GeneScape

Agent Based model of Gene Flow from crops and its implications for sustainability of Genetic Modification

Graduation thesis by Igor Nikolić for obtaining the Ir. degree at the Faculty of Applied Sciences, TU Delft on the 2^{nd} of November 2001

Executed jointly at Substances & Products section of the Centre for Environmental Science in Leiden and the Process Systems Engineering group of the DelftChemTech at TU Delft, in the period from 1.12.2000 to 2.11.2001.

Professor responsible for graduation: Prof. Ir. G.J. Harmsen, Hoogewerff-chair for Sustainable Chemical Technology, TU Delft

Project by:	Details
Igor Nikolić	TUD Student number : 9605013 Address: Korvezeestraat 375 2628 DS, Delft, The Netherlands Email: inikolic@tnw.tudelft.nl Telephone: +31 (0)624652486
Supervisors:	
Prof. ir. G.J. Harmsen	Email: g.j.harmsen@tnw.tudelft.nl Telephone: +31 (0)152784355
Drs. R. Huele	Email: huele@cml.leidenuniv.nl Telephone: +31 (0)715275648
Ir. G. Korevaar	Email: g.korevaar@tnw.tudelft.nl Telephone: +31 (0)152784466
Dr. E. van de Voet	Email: voet@cml.leidenuniv.nl Telephone: +31 (0)715277480

Abstract

If Genetic Modification is to be accepted as a sustainable technological development its social, ecological and economical impacts must be understood.

One important tool in determining the environmental and human risks of a technological development such as Genetical Modification (GM), is the Quantitative Risk Analysis (QRA). QRA can quantify the risks of a new technology, so that the society can make a choice whether the benefits outweigh the risks. Only if the risk of a new technological development is very small and acceptable, can it be sustainable.

One of the main risks from GMOs is the spread of their genes through Gene Flow into wild populations causing genetic pollution and potentially ecosystem disruption.

The current QRA is based on mass flow models. It does not consider the fact that law of conservation of mass does not hold for genes, which are essentially information. QRA also dos not take the imbedded open feedback loops present in life forms, i.e. the ability to self-amplify and reproduce, into consideration. Because of these shortcomings, the current risk assessment tools are probably inadequate to answer the question whether Genetic Modification is a sustainable technological development.

Next to the lack of scientific knowledge about the impacts, there is a lack of societal acceptance of GM. Part of the reason for the lack of acceptance is the unacceptability of the tools for risk prediction. There is a need for a risk analysis tool that is both acceptable and able to describe key aspects of life.

Agent Based Models (ABMs) are a candidate modeling framework for such a tool because they are isomorphic to living organisms. Furthemore, they are intuitive and easy to understand, forming a basis for societal acceptance.

By defining evolutionary processes and plant entities in Agent terms an Agent Based Model called GeneScape has been developed. The model explicitly describes Plants as entities with states and interactions. Plants populate a geometry, the Field, through which the environment, The World, enforces the rules for pollen distribution, mating and genetic inheritance. The World also determines how the information content of the plant, its genome, is to be translated into a fitness, and thus power to reproduce.

While there is a shortage of good experimental data for validation, GeneScape is evaluated and found to be coherent and biologically sound. It is however not a quantitative prediction tool for Gene Flow, but an explicit and graphical representation of the mental models of Darwinian selection, Gene Flow and plant ecology. Its strength lies in the fact that it allows the visualization of thought experiments on the behavior of GMO crops

and the neighboring plant populations. It can serve as a basis for further development of a tool for risk assessment for GMOs.

GeneScape can only be useful for answering questions about the sustainability of Genetic Modification if it is socially acceptable itself. Therefore, a societal acceptance hypothesis is formulated. The hypothesis states that GeneScape is not yet societally acceptable because it is too complex and the outcomes are open to interpretation. However, it is easily understandable and illustrative, which forms a basis for acceptance.

The most important conclusion from the use of GeneScape is the insight into the importance of the environment a GMO gene is placed in. The most important variables governing Gene Flow (Pollen spread distance, fitness increase by the GMO gene and the relatedness level of the surrounding populations) are dependent on environmental factors. Since the environment can not be controlled Genetically Modified Organisms can only be sustainable in a closed, controlled environment.

The model is implemented in the Java 2 programming language, making use of the Ascape class library developed by the Brookings Institute. It is available for online execution on www.IgorNikolic.com

Contents

List of Tables

List of Figures

Chapter 1

General Introduction

In this chapter I will first discuss the background question of this graduation project. Further, its key components, Genetically Modified Organisms (GMOs) and Agent Based Models (ABMs) will be quickly introduced. Finally, the structure of the report will be presented. But first, the background question.

1.1 The background question

The central background question to this project is: *Is genetic modification a sustainable technological development ?*

The central question is a rather broad one. It implies several assumptions. By Genetic Modification I mean a modification of the genome of an organism in a way that is not achievable by natural means. I also assume that such an organism is not kept isolated, but is placed in a natural environment. Also, I will exclude the inevitable application of genetic modification of the human species from the discussion.

By sustainable is meant that the effects of this modification on the social, ecological, economic or any other system that the society consists of are such that the existence of the society is not threatened by them and that the society accepts any other, non-existence threatening effects.

1.2 GMOs and sustainability

GMOs as a solution Use of Genetically Modified Organisms (GMOs) is claimed to be one of the technologies that will be instrumental in achieving a sustainable world. It should provide the means to support an ever increasing population with an ever increasing need for wealth and comfort. This can, for example, be achieved by an increase in food production efficiency through use of highly efficient GMO crops. Novel medicines & health protection treatments can be engineered into plants in order to achieve cheap and effective health care. The production of greenhouse gases could be combated using bio-fuels from highly efficient GMO energy crops. With this plethora of benefits, it seems inevitable that GMOs will have to have a place in a future, hopefully sustainable, world.

GMOs as a problem However, just as any sharp tool that can be used to build something, wrong use of GMOs might wreak havoc in the societal systems. First and foremost, the ecological systems could be at risks. There is real risk of GMO crops and their out-crosses with wild relatives becoming invasive and weedy. GMO crops also cause "genetic pollution" of nearby fields, making them loose the more and more desired "GMO-fee" status. Further, there are possibilities for direct and indirect effects on beneficial and native organisms. In the most extreme case, ecosystems might get destabilized, with a range of consequences.

Unequity Since Genetic modification is a highly complex technology, only large corporations can afford the necessary investments. It is therefore not surprising that a lot of effort is invested to keep the interests and protect the ownership of technology. Unfortunately, protecting these interests often clashes with the needs and wishes of the people, especially in the third world. Corporations who patent traditional crops, and engineer them so that they can not be re-sawn next year, undermine the existence of poor farming communities and further increasing the income unequity.

Ethics Next to these issues there is the great question of ethics and morality of modifying life. Many people find interfering with Gods creations or messing with life religiously objectionable and morally unacceptable. Such views make Genetic Modification a technology that can not have a place in the lives of those people. However, society at large must come to terms with the advantages and disadvantages of technology and make a decision whether to pursue a certain technological development and how to control it.

Technology acceptance If society is to accept genetic modification, it must also accept its potential risks to human health and the environment. The risks of genetic modification cannot be evaluated before the knowledge of possible impacts is sound, and scientific consensus arises. And the impacts cannot the properly evaluated before a good understanding of the spreading of the "results" of genetic modification is known. Designing a model for the description of such spread is one of the goals of this project.

Qualitative Risk Analysis One of the ways to estimate and measure the ecological aspects of technology is the Quantitative Risk Assessment (QRA) method. QRA attempts to quantify the risks to humans and "nature" in such a way that rational (policy) choices can be made by the society about a technological development. These choices also have a very strong economical side to them.

Acceptance of QRA results If society is to accept the results of a risk analysis process, the predicted impacts must be "believed". In order for the results to be believed the technologies used for the predictions of the impacts must be acceptable [1] as well. Since societal acceptance of a technology is such a important issue, it should be something that it must be taken into account when developing the technology in the first place.

1.3 Agent Based Models

Agent Based paradigm Agent based models are a relatively novel approach to modeling. The main characteristic of ABMs is the fact that they are composed of distinct units, Agents, that have states and are capable of interaction, either with themselves or with other Agents. Agents populate some environment that allows the Agents to interact and adapt. When ABMs are constructed of natural phenomena, the agent boundaries and definition usually overlap with the boundaries of the natural entity that is being modelled. This makes ABMs very well suited for biological applications.

Application to biology ABMs are natural candidates for biological systems. Biological entities have clear boundaries. They ara capable of interaction with themselves, through growth, reproduction and mutation. They are also capable exerting influence on other entities, and even on their environment, as demonstrated by Lovelock [24].

Intuitive ABMs tend to be much easier to understand than equation based models. Their analogy with the real world, where most things appear as distinct units experiencing interaction makes them inituitively close and easy to understand.

For the resons biological suitability and expected ease of understanding ABMs offer an exciting possibiliy to model the risks of GMOs.

1.4 Project

Focus on crops As implicitly presented in the previous section, the focus of this research are GM plant crops. Agriculture not only forms the basis of

---

[1] And accepted

the human food chain but is also one of the most environmentally disruptive human activities.

Lines of thought Because of the complexity of the background questions, it may be useful to present the two lines of thought that are running through the project :

Scientific Can GM be part of a sustainable world? → What kind of thinking is necessary to understand the sustainability of GM → Development of an appropriate model → Model predictions → Conclusions about GM and sustainability

Social GM safety → Public concern about GM safety → Development of Tool to evaluate GM risks → Use of Tool to facilitate public discussion

These two lines of thought are woven throughout the report, and it should be read in the light of them.

Research Questions Following from the background question, two central research questions can be raised :

1. How can Agent Based models be used to acquire insight in the gene flow from 'genetically modified organisms', and can these models eventually be used to estimate the environmental impacts associated with GM?
2. What is the response of the stakeholders to the use of the predictive capabilities of the model in the public debate about the safety and environmental impacts of genetic modification ?

The answer to these two questions will be given through the report. The structure of it is presented below.

Structure of the report

- What is the problem, and why is it a problem ?

  Chapter 1. Presents the background question and the project goals.

  Chapter 2. Discusses the What, Why and How of Genetic Modification and Gene Flow of plants.

  Chapter 3. Discusses the societal perception of the GMO risks and the basics of the Qualitative Risk Analysis

- How is the problem solved ?

  Chapter 4. Presents the basics of Agent Based Modeling

Chapter 5. Description of GeneScape and its implementation

- How good is the solution?

Chapter 6. How does GeneScape look like, how does it behave, and how can it be used through a case.

Chapter 7. The results and discussion of GeneScape.

- What can be learned from it ?

Chapter 8. Conclusions and Recommendations

After presenting the structure of the report, I would like to begin with the introduction of Genetically Modified Organisms.

Chapter 2

Introduction to GMOs

2.1 Genetic Modification: What, Why and How ?

In the following sections I will discuss the basics of Genetic Modification. After discussing the What, Why and How of the technology, I will present the most important advantages and disadvantages.

2.1.1 What is Genetic Modification?

Genetic Modification can be defined as [27]:

- the deletion, changing or moving of genes within an organism, or
- the transfer of genes from one organism to another, or
- the modification of existing genes or the construction of new genes and their incorporation into any organism.

Genetic Modification is a profound alteration of an living organism. The reasons for doing it are presented in the following section.

2.1.2 Why using Genetic Modification?

The simple reason for developing GMOs is the addition of extra properties they have. If we focus on plants, there are roughly 4 types of GMO plants currently being introduced [17]. These are:

Plants resistant to broad-spectrum herbicides This type of plants has genes for the protein that breaks down certain types of herbicide. They can therefore survive an application of the herbicide that destroys normal plants.

Insect-herbivore resistant plants This type of transgene plants expresses a protein toxic to insects. Most commonly the gene for production of the *Bacillus thuringiensis* (Bt) bacterial toxin is introduced. It causes damage to the mid-gut of the *Lepidoptera* genus, to which mainly caterpillars belong, and which results in insect death.

Plants resistant to viral pathogens Resistance to a plant virus can be conferred by a gene coding for the protein coat of the pathogen itself. Expression of low levels of the coat protein prevents disease symptoms from developing, effectively vaccination the plant.

Plants with increased environmental resistance These plants are engineered to have an increased resistance to drought, increased salinity or resistance to high levels of heavy metals in the soil.

2.1.3 How is Genetic Modification done?

Humans have been modifying the characteristics of the plants that they are growing for approximately 20000 years [30]. Techniques as seeding, cultivating, harvesting and storing have been exerting evolutionary pressure upon the domesticated plants. The effect of which is that domesticated plants have remarkably different properties than their wild relatives.

Breeding technologies can be roughly subdivided in three categories. These are:

- Classical selective breeding
- Plant tissue culture
- Genetic engineering (modification)

I would especially like to discuss Genetic Engineering or Genetic Modification, and highlight the aspects that set it apart.

Method Genetic modification is a subtle approach to organism breeding. First, the gene coding for a desired characteristic is isolated from whichever organism carries it. Together with a selection marker (usually a antibiotic resistance) it is inserted in a T plasmid [1]. This plasmid is inserted into a crippled *Agrobacterium tumefaciens*. This bacterium is a plant pathogen, which normally produces tumors in plants. It does this by inserting DNA from the plasmids it carries into the genome of the plant cells. The crippled *Agrobacterium tumefaciens* has the tumor-causing genes removed, but the DNA insertion machinery intact.

---

[1] Plasmid is a circular piece of DNA that can replicate itself

Plant tissue samples are infected with the bacterium, and are grown on a medium both promoting cell growth (growth hormones) and containing a selection agent (antibiotic). The cells that survive and grow contain the DNA insert. These cell cultures are allowed to germinate and are planted to develop a mature plant. This plant, if fertile, can serve as a basis for a new crop.

Advantage The advantage of this technique is that is very efficient, the property of interest is directly introduced, and it allows for a limitless recombination of properties. There are some technical issues involved, but it is practically possible to express any conceivable protein in a plant regardless of the species of origin.

Disadvantages This technique can not be called natural, since it allows for interspecies gene exchange which cannot occur in nature. Also there are many questions about the behavior of selection markers inside the plants. Furthermore, it is so far unknown what the environmental effects of the Gene Flow of the introduced genes are.

2.2 Threats and Benefits of GMOs

The discussion on the advantages and disadvantages of the technique are central to this work, so I will discuss them in greater detail in the next section.

2.2.1 Benefits of GMOs

The advantages of introduction of GMO plants seem quite obvious. However, since the benefits of the technology must be weighed against the possible risks, I would like to discuss them in the following section in some more detail.

Plants resistant to broad-spectrum herbicides Because these plants remove the need for pre-emergence spraying, this type of crops can better be integrated in zero or minimum tillage systems [17],[33]. This reduces soil erosion, improves soil moisture retain and preserves the micro-fauna and flora. When herbicides are eventually applied, the most common resistance conveyed is to glyphosate or glufosinate. These herbicides are less persistent than many others, and thus help reduce the toxic residue in the environment. In the most extreme case, if a weed resistant to a common herbicide has overtaken a field, crops resistant to different herbicide can be planted and thus provides the means for controlling the weed invasion.

Insect-herbivore resistant plants By using plants resistant to their common insect pests, reduction in use of toxic chemical insecticides can be achieved. Next to the effect in reduction in chemicals used, there is a reduction in the impact on non-target insects, which usually suffer from non-specific insecticide usage. Also, by using GMO crops, a more effective control of the pest can be achieved, since the insecticide is expressed in all tissues and at all times. This way no insecticide is lost to leaching, the plant is protected "top to toe", also from for example underground pests which can be very difficult to control.

Plants resistant to viral pathogens Most plant diseases are transported through insect vectors. Making GMOs resistant to viruses a reduction in the use of toxic chemicals to control the insect vectors can be achieved. Concomitantly, there is a reduction in the impact on non target organisms. Further, because plants are vaccinated, this provides an effective control strategy of plant disease. This effect is manly because the population in which the disease can establish is reduced. Vaccination also offers a more enduring resistance for the plant in the co evolutionary "arms race"

Plants with increased environmental resistance This type of GM crops give an increased yield in unfavorable conditions like high salt content, low water availability etc. This is especially important in a world where there is an increasing loss of good quality arable land. It offers great possibilities for food production in Third World countries which often do not have good quality soil. This might be a possible solution to shortage of food by extending the range of arable land to previously unavailable areas. Further, these plans offer possibilities for phytoremediation. Using plants to improve the environment through heavy metal uptake or waste water treatment is environmentally preferred and cheaper than any process-technological means.

2.2.2 Threats from GMOs

As presented in the previous section, GMOs offer great possibilities for improving the environmental impact of agriculture, relieving world hunger and cleaning up the environment. But just as any technology, GMOs are a double edged sword. Next to the possible benefits there is a number of potential problems with GMO crops. According to Wolfenbarger [33], Harding [18] and Altieri [2] the risks of genetically modified plants can be grouped in three main categories:

- Risk of invasiveness, weediness and gene-pollution
- Direct non-target effects on beneficial and native organisms

- New diseases through gene transfer

To examine the risks in more detail, I shall discuss them per type of GMO crop, as presented in the previous section.

Plants resistant to broad-spectrum herbicides One commonly observed effect is the introgression of transgenes into weedy relatives that cause herbicide resistant weeds. These weeds must be treated with more aggressive herbicides". Unfortunately, first cases of "Gene stacking", whereby a weed becomes resistant to multiple herbicides are already observed. The problem is aggravated by "Spray drift". It occurs with application of any herbicide, but since the genes for resistance are present in the field borders due to Gene Flow, it helps establish resistant weed colonies outside the field.

Another problem is caused by the behavior of farmers. Because the crop is resistant to the herbicide farmers see it as an "open invitation to spray" and are encouraged to use "Squeaky clean field" policies. Often farmers do not respect the prescribed herbicide application moments but spray at their convenience. Especially in the US this has lead to an increase in overall herbicide use.

Insect-herbivore resistant plants In case of Bt toxin, evolution of resistant insects does not only render the use of the transgenic plants useless, but also the application of Bt as a surface insecticide as well. The speed of resistance evolution is far greater that the commercialization of new insecticides or transgenes. The battle seems lost even before it has started. Further, there is evidence of impact on non-target organisms, even though in theory only pests should suffer. This impact can be direct through for example pollen ingestion by butterflies. Indirect routes are further up in the food chain, where caterpillars who have ingested non-lethal amounts of Bt toxin are toxic to larvae of beneficial wasps who lay their eggs in them.

Next to these issues, one of the greatest worries is the fact that such a gene conveys a great evolutionary advantage in almost every ecological situation so that the spread of gene is inevitable.

Plants resistant to viral pathogens This issue should be viewed separate from the usual process of vaccination, where a piece of the virus protein coat is used to provoke an immune reaction. In normal vaccination, there is no transport of genetic information.

According to Margulis [25], viruses are an integral part of a ecosystem. Making large parts of the plant population resistant may have unknown effects on the plant survival and fecundity. Little is known about virus role in regulating plant populations in natural communities.

Naturally occurring process of tanscapsidation occurs when a plant virus infecting the host encapsulates a piece of plant DNA. If a plant already possesses a code for a virus protein coat, another virus might enclose it, and thus gain access to the (insect) vectors and hence new plant hosts. Also, recombination is possible. The mRNA of the coat protein might recombine with other plant viruses infecting the host, and thus give rise to novel viruses. The frequency of this happening is unknown.

Also, there are worries of synergism. Synergism takes place when other infecting viruses interact with the transgenic product and produce disease symptoms of greater severity than the pure infection.

Plants with increased environmental resistance The main problem is that plants that have enhanced fitnesses might become invasive and disrupt ecosystems. Furthermore, extreme habitats are often very fragile and contain very specific groups of organisms. If GMOs are able to withstand extreme conditions, it is very likely that agriculture will spread into those habitats and destroy them.

2.3 Gene Flow: What, Why and How ?

As presented in the previous section, widespread use of GMOs has many possible risks attached to them. One of the more complex issues to deal with is Gene Flow. It is responsible for the most poorly understood and potentially most disruptive effects. I would therefore like to examine Gene Flow in far more detail in the next section. In order to present a sense of the importance and occurrence of Gene Flow, please consider the following quote from Ellstrand [15] :

> Domesticated plant taxata cannot be regarded as evolutionary discrete from their wild relatives. Most domesticated plant taxata mate with wild relatives somewhere in the world, and gene flow from crop taxata may have a substantial impact on the evolution of wild populations.

Further, a quote from Stewart [22]

> Fitness-conferring transgenes may have their greatest effect when they are in a host that may persist outside of cultivation, and/or may be spread from an agricultural host to a weedy relative.

2.3.1 What is Gene Flow

Gene Flow can be defined as [15] : *The incorporation of genes into the gene pool of one population from one or more populations.*

Effect of hybridization Gene Flow is a direct effect of hybridization. In the case of plants, related species are able to fertilize each other producing hybrid offspring. Unlike animal hybrids, plant hybrids are often not sterile and may have an increased fitness compared with their parents [15]. This process is one of the most important processes in plant evolution. It is estimated that more than 70 % of plant species may be descended from hybrids [15]. Because of frequent mating between species, genes from one population can and will enter the other population.[2] Form this hybrid the genes will spread into the population through a so called back-cross. Back-cross is a hybrid between the hybrid plants with one of the parent species. This way a gene enters and spreads through the population, altering its genetic makeup.

2.3.2 How does it take place?

In order to properly understand Gene Flow, it is important to discuss the conditions for gene flow. These are summed up below :

- Temporal proximity
- Spatial proximity
- Cross compatibility
- Embryo development & seed viability
- F1 population fertility

I will discuss these factors in the following paragraphs.

Temporal proximity In order for hybridization to take place, both involved species must flower in the same time. Only when both pollen and stigmas are available simultaneously can a flower be fertilized.

Spatial proximity Furthermore, the pollen source and the pollen receiver must be within a distance that the pollen can bridge. This distance can range from a few meters to several thousand kilometers, depending on the species and the environmental factors.

---

[2] The use of term species is somewhat disputed in this context since species are defined as groups of organisms that can not mate with any other species. Since interspecies breeding in plants is more a rule than exception, species concept is highly problematic when discussing plants.

Cross compatibility In order for pollen to germinate and transfer its genetic material to the egg, biochemical compatibility between pollen en the egg must exist. If the pollen grain has successfully germinated and reached the egg deep inside the stigma, it still has to be able to fertilize the egg. Further biochemical incompatibilities can prevent this from happening too.

