

The University of York

Department of Computer Science

**Submitted in part fulfilment for the degree of BEng.**

# **Evolutionary agent-based simulation modelling of human life-history evolution**

Caleb J. H. Riley

Version 0.01, 2016-November-15

Supervisor: Daniel W. Franks

Number of words = 2001, as counted by `wc -w`.  
This includes the body of the report only.



### **Abstract**

This is an abstract. Should be about 500 words long.



# Contents

<b>1</b>	<b>Introduction</b>	<b>9</b>
1.1	Motivation . . . . .	9
1.2	Aims . . . . .	9
1.2.1	Check validity of computational model . . . . .	9
1.2.2	Rewrite model into an object oriented language . . . . .	9
1.2.3	Extend the model to allow co-evolution of preference . . . . .	10
1.3	Thesis Outline . . . . .	10
1.4	Statement of Ethics . . . . .	10
<b>2</b>	<b>Literature Review</b>	<b>11</b>
2.1	Menopause . . . . .	11
2.1.1	Patriarch Hypothesis . . . . .	11
2.1.2	Grandmother Hypothesis and the Mothering Hypothesis . . . . .	13
2.1.3	Reproductive Conflict Hypothesis . . . . .	13
2.1.4	Other Hypotheses . . . . .	14
2.2	Evolution . . . . .	14
2.2.1	Key Concepts of Evolution . . . . .	15
2.3	Modelling in Biology . . . . .	16
2.3.1	Differential Equations . . . . .	16
2.3.2	Multi-Agent Systems . . . . .	17
2.3.3	Genetic Algorithms . . . . .	17
<b>3</b>	<b>Problem Description</b>	<b>19</b>
3.1	Constraints of the Patriarch Hypothesis . . . . .	19
3.2	Why Men Matter: a mathematical model . . . . .	19
3.3	Mate Choice and the Origin of Menopause: a computational model . . . . .	21
<b>4</b>	<b>Design and Implementation</b>	<b>22</b>
4.1	Class Structure . . . . .	22
4.1.1	Person Class . . . . .	22
4.1.2	MortalityDistribution Class . . . . .	25

## *Contents*

4.1.3	ReproductionDistribution Class . . . . .	25
4.1.4	PreferenceDistribution Class . . . . .	25
4.1.5	Genetics Class . . . . .	26
<b>5</b>	<b>Results and Evaluation</b>	<b>28</b>
<b>6</b>	<b>Conclusion</b>	<b>29</b>
.1	Main.py . . . . .	32

## List of Figures

2.1	Verhulst-Pearl Equation . . . . .	16
3.1	Marginal fertilities of females $G_n^F$ and males $G_n^F$ . . . . .	20
3.2	Stable growth equation. . . . .	20
3.3	Harmonic mean model . . . . .	20

List of Tables

4.1 Person Class Attributes . . . . . 23

4.2 Person Class Methods . . . . . 24

4.3 MortalityDistribution Class Methods . . . . . 25

4.4 ReproductionDistribution Class Methods . . . . . 26

4.5 PreferenceDistribution Class Methods . . . . . 26

4.6 Genetics Class Methods . . . . . 27



# 1 Introduction

## 1.1 Motivation

The senescence of organisms is a long standing evolutionary puzzle: why would genes which cause an organism to degrade over time be selected for? In particular, why could genes which cause a reduction in fertility with age provide evolutionary benefit versus reproducing until death? Menopause is of particular interest since it both only occurs within females, but within limited number of species.

Although we know the the biological cause of menopause, there are many theories about why it is beneficial for a species to cease reproduction long before death.

## 1.2 Aims

### 1.2.1 Check validity of computational model

This project focuses on reproducing a computational model [1] that implements the patriarch hypothesis [2], a theory that males' preference for younger females caused long post-reproductive lifespans. Replicating models is useful to check their validity and discover any flaws they may possess. Replication also provides a jumping off point for improving the model or for adapting it to other hypotheses to do with the evolution of menopause or population modelling in general.

### 1.2.2 Rewrite model into an object oriented language

The model is also being adapted from C, a performant but verbose and old imperative language, into Python, a modern object oriented language designed to be as readable as possible. Python is also popular among scientific research, so this will make the model more accessible.

### 1.2.3 Extend the model to allow co-evolution of preference

Whilst mortality and fertility are both able to evolve under the current model, the mating preferences of the males are fixed. Since the hypothesis relies on the preference of older males for younger females, it would be useful to see how allowing this preference to evolve over time would affect the results of the model.

## 1.3 Thesis Outline

**Chapter Two Literature Review** The literature review provides a critical overview of different hypotheses for the origin of menopause, as well as background information on evolution and modelling in biological contexts.

**Chapter Three Problem Description** Is an overview of the model that already exists, how it models the hypothesis and shortcomings with the model.

**Chapter Four Design and Implementation** Is a description of how the solution to the problem was designed and implemented.

