

AnimalMovementSimulations

7 Feb 2014

S. J. Lycett

Introduction

The AnimalMovementSimulations Java project generates simulated animal movement data for use with Broadwick, particularly for the Broadwick tutorial examples. Since the examples are based upon the implemented Soho model (which uses Broadwick), this project focuses on simulating cattle movements and CTS data, although different livestock scenarios might be added later.

Simulation Scenario

A number of farms have a variable number of animals of fixed lifespan. At each time step there are births, movements and deaths (due to old age); which are recorded in the output files. The main loop of the simulation is:

1. Remove deceased animals from each farm
2. If it is the birth season, then for each farm allow a random number of births (uniform between limits)
3. If it is the moving season, then choose how many movements to make (uniform between limits)
4. For each movement to be made, randomly choose two different farms and number of animals to move (uniform between limits). Move as many of these animals as possible; so if there were supposed to be 20 going from A to B, but there were only 15 at A, then all 15 of them are moved. The animals are moved according to age, so the oldest ones are moved first. Note it would be possible to have some animals born to farm A and then immediately moved to farm B within the same time step.
5. Increment the date

Additional notes:

- Birth Season - Currently hard coded to allow births in the months of April to September inclusive in the RandomMovementSimulator class.
- Moving Season - Currently hard coded to be all year round in the RandomMovementSimulator class.
- Connections between farms - Currently this is random, and there is no explicit network structure between farms.

Running a simulation

The project is a Java Eclipse project with no external dependencies (apart from the normally included JRE System Library), and can be built and run within Eclipse – the RandomMovementSimulator class has a main method which runs one simulation with default parameters to the default output file location (simulations\\simMov*).

To run one default simulation from the command prompt type: (this will echo to screen)

```
java -jar RandomMovementSimulator.jar
```

or to specify a new output location type: (no screen echo)

```
java -jar RandomMovementSimulator.jar myDirectory\\myOutputFileStem
```

Input parameters

The simulator uses its default input parameters at present (although in the future it may read these from a configuration file)

Parameter	Type / Class	Description	Default Value
species	Species	Animal host species	CATTLE
startDate	Calendar	Start date of the simulation (dd/MM/yyyy)	01/01/2000
endDate	Calendar	End date, last movement occurs just before this date.	02/01/2014
numberOfFarms	int	Number of farms (fixed)	20
numberOfAnimalsPerFarm	int	Initial number of animals per farm	100
animalAge	int	Animals only live this long, at the moment every animal lasts 10 years	10
maxToMove	int	Maximum allowed number of animals to move from one farm to another in the time step	25
minToMove	int	Assuming that a movement is going to take place, this is the minimum of animals that will be moved from one place to another in the time step	5
maxBirth	int	Maximum number of animals that can be born on a farm in the time step	10
minBirth	int	Minimum number of animals to be born per farm per time step	0
maxMovsPerTime	int	Maximum number of farm to farm movements allowed per time step	2
minMovsPerTime	int	Minimum number of farm to farm movements allowed per time time	0
timeIncrementType	int	The type of time increment (days, months, years)	Calendar.MONTH
timeIncrementValue	int	The value of the above time increment. The default is 1 time step per month.	1

Output files

- All output files are comma (,) separated text files unless otherwise specified.
- All dates are day, month, year separated by forward slash (/), e.g. 31/12/2008
- For each simulation, 5 result files are generated as detailed below.
- The default output file name stem is “simulations\\simMov” and this assumes that the simulations directory exists in the project folder next to the “src” and “bin” folders.

Farm Locations

Output file = *_farmLocations.csv
Farm Locations column headers (3) = farmID, Easting, Northing
See Farm class.

Life History

Output file = *_lifeHistories.csv
Life History column headers (6) = animalID, date of birth, farmID of birth,
date of death, farmID of death, species

See LifeHistoryRecord class.

Movement Records

Output file = *_movementRecords.csv
Movement records column headers (5) = animalID, species, farmID, date, movementType

The movement records file contains just “single” ended movements, and the allowed types in this file are: ON (to a farm), OFF (from a farm), BIRTH, DEATH.

See MovementRecord class and MovementType enumerated class.

Movement Records – Paired From and To

Output file = *_movementRecords_paired.csv
Paired records column headers (6) = animalID, species, from farmID, from date, to farmID, to date

The paired movements records file contains “double” ended movements, recording where each animal came from and where it went to in a single line. This file does not contain any of the BIRTH or DEATH events, only the actual movements.

See MovementRecord class and MovementType enumerated class.

Farm Status

Output file = *_farmStatus.csv
Farm status column headers = farmID, Easting, Northing, number of animals, date

The farm status file contains the number of animals at each farm at each time point of the simulation. This file can be used as a sanity check of the movement record files above. Note that the number of animals recorded is the number that are present after all the movements for the previous time step – i.e. status at 01/04/2002 means the number of animals present on the farm at the start of April after all of the movements that have happened in March.

See the info method in the Farm class and the main loop (in run) of the RandomMovementSimulator class.