Embryo development & seed viability If the egg has been fertilized, the embryo has to develop properly. If the genetic distance is too great, this may fail. Furthermore, if seed develops, it must be able to grow into a mature plant.

F1 population fertility Finally, only if the developed plant is not sterile and able to back-cross with one of the parent species can Gene Flow take place.

2.3.3 Why is Gene Flow important?

From the previous paragraphs it would seem that hybridization is an very improbable process. However, it happens very often and at a large scale. It is therefore important to understand its effects. They can roughly be divided into three types. Gene Flow can be beneficial, neutral or detrimental.

Beneficial In a natural situation, Gene Flow is beneficial because it enables a beneficial mutation, or an advantageous trait from one plant species to be introduced into another one. In this case the fitness of the receiving population increases.

Neutral The effect maybe also be neutral. By continually acquiring and removing genes from a population, the genetic makeup of a population may change in time, experiencing the so called genetic drift. This is especially observable in small populations.

Detrimental Gene Flow can also have negative effects on a population. If a very large population of a certain species is present next to small population close buy it is possible that the small population will experience pollen overload. Because of the pollen overload the probability of mating with an other individual of the own species becomes very small, and the small population may effectively become extinct. This effect is called swamping, and is especially pronounced nearby large agricultural monoculture populations.

2.3.4 Gene Flow in crops

After discussing the general types of effect, I will concentrate on the problems arising as an effect of Gene Flow from agricultural applications. There are two main problems. Gene Flow from crops is causing increased weed evolution and is causing the extinction of wild relatives.

Weed evolution Plant species living in the same ecosystem as crops are often regarded as weeds, since they occupy resources like space and nutrients from the ground that would otherwise be used by the crop. Also they lower the quality of the harvested product be contaminating it. Large parts of agriculture are focused at controlling and eradicating these species. These controlling measures, either mechanical, chemical and biological exert a continuous selection pressure on the weeds. If mechanical control is implemented, species evolved into a morphology similar to the crop. If the crop is made resistant to a chemical or biological agent, the selection pressure will select for weeds that have similar properties. Through hybridization, genes responsible for these special properties will Flow into the population, rendering the controlling method useless. There are many examples of this [15]. The most vivid one is the example of rice with a red pigment developed in India, that aids the manual weeding of similar, non edible grasses cohabiting rice paddies. After several generation the grasses acquired the red pigment, and could not be distinguished from crop rice anymore.

Extinction of wild relatives Because of the scape of application of modern monoculture crops, there is a real danger of pollen overload. Wild species do not have the chance to mate with their won species because of the sheer numbers of pollen available form the crops. This way the wild population incorporates more and more genetic material from the crop, loosing its identity. An example of species extinction is the case Taiwanese rice *Oryza rufipogon ssp. formosana* and another subspecies *Oryza nivara* that have become exceedingly rare because of gene introgression form domesticated rice *Oryza sativa*. This effect has been observed throughout Asia.

Chapter 3

Introduction to Risk Perception and QRA

3.1 Public Risk Perception & GMOs

After having discussed the risks and benefits of GMOs, it is important to understand how these risks are perceived by the society and how the public opinion is formed. Public opinion is crucialy important since we live in a predominantly democratic world, were public opinion eventually translates to political and legislative action. Whether this actually leads to effective and good policy is doubtful, Cross [9], but an elaboration of this doubt falls outside the the scope of this research. Background theory about risk perception is presented in Appendix A.

3.1.1 Issues and background

Considering the current public discussion on GMOs, it is interesting to note that there are several issues that dominate the debate. These are presented below:

- There is extensive focus on human health risk from GM food, while the scientific expectation is that this risk is negligible
- Ecological impacts are largely neglected, while the scientific expectations are that they are very considerable
- Economic aspects are seen as "evil" and inhumane, especially the interaction between the life science multi nationals and Third World farmers.
- After the initial outcry, there is little attention of ethical, philosophical and religious issues. If these discussions do take place, they are mostly in the realm of the ethics of human cloning.

- There is an almost hysterical public reaction on any food that is suspected to contain GMOs or their derivatives.
- There is a large international disagreement on how the issue should be addressed, especially between the US and Europe. The discussion is a Precautionary Principle vs. Adaptive Management one.

Backgrounds to Objections Next to examining the issues in the public discussion, it is important to understand what is the background of the objections public has against genetic modifications. These backgrounds have been presented by Evenhuis [16] :

- Not all possible hazards are known.
- Not all effects of known hazards can be predicted.
- Ethical/religious dilemma: Is man allowed to improve upon nature?
- The issue of lack of justness or equity of the distribution of the risks and rewards of the GM technology is not addressed by Evenhuis as a possible cause. I would suggest that it does belong to the backgrounds.

3.1.2 Perception of GMOs

As presented in the previous section, virtually all public discussion of GMOs is focused on their application as food, and food safety issues. While the topic of this research is the sustainability of GMOs and their environmental impacts in particular, it is very useful to understand the discussion on food, since it may offer a understanding on how the environmental discussion might develop in the future.

Applicable semantic image Of the semantic images suggested by Renn [31], the one of the Pandora's box (see Appendix A) is the most applicable one. The risk is seen as invisible threat to one's health. The effects are delayed, and very few people are affected at the same time. There is no personal knowledge about the risk. All of the information about the risk is received from external sources.

Information vs. experience External source of information in itself would not be problematic if the supply was objective and value-free. However, the opponents of the technology, mainly Non Governmental Organizations (NGOs), use the still relatively great trust the public has in them to "play" the attention and selection filters of the public through media techniques. The proponents of the technology, industry and sometimes governments attempt to use media in the same way, but since the trust public has in them is minimal, the achieved effect is also minimal. Since in (ideal) democracies public opinion determines policy, continuing negative perception of risks a technology will lead to policy limiting or maybe even outlawing the use of the technology. This can have very far-reaching consequences.

Inherent bias However, Cross argues [9] that risk management policy that is based on public values and opinion win necessarily be skewed towards to needs and want of the most vocal group. This, especially in the US, means the white, middle aged, middle class suburban population. In effect, Cross further argues, the risks will be un-equitably distributed on the minorities with less voice or political weight. Since, management potential and funds are limited, this will lead to suboptimal risk management, which will end up saving less lives that it could have.

Perception of GMOs In the case of GMOs, and especially food, we see that it could be quite difficult to imagine a problem that would have more negative perception. There is very little personal control involved in avoiding ingestion, because (especially in Europe) consumers do not trust the institutions to provide them with information about which food is non-GMO and which not. The risk in non-voluntary, since everybody must eat, and the familiarity is minimal. The equity or risk division is non-favorable, since richer people can afford ”good food, and the poor must buy what they can afford. The risk is purely artificial (GMOs are perceived to be ”unnatural”) and the possible effects are dreaded.

Concluding remarks From the previous, it is made clear that public perception of a technology is crucial for acceptance and eventually creation of policy to manage it. Therefore, a method for objectively examining and managing the risks of GMOs that is acceptable to the public is necessary to manage the introduction and application of such technology. One of the tools available is the Qualitative Risk Assessment. I will discuss it in the following section.

3.2 QRA and GMOs

As presented in the previous section, there is a need for an objective Risk Assessment describing the risks of GMOs. One of the available tools is the so called Quantitative Risk Assessment (QRA). In the following section I will describe its basic steps and discuss its inability to deal with life processes.

3.2.1 QRA vs. Life

The basic parts of a traditional Quantitative Risk Assessment are presented below :

- Risk Identification: What, Where, How
- Emissions(Liquid, Gas, Solid, Radiation)
- Exposure (Human, Environmental)
- Exposure-Risk relation
- Number of People exposed / Degree of environmental exposure
- Risk Calculation
- Risk Presentation

By methodically examining the different stages once can come to an objective, quantitative estimate of the risk. This would be highly desirable for GMOs. If a clear, objective and quantitative risk estimate can be given for application of GM techniques, it would aid the public discussion and possibly help answer the question of sustainability of GMOs. However, if one attempts to apply the method to living organisms, one comes across unsurmountable problems. These are discussed in greater detail below.

Why QRA does not work for GMOs QRA as is not able to estimate and predict the risks of a release of a GMOs into the environment. The most important reason for that is the fact that the:

Law of conservation of mass does not apply to genes, since they are information and not matter

It is obvious that by information genetic information is meant, and therefore Life, Dawkins [12].

There are several essential differences between matter and information that are relevant for this discussion. These are summed up below.

- Information can be multiplied
- Information can disappear
- Information can not be diluted
- Information is context sensitive

These point might seem very trivial, but they deserve a closer examination. I shall discuss them in the following paragraphs.

Multiplication One essential aspect of information is that it can, theoretically, be reproduced unlimited number of times. Information does not degrade upon being copied, and the copies have all the properties of the original. This is obviously not a property matter possess. Furthermore, genetic information, as contained by living being, has the capacity to multiply itself. Without environmental constraints the number of copies grows exponentially.

Disappearance Matter can not dissappear. If one has a piece of candy in a drawer it will still be there some days later [1]. An exception to this is a nuclear reaction, but this is not relevant for life processes. Information however, can disappear if no copies exist and the carrier of the original piece of information is destroyed. A species can, and often does, get extinct.

Dilution An old environmental "wisdom" states that "the best solution to pollution is dilution". A substance emitted by human activity in the environment is considered a pollutant it its concentration in the environment is higher that the naturally occurring level. Any toxic or disruptive effect of a substance can be mitigated by simply diluting it to non-toxic levels. Information can not be diluted. Only one copy is necessary to have an effect. An excellent example is the Polymerase Chain Reaction (PCR) technique, which can detect and amplify the information in a DNA molecule even if only a single molecule is present.

Context A gene only exists through its context it is in. The information contained in it will have dramatically different expression forms depending on the host organism. That means that it very much depends on whether it is placed in a soil bacterium or in a human being or in a plant. One famous example is a gene causing a color pattern on the forehand of a certain species of fish that causes a fatal cancer when engineered into a related species.

3.2.2 What should a QRA of GMOs look like?

From the discussion above, it is clear that the traditional QRA is not equipped to deal with risk from life processes. The Handbook of Environmental Risk Assessment and Management [5] presents the following areas of research needed for development of a good Risk Assessment for GMOs :

- Methods for detection, identification and enumeration of released GMOs.
- Methods for determination of the fate and transport of GMOs.

---

[1] Even though the law of conservation of matter does not always seem to hold true for candy

- Assessment of genetic stability, i.e. propensity for gene transfer.
- Hazard Assessment (pathogenicity to non-target species, including humans and disruption of environmental processes)
- Means of mitigation of possible effects

By specifying the above to the spread of genes from plants, if gene transfer by interspecific crossing is to be included in a QRA, it must deal with the following issues, Evenhuis [16] :

- Emission, dispersal and deposition of transgenic pollen.
- Introgression of the transgene into the wild species
- Stabilization and spread of the transgene in that wild species.
- Ecological effects of the transgene in the new host population.

Model design It is obvious that extending the QRA with genetic information and living organism is not a trivial matter. I would therefore like to contribute with the first step in the risk identification. Creating a model that describes the spread of genes from crop plants is the first step in improving the QRA.

Chapter 4

Introduction to ABMs

Agent Based Models are the heart of this project. It is important to discuss what are Agents, how they are used and with which tools they are developed. These issues will be discussed in the following section.

4.1 Agents

As a first step in discussing the field of Agent Based Modeling, it is important to define what is meant by an Agent. Definition is presented by Jennings [21]:

> An agent is an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives

For clear understanding of the definition, several points must be further elaborated [21]. Agents are:

- clearly identifiable "problem-solving" entities with well defined boundaries and interfaces.
- situated (embedded) in a particular environment; they receive inputs related to the state of their environment and they act on the environment through effectors.
- designed to fulfill a specific purpose; they have particular objectives (goals) to achieve.
- autonomous; they have control both over their internal state and over their behavior.
- capable of exhibiting flexible problem solving behavior in pursuit of their design objectives; they need to be both reactive (able to respond

in timely fashion to changes that occur in their environment) and active (able to act in anticipation of future goals).

A collection of such Agents that represent some part of reality is called an Agent Based Model.

Agent Based Modeling According to Parker [28] the most compelling argument for Agent Based Modeling can be summed up as follows: *Why don't we model it as it is in the real world?* It basically means that ABMs are useful because they offer us a possibility to create models that do away with generalizations and allow us to explore the world through discrete objects and their interactions. By modeling known interactions between natural Agents we can explore different aspects of the world. This approach reduces the amount of information that is necessary to describe a system, thus simplifying analysis.

4.2 Java

As presented above, Agents are software implementations. The choice of a software environment (also referred to as programming language) is not trivial, since every programming language is laden with hidden structural and relational assumptions, just as human languages are [1]. These assumptions work through in the implementation and may force a certain approach to implementation of the model. For example, the LISP programming language is a based on the conception that everything is a list. If an agent based model would be based on such a language, it would look very differently than if it was developed in Java, which is purely Object Oriented.

What is Java Java is one of the most modern programming languages that is developed by the Sun corporation [7]. Developers of Java kept pure Object Orientedness as a guiding principle. Java is developed in the era of Internet, and is made completely platform independent. This makes online execution and code transfer between systems and platforms easy.

All of these issues, together with an existing framework for development of Agent Based Models called Ascape, made Java the language of choice.

Objects The most important aspect of Java is that it is a Object Oriented Programming language (OOP). Therefore, it is essential to understand what Objects are, and how they influence the thinking about Agent Based modeling. All Objects have 5 key properties [14]. These properties are what sep-

---

[1] For example, some philosophers argue that it is impossible to discuss the thinking of Nietzsche in a non Germanic language like French, since it misses the key concepts he uses

arate Object oriented programming apart from the linear based approach. These properties are :

- Everything is an Object
- Objects can communicate
- Object have a state
- Every object has a type
- Communication is standardized within a type

Understanding of these aspects is basic to understanding how Java works. Since Java is the framework for the model, understanding these basic concepts will clarify the implicit, language inherent properties that Agents developed under Java will have. Since not all readers might be interested in these issues, I will present them in Appendix C

4.3 Ascape

Ascape is a collection of Java classes developed by Miles T. Parker at the Brookings Institute [28]. In Java terminology, Class is a collection of data, procedures and interfaces bundled in a Object.

Ascape is full modeling environment for development, analysis and distribution of Agent Based Models. The design goals include abstraction and generalization of key agent modeling concepts, ease of use, configurability and performance [28]. In the following paragraphs I will discuss the specificities of Ascape.

Abstractions Ascape relies heavily on three abstractions that determine model structure and layout. These are :

- All scapes are themselves agents
- Scape structure is hidden from agents
- Behavior occurs across scapes, as ”rules”

Because these abstractions lay at the hart of Ascape. They are of key importance when thinking about Agents Based Modeling and implementation. I shall therefore examine them in more detail.

All scapes are themselves agents This abstraction could read "Everything belongs to something, even the something". By requiring that all scapes are Agents, it is assured that all possible elements of the model are defined within the ABM paradigm. Basically, everything is an Agent. Effect of the abstraction is that it requires all the boundaries between Agents to be explicitly defined and must carry the same privileges. This means that the model does not carry any implicit assumptions except the strictly hierarchical defined ones [2]. The effect of this is that the model becomes strictly hierarchical and highly composable, where each system can impose any topology (form) on its constituent subsystem, and no "hidden" links exist which could create an undesired interaction.

Since all scapes (Agents) belong to the root Scape, all system behaviors (such as statistic collection, iteration rules, visualization) are managed systematically.

Scape structure is hidden from agents This abstraction means that agents do not directly interact with the "space" they exist in. Theoretically, they do not have means of directly accessing information about their surroundings. The advantage is that agents use the information that the upper level scapes provide them when this information is necessary. In other words, if an agent in a grid needs to know whether a certain resource is in the neighborhood, it will ask the grid for the information, and not look itself. It allows the agents to be "space independent".

Also, this abstraction makes sure that all the algorithms are placed in scapes they belong to. This creates logical transparency and reduces the amount of code necessary, and keeps it at one place, making modifications highly effective.

Behavior occurs across scapes, as rules In an Ascape model, Scape hierarchy is structurally determined. Agents are organized in scapes because they share the same type of behavior. Since scapes are hierarchically organized, their behavior follows the same order. Therefore, the behavior of Agents is executed collectively across a Scape. If necessary, rules can be applied at a single Agent. The effect of this is that subsystems can have their own independent rules for behavior subjected to the higher level ones.

---

[2] i.e. which Scape is whose subsystem.

Chapter 5

Model Development and Implementation

In the following chapter I shall discuss the development of GeneScape, the Agent Based Model of Gene Flow. I will present the model layout, its components, the model implementation, the in- and output variables and several interesting aspects.

5.1 Model Layout

GeneScape consists of three levels of hierarchy. The World, the Field and the Plant. World, the top level, contains the Field, which in turn contains the Plant(s). The conceptual model layout is presented in figure 5.1.

World The World is the top-most layer of the model. It provides the basic deffintions , for example what it means to be fit, how does the effector (pollen) behave etc. The World also contains the Field that Plants live on.

Field The Field is a holder for the Plants. It provides spatial discretization to the model. Through the Fiels specific individual interaction on a pseudo-spatial scale are possible. Thus, the Field, together with the World rules limits the geometry of Plant interaction. It is trough the lattice the World fitness rules are enforced.

Plant The Plant, as the "inhabitant" of the Field had no notion of space, or fitness. The Plant only has a state i.e. a genome, and is capable of interaction (i.e. mating) with other Plants. The interaction is executed through the effector (Pollen) and the memory (Egg). Pollen is capable of Mendelian permutation of the state (content) and is capable of influencing other Plants at a distance allowed by the World (interface). The Plant experiences the influence through it's selective in port (Egg). The Egg also

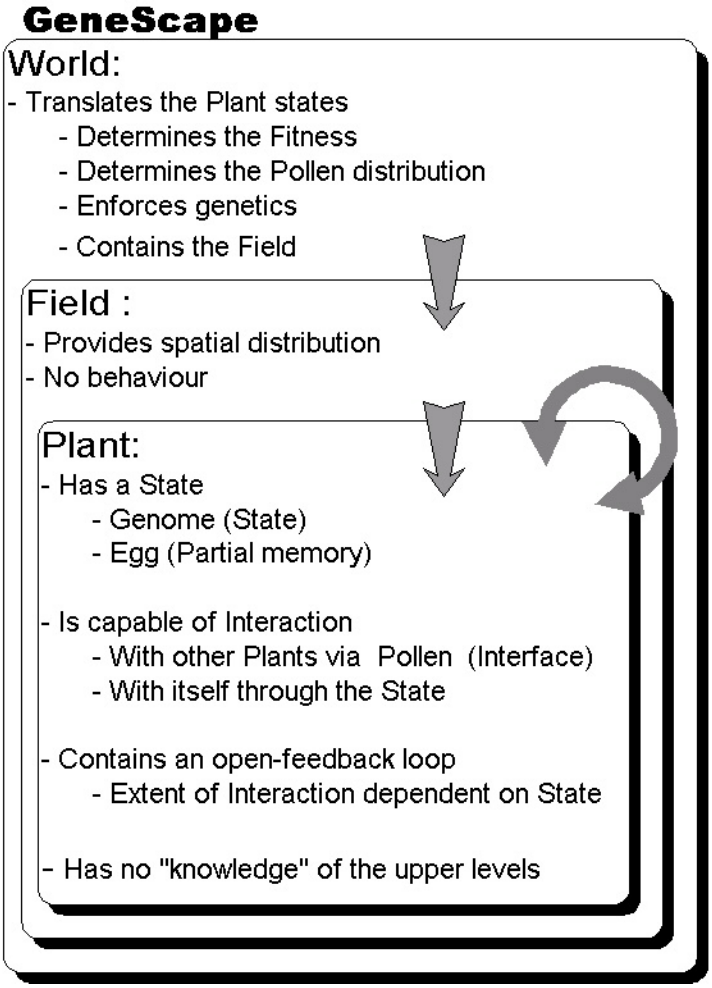


Figure 5.1: Conceptual model layout

carries a Mendelian permutation of the state, and has ”preferences” for accepting Pollen, depending on the amount and the kind of it.

Model behavior Model is iterated in annual cycles. Every year the fitness of a plant is evaluated, it is assigned a certain ”strength” or amount of pollen. This pollen is distributed, and the Egg scores all the pollen that could be falling on it for attractiveness and distance. It takes the most attractive one and combines the genomes. With that a new state (Plant) is created, and the process is iterated.

5.2 Model components

In the following section I will present the different components GeneScape consists of.

5.2.1 Model world

The situation being modeled is that of a GMO Crop field planted in a middle of a ”natural” environment. The natural environment consists of a number of more or less related plants Each year the same type of GMO crop is planted, while the ”nature” grows and reproduces without direct interference.

Space and Distance All the distances in GeneScaoe are expressed in the number of neighbors. Agents (plants) have a structural ordering in a grid, and a neighbor is seen as being a number of plants away. The field the plants live in is a square grid with each Plant experiencing a so called ” Moore” neighborhood. Figure 5.2 represents the Moore neighbors (x) with the distance of two Agents from the origin (O). The dots (.) denote the grid points outside the neighborhood.

```
.........
..xxxxx..
..xxxxx..
..xxOxx..
..xxxxx..
..xxxxx..
.........
```

Figure 5.2: Moore Neighborhood model of space

5.2.2 Species

There are 4 different species of plants in GeneScape. These are:

- GMO plant (G)
- Wild relative (W)
- Remote relative (R)
- Non-relative (N)

The GMO plant and the Wild relative are very closely related, and the main difference is that the GMO is a domesticated subspecies that has a fitness increasing gene inserted into it. The Remote relative is as the name says, remotely related to the GMO crop. It is somewhat closer related to the Wild relative. The fourth species is the Non-relative, which is completely unrelated to any of the previous species.

5.2.3 Genome

The following assumptions are made about the genomes of the plants :

- Plants are diploid, pollen, and egg are haploid.
- The genome consists of 8 genes:
    - The species determination genes
    - The GM gene G
    - Fitness gene A
    - Fitness gene B
    - Species Gene can have the GMO, W, R or N alleles.
- Genes are binary coded: 1 (gene present); 0 (gene absent).

I will discuss these in more detail below.

Genome as a data string Genome consists of two strings of information which have a compounded effect on the plant. Genome is no more than a binary string with a meaning attached to each position. The World later translates this string to a behavior of the plant.