**Chapter Five Results and Evaluation** Presents the results of the model created and takes a critical look at how the work progressed.

**Chapter Six Conclusion** making judgement of the solution and the results, suggesting new work.

## 1.4 Statement of Ethics

The project is implementing a computational model which is purely theoretical. Although it concerns human reproduction, it does not involve human subjects and implementing it with them would be impossible due to time constraints (as evolution take place over many many years), there are no real ethical concerns.

## 2 Literature Review

The literature review provides an overview of existing work surrounding the problem, as well as background information that will be relevant to its solution.

### 2.1 Menopause

Menopause is the process where females cease having menstrual periods and become infertile. Whilst reduction in fertility with senescence is not uncommon among species, the long lifespan after menopause (post-reproductive lifespans, or PRLSs) are a relatively rare trait, being only found in the wild in humans, short finned pilot whales and orcas (killer whales). Although we understand to some extent the physiological cause (a decrease in oestrogen and progesterone), the evolutionary benefits of a reduction in reproduction seem unclear as any reduction in fertility would seem to be a reduction in evolutionary fitness. As such this evolutionary puzzle has resulted in many different theories for its existence [3].

#### 2.1.1 Patriarch Hypothesis

The patriarch hypothesis [2] hypothesises that menopause came about due to older, high status males having access to younger female mates allowing them to reproduce for much longer. This increased the proliferation of genes linked to longevity, increasing both the lifespan of males and females. This is reliant on several factors:

- That females have a limited number of oocytes (immature ova) which are depleted over time, and that reproduction naturally comes to an end when they run out. [4] In early females, before female longevity increased, most females died before their supply of oocytes had been completely depleted, and so did not experience menopause.

## 2 Literature Review

- That longevity causing mutations are on the X rather than the Y chromosome. If the gene were on the Y chromosome then the increase longevity would only be present in the males. The paper suggests that female longevity (and therefore have long post reproductive lifespans) is a result of females being "dragged along" by male longevity being passed on through the X chromosome.
- That older men continue to reproduce and pass on their longevity causing genes. High status males (normally those with a better reputation for hunting and gathering) would start a new family with a second, younger wife once their first wife had undergone menopause. Thus males carrying longevity causing genes would have greater opportunity to pass them on.

A mathematical model of the hypothesis was created [5] which was a set of functions which relied on a group of females aged  $i$ , and a group of males aged  $j$ , the females' fertility at age  $i$  and the mating preference of females of age  $i$  for males of age  $j$  and vice versa. The main problem with this model is that it had a fixed age at which females stopped reproducing, rather than letting it evolve over time as you would expect in an evolutionary model. Indeed this model can be interpreted to suggest that males' preference for younger females came to be after the evolution of menopause, rather than as a cause of it.

To try and correct some of these shortcomings, a computational model [1] which does not have a fixed age where females become infertile. Each member of the population is modelled as an individual agent, with genes that affect either mortality or fertility, with genes acting either in a sex dependent or a sex indifferent manner. Pseudo-random numbers are used to determine births, deaths and partnerships against predetermined tables which are modified by genetics. One of the main shortcomings of the model is that it does not allow the preferences of the agents to coevolve with the change in fertility and mortality.

Overall the main issue with the patriarch hypothesis is that males' preference for younger females may itself be an epiphenomenon which emerged as a result of menopause, rather than the other way round. Knowledge that females would become infertile after a certain age would cause males to have a preference for younger females who had more fertile years left.

### 2.1.2 Grandmother Hypothesis and the Mothering Hypothesis

The grandmother hypothesis [6–8] theorises that the evolution of menopause came about due to the increased inclusive fitness of a population caused by grandmother being able to help her children raise their offspring through providing childcare and resources such as food. This is due to the grandmother being post-reproductive and not having any children that are of an age where they need care. This grandmothering improves the survival of grandchildren and allows her children to reproduce more frequently. Populations which had post-reproductive women would grow more rapidly as a result, meaning they would overtake populations where they did not occur.

One of the criticisms of the grandmother hypothesis is that the degree of increase to inclusive fitness has been overstated, and that whilst the benefit might be a factor, it is unlikely to be the only factor to have caused it. Another criticism is that if the benefits of grandmothering are so great, why do males remain reproductive throughout their lives? If the increase to inclusive fitness is so great from grandmothering then it seems unlikely that this benefit would be limited to just one sex.

The mothering hypothesis is a similar theory in which a species improves the survival of offspring by ceasing reproduction at a certain point so that resources could be focused on raising existing children, rather than continuing to focus on reproduction and newborn children. The reduced mortality rate of children would again result in an increase of inclusive fitness.

