkup. Immuunit = 0
 Parantumisen % = 5,
 Sairastumisen % = 60



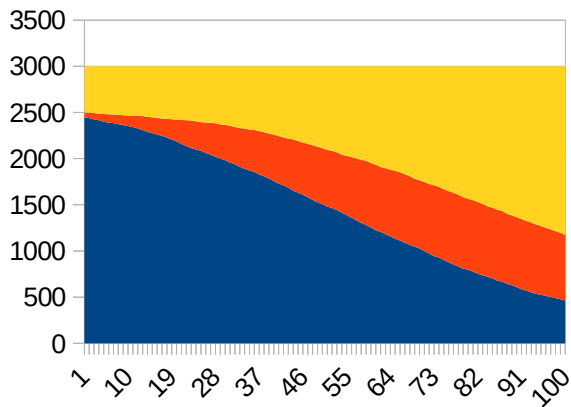
#5

N = 100, Aika-askeleet = 100,
 Väkiluku = 3000, Alkup. Sairaat = 15,
 Alkup. Immuunit = 500
 Parantumisen % = 5,
 Sairastumisen % = 60



#6

N = 100, Aika-askeleet = 100,
 Väkiluku = 3000, Alkup. Sairaat = 40,
 Alkup. Immuunit = 0
 Parantumisen % = 3,
 Sairastumisen % = 90



#7

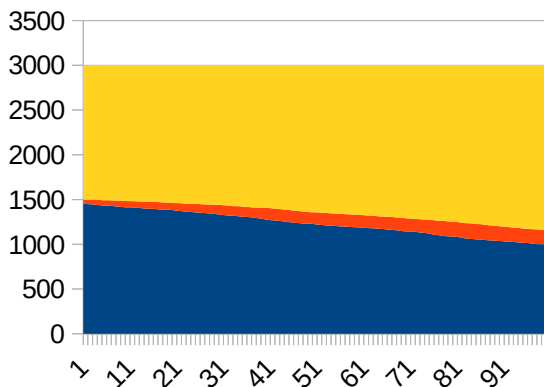
N = 100, Aika-askeleet = 100,

Väkiluku = 3000, Alkup. Sairaat = 40,

Alkup. Immuunit = 500

Parantumisen % = 3,

Sairastumisen % = 90



#8

N = 100, Aika-askeleet = 100,

Väkiluku = 3000, Alkup. Sairaat = 40,

Alkup. Immuunit = 1500

Parantumisen % = 3,

Sairastumisen % = 90

Tulokset näyttävät, että ensimmäisissä kuvaajissa (1-3), joissa on pieni populaatio, saadaan selkeästi epätarkempia tuloksia. Suuremman populaation kuvaajassa #4 huomataan, että suhteellisen tarttuva tauti aiheuttaa tasaista tartuntaa, mikä pystytään hallita jo hyvin 16,6% immunitetillä (jota voidaan ajatella rokotuskattavuutena). Rokotuskattavuuden vaikutus näkyy kuvaajassa #5.

Kuvaajassa #6 puolestaan esitellään huomattavasti tarttuvampi ja hitaammin parantuva tauti. Tämä aiheuttaa huomattavasti nopeamman tarttuvuuden, mihin ei aikaisemmalla 16,6% rokotekattavuudella ole juurikaan merkitystä kuvaajassa #7. Kuvaajan #8 kohdalla kuitenkin huomataan, että 50% rokotekattavuusa tekee tästä tarttuvammastakin taudista jo lähes merkityksettömän.

4 Yhteenveto ja jatkotutkimusehdotukset

Tehdyissä testeissä havaittiin, että tarkkojen simulaatiotulosten saamiseksi simulaatiosta tulisi tehdän realistisempi. Taudin tappavuutta ei simuloida lainkaan, mikä vaikuttaa suuresti tarttuvuuteen jos taudinkantajat saattavatkin kuolla. Tämän lisäksi rokotettavuutta ei simulaatiossa voitu kasvattaa ajan myötä, vaan väestö simulaatiossa aloittaa suoraan jostakin tietystä rokotekestävyydestä, minkä takia ei voida nähdä millaiset vaikutukset olisi esimerkiksi nopealla rokottamisella hitaaseen verrattuna.

Jatkotutkimusehdotuksena ehdotetaan näiden edellisten ongelmakohtien huomioimista simulaatiota tehtäessä, jotta voidaan saada monipuolisempia ja realistisempia tuloksia.