Both strings are identical in structure, but can differ in the information content. The genome is laid out sequentially, that is, the first diploid "gene" consists of two bits, that are the first bits of the haploid strings.

Position determines the meaning The meaning of the haploid genome positions is as follows: GMO-W-R-N-G-A-B. Each position can be a 1 or 0, depending on whether the gene is present or not, so a Pollen or Egg containing the genome 0-1-0-0-0-1-1 is a Wild relative that contains the fitness genes A and B. A complete Plant consists of a diploid genome with the meaning attached in the following way: GMO-GMO-W-W-R-R-N-N-G-G-A-A-B-B. So a 1-1-0-0-0-0-0-0-1-1-0-0-0-1 genome is a plant of GMO crop species that contains a full set of GMO genes and half of the fitness gene B.

Determination of species It is important to note that the positions GMO-W-R-N are considered as one gene that determines the species. Having two different species genes is not possible. This part of the genome abstracts the "rest" of the genome, and is therefore responsible for the differences between the plants. That is the reason mating preferences are based on the species of a plant.

Low level operations This representation is chosen because this simplifies the software operations on the genome, which is just a string of 1's and 0's. It also has a strong biological analogy. DNA is just a raw string of data which can be manipulated with the DNA machinery independent of the meaning. DNA replication, division and translation are low level operations on the genome, and are independent of the information content.

5.2.4 Fitness

The fitness of the plant depends on fitness determining part of the genome. The species does not influence the fitness.

- G gene gives a fitness of 15
- A gene gives a fitness of 10
- B gene gives a fitness of 5 [1]
- Fitness is cumulative
- The most fit plant is the one with genome x.11.11.11 ($2 \times 15 + 2 \times 10 + 2 \times 5 = 60$)
- The weakest plant in nature, x.00.00.00
- The fitness is directly related to the number of pollen that the plant makes

The fitness is a sum of total contributions of the fitness genes. This fitness effect of the genes G, A and B is an addition to the basic fitness of a plant that is set to 0. Thus, plants that would contain a fitness reducing gene would have a negative fitness compared to the reference, and vice versa.

Power to reproduce Fitness is considered as an abstract "power to reproduce". The rationale for this approach is the following: If fitness is the measure of the degree of adaptedness to the environment [10], it has directly an effect on the survival chances of that individual. Survival of an individual is only relevant for reproduction [12], thus the fitness is a measure of how many offspring one individual is able to produce. The model makes a slight simplification, since it only relates fitness to the number of pollen, and not to the actual number of individuals spawned, but this does not essentially change the principle.

---

[1] Plant fitnesses, and their rations are important model parameters. They are can be estimated from literature, see [15]

5.2.5 Pollen spread

Pollen spreads through the neighborhood up to a certain distance. This distance is defined in numbers of neighbors, not physical meters. GeneScape contains two rudimentary spread models. First model is a simple distance limit. Pollen is able to reach a certain number of neighbors, and after that distance its availability is 0. It is a sort of homogeneous cloud of pollen that spreads to a distance of x. Alternatively, pollen availability falls inversely with distance, until the maximum spread distance, when it becomes 0. Choice can be made between the two pollen spread models.

Egg horizon In GeneScape, there is no actual pollen movement. Instead, the pollen spread distance is implemented as an Egg horizon, or Egg "pollen experience". It means that Egg evaluates its environment, and "asks" the number of available pollen from each of its neighbors, up to the maximum distance, and than calculates the availability according to the spread model.

5.2.6 Mating

Mating is the actual mechanism of genetic exchange between plant agents. Mating means that the Egg takes the haploid genome from the pollen of the chosen neighbor, and combines it to create a new plant on the position where the mother plant resided.

Mate selection The most suitable neighbor is the one that has the greatest Attractiveness. Attractiveness is a product of the number of available pollen (through the spread model) and the Preference for mating between the species of the Egg and the species of the Pollen. Preference is expressed in a scale of 0 to 10. A Preference of 0 means that the mating between these species is impossible. Tha same species has a mating Preference of 10. These factors should be considered in relation to each other. They are not absolute values, but expressions of relative preferences.

Species Conservation It is important to note that the species of the mother plant is conserved by the Egg. The Pollen only contributes the fitness genes.

5.2.7 Behavior rules

GeneScape assumes that the plants are annual, and that there are no parent plants living together with the offspring. The life cycle of a plant (germination, maturing, mating, seed production and death) is abstracted in the following steps :

1. The GMO plant(s) are placed in the middle of the lattice.

2. The rest of the Lattice is filled at random with the other 3 species with random genomes.
3. The fitness of the plant is evaluated, and the number of pollen produced determined according the fitness rule.
4. The number of pollen produced is dependent on the plants fitness.
5. Plant is replaced by an Egg that is picked at random from the pool of possible haploid genomes that belong to the species the parent Plant belonged to.
6. The Egg evaluates the number of pollen from each species, multiplies it with the mating preference for that species, and together with the pollen that gets the highest score it forms a diploid Plant with a combined genome.
7. The fitness of the Plant is evaluated, and the number of allowed pollen determined according the fitness rule.
8. Process is iterated from step 3.

Rules In GeneScape, as discussed in the section on Ascape, all behavior occurs through Rules, of which the Agents have no notion. Behavior is defined by 4 rules, which are a summed up abstraction of a life cycle of a plant. These Rules are:

Planting GMOs rule Request a Plant agent to "Plant" itself if it is a GMO, replacing the plants on the GMO field with a new one.

Growth rule Request a Plant to perform the determination of the number of pollen available to it.

Pollen and Egg production rule Request a Plant to produce a Pollen and Egg genome.

Mating Rule Ask Egg to find almost suitable neighbor and exchange the genetic information with it, creating a new plant genome.

5.3 Model Implementation

Model implementation is the actual translation of the model layout and components into Java code. The code can be found in Appendix I.

Pseudo-code The code is inacessible to anybody without the knowledge of Java/Ascape. To help the interested reader in understanding the structure, I have translated the Java code into a "Pseudo-code". It is meant to present the logic and the buildup of the model code. It is presented in the Appendix H

5.4 Model Parameters

In the following section I will present the model parameters. Understanding these parameters is essential if the model is to be an effective environment for doing thought experiments about Gene Flow. All these parameters are World parameters, and are not "known" by the Plant, but are of course accessible to the user to adapt. They are presented in table 5.1 together with the default settings. The table can be found on the following page.

Parameter	Default value
Pollen Determination Model Used	0
Fitness To Pollen Factor	1
Pollen Spread Model Used	0
Relatedness Case	6
Field Height	60
Field Width	60
Field Layout	0
GMO Field Height	6
GMO Field Width	6
Population Structure Case	0
Fitness Of Gene A	15
Fitness Of Gene B	5
Fitness Of Gene GMO	10
Pollen Spread Distance	5
Mutation Rate	0
Preference GMO And GMO Mating	10
Preference GMO And Non Mating	0
Preference GMO And Remote Mating	1
Preference GMO And Wild Mating	8
Preference Non And GMO Mating	0
Preference Non And Non Mating	10
Preference Non And Remote Mating	0
Preference Non And Wild Mating	0
Preference Remote And GMO Mating	1
Preference Remote And Non Mating	0
Preference Remote And Remote Mating	10
Preference Remote And Wild Mating	2
Preference Wild And GMO Mating	8
Preference Wild And Non Mating	0
Preference Wild And Remote Mating	2
Preference Wild And Wild Mating	10

Table 5.1: GeneScape parameters and their default values

Most of the parameters are self explanatory. The descriptions and the rationale of the default value of each parameter are presented in Appendix D.

5.5 Input variables

All of the parametres can be used as Input Variables, depending what kind of experiment one wishes to do with the model. The ones selected for the evaluation in this project are presented below:

- Fitness increase by the GMO gene (referred to as Fitness)
- Pollen spread distance
- Relatedness level (Mating Preference XY)

The reasons for selecting these variables are presented below.

GMO gene fitness only The Fitnesses of Gene A and B are kept as parameters. The interesting issue to examine is the fitness increase by the extra gene, and not the value of the total fitness. We can assume that plants in a natural environment all have roughly the similar base fitness, and that only the added gene makes a difference between GMOs and wild-type. Since the effect of the GMO gene is strongly dependent on the environment it is placed in, it is interesting to vary the value of the added fitness in order to examine the effect on Gene Flow. The GMO gene fitness is therefore taken as a variable.

Spread distance Pollen spread distance can be vary very widely in nature, depending on the situation. This parameter is crucial in determining Gene Flow. Its is therefore taken as an input variable in order to examine its effect on the model.

Relatedness level Next to the Fitness of the GMO gene and Pollen spread diatance determines the Relatedness between plants the extent of Gene Flow. The Relatedness of the Crop to its plant surroundings varies widely depending where it is placed. It is therefore important to examine its effect. Relatednes level bundles the Mating preferences into a coherent biological set. These clusters are further described in tables 6.2 and E.1.

5.6 Output variables

Before I discuss the output variables, it is important to determine what is exactly the quantity we want to measure.

Background question translated to measurements The background question was about the sustainability of GMOs. Considering sustainability and Gene Flow, the following question should be answered : *What is the effect of Gene Flow from GMO crop application on the genetic composition of the surrounding populations?*

Answering the question The obvious answer to the question posed it to look at the image of the Field that is the primary GeneScape output. However, it is not a quantitative measure, and is therefore not useful for a scientific evaluation. However, next to the visual result there are two other relevant metrics available to measure the effect of Gene Flow. These are the:

1. Number of Plants Containing the GMO Gene
2. Average Population Fitness

I will discuss these metrics and their application in the following subsections.

5.6.1 Number of Plants with GMO gene

The first relevant metric in GeneScape that describes the effect as stated above is the Number Of Plants Containing the GMO Gene (NPCGG). By this I understand the plants that have at least one copy of the GMO gene in their genome.

Significant values There are three significant values for the NPCGG.

36 The default GMO Field size is 6x6. Thus, initialy the number of plants with the GMO gene is 36.

1200 There are 3600 plants in the Field, and the three species are equally distributed. NPCGG around 1200 represents the situation in which the gene has spread into only one of the populations completely.

2400 NPCGG around 2400 represents the spread into two population. These numbers are only approximate, since the exact numbers of each species are randomly determined, and can fluctuate slightly.

3600 3600 represents the situation in which all Plants have acquired the gene.

5.6.2 Effect of Gene Flow

Average population fitness Second important output variable is the Average Population Fitness (APF). From this the degree of effect of Gene Flow on the population can be estimated. However, APF varies with the value of the Fitness variables. Therefore, it must be made dimensionless, so that the relative effect of Gene Flow at a certain Fitness level can be determined.

Effect of Gene Flow The effect is expressed in equation 5.1. Let Fit_{Max} be the maximum increase in the average population fitness if two copies of the gene would be present in each Plant in the model. Also, let Fit_{Min} be the average fitness the population will achieve if all the other genes have spread through the population except the GMO gene. APF_t is the measured Average Population Fitness after t generations. The effect E is then calculated as follows:

$$E_t = \frac{APF_t - Fit_{Min}}{Fit_{Max} - Fit_{Min}} \tag{5.1}$$

Significant values Equation 5.1 can have a number of significant values providing insight in the extent of spread.

Negative values A value that is negative represents a state in which the population did not have the chance to spread the non-GMO genes to the base level. The more negative the number, the less equally the populations genes are spread through the population.

0 Value of 0 represents the situation at which there was sufficient time to spread all the genes between the wild populations, but there was no escape of the GMO gene. At this situation all Plants have both two copies of Gene A and B, but no copies of the GMO gene.

1 A value of 1 represents the situation at which all the genes have been equated through the population, and the GMO gene has entered the population to the full extent, that is, each plant has two copies of the GMO gene, next to two copies of both gene A and B.

Chapter 6

Model Evaluation

In the following chapter I will discuss the method for evaluating GeneScape. The approach to the evaluation of model appearance, model behavior and use will be presented. I will beginning by presenting the case study.

6.1 Introduction to model evaluation

Evaluation of the model should answer three basic questions.

1. What does the model look like ?
2. How does the model behave ?
3. How can and should the model be used ?

Each of these questions has a range of subquestions and aspects. These will be discussed in the following sections.

Model appearance Model appearance can be evaluated in a very straightforward fashion. One only has to look at the model interface and graphical output to evaluate it. This aspect will therefore not be discussed in the Evaluation section, but will be presented in chapter 7, Results.

Answers The answers to these three questions should help determine whether the model is scientifically valid, whether it is useful and whether it could be a socially acceptable part of a new approach to estimating the risks and sustainability of Genetic Modification of crops. These answers will be presented in chapter 7, Results.

In the following sections the model behavior and model use evaluation methods will be presented.

6.2 Model behavior evaluation

In this section the second question posed in the introduction will be answered: How does the model behave? First, different aspects of the analysis will be discussed. Subsequently the methods of the analysis will be discussed.

6.2.1 Aspects of analysis

When analyzing the behavior of the model, there are three aspects to be examined: Internal consistency, Reproducibility and the Rrange of possible Behaviors.

Reproducibility GeneScape contains a strong stochastic element. The position of plants, their species and their initial genome are randomly determined. These parameters will not be made variable within this project. When examining the behavior of the model as a result of change of other parameters, it is important to know whether the randomness of these parameters will affect the outcome.

Range of behaviors In order to use GeneScape as a prediction tool, it is interesting to test the model across the parameter space, to evaluate the range of possible behavior, and possibly use that to predict what would the system do.

Internal consistency The goal of the internal consistency evaluation is to check whether the theory that is intended to govern model behavior is correctly implemented. To test this, model behavior will be examined in extreme parameter settings, where the outcomes can be predicted by biological theory.

6.2.2 Analysis method

In the following subsection I shall describe the methods that shall be used to evaluate the aspects presented above. The results of these evaluations are presented in chapter 7, Results.

6.2.2.1 Reproducibility

If we assume that the model outcomes are normally distributed, than it is possible to test the hypothesis that the model is behaving consistently, by repeating one experiment a large number of times and examining the results statistically.

Hypothesis By estimating the average Average Population Fitness (APF) and the standard deviation of the APF we can test the hypothesis that a group of experiments has the same average value as another group. If the hypothesis is correct, it can be assumed that the model behaves consistently. In formals terms : $H_0 : \mu_1 = \mu_2$ *and* $H_1 : \mu_1 \neq \mu_2$. The test parameter used is $D = \bar{X}_1 - \bar{X}_2$. Because the variance of D is unknown, it will be estimated using equation 6.1.

$$S_D = \sqrt{\frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{(n_1 + n_2 - 2)}(\frac{1}{n_1} - \frac{1}{n_2})} \qquad (6.1)$$

To test the hypothesis with 95% confidence, we use the so called *t-test for the difference in averages* [32]. The H_0 hypothesis is rejected if the condition presented in equation 6.2 is met.

$$D \leq -t_{0.05/2}(n_1 + n_2 - 2)S_D \quad \cap \quad D \geq t_{0.05/2}(n_1 + n_2 - 2)S_D \qquad (6.2)$$

The same test will be executed on both the NPCGG and APF metric.

Parameters The test will be executed with the default set of parameters, as presented in section 5.4, in table 5.1.

Number of generations The model is run for 7 generations. This number is chosen because at this time not all plants have acquired the GMO gene. Considering that the steady state is reached at 10 generations, measuring the output variables at higher times would re measure the same steady state. By measuring earlier variations in behavior can be traced.

6.2.2.2 Range of behaviors

The model will be examined across the input variable space, to estimate how does the behavior change and to provide pointers to which combination of the key parameters is particularly able to cause spread. The input-variable space is presented in the table 6.1. The model is run at each combination of the variables. The output variables are recorder after 20 generations.

Fitness	Spread distance	Relatedness Level
1 to 20	1 to 20	I to V

Table 6.1: The Variable Space

The Relatedness Levels are defined in table 6.2

The values of the Mating Preferences for each Relatedness level and the parameter setup for the Range of Behaviors evaluation are presented in Appendix E.

Case number	Meaning
I	No relatedness between the different species
II	Low level of relatedness
III	Medium level of relatedness
IV	High level of relatedness
V	Effectively one species

Table 6.2: The Relatedness cases defined

6.2.2.3 Internal consistency

Internal consistency will be evaluated at extreme sets of variables. These extreme situations and their predicted outcomes are presented in table 6.3. The descriptions and the parameters of the test runs are presented in Appendix E, table E.3. The outcomes of the internal consistency analysis will be evaluated in the same manner as the range of behavior analysis.

Test	Situation	Expected behavior
1	Zero level relatedness between species, average spread distance , Average GMO Gene fitness	No Gene Flow
2	Zero pollen spread distance, Average relatedness and GMO gene fitness	No Gene Flow
3	Maximum relatedness, No fitness increase by GMO gene, Average pollen spread distance	Diffusive Gene Flow
4	Average level of relatedness, Average pollen spread distance, Extreme GMO gene fitness	Extreme extent of Gene Flow
5	Average relatedness, No fitness increase by GMO gene, Average pollen spread distance	Limited amount of Gene Flow

Table 6.3: Consistency tests and their predicted outcomes

6.3 Case study: *Brassica napus L.*

In this section the model use is evaluated through a case study of Canola, *Brassica napus L.*. Different aspects of the evaluation are discussed and the analysis methods are described. But first, the case study is described.

6.3.1 *Brassica napus L.*

Brassica family encompasses a large number of wild plants and vegetables grown in Europe. All rapes, cabbages, cauliflowers, Brussels sprouts, broccoli and mustard species, common in Europe, belong to it, as well as a large number of wild species. *Brassica napus L.* is a annual or biennial plant. It has slender, hard, long stems. It is branched, up to 1.5 m tall, often purple toward base [13]. Leaves are lobed, with petioles 10x30 cm long covered with a few bristly hairs. Flowers are pale yellow, 1.2 to 1.5 cm long. Figure 6.1 presents the stem and the flowers.

Figure 6.1: Stem and flowers of *Brassica napus L.* [8]

Crop application *Brassica napus L.* is worlds 10th most important crop [15]. Worldwide planted area is 24000 Kha. 11 % of all *Brassica* grown is GMO. This accounts for 6 % of the world total GMO area. It however accounts for 13 % of all GMO field trials.

Related species If we assume application in Europe, the related species most likely to be neighboring a GMO *Brassica* field are presented in table 6.4. These will be used as model species.

Type of modification There are two types of modification that can be considered in the case. One is the glyphosate resistance, and the other is the *Bt* toxin production. If we assume considerable spray drift from application of herbicide on the field edges, than the fitness increasing effect of herbicide resistance is similar to the *Bt* effect. Plants that have the gene for insect or

Model Species	Biological counterpart
GMO	*Brassica napus L.* modified
Wild	*Brassica napus* wild type
Remote	*Brassica rapa* a.k.a. *Brassica campestris*
Non	All other plants

Table 6.4: Model species and their biological counterparts

for herbicide resistance, have both an extreme fitness advantage above the plants not possessing it. The long term effects of spread through the nature would be indeed different, since the Bt toxic always confers an advantage. But considering the local scale of the model it does not matter which gene is taken. It does however widen the available literature for the case study.

6.3.2 Aspects of analysis

There are three aspects to evaluating model use through a case study. First is the examination of the usefulness and availability of literature for translating real world situations into model terms. The second aspect is the demonstration of the process of abstraction of biological data into Agent Based terms and GeneScape parameters. Third aspect is the evaluation whether GeneScape is able to reproduce the observed case study behavior.

Availability of literature It is important to examine the availability and type of literature about Gene Flow. Researchers experimenting with Gene Flow, pollen spread, plant relatedness and similar issues are trying to answer very specific questions, and data necessary for model validation might not be available.

Abstracting biological data If biological data are available, they have to be abstracted into model parameters. Since ABMs require a very specific type of information, it has has to be decoded from traditional experiments. Making useful abstractions requires a knowledge of how GeneScape operates. It is therefore important to provide the user with an example, a cook-book if you like, to how this is done.

Case behavior This third aspect is a problematic one. The goal of each model is to be able to reproduce a part of reality, and possibly predict the response of reality to some change. Whether this is possible with GeneScape depends strongly on the first two aspects. Only if there is clear data to compare with, and if the relevant biological aspects have been properly abstracted into GeneScape parameters can this comparison successful.

It should be noted that GeneScape is designed to make quantitative predictions. Prediction that can realistically be expected from the model are qualitative in nature. Nonetheless qualitative results are sufficient to evaluate the case behavior in the light of biological theory.

6.3.3 Analysis method

6.3.3.1 Literature use and Variable estimation

As presented in section 5.5, Input variables, the three input variables used are the Fitness increase of the GMO gene, Pollen spread distance and the Mating preferences. In order to implement the case, these variables must be estimated from literature.

Literature review The first step is a literature review. Focus will be on the literature about experiments directly describing the variables. If such literature is not available, indirect estimation paths will be used.

Parameter estimation After the literature review the found data will be abstracted to model parameters. These parameters will be used in examining the behavior of the case.

6.3.3.2 Case behavior

Case behavior evaluation will done by reproducing some possible real world situations and examining the outcomes of the model. The outcomes must not be interpreted as quantitative prediction of what might happen. They are rather illustrations of kind of behavior that is theoretically possible under such a situations. The experiments are described in the following paragraphs.

Steady state run To start the evaluation of case behavior, it is important to determine the basic behavior. It will serve as a blueprint to compare behavior under different situations to it. The case will be run until is reaches steady state.

Barren zone created than the pollen spread distance Barren zones are often suggested as a measure to limit Gene Flow. A barren zone is an area around the field of GMOs without any plants. Its width is greater than the pollen spread distance.

Barren zone smaller than the pollen spread distance Since pollen spread distance is strongly dependant on the environmental situation, fixed width barren zones may not always be effective. It can easily happen that the pollen is carried further than expected. To simulate this situation, model will be run with the width of the barren zone just under the pollen spread distance.

Single year planting Crop manuals advise not to plant GMO *Brassica* each year because of risk of creation of volunteer weeds and increased weed evolution. A situation that often occurs is that GMO *Brassica*, containing a for example herbicide resistance gene is planted one year, and the use is discontinued for several years. In the year that the crop is planted, the field and the surrounding area experience large selection pressure because of heavy herbicide application and spray-drift. In the following years there is no selection pressure. Therefore it is interesting to examine the behavior of the model in this situation. The experiment is described in table 6.5.