### 2.1.3 Reproductive Conflict Hypothesis

The reproductive conflict hypothesis [9, 10] is another hypothesis based on grandmothers: this time instead of them stopping reproduction to provide childcare, they cease reproduction so that their offspring are not competing for resources such as food with their grandchildren. Menopause reduces the amount of time where a mother is fertile at the same time as her offspring, reducing the likelihood that they have young children at the same time. This reduced competition for resources means that children will have a greater chance of survival, increasing the inclusive fitness of the group as a whole.

#### 2.1.4 Other Hypotheses

A non-adaptive hypothesis is that women have a fixed number of oocytes (immature eggs), and that menopause did not evolve, but is an epiphenomenon of us now living past the age where they run out. [11, 12] Proponents of this theory often use the evidence of menopause occurring in some primates in captivity – their increased lifespan means they now live well past becoming infertile. They suggest that human's improvements to medicine and nutrition have extended our lifespans in the same way, which is why humans now experience menopause. Counter to this however is the fact that menopause is experienced by orcas and short-finned pilot whales. Since they do not have access to healthcare, you can only conclude that although improved lifespan makes a woman more likely to live through menopause, it is not itself what causes it.

Another hypothesis is that menopause came about to guard against the negative effects of pregnancy at an advanced age. Increased maternal age can result in: miscarriage, chromosomal abnormalities, and pregnancy diabetes, among other effects [13]. These complications would result in an increased need for investment of resources in reproduction itself (with a chance that the baby would not make it to full term), as well as resulting in the mother's other offspring being left with fewer resources such as food and childcare, or potentially being left motherless. However the improvements made to medical care have had a greater impact on both infant and maternal mortality, and the effect is a steady increase with age, rather than a sharp threshold effect. This improvement to healthcare came about after menopause was present in humans, and so this cannot be the cause.

## 2.2 Evolution

Charles Darwin's theory of evolution, first proposed in *On the Origin of Species* [14] may be one of the most important and influential academic pieces of work in science. The theory supposes that organisms with traits favourable to their environment were more likely to survive and reproduce, increasing the prevalence of the trait over other less useful traits. This is often summarised as "survival of the fittest".

### 2.2.1 Key Concepts of Evolution

#### **Fitness**

Fitness is the measure of how well suited an organism is to an environment, and how likely it is to pass on its genetic information as a result. Note that "fitness" in this instance is relating to how well an organism "fits in" with its environment, and not athletic fitness.

#### **Inclusive Fitness**

Inclusive fitness is an extension of the concept of **Fitness**, in which you can improve your fitness by improving the survival of individuals with similar genetics to you, such as siblings, nieces and nephews or grandchildren. This increase to fitness is proportional to your relatedness to them. Improvement of inclusive fitness could come about due to provision of resources such as food and childcare.

#### **Selection**

Selection is the process in which survival and reproduction of a species is affected by the phenotype (physical expression of genetics). A trait that increases lifespan and allows more time for reproduction will cause there to be a greater chance for more offspring with that trait to be produced. By comparison a trait that decreases lifespan or realised fertility will tend to become less prevalent in the population as they have fewer offspring and these offspring if they have the same phenotype will inherit the lower survival chances. In essence useful traits survive, whereas traits that are less useful will gradually disappear.

#### **Crossover**

Crossover is part of the process that combines the genetics of two parents into the new set of genetics present in their offspring. For each chunk of genetic information, the offspring has a 50% chance of receiving that information from their mother, and a 50% chance of receiving the information from their father. This results in an offspring with traits of both their mother and father. When a gene at a particular location is the same in both the mother and father, this gene is transmitted to the child 100% of the time.

#### **Mutation**

Genetic mutation is the change of genetic information at random,

usually from mistakes when copying genes over from their parents. This random change results in the emergence of new traits since it can result in genetics that are not previously present in the population. These new traits can be deleterious, neutral or positive.

### **Coevolution**

Coevolution is the evolution of traits in a species in response to the evolution of traits in the same or another species. Think of it as a genetic arms race; a species that evolves a form of camouflage may cause a species that predate on

## **2.3 Modelling in Biology**

In biology, there are many problems that it is only possible to study through the use of modelling. This can be due to the difficulty in finding individuals to study (and therefore a lack of data), the amount of time required to conduct a study (especially for evolutionary effects which take place over many generations) or the ethical constraints of conducting experiments on living organisms (especially humans). Thus models provide a useful tool for examining in a theoretical space hypotheses that cannot be studied in the physical space.

### **2.3.1 Differential Equations**

Differential equations are a traditional tool for use in modelling in general, as they allow you to model the rate of change in a system (using its derivatives). The Verhulst-Pearl equation [15, 16] in Figure 2.1 is an old model for simulating population growth.

$$\frac{dN}{dt} = rN(1 - \frac{N}{K})$$

Figure 2.1: Verhulst-Pearl Equation

$N$  is the current population at time  $t$ ,  $r$  is the rate of population growth and  $K$  is the carrying capacity of the system, or the maximum number of individuals the environment is capable of supporting. As  $N$  approaches  $K$  the rate of population growth slows until an equilibrium is met. Differential equations are capable of modelling complex systems with several variables, but reaching a solution is not always possible.