Generation	GMO Gene Fitness
1	160
2 & up	0

Table 6.5: Single planting experiment parameter changes

Chapter 7

Results and Discussion

In this chapter I will present the results of the Model Evaluation and discuss their significance. Results of the appearance, behavior and use evaluation will be presented and through the results the final result of the project, GeneScape will be described. Finally, I will present some observations made during GeneScape development, implementation and evaluation.

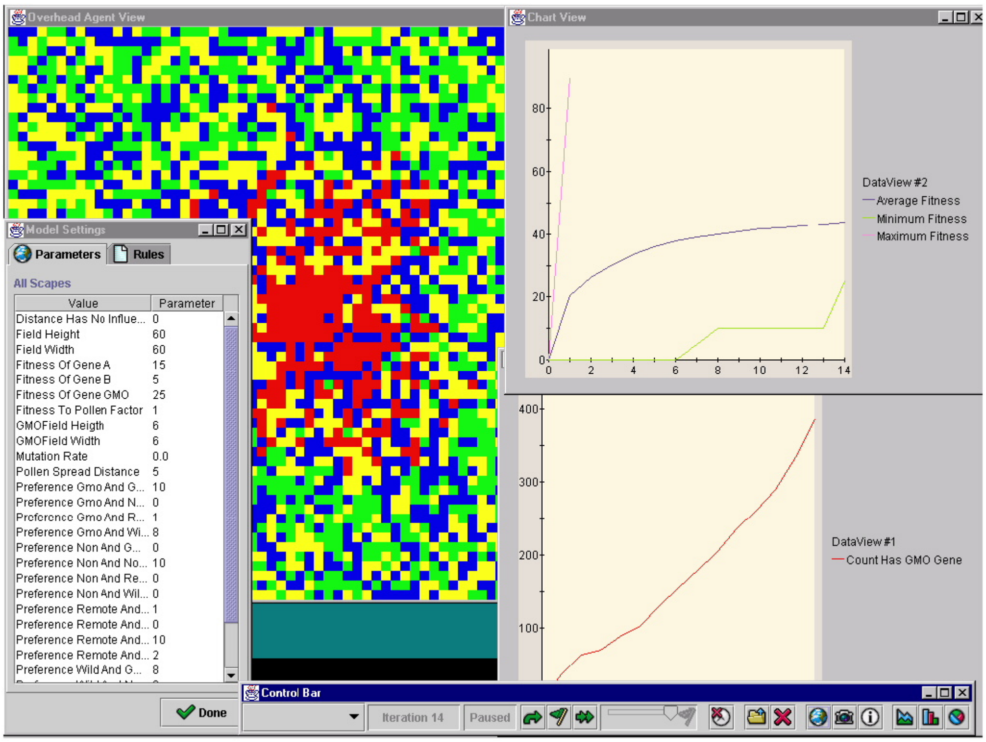
7.1 Model appearance

GeneScape is implemented in Java 2. This makes it is platform independent, and is accessible on all common hardware/operating system combinations. Furthermore, it is also web based, so that it is accessible from anywhere in the world. It has small download footprint of approximately 1 MB. This makes it accessible to users without fast Internet access. Operation of GeneScape is straight-forward through the control panel, and the interactivity is provided through the parameter input window, the Field view and the simple graphing interface. Figure 7.1 presents the desktop version of GeneScape during use.

The direct output of GeneScape is a movie showing Gene Flow. Both the screen captures and the movies of different runs can be found at www.IgorNikolic.com under the section Results. Alternatively it is possible to run GeneScape in the Web mode, directly from a web-site. In this mode the capabilities are somewhat reduced, since graph plotting is disabled. The online version can be found under the Run The Model section.

7.2 Model behavior evaluation results

In the following section I will present and discuss the behavior evaluation. First the Reproducibility Analysis results are presented. Subsequently the results of the Range of Behaviors and the Internal Consistency evaluation are presented.



Value	Parameter
Distance Has No Influe...	0
Field Height	60
Field Width	60
Fitness Of Gene A	15
Fitness Of Gene B	5
Fitness Of Gene GMO	25
Fitness To Pollen Factor	1
GMOField Heigth	6
GMOField Width	6
Mutation Rate	0.0
Pollen Spread Distance	5
Preference Gmo And G...	10
Preference Gmo And N...	0
Preference Gmo And R...	1
Preference Gmo And Wi...	8
Preference Non And G...	0
Preference Non And No...	10
Preference Non And Re...	0
Preference Non And Wil...	0
Preference Remote And...	1
Preference Remote And...	0
Preference Remote And...	10
Preference Remote And...	2
Preference Wild And G...	8

Figure 7.1: Screen capture of GeneScape in desktop mode

7.2.1 Reproducibility

In tables F.1 and F.2 of the Appendix F, the outputs of the *t*-test for difference in averages are presented. These data are produced by the Gnumeric v0.41 spreadsheet and data analysis software. The raw data used for the test can be found in the same Appendix in table F.3.

Confidence interval for NPCCG Taking the "t Critical two-tail" value from table F.1, the 95 % confidence interval for D is [-1.984, 1.984]. Considering that the value of D = 0.374 we can conclude that the $H_0: \mu_1 = \mu_2$ is not rejected, and that the averages of both sets of NPCGG data are not significantly different.

Confidence interval for APF Considering that the number of observations is the same as in the previous test, the 95 % confidence interval for D is [-1.984, 1.984]. Considering that the value of D = 0.273, we can conclude that the $H_0: \mu_1 = \mu_2$ is not rejected, and that the averages of both sets of APF data are not significantly different.

Effect on the output variables The presented result means that even though the initial genomes, location and species of the Plant agents are determined at random at each run, there is no significant effect on the output variables.

Alternatively, the outcome can be seen as the proof that the effect of the initial genome, plant location and species which does in fact influence Gene Flow was successfully excluded as in input variable by randomization.

Future experiments may consider explicit starting genome, plant locations and species distributions, but in that case the input variables need to be redefined.

7.2.2 Range of behaviors

In the following section I will present the results of the Range of Behavior evaluation. The data are presented and discussed per Relatedness Level, for both 2 and 20 generations.

Parameter Space Plots One way to represent the behavior across the state space is to plot the Effect and NPCGG after 2 and 20 generations against the Fitness of GMO Gene (Fitness) and Pollen Spread Distance (Distance), for each Relatedness Level. An example of the plots is given in figures 7.2 and 7.3. The plots of all Relatedness Levels are presented in Appendix G.

7.2.2.1 Relatedness Level I

2 generations The Parameter Plot for Relatedness Level I and 2 generations is presented in figure G.2. We see that the NPCGG plot shows that there is no spread after two generations whatsoever. This is logical, since the Relatedness case is I, and the plants are not related to each other at all. The Effect plot shows a negative value at low Distance value. In these situations Plants can not reach enough neighbors to mate with. Therefore the population as a whole has not had the time to equate the genes within a species. Distance of 2 plants already allows for sufficient neighbors so that the equalization of the genomes can take place.

20 generations The Parameter Plot for Relatedness Level I and 20 generations is presented in figure 7.2 and 7.3. The situation is very comparable to the 2 generation run. If we examine the scale of the Effect axis, we can see that the equation of genes has continued in a slow pace, and that the

---

[2] Reading the figures: The axis label is placed *below* the axis it belongs to. So the Fitness is plotted to the *left*, away from the Fitness label, and the Pollen Spread Distance axis is plotted to the *right*, away from the label.

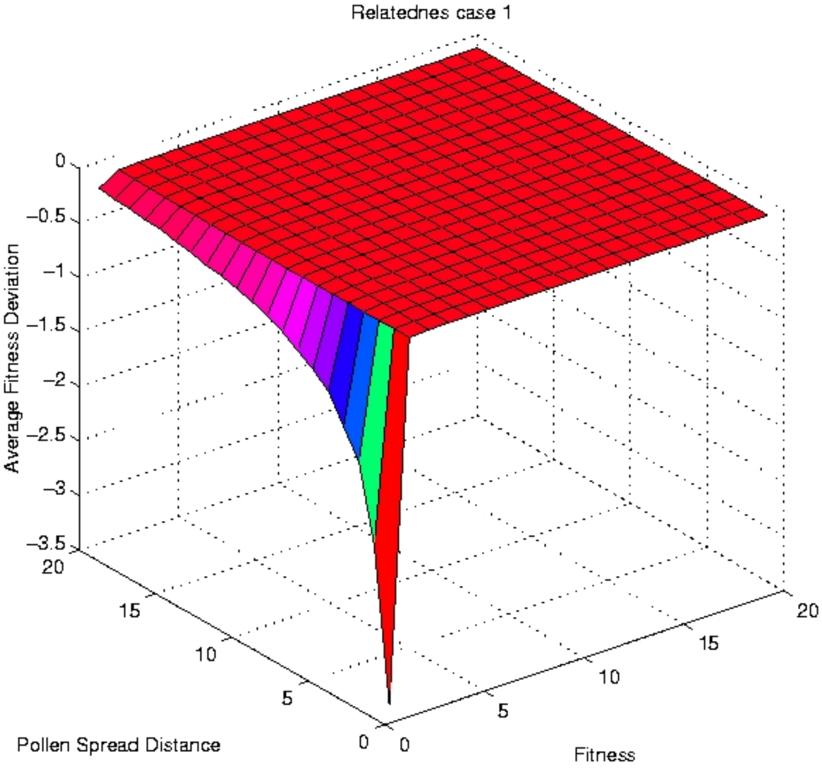


Figure 7.2: Effect vs Fitness and Distance, Relatedness Level I, 20 generations.[2]

inequality of maximum -14 in the 2 generation case is reduced to an inequality of maximum -3.5 in the 20 year case. This straightens the notion that the plants are very strongly limited in their options of mating partners to exchange the genes, and that it takes a long time for them to spread out through the population.

7.2.2.2 Relatedness Level II

2 generations The Parameter Plot for Relatedness Level II and 2 generations is presented in figure G.4. Just as with the previous case, the Effect increases with the Distance. At the distance of approximately 4, the distance does not have a limiting effect any more, and the genes can equate at the same rate across the space. The fitness has a very low effect, with the high fitness area's having only a approximately 1% higher effect than the low fitness areas.

Short-Distance effect The NPCGG of plants shows an interesting effect. The largest portion of the space shows no increase in the number

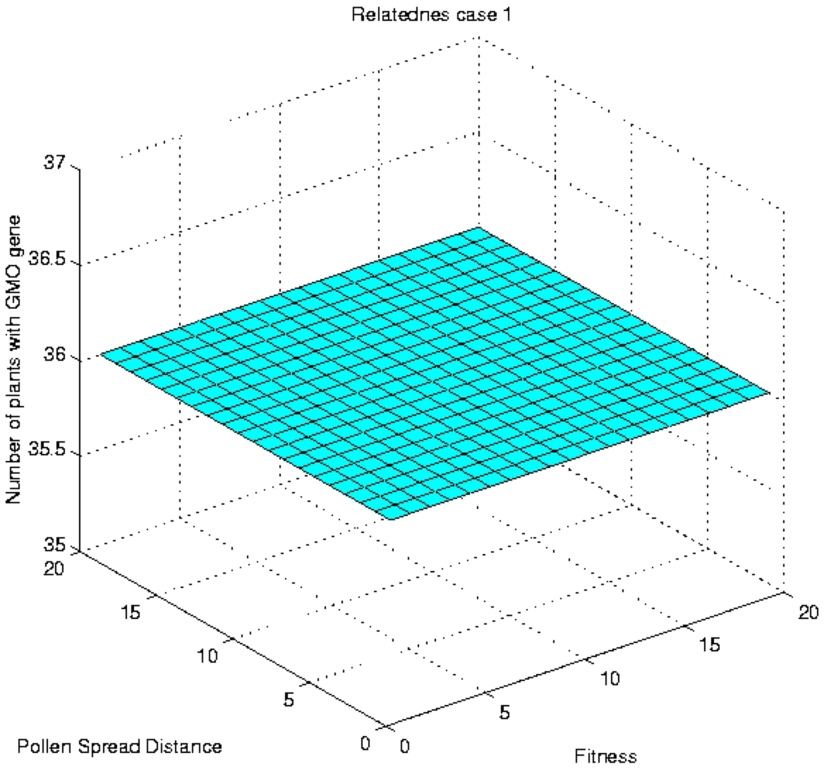


Figure 7.3: NPCGG vs Fitness and Distance, Relatedness Level I, 20 generations.

of Plants containing the GMO gene. A the very low spread distance areas however, the gene is able to cross the species barrier. This happens because each Plant must choose a mating partner, and if the spread distance is limited to 1 and maximum 2 plants, a mating partner of another species will be chosen if there are no partners of the own species available. At higher distances there is always at least one partner of the own species available. The effect is caused by the random division of the plants, which not being perfect, allows irregular division at a local scale. The effect is noticeable if it takes place near the border of the GMO field.

20 generations The Parameter Plot for Relatedness Level II and 20 generations is presented in figure G.3. Here we see that the Short-Distance effect still dominates the behavior of the Effect. The population is very close to being totally equated. The average Effect is around -0.15, whereas at 2 generations it is around -0.85. We can also see the short distance effect on the NPCGG. In some situations the GMO gene is able to spread to a large number of plants, whereas in other cases it stays limited because of the random plant division which can create boundaries of incompatible plants, especially of Non Related species. These boundaries can limit the spread by containing the gene.

7.2.2.3 Relatedness Level III

2 generations The Parameter Plot for Relatedness Level III and 2 generations is presented in figure G.6. First, we see the Short-Distance effect again, in both the Effect and the NPCGG plots. Secondly, another effect also presents itself in this plot. At the highest Fitness value of 20, both the Effect and NPCGG suddenly increase. This is better observable at the 20 generations plot.

20 generations The Parameter Plot for Relatedness Level III and 20 generations is presented in figure G.6. In the 20 generations plot the Short-Distance effect can also be observed.

Critical Fitness Value effect More pronounced than in the 2 generations plot, we can see the effect of Critical Fitness Value. When fitness reaches a value of 20, the Attractiveness for a Plant of a further related species containing the GMO gene can become larger than the Attractiveness for a Plant of own species without the gene. And since the most Attractive Plant is chosen for mating, Plant containing the GMO gene of a Critical Fitness Value will be able to cross the species barrier.

Combined effects This is especially noticeable at lower Fitness and Distance values. If the plant can not find an mate of the own species that has the maximum base fitness, it will choose a GMO in the neighborhood which has a higher attractiveness because it contains the very strong GMO gene. So the fitness of the GMO gene overpowers the Preference. We can see that this effect stops at distances greater than 5, because there seem to be enough plants of the own species with a high base fitness.

Change of Steady States The model changes behavior completely only at the fitness of 20 and above, because the Critical Fitness Value effect comes pronounced. In this case plant the GMO Plant is always the most Attractive partner. When this happens the model switches to a different steady state. This can be observed especially well when examining the NPCGG. Whereas the first steady state is at 36, the second is at 2500, where both the Wild and Remote have acquired the gene. There is also a third, metastable point possible at 1200, where only one population has acquired the gene. However, this can happen only if the Mating Preferences have large differences.

7.2.2.4 Relatedness Level IV

2 generations The Parameter Plot for Relatedness Level IV and 2 generations is presented in figure G.8. In Case 4 we can observe both Short-Distance and the Critical Fitness Value phenomenon. When examining the Effect plot we see that the strongest effect is at the highest Fitness and Distance values, as expected. Also, 2 generations are even at this high relatedness level not enough to equate the genes.

Multiple states When examining the NPCGG plot, we see two cases of the Critical Fitness Value phenomenon. At the low plateau, there is no species crossover possible up to fitness of 5. Here the added extra fitness is less than 20 %, while there is a 20% less attractiveness of a Wild for the GMO that for the own species. Between 5 and 10, the Wild species can always be entered. But the Remote is still too little related. Above 10 both species are entered, and the number rises much faster. It seems that the 5 to 10 Fitness section is going to the intermediate steady state. However, an interesting aspect occurs when the situation is examined at 20 generations.

20 generations The Parameter Plot for Relatedness Level IV and 20 generations is presented in figure G.7. When examining the Effect plot after 20 generations we see that the effect is 0 or higher everywhere except at the very lowest Distance. We observe the 0 effect region, which corresponds with the lowest steady state of 36 plants. Also the effect now clearly increase with the Fitness, showing that the high Fitnesses can increase the rate of spread.

Metastable Steady State overshoot If we examine the NPCGG plot, we see that the metastable region the intermediate Fitness seemed to be going to is overshot, and the model ends in the high steady state of 2500. This is because after entering the Wild population, the gene increases the total fitness of a Wild plant so much that it can cross the second barrier to the Remote. So the Gene step-stones into the whole population.

7.2.2.5 Relatedness Level V

2 generations The Parameter Plot for Relatedness Level V and 2 generations is presented in figure G.10. Just as Relatedness Level I, V is a special situation. Here, the three species can be considered as one population. The relatedness levels are all equal. Considering the Effect plot, we see the Short-Distance effect takes place. Further there is a monotonous increase of the Effect with the Fitness and Distance. The NPCGG plot shows a monotonous increase with the Distance. The Fitness has no effect on the spread anymore, since even the smallest extra Fitness will spread through the population. Only the distance affects the speed of the spread.

20 generations The Parameter Plot for Relatedness Level V and 20 generations is presented in figure G.9. If we consider the Effect plot, we see the combined effect of Distance and Fitness, without interference of the Relatedness levels. In the NPCGG plot we see that all plants have taken up the gene, except at the very smallest Distance, where the time was not sufficient.

7.2.2.6 State-space plots

State space plot presents all of the output variables in one figure. It represents trajectories, time series of model states. Plotting the trajectories of all end points shows how that steady state was achieved, and allows for observations of model dynamics.

Steady States In figure 7.4 we can see all the steady states achieved in the Behavior evaluation. In this plot the trajectories are omitted.

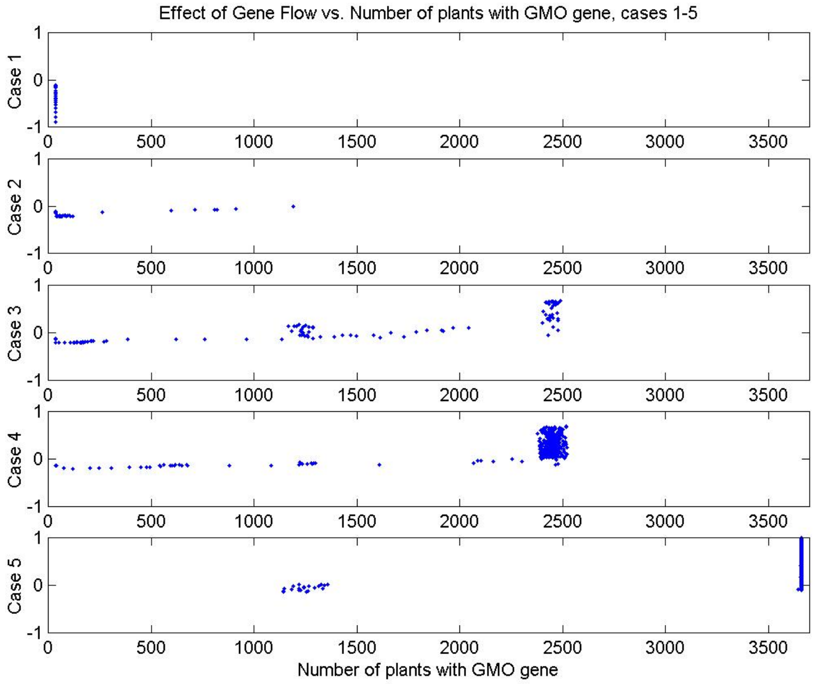


Figure 7.4: State-space plot of the endpoints of the parameter sweep, 20 generations, per Relatedness Level.

For Relatedness level I it is clear that all parameters ended in the steady state of 36, while the Effect is in the range on -1 to 0. In Level II, next to the steady state at 36, some runs are heading to a different steady state. It is not clear which one that would be. In Level 3 we observe three steady states, at 36, 1200 and 2500. We also observe a large number of runs between the states. In Level IV most runs have reached the 2500 steady state. In Level V basically all the runs end up in the highest steady state, with a few runs at low Distances which have not reached it yet.

While presenting the division of end states per case, this plot shows us little model dynamics. Therefore it is interesting to examine the trajectory plots.

Trajectory plots of the Parameter Space In figure 7.5 we see the State-space trajectories in a 3D plot for 100 generations. It represents a selection of the runs described above.

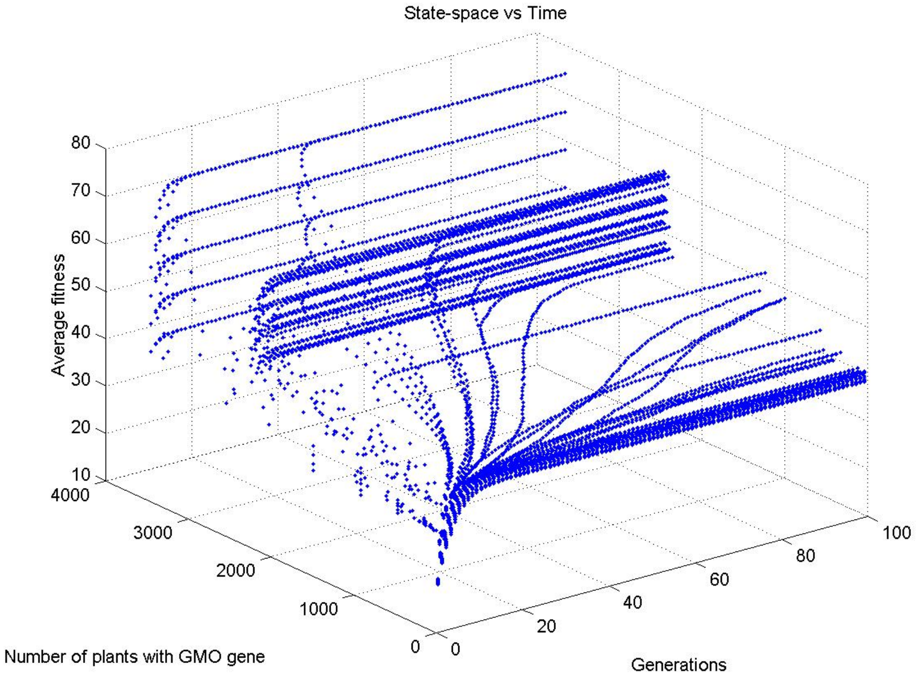


Figure 7.5: State-space plot of the parameter sweep, 3D view.