### 2.3.2 Multi-Agent Systems

Multi-agents systems are used to model individuals that interact with each other in a similar way to how organisms interact with each other. Rather than modelling a population through differential equations, you could model it by having a set of agents, who interact to form couples and produced offspring based on their inherent properties such as fertility, mortality and age. The population growth would then become apparent through the emergent properties of the system, rather than through a set, programmed idea.

More detail/ideas here.  
References needed

### 2.3.3 Genetic Algorithms

Genetic algorithms are mathematical representations of the process of evolution, with the genetics being used to set parameters for program behaviour, rather than the traits of an organism. An initial pool of individual genetics (typically binary strings are used) are created. These gene strings are then used to set parameters for whatever computation is being done. The best performing sets of genes are selected using an arbitrary heuristic fitness function (such as shortest path) to create the next generation of genetics.

Random parents are selected from the best performers, and the next generation are created by crossing over the genes of the parent pair. The next generation of individuals also undergo random mutation of their genes (i.e. by flipping random bits of the binary string) to prevent the algorithm from getting stuck on a single solution and to produce potentially beneficial genes which may have not appeared in the population yet (and so cannot be acquired from their parents). These stages all correspond to the concepts of biological evolution, and so can be useful when modelling genetic change. Genetic algorithms are good at working out optimal solutions to problems with lots of parameters as the algorithm combines parts of the best solutions from each generation to produces new, potentially better solutions.

## 2 *Literature Review*

More detail/ideas here. References needed

## 3 Problem Description

### 3.1 Constraints of the Patriarch Hypothesis

The patriarch hypothesis [2] states that menopause came about due to older males' preference for younger females. Since humans have relatively long lifespans and the trait of long, postreproductive lifespans is present in all humans, it is necessary to use a model to study the validity of this hypothesis. In the paper, 3 main conditions that the hypothesis depends on are set out – it follows that any model examining the hypothesis should model these constraints. The constraints are as follows:

1. That the depletion of oocytes (undeveloped ovum) must be a constraining factor on when menopause occurs. Menopause was not previously observed until longevity increased to the extent that females lived long enough that depletion could occur.
2. That longevity causing genes are present on the X chromosome, rather than the Y. If it were present on the Y chromosome then a general increase in longevity would only be present in males. Thus the increase in longevity must be on the X chromosome.
3. That older, high status males still reproduce. This is required for the accumulation of the longevity promoting gene, as if a male with the gene who lives for a long time only reproduces over the same time period as one without the longevity promoting gene, then they will have an equal chance of passing their genes on and the longevity gene will not become more prevalent.

### 3.2 Why Men Matter: a mathematical model

A mathematical model was created [5] based on the two sex demography work of Schoen [17] and Pollak [18] which link together a group of females and a group of males of different ages. The marginal fertilities for females and males are constructed in a similar manner (as seen in

### 3 Problem Description

Figure 3.1) but marginal male fertility is assumed to be positive when realised male fertility is non-zero. These marginal fertilities do not represent an individual's virility, rather their biological ability to procreate combined with the chance they are able to mate.

$$G_n^F = \sum_{ij} \frac{\delta B_{ij}}{\delta F_n}, \quad G_n^M = \sum_{ij} \frac{\delta B_{ij}}{\delta M_n}$$

Figure 3.1: Marginal fertilities of females  $G_n^F$  and males  $G_n^M$

These marginal fertilities make up part of the equation for stable growth (Figure 3.2), where  $l_n^F$  and  $l_n^M$  correspond to the female and male life expectancies respectively, and  $\sigma$  is the male-to-female sex ratio at birth.

$$1 = \sum_{n \geq 1} (G_n^F l_n^F \lambda^{-n} + \sigma G_n^M l_n^M \lambda^{-n})$$

Figure 3.2: Stable growth equation.

A harmonic mean model similar to that of Schoen's [17] was constructed, where  $b_i$  is the fertility of a female aged  $i$ , and  $\alpha_{ij}$  are mating preference weights determined by real world data.

$$B_{ij} = \frac{b_i \alpha_{ij} F_i M_j}{F_i + M_j}$$

Figure 3.3: Harmonic mean model

When it comes to the three constraints imposed by [2] this model implements a reduction in female fertility in time but does not specifically link this to the oocyte depletion. The paper does however note that it is important to model the fertility and mortality of both sexes as any mutation that increases mortality past the age of menopause has no effect on population growth when only dealing with females. The model treats this mutation as being sex indifferent (meaning it would have to be located on the X chromosome). For the third constraint the paper largely relies on citing real-world data of hunter-gatherer societies and their social structures as evidence for continued male reproduction later

### *3.3 Mate Choice and the Origin of Menopause: a computational model*

in life. Data from the Dobe !Kung society [19] and preconstructed life tables [20] were used in the harmonic mean model (Figure 3.3) to this effect.

The main issue with this model is the fixed age of the end of reproduction, as mutations are only used to alter mortality, when they should also be used to affect the fertility of females.

### **3.3 Mate Choice and the Origin of Menopause: a computational model**

Existing model overview - relate to code What has been done Limitations of their model How this can be fixed
-----------------------------------------------------------------------------------------------------------------------

## 4 Design and Implementation

### 4.1 Class Structure

When designing a multi-agent system, it can be useful to model each agent as being an instance of a class, each with their own attributes (variables belonging to a class instance that are an abstract representation of the differing traits an agent has) and class methods (representing the actions an agent can take to interact with its environment).

#### 4.1.1 Person Class

The `Person` class provides the container for the attributes and methods necessary to build the model. Many of its attributes are instances of the classes used to hold probability distributions in control of mortality and fertility, as well as the person's mating preferences and genetics. Each time a new child is born, a new instance of `Person` is created, with the parameters for `__init__()` (see Table 4.2) based on their parents' attributes.

Table 4.1: Person Class Attributes

Attribute	Type	Purpose
age	int	Age of the person
sex	str	Sex of person ("f" for female and "m" for male)
alive	bool	Whether the person is currently alive
mortality	MortalityDistribution	The mortality distribution of the individual.
reproduction	ReproductiveDistribution	The mortality distribution of the individual.
pref_dist	PreferenceDistribution	The mating preference distribution of the individual.
genes	Genetics	The genetic information of the individual

Table 4.2: Person Class Methods

Method	Purpose
<code>__init__(self, age, sex, mort_dist, repr_dist, genes=None)</code>	Used when instantiating the class, setting up initial values for the properties based on those passed as parameters
<code>__repr__(self)</code>	Returns a string representation of the class
<code>increase_age(self)</code>	Assigns a person to next age class if they survive the current generation (based on a psuedo-ransom number and their mortality distribution), otherwise sets <code>alive</code> to <code>False</code> .
<code>calculate_age_class(self)</code>	Returns the derived attribute <code>age_class</code>
<code>reproduce(self)</code>	Returns <code>True</code> if a person is currently able to reproduce (based on a psuedo-random number and their reproductive distribution).
<code>calc_preference(self, other)</code>	Calculates and returns the mating preference that this person has for the other person (using their ages classes and the person's preference distribution)



### 4.1.2 MortalityDistribution Class

The `MortalityDistribution` class serves as a lookup table for the probability of death a given age, based on data from the input files `female_data.csv` and `male_data.csv`. There are two base instances of this class created `female_mortality` (using `female_data.csv`), the mortality distribution used by females and `male_mortality`, for males (using `male_data.csv`). The class also contains methods for getting survival probabilities based on age (see Table 4.3).

Table 4.3: MortalityDistribution Class Methods

Method	Purpose
<code>__init__(self, input_file)</code>	Used when instantiating the class, reads distribution input from file
<code>__repr__(self)</code>	Returns a string representation of the class
<code>get_survival(self, age)</code>	Returns mortality probability value based on age.
<code>get_survival_by_age_class(self, age_class)</code>	Returns mortality probability value based on age class.

### 4.1.3 ReproductionDistribution Class

The `MortalityDistribution` class serves as a lookup table for the probability of being able to reproduce at a given age, based on data from the input files `female_data.csv` and `male_data.csv`. There are two base instances of this class created `female_mortality` (using `female_data.csv`), the mortality distribution used by females and `male_mortality`, for males (using `male_data.csv`). The class also contains methods for getting survival probabilities based on age (see Table 4.4).

### 4.1.4 PreferenceDistribution Class

The `PreferenceDistribution` class is used to store the mating preferences based on age class of each individual, and to provide access to the data. There are three instances of this class created `female_preference` representing females' mating preferences (using data from `female_pref.csv`), `male_map_preference` representing males who have a preference for

Table 4.4: ReproductionDistribution Class Methods

Method	Purpose
<code>__init__(self, input_file)</code>	Used when instantiating the class, reads distribution input from file
<code>__repr__(self)</code>	Returns a string representation of the class
<code>get_reproduction(self, age)</code>	Returns reproduction probability value based on age.
<code>get_reproduction_by_age_class(self, age_class)</code>	Returns reproduction probability value based on age class.

females their own age (using data from `male_map.csv`) and `male_myp_preference` representing male preference for younger females (using data from `male_myp.csv`).

Table 4.5: PreferenceDistribution Class Methods

Method	Purpose
<code>__init__(self, name, input_file)</code>	Used when instantiating the class, reads distribution input from file
<code>__repr__(self)</code>	Returns a string representation of the class
<code>get_preference(self, age_class_one, age_class_two)</code>	Returns the mating preference of a person <code>age_class_one</code> , <code>age_class_two</code> .