In figure 7.5 we can observe the differences in the time it takes for different runs to reach the steady state. We can also observe the runs passing the intermediate steady states before reaching the final one. It also shows the existence of the metastable state around 1200 Plants, with several trajectories passing it, but just not staying at it. Since the distance between the points is proportional with the distance of spread, we can observe the different dynamics of the runs as well.

7.2.3 Internal consistency

In this section I will discuss the results of the Internal Consistency evaluation. The model is run at the parameters described in Appendix E, table E.3, for 50 generations. The description of the different tests is presented in section 6.2.2.3, table 6.3.

7.2.3.1 Test 1.

In figure 7.6 the state space plot of Test 1 is presented. In this test there is no relatedness between the species. The GMO Gen Fitness and Pollen Spread distance have an average value. As we can see in figure 7.6, the state space is a vertical line. This indicates that the average fitness rises from the minimum of 30 up to the theoretical maximum through gene exchange within one species, while there is no increase in the number of individuals containing the GMO gene. This confirms the assumption that there is no interspecies Gene Flow between populations if the organisms are not related to each other and thus unable to mate. Also, in the decreasing distance between the subsequent points we can see that the speed of the APF increase slows down.

7.2.3.2 Test 2

Test 2 is designed to test whether there is Gene Flow possible if there is no pollen movement. Executing GeneScape with Pollen Spread Distance = 0 produces a run-time error.

It means that when the Spread Distance is 0, there are no neighbors to mate with, and the model stops functioning. It can be solved by coding and error-catching routine that describes the behavior when the distance is 0. This amounts to building the desired behavior in the model. Correcting this error is an issue of further development.

7.2.3.3 Test 3

In this test there is no fitness increase conferred by the GMO gene. The spread distance is average, and the relatedness of the plants is maximal. This means that the whole Field can be considered to be populated by one species. Since there is no fitness increase, the gene is incorporated through random mating. Since the source of the gene is renewed each generation, and there is no flux non-GMO genes into the crops, the total number of genes increases. We can see in figure 7.7 that the fitness increases for the initial minimum value to the average for the genes A and B (40).

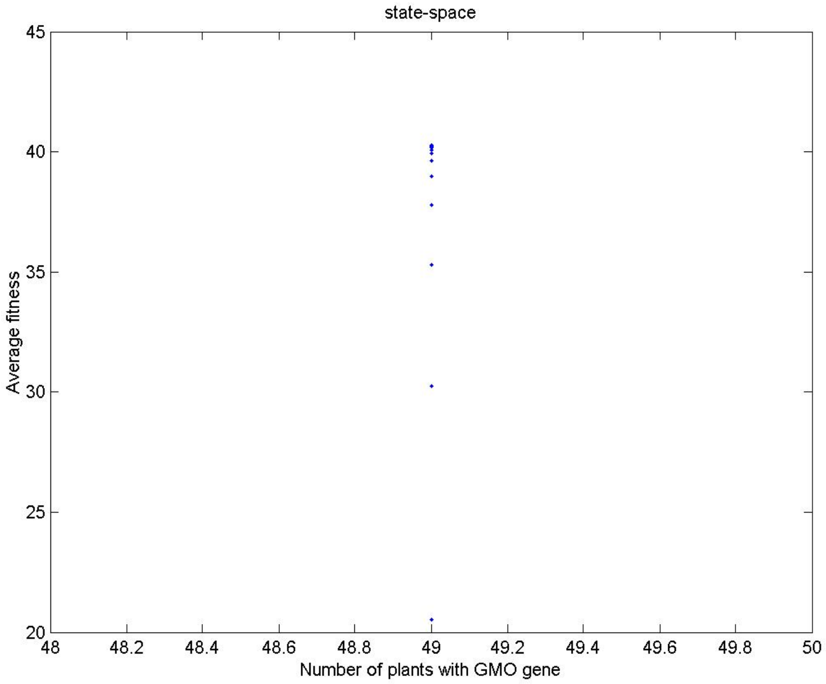


Figure 7.6: State-space plot of the Consistency test 1.

We can also observe that the speed of the increase of the number of plants fluctuates. This can be seen on the diverse distances between the points in the right hand part of the figure. This can also be observed between the second and third generation (point), where the speed of spread dramatically increases and than drops again. The same situation happens between the 5th and 6th point. This points at the random, diffusion-like propagation of the gene through the population.

7.2.3.4 Test 4

In this test the added fitness by the two copies of the GMO gene is 200, while the base fitness is 40. This extreme fitness difference causes a fast Gene Flow. The state space of this test is presented in figure 7.8.

There are several interesting aspects of this GeneScape captured in this figure. First, it is interesting to notice that the curve is S shaped. The APF increase is rather high, while the NPCGG grows slowly. This is because the pollen spread distance is only 5 plants, and before the GMO gene had the chance to spread widely, Gene A and B are being exchanged in the population beyond the reach of the plants with the GMO gene. When the GMO gene has spread widely enough to be reached by all Plants in the the

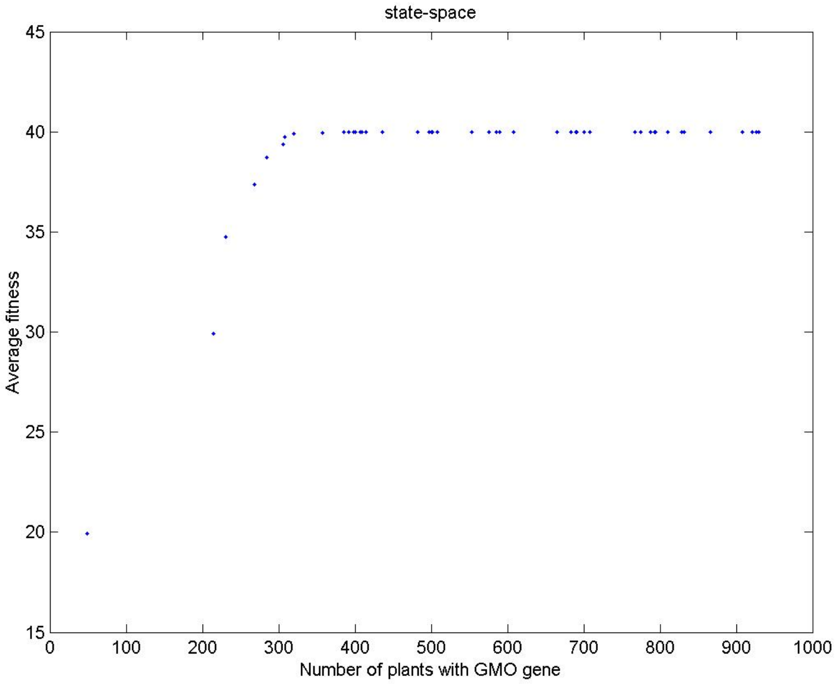


Figure 7.7: State-space plot of the Consistency test 3.

field, the speed of the NPCGG increases, since all the plants who do not have the gene yet will choose a partner with the the GMO gene. After all the plants have acquired the GMO gene, the curve becomes vertical, and the remaining plants who do not have the Gene A or B acquire them as well. This explains the slow increase of the fitness after all the plants have the GMO gene. This shows that at extreme Fitness advantages the spread of the GMO gene can be much faster than normal Gene Flow.

7.2.3.5 Test 5

Test 5 was designed to examine the situation where there is no added advantage to the GMO gene, and there are some barriers to mating.

We see that the plot is vertical line again, with a decreasing speed of APF increase. It shows that there will be no Gene Flow if there is no advantage in having the GMO gene.

This is a situation, while consistent with GeneScape design, that will not happen in nature. Even if a gene conveys no advantage, there is a real chance that it will be transferred in a hybridization process. The model works with a step-over Attractiveness threshold that is not completely biologically correct. There should be a possibility to hybridize even when there is a low

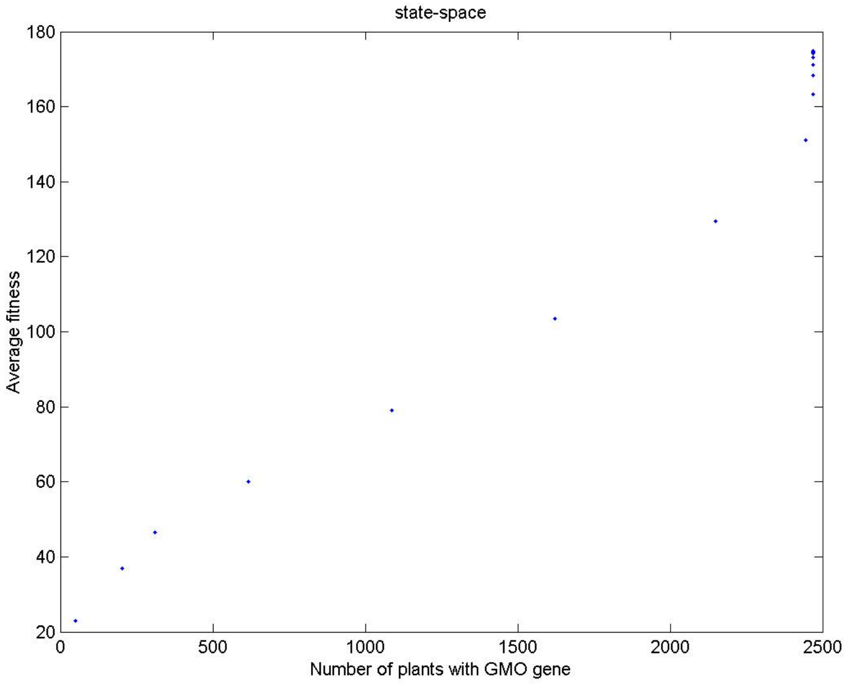


Figure 7.8: State-space plot of the Consistency test 4.

relatedness, and it happens purely by chance. This causes GeneScape to underestimate the amount of Gene Flow. This point is further discussed in section 8.4.1, Recommendations.

7.3 Case study evaluation results

In the following section I will present the results of the Case Study. I will first present and discuss the finding of the Literature Analysis and Parameter Abstraction. After that the results of the Case behaviour evaluation will be presented and discussed.

7.3.1 Literature analysis and parameter estimation

In the following subsection, I will discuss the available literature and the parameter estimation made from it for each of the three input variables.

7.3.1.1 Fitness

Fitness data are not widely available.Them ain reason seems to be that there are great difficulties determining the extra fitness, and so far there is

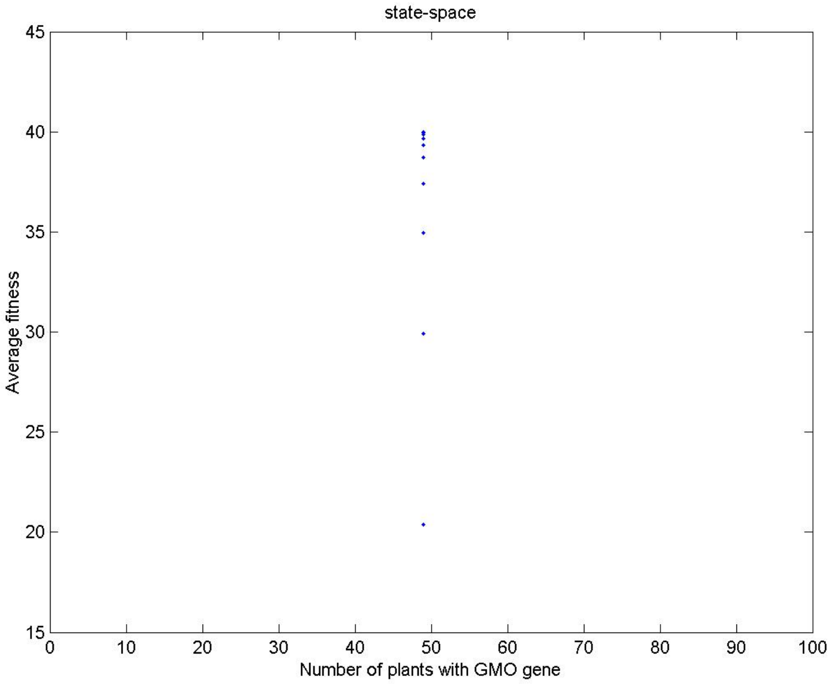


Figure 7.9: State-space plot of the Consistency test 5.

no consensus on what a good fitness measurement should consider. Stewart [22] considers the number of seeds produced by a plant a direct measurement of its fitness.

Type of study Stewarts study is a tented field try, with a controlled herbivore insect availability. It is aimed at determining the fitness increasing effect of the presence of the *Bt* gene.

Main findings There was no fitness difference observed between transgenic and non-transgenic plants in case of no selection pressure. Under insect selection pressure, there is an extreme difference of survival and seed production between transgenic and not-modified plants.

Non-transgenic plants under insect selection did not produce more than 100 seeds in the study. Under the same conditions GM plants produced seeds in the range from 100 to 800 seeds. On the average, the GMOs produced roughly 4 times more seed than the non-transgenics.

Parameter estimate From the data presented, it can be said that it is not possible to purely estimate the fitness increase, since it very strongly depends on the environment. However, since the fitness, as implemented in GeneScape, is directly related to the ability to reproduce, values can be estimated from the seed numbers. If we assume that there is no insect pressure, the Fitness Of Gene GMO variable should have the value of 0, independent of the other Gene Fitness values. In the case of selection pressure, the value of Fitness Of Gene GMO variable should be set to 4 times the combined value of the other two genes. This is a direct coupling to the difference in the number of seeds produced.

7.3.1.2 Pollen Spread Distance

Pollen spread data are much more widely available that fitness. This is mostly because there is general agreement that pollen spread is one of the key contributors to Gene Flow. Furthermore, it is relatively straight-forward to determine. There are three authors describing pollen spread from GMO *Brassica napus*. I shall compare their findings in table 7.1

Model space to distance translation The Agronomy Guide of Purdue university [6] states that optimally planted *Brassica napus* has 6 to 8 plants per square foot. That is roughly 75 plants per m^2, i.e. roughly 8 plants per meter. This way distance measurements can be translated in the number of neighbors in GeneScape.

	Morris [26]	Brown [4]	Lavigne [23]
Type of study	Gridded field study with pollen movement limiting measures implemented, experiments in two geographic areas. Sampled for transgenic plants	Patch of GMOs in a field of non-transgenics, field sampled along wind directions for transgenic seeds.	GMO Patch in a field, grid sampling for transgenic seeds.
Main findings	Most of hybrids formed within 4m.	Most of hybrids occurred within 1m, Hybrids observed up to 26m	50 % pollen within 3m, 90 % within 9m. Up to 100 m observed
Parameter estimate	Since most of the pollen falls within 4 m, the pollen spread distance can be estimate at 35 plants.	Spread distance can be estimated to be 8 plants.	Bulk of pollen falls within 9 m, thus estimated distance is 72 plants.

Table 7.1: Pollen spread studies compared

Considering the wide range of possible pollen spread, a choice needs to be made. Considering the data from literature, a distance of 5 m or 40 plants would be a realistic value.

7.3.1.3 Mating Preferences

The third important group of parameters for describing the case are the Preference X and Y Mating set. These parameters are the most difficult ones to estimate. They are derived from the tunneling speed that the germinating pollen grain has while entering the stigma of a flower. Comparative research of such processes is not widely available in the literature.Therefore, a different approach is necessary.

First, phylogenetic analysis can provide the differences in the genetic makeup between the species of interest. The distances can be used to estimate the ration of mating preferences.

Secondly, phenotypic analysis, describing the external characteristics of a plant could be used to estimate the similarities between species and derive an estimate for the preference. However, phenotypic similarities can be very misleading, since adaptation to a certain environment can produce similar morphologies with very different genetic background.

However, all these approaches require an in-depth knowledge of the involved disciplines, and their review falls outside the scope of this research.

The Relatedness is therefore estimated from literature that discusses plant relatedness in more general terms. I will discuss three publications in more detail. They are summed up in table 7.2 on the next page.

	Ellstrand [15]	Hauser [19]	Parkin [29]
Type of study	Literature review	Hybridization & progeny fitness study	Genetic marker conservation across chromosome pairing and crossing
Main findings	”It is uncertain whether or not *B. napus* exists in truly wild form, except for escapes from cultivation (85), but reproductive barriers that would prevent spontaneous hybridization between wild and cultivated *B. napus* are unlikely.” Field experiments examining the hybridizations rates between *B. napus* and *B. campestris* by growing the latter in a field of the first one, found that between 9 and 93 % of progeny to be hybrid, depending on the experiment setup	*B. rapa* is the most likely candidate for hybridization of *B. napus*. Hybrid seed can be found up to 13% of weedy *B. rapa* populations. The data are highly dependant on which brand *B. napus* is taken, and which population *B. rapa* is examined.	Genomes of the diploid progenitors of amphidiploid *B. napus* have remained essentially unaltered since the formation of the species and that the progenitor genomes were similar to those of modern-day and *B.oleracea*.
Parameter estimate	From these data it can be concluded that the relatedness level between *B. napus* and *B. rapa* is approximately 8.	These data would suggest a lower relatedness score, approximately around 1. However, since *B. rapa* can pollinate with its own species, this number is less reliable than the one estimated from the Ellstrand study.	These finding suggest a very high level of relatedness between crop and wild *B. napus*, approximately around 9.

Table 7.2: Relatedness studies compared

7.3.1.4 Case parameters

Estimated parameters are presented in table 7.3. They assume an insect infestation or extensive spray drift, base fitness of 40, 5 m pollen spread distance and a relatively high relatedness level.

Parameter	Estimated parameter value
GMO And GMO	10
GMO And Non	0
GMO And Remote	8
GMO And Wild	9
Non And GMO	0
Non And Non	10
Non And Remote	0
Non And Wild	0
Remote And GMO	8
Remote And Non	0
Remote And Remote	10
Remote And Wild	9
Wild And GMO	9
Wild And Non	0
Wild And Remote	9
Wild And Wild	10
Pollen spread distance	40
GMO gene fitness	160

Table 7.3: Estimated mating preferences

7.3.2 Case behavior

In the following subsection I will discuss the behavior of the case under different situations. The situations have been presented in section 6.3.3.2.

Test run First it is interesting to describe how the case behaves when reaching steady state in the default layouts. Figure 7.10 presents the state-space of the run.

The run is 24 generations long. At this time the steady state has been achieved. As we can see from figure 7.10, the spread proceeds a very high rate, almost linearly. Only when all plants have acquired the very advantageous GMO gene do the other genes start to spread. This situation can be expected, since if there is an extreme section pressure only the plants containing the GMO gene will be able to survive and reproduce.

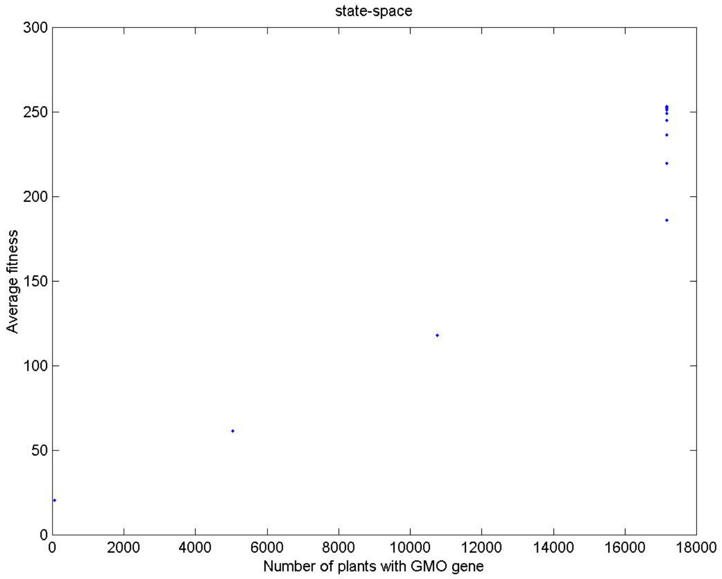


Figure 7.10: State-space of the case run, 24 generations.

Barren zone greater than the pollen spread distance Barren zones are often suggested as a measure to limit Gene Flow. A barren zone is a border around the field of GMOs that has no plants, and has a width greater than the pollen spread distance. The results of this situation is presented in figure 7.11

As it can be seen from the plot, there is no spread of the GMO gene. This is indeed how the barren zone should work. However, in practice pollen can spread to grater distances than expected. That situation is examined in the following experiment.

Barren zone smaller than the pollen spread distance Since pollen spread distance is strongly dependant on the environmental situation, fixed width barren zones may not always be effective. In the following figure the state-space plot of a run is presented with a barren zone 1 plant smaller than the pollen spread distance.

As we can see, there is extensive degree of Gene Flow. The situation is very similar to the one presented in figure 7.10, where there is no limitation to Pollen movement. The difference is that in this case the spread is somewhat slower, as can be seen at the value of the APF at the 4th generation. The total NPCGG is lower in this case, since there is a number of plants missing in the barren zone compared to the test run.

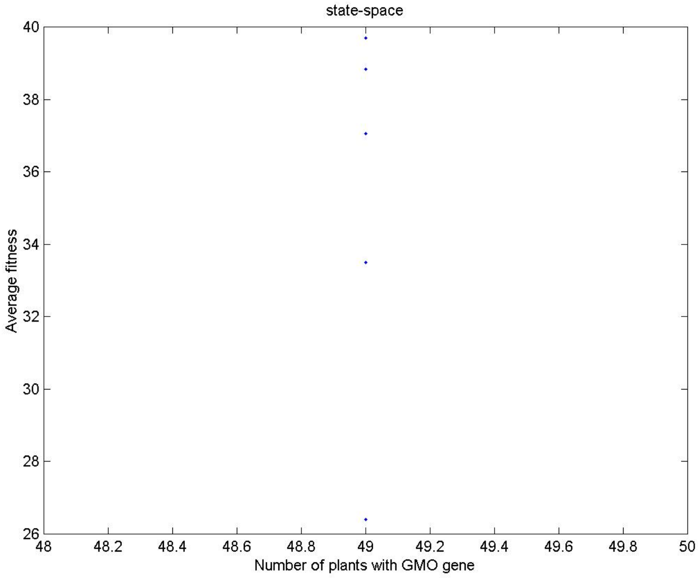


Figure 7.11: State-space of the case run with a large barren zone, 6 generations.