#### 4.1.5 Genetics Class

Table 4.6: Genetics Class Methods

Method	Purpose
<code>__init__(self, genes=None)</code>	Used when instantiating the class. Uses genetic string passed or generates random gene string using the <code>random_genes</code> method.
<code>__repr__(self)</code>	Returns a string representation of the class
<code>random_genes(self, size)</code>	Returns a random string of "1"s and "0"s of length equal to <code>size</code> .
<code>crossover(self, other)</code>	Returns a gene string made of of random segments of the two gene strings.
<code>mutate(self, rate)</code>	Pseudo-randomly flips "0"s and "1"s in the gene string with a probability equal to <code>rate</code> .

## 5 Results and Evaluation

Sample output - maybe some graphs What could have been done better  
- achieved with more work/time

## **6 Conclusion**

Suggestions for further work.

## Bibliography

- [1] R. A. Morton, J. R. Stone, and R. S. Singh, "Mate choice and the origin of menopause," *PLOS Computational Biology*, 2013. [Online]. Available: <http://dx.doi.org/10.1371/journal.pcbi.1003092>
- [2] F. Marlowe, "The patriarch hypothesis," *Human Nature*, 2000. [Online]. Available: <http://dx.doi.org/10.1007/s12110-000-1001-7>
- [3] D. P. Croft, L. J. Brent, D. W. Franks, and M. A. Cant, "The evolution of prolonged life after reproduction," *Trends in Ecology & Evolution*, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0169534715001044>
- [4] J. Wood, S. Weeks, G. Bentley, and K. Weiss, "Human population biology and the evolution of aging," *Biological Anthropology and Aging: Perspectives on Human Variation over the Life Span*, 1994.
- [5] S. D. Tuljapurkar, C. O. Puleston, and M. D. Gurven, "Why men matter: Mating patterns drive evolution of human lifespan," *PLOS ONE*, 2007. [Online]. Available: <http://dx.doi.org/10.1371/journal.pone.0000785>
- [6] H. P. Alvarez, "Grandmother hypothesis and primate life histories," *American Journal of Physical Anthropology*, vol. 113, no. 3, pp. 435–450, 2000. [Online]. Available: [http://dx.doi.org/10.1002/1096-8644\(200011\)113:3<435::AID-AJPA11>3.0.CO;2-O](http://dx.doi.org/10.1002/1096-8644(200011)113:3<435::AID-AJPA11>3.0.CO;2-O)
- [7] P. S. Kim, J. S. McQueen, J. E. Coxworth, and K. Hawkes, "Grandmothering drives the evolution of longevity in a probabilistic model," 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0022519314001465>
- [8] P. S. Kim, J. E. Coxworth, and K. Hawkes, "Increased longevity evolves from grandmothering," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 279, no. 1749, pp. 4880–4884, 2012.

- [9] M. A. Cant and R. A. Johnstone, "Reproductive conflict and the separation of reproductive generations in humans," *Proceedings of the National Academy of Sciences*, vol. 105, no. 14, pp. 5332–5336, 2008.
- [10] D. P. Croft, R. A. Johnstone, S. Ellis, S. Nattrass, D. W. Franks, L. J. N. Brent, S. Mazzi, K. C. Balcomb, J. K. B. Ford, and M. A. Cant, 2017.
- [11] L. Van Valen, "Ovarian excess and the evolution of menopause," *Evol. Theory*, vol. 12, pp. 131–153, 2003.
- [12] G. S. Cooper and D. P. Sandler, "Age at natural menopause and mortality," *Annals of epidemiology*, vol. 8, no. 4, pp. 229–235, 1998.
- [13] J. Cleary-Goldman, F. D. Malone, J. Vidaver, R. H. Ball, D. A. Nyberg, C. H. Comstock, G. R. Saade, K. A. Eddleman, S. Klugman, L. Dugoff *et al.*, "Impact of maternal age on obstetric outcome," *Obstetrics & Gynecology*, vol. 105, no. 5, Part 1, pp. 983–990, 2005.
- [14] C. Darwin, *On the origin of species*. John Murray, 1859.
- [15] J. Garnier and A. Quetelet, *Correspondance mathématique et physique*. Impr. d'H. Vandekerckhove, 1838, no. v. 10. [Online]. Available: <https://books.google.co.uk/books?id=8GsEAAAAYAAJ>
- [16] R. Pearl and L. J. Reed, "On the rate of growth of the population of the united states since 1790 and its mathematical representation," *Proceedings of the National Academy of Sciences*, vol. 6, no. 6, pp. 275–288, 1920.
- [17] R. Schoen, "The harmonic mean as the basis of a realistic two-sex marriage model," *Demography*, vol. 18, no. 2, pp. 201–216, 1981.
- [18] R. A. Pollak, "Two-sex demographic models," *Journal of Political Economy*, vol. 98, no. 2, pp. 399–420, 1990.
- [19] N. Howell, *Demography of the Dobe! kung*. JSTOR, 1979, vol. 35.
- [20] M. Gurven and H. Kaplan, "Hunter-gatherer longevity: Cross-cultural perspectives," *Population and Development Review*, vol. 33, no. 2, pp. 321–365, 2007.

## **.1 Main.py**

```
import csv
import random
```

```
class Person:
    def __init__(self, age, sex, mort_dist, repr_dist, pref_dist, genes):
        self.age = age
        self.sex = sex
        self.alive = True
        self.mortality = mort_dist
        self.reproduction = repr_dist
        self.pref_dist = pref_dist
        if genes is None:
            self.genes = Genetics()
        else:
            self.genes = genes

    def __repr__(self):
        return ("AGE:_" + str(self.age).zfill(2)
        + ",_SEX:_" + str(self.sex)
        + ",_ALIVE:_" + str(self.alive)
        + ",_GENES:_" + str(self.genes))

    def increase_age(self):
        if random.uniform(0.0,1.0) < self.mortality.get_survival(self.age):
            self.age += 5
        else:
            self.alive = False

    def calculate_age_class(self):
        return ((self.age + 5) / 5)

    def reproduce(self):
        return self.reproduction.get_reproduction(self.age) > random.uniform(0.0,1.0)

    def calc_preference(self, other):
        return self.pref_dist.get_preference(self.calculate_age_class(other))
```



```
# Set up MortalityDistribution

class MortalityDistribution:
    def __init__(self, input_file):
        self.probability_table = []
        with open(input_file, "rb") as f:
            reader = csv.reader(f)
            for row in reader:
                self.probability_table.append(float(row[3]))

    def __repr__(self):
        str_representation = ""
        for row in self.probability_table:
            str_representation += str(row).zfill(6) + "\n"
        return str_representation

    def get_survival(self, age):
        if age < len(self.probability_table) and age >= 0:
            return self.probability_table[age]
        else:
            return 0.0

    def get_survival_by_age_class(self, age_class):
        return get_survival((age_class * 5) - 1)

female_mortality = MortalityDistribution("female_data.csv")
#print(female_mortality)

male_mortality = MortalityDistribution("male_data.csv")
#print(male_mortality)

print("Initial_mortality_distributions_set_up")

# Set up ReproductiveDistribution

class ReproductiveDistribution:
    def __init__(self, input_file):
```

## *Bibliography*

```
        self.probability_table = []
        with open(input_file, "rb") as f:
            reader = csv.reader(f)
            for row in reader:
                self.probability_table.append(float(row[2]))

    def __repr__(self):
        str_representation = ""
        for row in self.probability_table:
            str_representation += str(row).zfill(6) + "\n"
        return str_representation

    def get_reproduction(self, age):
        if age < len(self.probability_table) and age >= 0:
            return self.probability_table[age]
        else:
            return 0.0

    def get_reproduction_by_age_class(self, ageClass):
        return get_reproduction((ageClass * 5) - 1)

female_reproduction = ReproductiveDistribution("female_data.csv")
#print(female_reproduction)

male_reproduction = ReproductiveDistribution("male_data.csv")
#print(male_reproduction)
print("Initial_reproductive_distributions_set_up")

# Set up PreferenceDistribution
class PreferenceDistribution:
    def __init__(self, name, input_file):
        self.name = name
        self.distribution = []
        with open(input_file, "rb") as f:
            reader = csv.reader(f)
            for row in reader:
                new_row = []
                for cell in row:
```

```
        new_row.append(int(cell))
        self.distribution.append(new_row)

    def __repr__(self):
        str_representation = ""

        for row in self.distribution:
            row_str = ""
            for cell in row:
                row_str += str(cell) + ",_"
            row_str += "\n"

            str_representation += row_str

        return str_representation

    def get_preference(self, age_class_one, age_class_two):
        return self.distribution[age_class_one][age_class_two]

female_preference = PreferenceDistribution("female", "female_pref.csv")
#print(female_preference)

male_map_preference = PreferenceDistribution("male_map", "male_map.csv")
#print(male_map_preference)

male_myp_preference = PreferenceDistribution("male_myp", "male_myp.csv")
#print(male_myp_preference)

print("Initial_preference_distributions_set_up")

class Genetics:
    def __init__(self, genes=None):
        if genes is None:
            self.genes = self.random_genes(8)
        else:
            self.genes = genes

    def __repr__(self):
        return str(self.genes)
```