Single year planting Crop planting manuals [6] advise not to plant GMO *Brassica* every year because of risk of creation of volunteer weeds and weed evolution. A situation that often occurs than is that GMO *Brassica*, containing for example a herbicide resistance gene is planted single time and the use is discontinued for several years. In the year that the crop is planted, there is a very large selection pressure on the surrounding population because of heavy herbicide application and spray-drift. The state-space plot of this experiment is presented in figure 7.13

We can observe a rapid increase of the NPCGG in the first generation. The gene flows out the population, increasing the APF. After that generation, the Fitness of the GMO gene is decreased to 0, and no new plants are planted in the field. As an immediate effect the APF drops to around 35. Since plants that already have the GMO also acquired other genes, they are still attractive as mating partners, and the NPCGG increases further. In the subsequent generations there is an exchange of genes A and B which increase the APF. Since there is no selection pressure for the GMO gene, we see the number fluctuate.

Since this figure is quite complicated, time plots of both output variables are presented in figures 7.14 and 7.15.

The fitness plot increases drastically with the flow of the GMO gene. After its value is set to 0 we see a gradual increase of the APF in time.

After initial extreme spread, the selection pressure for the GM gene is not

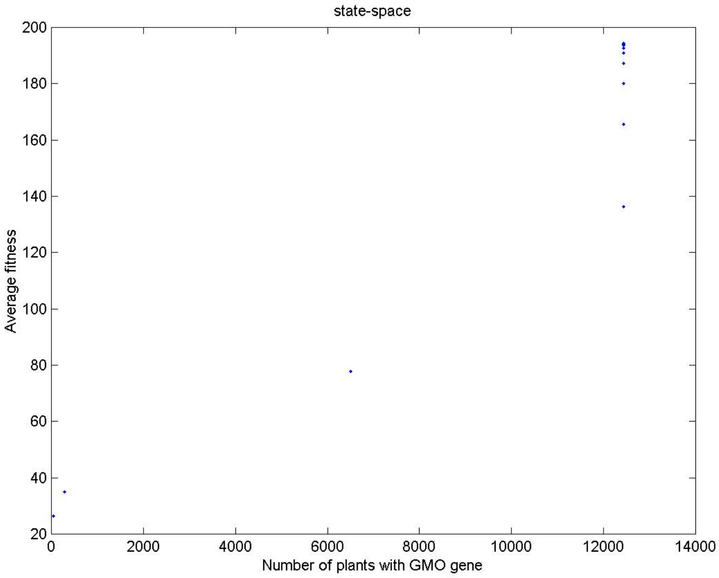


Figure 7.12: State-space of the case run with a small barren zone, 14 generations.

present. The population, which received an single input of a large number of copies of the GMO gene is balanced between in- and outflow of the gene from plants. Since after 6 generations all the Plants have the same fitness a plant containing the GMO gene might be chosen as a mating partner randomly. This way a plant can acquire the gene. In the next generation it may mate with a plant without the gene, and thus loose the GMO gene again. These fluctuations are clearly observable in the time plot. It is also interesting to note that the NPCGG stays roughly constant.

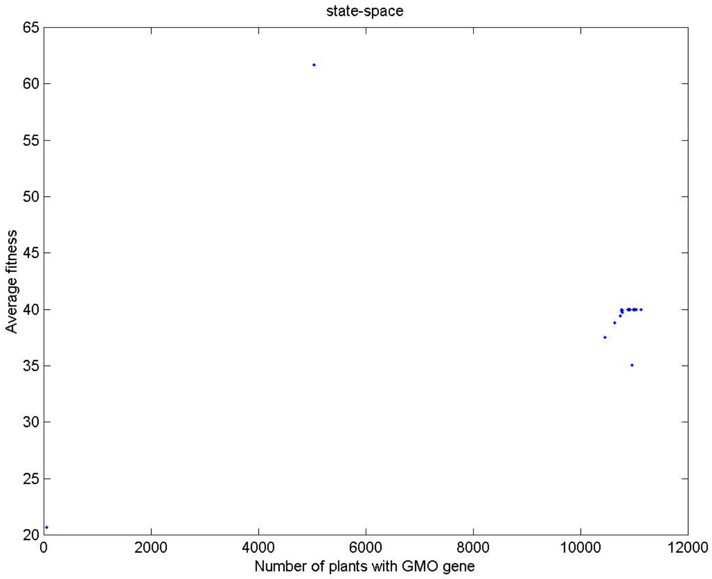


Figure 7.13: State-space of the case run with single year GMO planting, 24 generations.

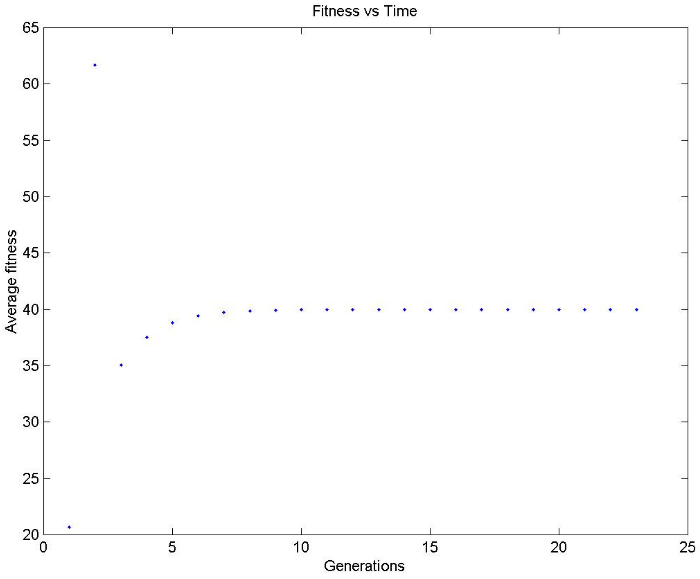


Figure 7.14: Fitness vs. Time plot, case run with single year GMO planting, 24 generations.

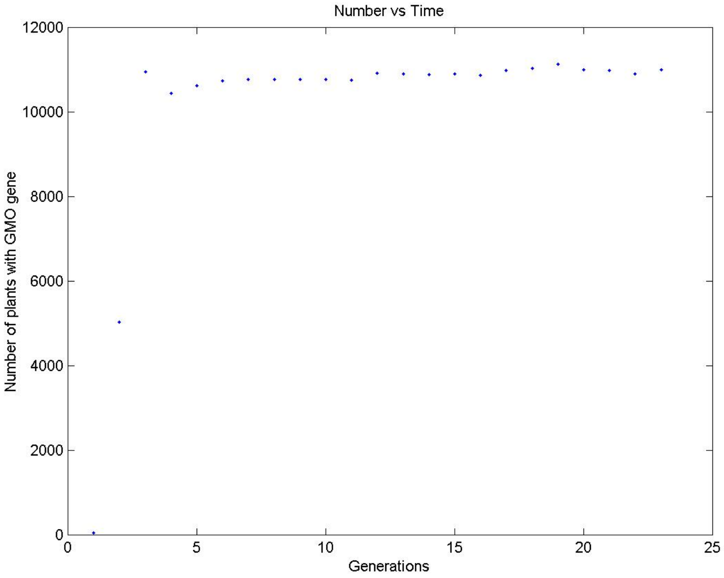


Figure 7.15: NPCGG vs. Time, case run with single year GMO planting, 24 generations.

Chapter 8

Considerations, Conclusions and Recommendations

In the last chapter of the thesis I will present the societal acceptance hypothesis, a number of considerations from which the conclusions will follow. Finally, recommendations for further work will be presented.

8.1 Social Acceptance hypothesis

As presented in section 3.1, there is a need for a tool that will estimate risks of GM quantitatively and objectively. Such a tool must itself be acceptable. Gene Scape is by no means the definitive risk prediction tool, but is it certainly a first step towards the development of a contribution to the QRA.

In this section I would like to present a hypothesis about the expected public acceptance of GeneScape. It is based on theoretical considerations presented in section 3.1 and Appendix A and on personal experience in presenting GeneScape to different audiences.

This hypothesis should be examined through rigorous social research for which there was no possibility within the scope of this project. However, the hypothesis can serve as a basis for further work.

Assumptions & Observations In the following paragraphs I will quickly present the assumptions and observations about the acceptability of GeneScape.

Intuitive but not simple GeneScape is an intuitive model. Because it models explicitly distinct entities such as the Plant, the Field and the World it is easy to grasp the concept and the workings of the model. However, it is not simple to use. It requires a high level of expertise to select the parameters and make sense of the results.

Accessible and Representative Because of the way it is implemented through the Internet and as a stand-alone application is easily accessible and representative. It produces evocative images and the the interface is relatively straight-forward.

Valid within boundaries From the evaluation it can be concluded that as long no quantitative spatial and temporal data are required, GeneScape is biologically sound.

Value judgment GeneScape carries no explicit value judgment. However, examining the implicit undertone of GeneScape one can detect slight negative connotations about the use of GMOs and their spread. For example, the color of the GMO gene in the model is red. Seeing a red "invasion" on the screen carries an emotional message. While this was not consciously implemented, such aspects might be important in acceptance.

Independent source GeneScape is an independent source of information about Gene Flow. The motives for developing GeneScape are purely scientific interest, and there is no influence of involved actors like NGOs or life-science corporations.

Hypothesis From the points discussed in the previous subsection I suggest the following societal acceptance hypothesis :

> GeneScape in its current state is not a societally acceptable tool for predicting risks of Gene Flow. It requires a high level of knowledge to operate and interpret. Its working, its inputs and outputs are too complicated to explain in clear and unambiguous way. It is a research tool that can be developed to become acceptable, since the basic acceptability premises are present.

8.2 Consideration

In the following section I will present several considerations arising from the presented work.

Natural selection as an emergent property An interesting aspect is that all genes can and do flow. This seems as a trivial observation, but it is an important one. Since the genomes of the wild population are randomly distributed, it can be observed in the total fitness graphs that there is an information exchange in the wild population that goes on until all plants have acquired all of the fitness increasing genes. This can be seen in figure 8.1

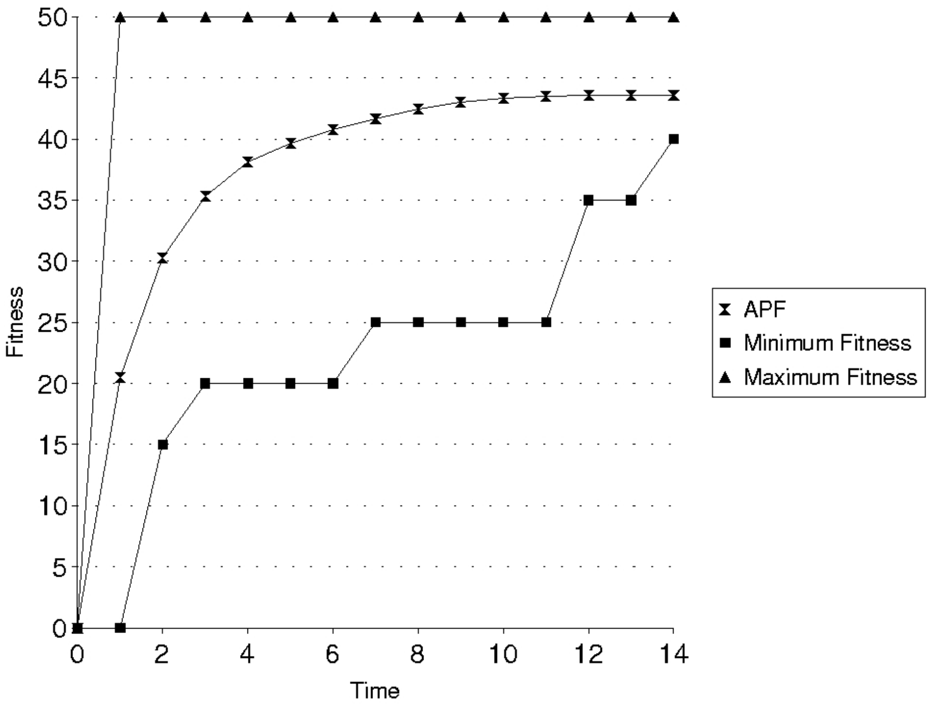


Figure 8.1: Change of fitnesses in the population. Lowest line presents the fitness of the least fit Plant, the highest line the Plant with maximum fitness, the middle line presents the APF

We can see that the plant with the minimal fitness slowly acquires more and more fitness increasing genes, until it reaches at least the average population fitness. It does not necessarily have to reach the maximum, since mating barriers might prevent it from gaining the GMO gene.

This means that even that there is no explicit goal of "must be as fit as possible", the mechanism of natural selection is implicit in the system. It can be seen as an emergent property of the model, since the only link to natural selection in the model is the statement that a more fit plant makes more pollen, and that the Egg will select the neighbor that has the most pollen.

Information spread network When a gene flows out of the GMO population and enters a particular species, it spreads quickly through it, much faster than if it was spreading at random. The spatial connectedness of for example the Wild relatives serves as a pathway for spread. Since every time a Wild relative plant becomes more fit because it contains the GMO gene, it becomes more attractive as a partner, and is more or less certain to pollinate the neighboring plants of the same species. This way the gene influences its own spread, since it amplifies the ability of the plant to spread it. Because of this, even after a few generations the gene can spread through the entire Wild population. If the gene is able to jump the species barrier by accident, it will suddenly experience an increase in spread again through the second population.

Importance A biologically relevant conclusion can be made from this observation. Care must be taken when modifying GMO which have relatives who can pollinate over great distances. Since if and when the gene enters the far-pollinating population, the flow through the population might be very fast, and cover large distances. On the other side, if the relative the gene enters has a very localized pollinating potential, the threat might be less.

Loss of information In the situation where all members of a population have acquired a certain gene, and there is no more possibility of the gene entering from a different population, the system as a whole looses the information for not possessing the gene. Interesting experiment is the situation where after the gene has enter all individuals, one makes the gene convey a negative advantage.

The APF will drop and if there is no individual that contains the ”do not have th GMO gene” gene, it will stay low. Since the population has no source of information to ”forget” about having the GMO gene it stays permanently damaged. It has to rely on mutation to destroy the GMO gene, making the carrier of the destroyed gene suddenly more fit in order for Gene Flow form the mutant to improve the APF again.

Since this is a consequence of basic genetic principles, one can imagine a situation where a population, once it picks up a certain gene under advantageous regimes might be permanently damaged if it has no source of alternative genes because it is isolated.

In reality this of course would not happen very easily since the real plants are far more complex, but the principle is not impossible.

Importance From the following one could conclude that exerting a continuous artificial selection pressure on a non-target population carries a risk. This might cause the entire population to become less fit with the selection pressure is removed. In that case the population might disappear because a fresh intruder species out-competes it. A hypothetical example would be a case where for example a grass is modified take an advantage of the high nitrogen levels in the soil in the Netherlands, as an attempt to control the Dutch nitrogen problem/ Since nitrogen is applied to the field and it leaches out to the surrounding area, populations in the area around the field are likely to take up the gene, since they also experience high nitrogen. If after a long time nitrogen supply is reduced, the wild populations who might have incorporated the nitrogen gene will suffer, and the ecosystem might face a change, with an array of effects.

Gene with no fitness advantage Another observation is that when a gene offers no advantage what so ever, it can still participate in Gene Flow. On purely statistical grounds the gene will spread through the population and after a certain amount of time, depending on the initial gene input, might enter the whole population. When the input of genes is stopped, it is extremely unlikely [1] that the gene will leave the population.

Importance Consider a for example a gene that changes the color of flowers for esthetic reasons at very low fitness cost. It this gene is introduced in a flower field for generations, it might flow out to the neighboring populations of related species. The gene will linger on, and with continued input will eventually enter the whole population. Even though it does no direct harm, it is an case of genetic pollution and loss of bio-diversity.

8.3 Conclusions

The research questions presented in the General Introduction were:

1. How can Agent Based models be used to acquire insight in the gene flow from 'genetically modified organisms', and can these models eventually be used to estimate the environmental impacts associated with GM?
2. What is the response of the stakeholders to the use of the predictive capabilities of the model in the public debate about the safety and environmental impacts of genetic modification ?

From the discussions on the sustainability of GMOs, public risk perception, Gene Flow, Agent Based Models, Gene Scape development, implementation, evaluation and consideration, the following can be concluded:

1. Agent Based Models are a useful tool for modeling Gene Flow.
2. GeneScape is a valid basis for further development of a tool for predicting environmental effects and risks from GMOs
3. Since the most important input variables governing Gene Flow are dependent on the environment, Genetically Modified Organisms can only be sustainable in a closed, controlled environment.

---

[1] but theoretically possible

4. GeneScape is not yet, but can be a societally acceptable tool.

With the above presented, it can be concluded that the project was a success

8.4 Recommendations

In the last section of this report I would like to discuss the recommendations for the future work. These present the aspects of the model that should be developed further and directions in which further research would be desirable.

8.4.1 Model improvements

There are several aspects of GeneScape that should be extended and adapted in order to improve it. These are correcting the 0 pollen distance error, refining the pollen spread model, improving hybridization behavior, adding seed movement and including F1 progeny fitness and back-crosses. I shall discuss these issues in more detail below.

0 spread distance As observed in section 7.2.3.2, Result of model behavior evaluation, Test 2, GeneScape is not able to handle a spread distance of 0. While this is not crucial for the operation of the model, it is an unpleasant error.

Improvement of pollen spread model Implementation of the Pollen Spread Distance in GeneScape means that pollen is not able to travel further than a set value. Field trials observe an average pollen spread distance, which corresponds to Pollen Spread Distance in GeneScape, but they also observe hybrids at many times grater distances as well. These hybrids are very important for Gen Flow and the extent of gene spread. Implementation of such rare extreme spread distances would add a stronger biological reality to the model.

Refinement of hybridization When selecting a mating parter, GeneScape only takes the most Attractive partner. It is often observed in nature that hybridization occurs even at very unfavorable combinations. This means that the extent of Gene Flow described by GeneScape is an underestimate. It would be interesting to implement the possibility of accidental hybridization, even when the Attractiveness is low.

Seed movement As currently modeled, there is no seed movement in the model. Where the mother plant stood, a new plant is created with the same species but with a new set of fitness genes. This solution was chosen for its simplicity, while retaining enough biological relevance. If more quantitative description is desired, seed movement should be introduced. Considering that seed movement not only changes the spatial distribution of the plants, but is also dependant on the fitness it contributes to the multiplication of the genes. This effect should be considered in future improvements of the model.

F1 progeny fitness and back-crosses Event hough there is conflicting evidence about the direction, literature suggests a change in fitness of the F1 and successive generations. A much less, or much more fit F1 progeny may have a large influence on Gene Flow. Adding parenthood tracking and related to it changes in fitness would increase the realism of the model.

8.4.2 Further research

From the evaluation of model use, especially the literature review, several aspects of the model were identified which were not suitably covered by experimental data. Further experiments in those areas would not only provide experimental data for use of GeneScape but would contribute to overall understanding of the risks of GMOs and Gene Flow.

Competitive pollen germination Especially in ecosystems near massive monoculture a wild species might experience severe pollen competition. Competitive pollen development studies might indicate the chances of hybridization, and thus the risks of Gene Flow. By examining the development of the crop pollen on the wild species stigma compared to own pollen development insight will be acquired to the extent of hybridization risk. Such data can also be directly used in GeneScape as Mating Preferences.

Fitness increasing effect of GMO genes There is a very limited number of studies examining the effect of the GMO gene on the overall fitness of the plant. Considering that the fitness increasing effect is completely dependent on the environment is very important that these studies include different environments and different selection pressures. Seed production and their fecundicity might be good metrics to examine. Such data might provide insight to how "risky" GMOs really are, since the greater the fitness increase, the greater the chance of escape. Such data can be used directly in GeneScape.

Importance of increased fitness vs. mating preference for hybridization GeneScape provided an insight that Gene Flow can be driven either by an great fitness increase by the GMO gene, or by a high degree of relatedness between species. The balance of these parameters should examined in a study comparing the Gene Flow of different fitness increasing genes within a fixed population, and the Gene Flow of a single gene in different population structures. It can provide insight in measures needed to prevent it.

Embedding of Gene Flow in QRA Further research should be done on the integration of life/information aspects in the QRA. Especially the presence of open-feedback loop, the ability to self amplify and the context dependence must be investigated in depth. So far it has proved to be very difficult, but such activities are essential.

Gene - environment interaction So far, only the spread of genes received attention. However the spread itself might not necessarily be the only problem. The translation of specific genes to environmental effect must be understood if a good risk assessment is to be made.

Social acceptance hypothesis testing For further development of GeneScape it is important to test the social acceptance hypothesis, and use the feedback from such an study to improve upon the aspects that make it unacceptable, or diminish it acceptability. Only when such an examination is done will GeneScape be able to contribute to deeper understanding of the sustainability of GMOs.

Bibliography

[1] T.F.H. Allen, J.A. Tainter, and T.W. Hoekstra. Supply-side sustainability. *Systems research and Behavioural Science*, 16:403–427, 1999.

[2] M.A. Altieri. The ecological impacts of transgenic crops on agroecosystem health. *Ecosystem health*, 6(1):13–23, March 2000.

[3] S. Barber. Transgenic plants: field testing and comercialisation including a consideration of novel herbicide resistant oilseer rape *(Brassica napus L.). Gene Flow and Agriculture: Relevance for transgenic crops. BCPC Symposium Proceedings*, 1(72):3–13, April 1999.

[4] J. Brown, D.C. Thill, A.P. Brown, C. Mallory-Smith, T.A Brammer, and H.S. Nair. Gene transfer between canola (brassica napus l.) and related wild species. *www.nbiap.vt.edu*, 1995.

[5] P. Calow. *Handbook of Environmental Risk Assesment and Management.* Blackwell Science, 2nd edition, 1998.

[6] E.P. Christmas and S.S. Hawkins. Winter canola – an alternative crop in indiana. *Agronomy Guide, Purdue University Cooperative Extension Service*, 1992.

[7] SUN Corporation. *The source of JAVA technology.* http://java.sun.com, 2000.

[8] Canadian Canola Council. Image database. *http://www.canola-council.org/*, 2001.

[9] F.B. Cross. Facts and values in risk assesment. *Reliability Engineering and System Safety*, 55:27–40, 1998.

[10] Charles Darwin. *The Origin of the Species.* Penguin books, 1985.

[11] Paul Davies. *The Fifth Miracle, the search for the origin of Life.* Penguin Books, 3rd edition, 2000.

[12] Richard Dawkins. *The Selfish Gene.* Oxfrd University press, 1990.

[13] J.A. Duke. Handbook of energy crops. *Unpublished*, 1983.

[14] B. Eckel. *Thinking in Java*, chapter 1, pages 31–32. Prentice Hall PTR, 2000.

[15] N.C. Ellstrand, H.C. Prentice, and J.F. Hancock. Gene flow and introgression from domesticated plants into their wild relatives. *Annu. Rev. Ecol. Syst.*, 30:539–563, 1999.

[16] A. Evenhuis and J.C. Zadoks. Possible hazards to wild plants of growing transgenic plants. a contribution to risk analysis. *Euphytica*, 55:81–84, 1991.

[17] R.S. Hails. Gentically modified plants- the debate continues. *TREE*, 15(1):14–18, January 2000.

[18] K. Harding and P.S. Harris. Risk assesment of the release of genetically modified plants: A review. *Agro-Food-Idustry Hi-Tech*, pages 8–13, November/December 1997.

[19] T.P. Hauser, R.G. Shaw, and H. Oestergaerd. Fitness of f1 hybrids between weedy brassica rapa and oilseed rape(b.napus). *Heredity*, 81:429–435, 1998.

[20] C. James. Global review of comercialized transgenic crops: 2000. *ISAAA Briefs*, 21, 2000. ISAAA, Ithaca, NY. USA.

[21] N.R. Jennings. On agent-based software engineering. *Artificial Intelligence*, 117:277–296, 2000.

[22] C.N. Stewart Jr., J.N. All, P.L. Raymer, and S. Ramachandran. Increased fitness of transgenic insecticidal rapeseed under insect selection pressure. *Molecular Ecology*, 6:773–779, 1997.

[23] C. Lavigne, E.K. Klein, P. Vallee, J.Pierre, B. Godelle, and M. Renard. A pollen-dispersal experiment with transgenic oilseed rape. estimation of the average pollen dispersal of an individual plant within a field. *Theoretical and Applied Genetics*, 96:886–896, 1998.

[24] J. Lovelock. *Gaia : A New Look at Life on Earth.* Oxford University Press, 3rd edition, 2000.

[25] L. Margulis. *Symbiotic Planet : A New Look at Evolution.* Perseus Books, 1st edition, 2000.

[26] W.F. Morris, P.M. Kareiva, and P.L. Raymer. Do barren zones and pollen traps reduce gene escape from transgenic crops. *Ecological Applications*, 1(4):157–165, 1994.

[27] Royal Commission on Genetic Modification. *Report of the Royal Commission on Genetic Modification.* Government of New Zeland, http://www.mfe.govt.nz, July 2001.

[28] M.T. Parker. Ascape: Abstracting complexity. *http://www.brookings.edu/ dynamics/ models/ ascape/ Abstract-Complex.doc*, pages 1–17, March 2000.

[29] I.A.P. Parkin and D.J. Lydiate. Conserved patterns of chromosome pairing and recombination in brassica napus crosses. *Genome*, 40(4):496–504, 1997.

[30] K.P. Pauls. Plant biotechnology for crop improvement. *Biotechnology Advances*, 13(4):673–693, 1994.

[31] O. Renn. *Risk Perception and risk behaviour: Proceedings of the SRA Annual Conference in Stuttgart*, volume 2, chapter 1, pages 13–76. Society for Risk Analysis-Europe, 1997.

[32] C.M. van der Bleek, P.J.T. Verheijen, and B.C. van Zomeren. *Statistiek van Proefopzetten.* Tu Delft, 6th edition, August 1996.

[33] L.L. Wolfenbarger and P.R. Phifer. The ecological risks and benefits of genetically engineered plants. *Science*, 290:2088–2093, 15 December 2000.

Appendix A

Risk perception

What is risk? A quote from Renn [31] presents a broad definition of risk:

> All risks have one element in common: The distinction between reality and possibility. If the future is either predetermined or independent of present human activities, the term ”Risk” makes no sense. Assuming that the distinction between reality and possibility is accepted, the term **”Risk ”denotes the possibility of an undesirable or desirable state of reality occurring as a result of natural events or human activities.**

So the chance of something happening becomes a risk depending on how ”desired or undesired” some event is. Therefore it is important to understand how this sense of risk is formed

Perception factors Several factors play a role in risk perception. These are according to Renn [31]:

- Attention and selection filters
- Intuitive heuristic
- Semantic images
- Qualitative risk characteristics
- Role of media

I would like to quickly discuss these issues, since they greatly help understand how and why does risk perception form. This knowledge helps place GMOs in a different light. The discussion is based on the discourse that Renn presents.

Attention and selection filters Very few risks in the modern society are experienced first hand. Mostly, people learn about them through communication. It is estimated that an average person is exposed to roughly 7000 bits of information daily. Of it, 700 are perceived, 70 acknowledged, 7 stored in the short term memory, and probably only 1 remembered in the long term. Therefore, the process of information selection is of crucial importance for risk perception. Table A.1 presents the factors influencing the selection of information.

Conditions	Elements of Conditions
Ability	Physical access to information Time to process information Absence of sources of distraction
Motivation	Reference to personal interests, salient values or self esteem Inducement of personal involvement with the issue, the content or the source

Table A.1: Conditions and requirements for information selection

From the previous it is relatively easy to see that reading a simple message like "Eating Frankenfood is dangerous" in the poster on the wall or from a demonstrators banner might cause the message to be remembered. Remembering however is not enough. Before the information is internalized, several other steps need to take place. These are described in table A.2

Often, because of lack of time, this route is short-circuited. In that case decisions are made on the base of "quick-and-dirty" considerations. These manifest themselves in the intuitive heuristic.

Intuitive heuristic Once information has been received, common sense most often processes the information, since a thorough processing like described above requires too much time. This is very important in the case of risk perception since these models relate to assessment of probabilistic information. These "common sense" models are often inadequate, since they relay on momentarily accessible knowledge. They replace knowledge with general convictions that are strongly influenced by the social context. These models crate a bias towards certain types of risk perception. These biases are presented in table A.3

These heuristic biases might not always take place, especially when enough information about a risk is available. However considering that

---

[1] This step is a crucial one. It involves the evaluation of the perceived consistency with existing beliefs (Cognitive dissonance) and with reference group judgments (to avoid social alienation).

Steps	Description
Passing of attention filters	Selection and further processing of signals from the environment
Decoding of signals	Investigation of factual content, sources of information, value statements overt and hidden intentions of sources and transmitters
Drawing one's own inferences	Conclusions about the above, applying intuitive heuristic (common sense reasoning) and judging the seriousness of information
Comparing the decoded message with encoded attitudes and beliefs	Analyze the meaning in the light of related knowledge or attitudes
Evaluating the message	Rating the importance, persuasiveness and potential for personal involvement [1]
Forming specific beliefs	Generation or change of beliefs about the subject
Taking corresponding action	Generation of intentions for future actions in accordance with above

Table A.2: Steps of individual information processing

there is almost never enough information, these biases form general public opinion, and as such they are important.

Semantic images After the information has been processed, and maybe internalized, the role of the semantic images in risk perception becomes important. Semantic images are mental stories that represent and simplify and categorize thinking about a type of risk. The four main types of semantic images in public risk perception are presented in table A.4

Next to the semantic images, individual risk estimations are dependent of several characteristics that "fine-tune" the perception.

Qualitative risk characteristics The characteristics of risk that further influence risk perception are presented in table A.5. These characteristics are partly grouped around the semantic images, and are partly dependent on the individual.

After having discussed the way risk perception is formed, it is interesting to do a "perception analysis". I will examine the image forming around the GMO issues through the issues presented above.

Biases	Description
Availability	Events that come into mind immediately are rated as more probable
Anchoring effect	Probabilities are adjusted to the information available or the perceived significance of the information
Representativeness	Singular events experienced in person or associated with properties of an event are perceived more important
Avoidance of cognitive dissonance	Information that causes cognitive dissonance will be downplayed or ignored

Table A.3: Conditions and requirements for information selection

Pending Danger (Damocles Sword)
Artificial risk source Large catastrophic potential Inequitable risk distribution Perception of randomness as threat
Slow Killers (Pandora's Box)
(artificial) ingredient in food, water, soil Delayed effects, non-catastrophic Contingent on information rather than experience Quest for deterministic risk management Strong incentive for blame
Cost-Benefit ration (Athena's Scale)
Confined to monetary gains and losses Orientation towards variance of distribution rather than expected value Asymmetry between risks and gains Dominance of probabilistic thinking
A vocational Thrill (Hercules Image)
Personal control over degree of risk Personal skills necessary to master danger Voluntary activity non-catastrophic consequence

Table A.4: Semantic images of public risk perception

Qualitative Characteristic	Influence of perception
Personal control	Increases risk tolerance
Institutional control	Depends on the confidence in institution
Voluntariness	Increases risk tolerance
Familiarity	Increases risk tolerance
Dread	Decreases risk tolerance
Inequitable distribution or risk and benefits	Depends on individual utility
Artificiality of risk source	Amplifies attention to risk, often decreases risk tolerance
Blame	Increases quest for social and political response

Table A.5: Qualitative risk characteristics and their influence of perception

Appendix B

Scope of GMO application

In this Appendix I will examine and present the most recent figures on GMO crops. Total areas, growth, divisions per country, trait, species and similar are presented. these data are meant to present the reader with a feeling of the GMO situation in the world.

Species From the first experimental release of transgenic crops in 1986, more than 60 species have been transformed using GM techniques. Most of these are not commercialized [3]. 98.3 % of all experimental releases of GM organisms, are transgenic plants. The most numerous experimental releases are presented in table B.1.

Species	% share
Corn	38
Rape seed	13
Potato	12
Tomato	10
Soybean	9
Cotton	7
Tobacco	5
Sugar beet	2
Other	4

Table B.1: Most common experimental releases in [3]

The *Other* category contains 50 species, ranging from kiwi fruit, papaya, mustard, grapes. strawberry, pine trees and flowers.

Total area The estimated global area of transgenic crops for 2000 is 44.2 million hectares (442000 km^2) [20]. This amounts to little more than 10 times the area of the Netherlands [1] . This area grew 25-fold since 1996, and

---

[1] Area of the Netherlands is 41.532 km^2

by 11 % since 1999.

Growth Total transgenic crop area is estimated to have increased from 39.9 million hectares in 1999 to 44.2 million hectares in 2000 (11% growth over 1999). Of this area, 3.6 million hectares (84%) was in the developing countries, and only only 0.7 million hectares (16%) in the industrial countries.

1st vs. 3rd world The number of countries growing GMO crops grew from 6 in 1996 to 13 in 2000 [20] From 1996 to 2000 up to 85% of global transgenic crops has been grown in industrial countries. However, the proportion of transgenic crops grown in developing countries has increased consistently, see table B.2

Year	% share
1997	14
1998	16
1999	18
2000	24

Table B.2: Proportion of the total GMO crop area in the third world [20]

Thus, in 2000 approximately one quarter of the global transgenic crop was grown in developing countries. The growth continued to be strong between 1999 and 2000, in contrast to the expected plateauing that is evident for the industrial countries [20].

The area of transgenic crops in developing countries grew by 51% from 7.1 million hectares in 1999 to 10.7 million in 2000, compared with a 2% growth in industrial countries where it increased from 32.8 million hectares in 1999 to 33.5 million hectares in 2000.

Per country In 2000, transgenic crops were grown commercially in all six continents of the world. Of the top four countries that grew 99% of the global transgenic crop area, the division is presented in table B.3.

Country	% share
USA	68
Argentina	23
Canada	7
China	1
Other	1

Table B.3: Distribution of the global GMO area by country [20]

The *Other* 1% was grown in the remaining 9 countries, with South Africa and Australia being the only countries in that group growing more than 100,000 hectares of transgenic crops.

Per crop The distribution of the global transgenic crop area per crop in year 2000 is presented in table B.4 [20]:

Crop	% share
Soybean	58
Corn	23
Cotton	12
Rape seed	6
Other	1

Table B.4: Division of global GMO area by crop [20]

By Trait The division of the main traits is presented in table B.5

Trait	% share
Herbicide tolerance	74
Insecticide resistance (Bt)	19
Stacked resistance	7
Rape seed	6
Other	1

Table B.5: Distribution of area per trait in 2000 [20]

It is interesting to note that the area of herbicide tolerant crops has increased between 1999 and 2000 by 14 % (28.1, to 32.7 million hectares), the crops with stacked genes for herbicide tolerance and Bt by 9 % (2.9 million hectares in 1999 to 3.2 million hectares in 2000), whereas the global area of insect resistant crops has decreased by 8% (from 8.9 million hectares in 1999 to 8.2 in 2000). The trend for stacked genes to gain an increasing share of the global transgenic crop market is expected to continue.

Adoption of Transgenic crops Finally, it is interesting to see the degree that GMO crops have penetrated in to the main crop use. The data are presented in table B.6

If the global areas of these four crops are aggregated, the total area is 271 million hectares, of which 16%, (44.2 million hectares) is estimated to be transgenic.

Crop	% GMO
Soybean	36
Cotton	16
Rape seed	11
Corn	7
Global	16

Table B.6: Fraction of GMO crop in 2000 [20]

Appendix C

Java Object properties

In this Appendix I will discuss the properties of Objects as presented in section 4.2.

Everything is an Object In an Object Oriented language, everything that can be seen is an Object. This practically means that every part of a model or program must be defined as a unit with clear boundaries and functionality. In itself it is not a very drastic thing, but coupled with the other four aspects the power of Objects emerges.

Objects can communicate This aspects relates to the interaction between Objects. Each Object can send a request to another Object and it can conversely accept a request from another one. Since Objects contain methods, i.e. can do something, one object can request an Object to perform a Method, and the results of that action can be communicated back to the requestor.

Object have a state The aspects I will discuss last is the notion of states. Next to being a bounded unit being able to do something, every object can contain information. Objects can be used as data storage elements, just as they can be used as packaging or tools.

Every object has a type Each Object has a property called Type. The type is a description of what kinds of states and interactions the Object is capable of. For example, and Object of Type ”Bag” is something that can contain items, has a capacity and has the ability to have items placed in and taken out through the opening. An Object of Type ”Dog” is able to run, fetch a ball, wiggle its tail when given food, bark when kicked and sleep when tiered.

Communication is standardized within a type This property means that every Object of a certain Type will communicate the same way. That is, putting an item in a bag is done basically the same way for every bag, and every dog will fetch a ball if it is thrown at it.

Appendix D

GeneScape Parameters

Field Height This parameters sets the height, or the Y axis, of the Field, i.e. it sets the size of "Nature". It is expressed in a number of plants. If modeling of an actual field is desired, estimates of number of plants per meter must be made. The reason is that, since GeneScape is an individual-explicit model, it as of yet can not deal with physical length. The default value is 60 plants. This is chosen purely as a convenient value because it fits on the most common screen resolutions.

Field Width This parameter sets the width, or the X value, of the field. The field does not have to be rectangular. The default value is 60 plants.

GMO Field Height This parameter describes the height of the GMO "patch" in the middle of the field. As the Field Height and Width, it is expressed in a number of plants. The default value is 6. It is chosen in accordance with the maximal Pollen Spread distance, so that the length of the GMO field is at least of the same size as the maximal spread. This is done in order to assure that the GMO plant in the middle of the field would only experience pollen form other GMO plants. This situation resembles the real field situation the most. Since usually the GMO planting rule is active, this effect is not noticeable, but if a user decides to let the GMOs be planted only once, there must be at least one GMO plant that has to mate with another GMO.

GMO Field Width This parameter describes the with of the GMO "patch". The rationale is the same as with the GMO Field Height. The default value is 6.

Fitness Of Gene A This parameter describes the extra fitness increase of an plant when it possesses a single copy of gene A. The total Fitness increase is a sum of all gene fitness effects. Fitness increase can also be negative, for simulating a gene that has a fitness reducing effect. It is important to notice that a plant with a fitness of 0 is not "dead" but it has no genes that would extra increase its fitness, consequently, negative total fitness means a reduced overall fitness. The default value is 15. This value is arbitrary, and should be set relative to the other genes. It should be chosen keeping the relationship to the Fitness To Pollen factor.

Fitness Of Gene B This parameter describes the fitness increase effect of the gene B. The rationale is identical to the one of gene A. Its default value is 5

Fitness Of Gene GMO This parameter describes the fitness-increase effect of the GMO gene. There is no essential difference between this gene and gene A or B, except that plants that carry the GMO gene are presented in red color, and the changes in the possession of this gene are easily observable. The default value is 10.

Pollen Determination Model Used This parameter describes whether the Number of Pollen Determination model should be used. This model translates the total fitness of a plant to a number of pollen that plant can produce. The parameter options are 0 or 1. For the time being, Pollen Determination Model is a simple proportionality, that uses the Fitness To Pollen Factor to multiply the Total Fitness to get the number of pollen that a plant may produce. The default value of this parameter i 0. In this case, the number of pollen is equal to the total fitness value.

Fitness To Pollen Factor This parameter describes the factor that the total plant fitness will be multiplied by to get the total number of pollen produced. The default value is 1.

Mutation Rate This parameter describes the chance that a plant will experience a mutation during Growth phase. It is a number between 0 and 1. Mutation is a flipping of a gene's state (from 0 to 1 and vice versa) in the Fitness determination part of the genome. The default value is 0.

Pollen Spread Distance This parameter describes the distance in number of Plants that the pollen will be able to reach. It is the distance that the Egg can "look" to find a mating partner. The default value is 5 plants. There is no upper limit, but the calculation time rises very quickly with the number. This parameter greatly influences behavior, especially is the Pollen Spread Model used is turned off, see next paragraph.

Pollen Spread Model Used This parameter, which can have the value of 0 or 1 determines whether the Pollen Spread Model will be used. Model assumes a 1/distance reduction in Pollen availability/ The distance is measure from the plant of origin, until the Pollen Spread Distance, after which it become 0. If the Model is turned off, pollen is equally available all through the circle defined by the Pollen Spread Distance. The default is off, or 0.

Preference X And Y Mating This parameter describes the relative mating preference for mating between tow species. The X represents the Egg, Y the Pollen. Mating between the egg and the pollen of the same species is take to be the maximum, and other values should be related to. Preference of 0 makes mating impossible. Asymmetric preference are possible. Defaults are arbitrary and symmetric. Defaults are presented in table D.1.

	G	W	R	N
G	0	8	1	0
W	8	10	2	0
R	1	2	10	0
N	0	0	0	10

Table D.1: Preference X and Y mating parameter defaults

Field Layout The Field Layout chooses between two possible Field setups. The default 0 sets up a field with the dimensions of Field Height and Field Width with a GMO Crop patch in the middle with dimensions of GMO Field Height and GMO Field Width. Option 1 sets up a border of 40 pants wide around the GMO crop patch.

Relatedness Case Relatedness Case option can have values between 0 and 6. The relatedness cases are described in tables 6.2 and E.1. Case 0 is the default test case as presented above, and 6 chooses the Cases study values as presented in table 7.3.

Population Structure Case The Population Structure Case offers the values of 0 and 1. 0 chooses the default distribution of species in the Field. In this case Wild, Remote and Non -related they are chosen at random with an equal probability. If the option 1 is chosen the plants are distributed with a 1/6 Wild, 2/6 Remote and 1/2 Non-related ratio.

Appendix E

Model Evaluation parameters

In this Appendix the values of the mating preferences for the Relatedness Levels and the model parameters used in th Range of Behaviours analysis, presented in section 6.2.2.2, are given.