## Bibliography

```
def random_genes(self, size):
    return "".join(str(random.randint(0, 1))
                    for x in xrange(size))

def crossover(self, other):
    # Crosses over this set of genes with another set of genes with
    crossed = ""

    for i in range(0, len(self.genes)):
        if 0.5 > random.uniform(0.0, 1.0):
            crossed += self.genes[i]
        else:
            crossed += other.genes[i]

    return crossed

def mutate(self, rate):
    new_genes = ""
    for gene in self.genes:
        if rate > random.uniform(0.0, 1.0):
            if gene == "0":
                new_genes += "1"
            else:
                new_genes += "0"
        else:
            new_genes += gene

    self.genes = new_genes

# Set up base female population
print("###_Setup_Female_Population_###")

female_population = []

for i in range(0, 500):
    age = random.randint(0, 18)*5
    female_population.append(Person(age, "f", female_mortality, female

# for female in female_population:
#     print(female)
```

```
#
# print(female_population[25].reproduction)

for i in range(1,101):
    born = 0
    died = 0
    for female in female_population:
        female.increase_age()

        if not female.alive:
            female_population.remove(female)
            died += 1

    while len(female_population) < 500:
        female = random.choice(female_population)

        if female.reproduce():
            female_population.append(Person(0,"f", female.mortality, female))
            born += 1

    #print(str(i).zfill(2) + " Population " + str(len(female_population))).

female_population.sort(key=lambda x: x.age, reverse=False) #

age_groups = {}
for female in female_population:
    if female.age in age_groups:
        age_groups[female.age] += 1
    else:
        age_groups[female.age] = 1
    #print(female)

age_indices = []

for age_group in age_groups:
    age_indices.append(age_group)

age_indices.sort()

for age_group in age_groups:
```

## *Bibliography*

```
        print(str(age_group) + ":_ " + str(age_groups[age_group]))

print("###_Female_Burn_In_Period_Ended_###")

# Set up base male population
print("###_Setup_Male_Population_###")
male_population = []

for i in range(0,500):
    age = random.randint(0,18)*5
    male_population.append(Person(age, "m", male_mortality, male_repro

# for male in male_population:
#     print(male)
#
# print(male_population[25].reproduction)

for i in range(1,51):
    born = 0
    died = 0
    for male in male_population:
        male.increase_age()

        if not male.alive:
            male_population.remove(male)
            died += 1
    while len(male_population) < 500:
        male = random.choice(male_population)

        if male.reproduce():
            male_population.append(Person(0,"m", male_mortality, male_
            born += 1

    #print(str(i).zfill(2) + "   Population   " + str(len(male_population

male_population.sort(key=lambda x: x.age, reverse=False)

age_groups = {}
for male in male_population:
    if male.age in age_groups:
```

```
        age_groups[male.age] += 1
    else:
        age_groups[male.age] = 1
    #print(male)

for age_group in age_groups:
    print(str(age_group) + ":" + str(age_groups[age_group]))

print("###_Male_Burn_In_Period_Ended_###")

print("Female_Population_Size:" + str(len(female_population)))
print("Male_Population_Size:" + str(len(male_population)))

### Begin MAP simulation ###
print("###_Begin_MAP_Simulation_###")
map_output = open("map_simulation_output.csv", "wb")
writer = csv.writer(map_output, delimiter=",", quotechar='"', quoting=csv.QUOTE_MINIMAL)

map_females = female_population[:]
map_males = male_population[:]

for gen in range(1,51):
    pairings = 0.0
    born = 0.0
    died = 0.0

    map_females.sort(key=lambda x: x.age)
    map_males.sort(key=lambda x: x.age)

    # Aging Phase
    for female in map_females:
        female.increase_age()
        if not female.alive:
            died += 1.0
            map_females.remove(female)

    for male in map_males:
        male.increase_age()
        if not male.alive:
            died += 1.0
```

```
map_males.remove(male)

# Reproduction Phase
while (len(map_males) + len(map_females)) < 1000:
    female = random.choice(map_females)
    male = random.choice(map_males)

    if (female.calc_preference(male) == 1 and male.calc_preference(female) == 1):
        if female.reproduce() and male.reproduce():
            born += 1
            if 0.5 > random.uniform(0.0, 1.0):
                new_person = Person(0, "f", female_mortality, female_mortality)
                map_females.append(new_person)
            else:
                new_person = Person(0, "m", male_mortality, male_mortality)
                map_males.append(new_person)

population = len(map_males) + len(map_females)

map_age_groups = {}

for male in male_population:
    if male.age in map_age_groups:
        map_age_groups[male.age] += 1
    else:
        map_age_groups[male.age] = 1

for female in female_population:
    if female.age in map_age_groups:
        map_age_groups[female.age] += 1
    else:
        map_age_groups[female.age] = 1

map_age_indices = []

for age_group in map_age_groups:
    map_age_indices.append(age_group)

map_age_indices.sort()
```



```
print("\n_###_GENERATION_" + str(gen) + "_###")
for age_group in map_age_indices:
    print(str(age_group) + ":_ " + str(map_age_groups[age_group]))

#print(str(gen).zfill(2) + " - Population: " + str(population).zfill(4))
writer.writerow([gen, population, born, died, (born-died)])
```