The relatedness cases are defined as follows :

Mating preference	I	II	III	IV	V
GMO And GMO	10	10	10	10	10
GMO And Non	0	0	0	0	10
GMO And Remote	0	1	5	7	10
GMO And Wild	0	3	5	8	10
Non And GMO	0	0	0	0	10
Non And Non	10	10	10	10	10
Non And Remote	0	0	0	0	10
Non And Wild	0	0	0	0	10
Remote And GMO	0	1	5	7	10
Remote And Non	0	0	0	0	10
Remote And Remote	10	10	10	10	10
Remote And Wild	0	3	5	8	10
Wild And GMO	0	3	5	8	10
Wild And Non	0	0	0	0	10
Wild And Remote	0	3	5	8	10
Wild And Wild	10	10	10	10	10

Table E.1: The Relatedness Level Values

Parameter values

Parameter	Value
Stop Period	20
Field Height	60
Field Width	60
Fitness Of Gene A	15
Fitness Of Gene B	5
Fitness To Pollen Factor	1
GMO Field Height	6
GMO Field Width	6
Mutation Rate	0
Pollen Determination Model Used	0
Pollen Spread Model Used	0
Field Layout	0
Population Structure Case	0

Table E.2: Parameters values during behavior analysis

Test	Fitness	Spread distance	Relatedness level
1	10	5	I
2	10	0	IV
3	0	5	V
4	100	5	IV
5	0	5	IV

Table E.3: Consistency test parameters

Appendix F

Model reproducibility analysis

	Variable 1	Variable 2
Mean	1505.32	1516.76
Variance	23258.058776	23634.594286
Observations	50	50
Pooled Variance	23242.543838	
Hypothesized Mean Difference	0	
df	98	
D =	0.373559	
P(D≤t) one-tail	0.35477	
t Critical one-tail	1.660551	
P(D≤t) two-tail	0.70954	
t Critical two-tail	1.984467	

Table F.1: *t*-test for difference in averages. Analysis of Number of Plants With GMO Gene

	Variable 1	Variable 2
Mean	30.962384	30.969395
Variance	0.015743	0.017156
Observations	50	50
Pooled Variance	0.016296	
Hypothesized Mean Difference	0	
df	98	
D =	0.273345	
P(D≤t) one-tail	0.392582	
t Critical one-tail	1.660551	
P(D≤t) two-tail	0.785163	
t Critical two-tail	1.984467	

Table F.2: t-test for difference in averages. Analysis of Average Population Fitness

Number of GMOs I	Number of GMOs II	AVG Fitness I	AVG Fitness II
1587	1677	31.03090847	31.082650273
1592	1599	30.991632514	31.129952186
1569	1595	31.113900273	30.983606557
1542	1587	30.929986339	30.90385929
1578	1521	30.977459016	30.952527322
1381	1668	30.90932377	31.165642077
1497	1598	31.034665301	31.034665301
1727	1373	31.059938525	30.815061475
1376	1087	30.827356557	30.657445355
1604	1581	31.004610656	31.107240437
1443	1673	30.764685792	31.090334699
1578	1164	30.96840847	30.694672131
1285	1607	30.890368852	31.043203552
1411	1557	30.785519126	31.092554645
1619	1565	31.103312842	31.090846995
1449	1478	31.042349727	31.021174863
1462	1564	30.950648907	31.002561475
1622	1681	30.999146175	31.058401639
1418	1564	30.795252732	30.912226776
1584	1094	30.97045765	30.716530055
1792	1471	31.121072404	30.847165301
1642	1573	31.040642077	31.00034153
1510	1419	31.037568306	30.982752732
1182	1275	30.754269126	30.772711749
1699	1377	31.122780055	30.883709016
1820	1713	31.134392077	31.128756831
1238	1682	30.764856557	31.075990437
1307	1445	30.821892077	30.856045082
1489	1321	30.986338798	30.811304645
1815	1751	31.22148224	31.057547814
1423	1475	30.842896175	30.998804645
1447	1643	30.938353825	31.044569672
1673	1661	31.102800546	31.204918033
1606	1475	31.115095628	30.929474044
1579	1588	31.011270492	30.895833333
1493	1449	30.932889344	30.896687158
1650	1694	31.12260929	31.192110656
1539	1351	31.046789617	30.809084699
1318	1475	30.862192623	31.060280055
1394	1462	30.860484973	30.935621585
1336	1608	30.795935792	31.025614754
1322	1675	30.871243169	31.122438525
1565	1439	31.03227459	30.944501366
1592	1603	30.981557377	31.044740437
1393	1628	31.015368852	30.981898907
1377	1299	30.889515027	30.756318306
1309	1555	30.848189891	30.894125683
1410	1488	30.761270492	30.945355191
1722	1397	31.182718579	30.823428962
1300	1613	30.750512295	30.992486339

Table F.3: Data for t-tests

Appendix G

Parameter Sweep Results

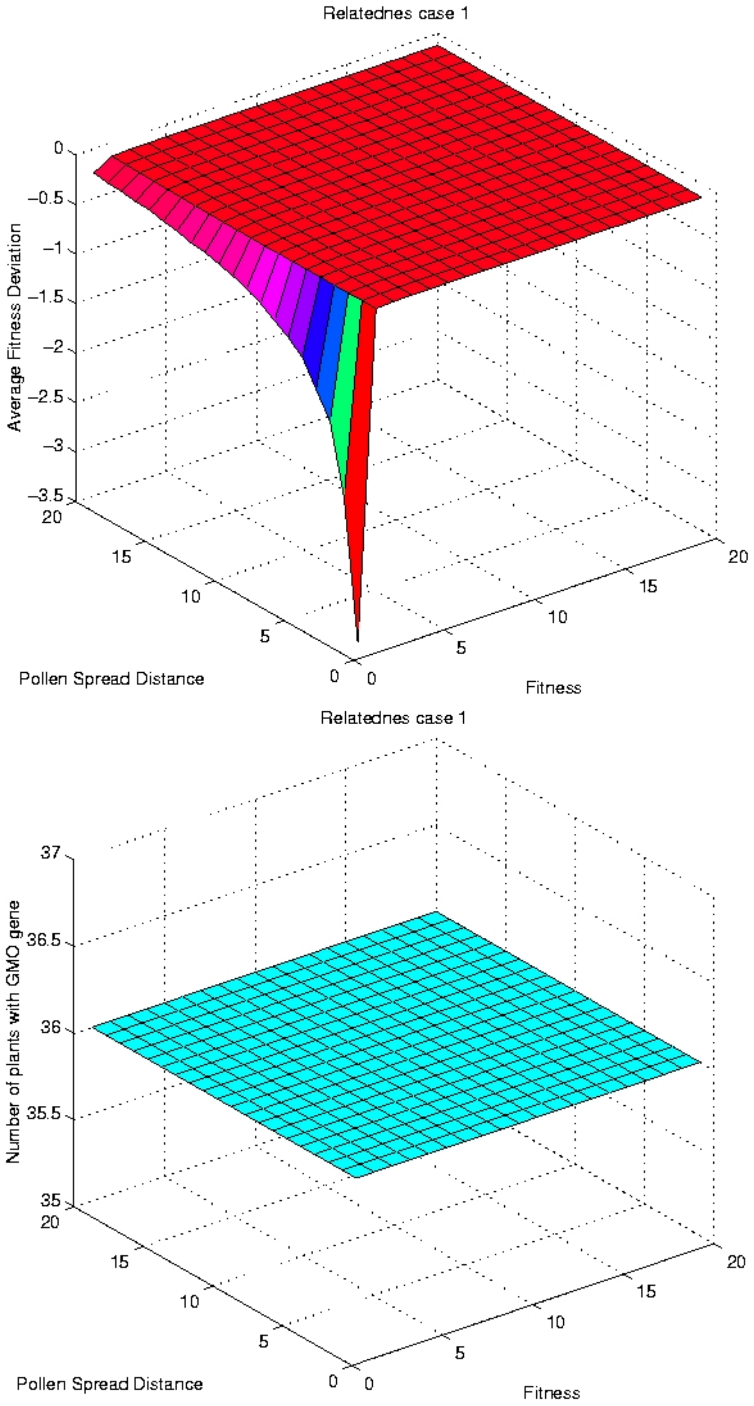


Figure G.1: Parameter sweep results, Relatedness level I, 20 generations.

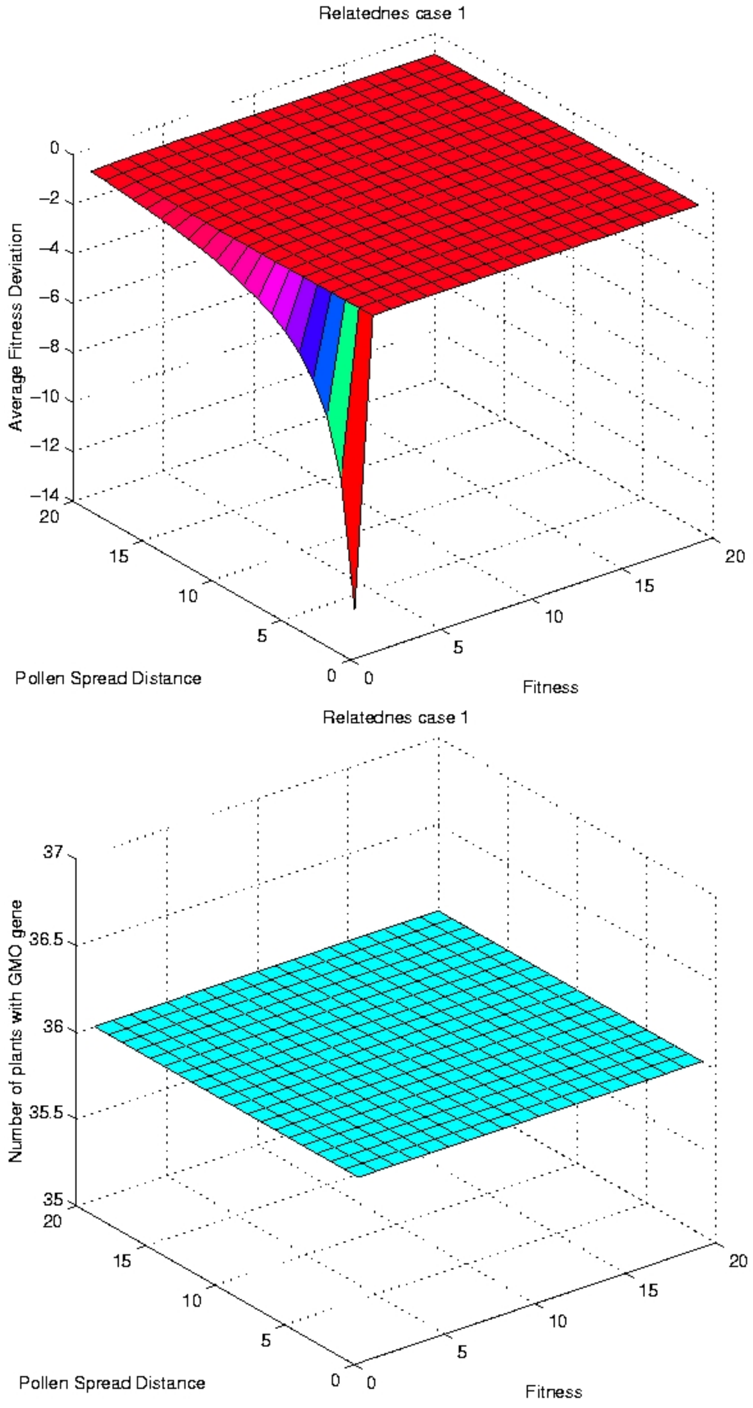


Figure G.2: Parameter sweep results, Relatedness level I, 2 generations.

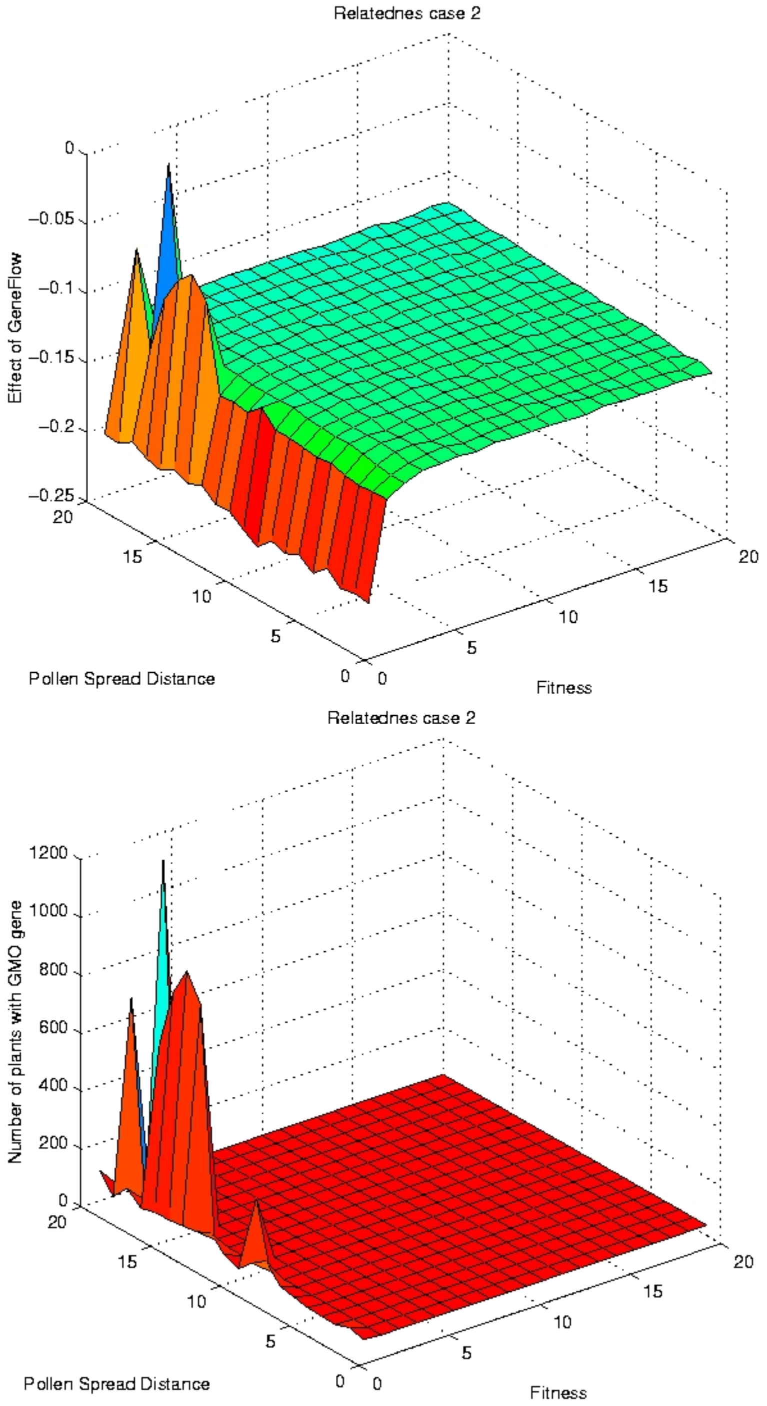


Figure G.3: Parameter sweep results, Relatedness level II, 20 generations.

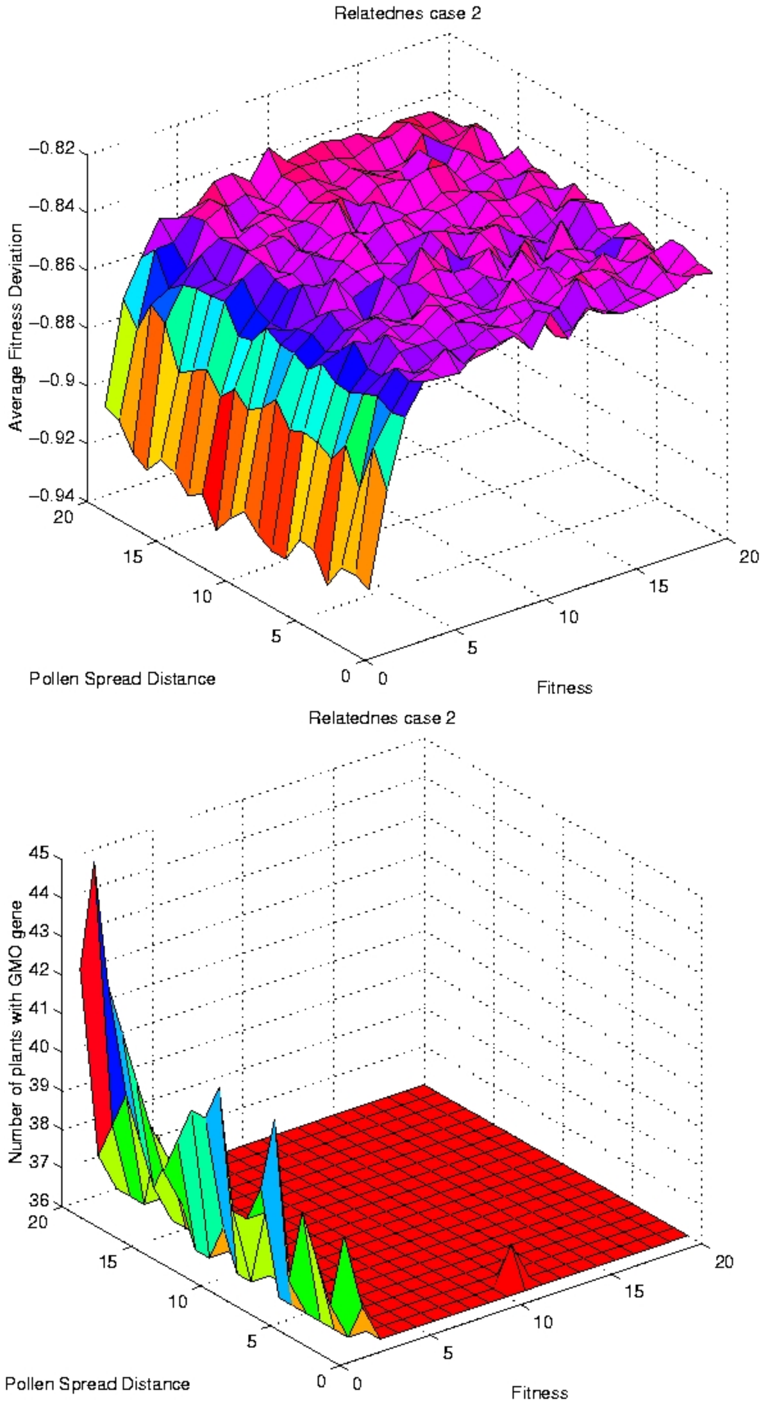


Figure G.4: Parameter sweep results, Relatedness level II, 2 generations.

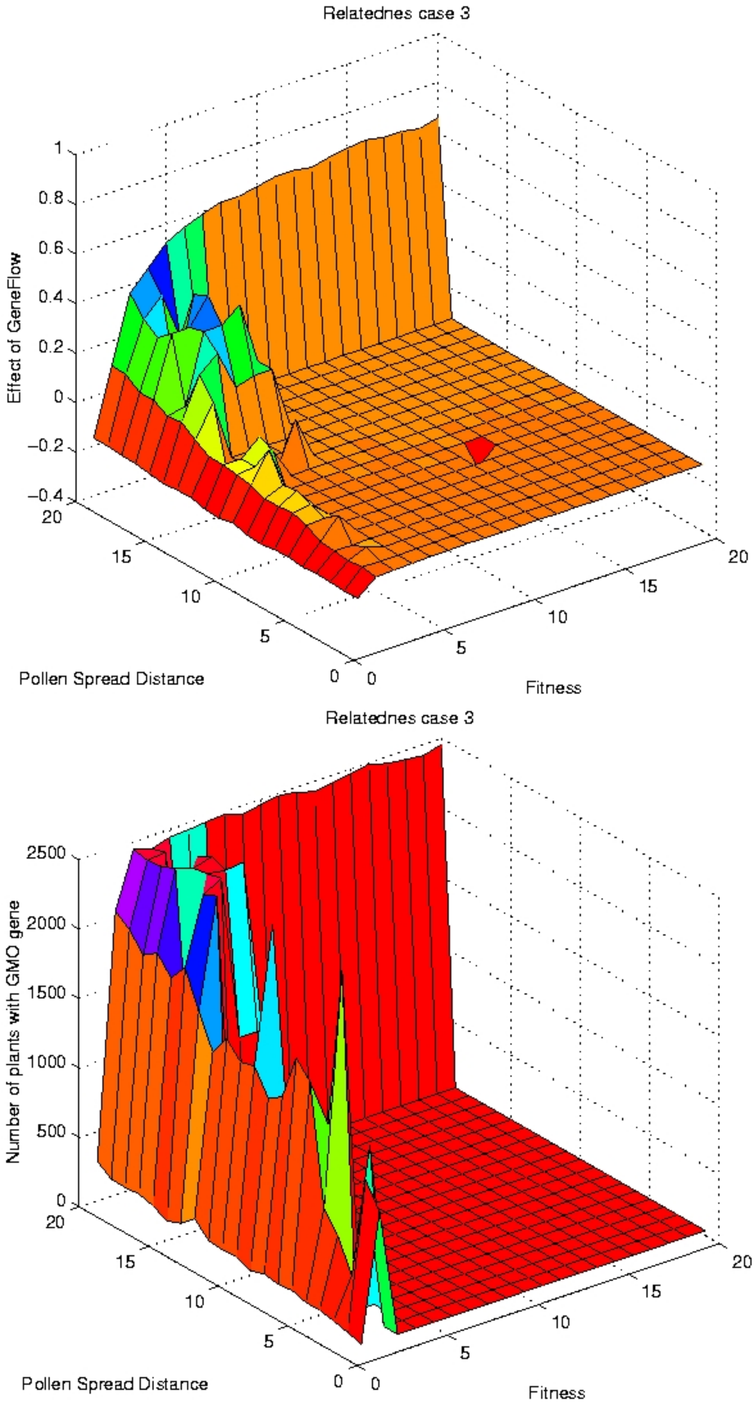


Figure G.5: Parameter sweep results, Relatedness level III, 20 generations.

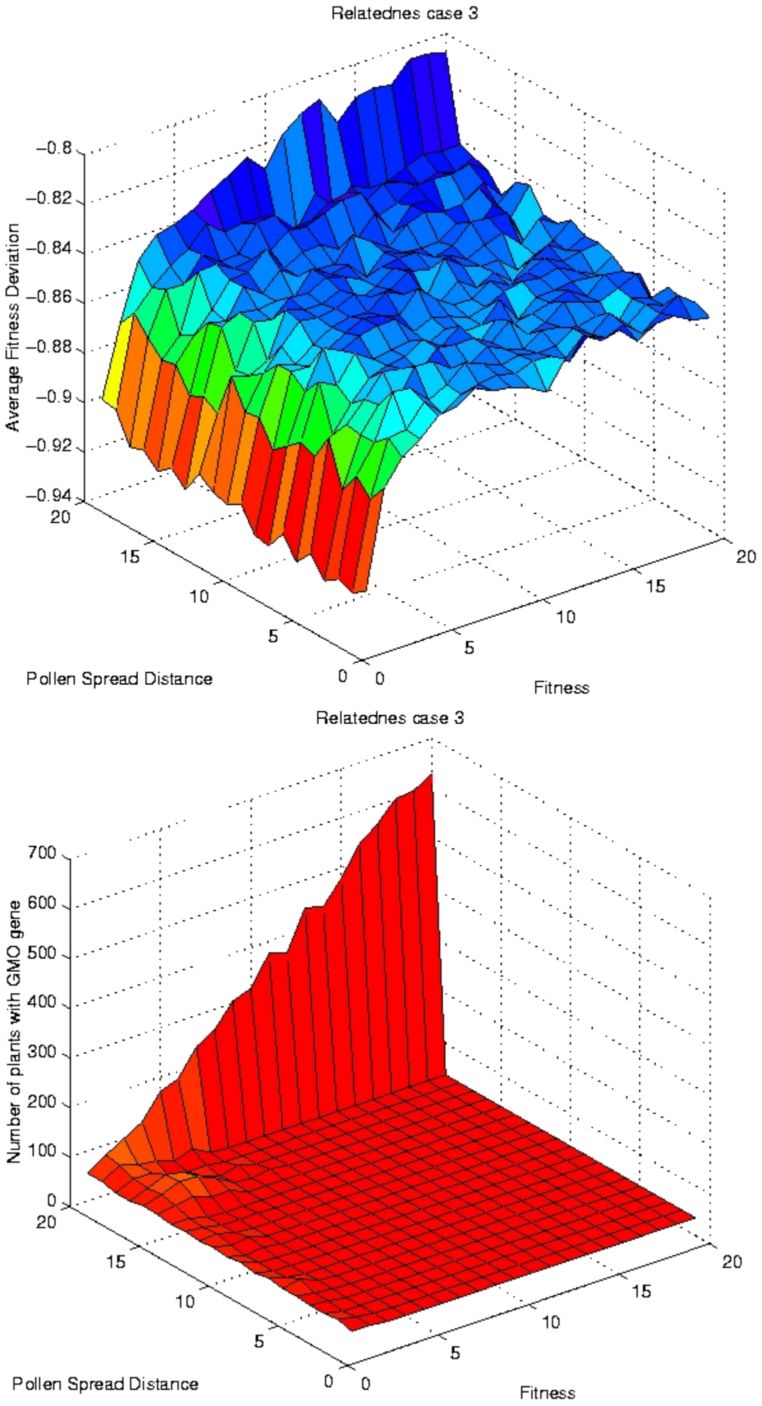


Figure G.6: Parameter sweep results, Relatedness level III, 2 generations.

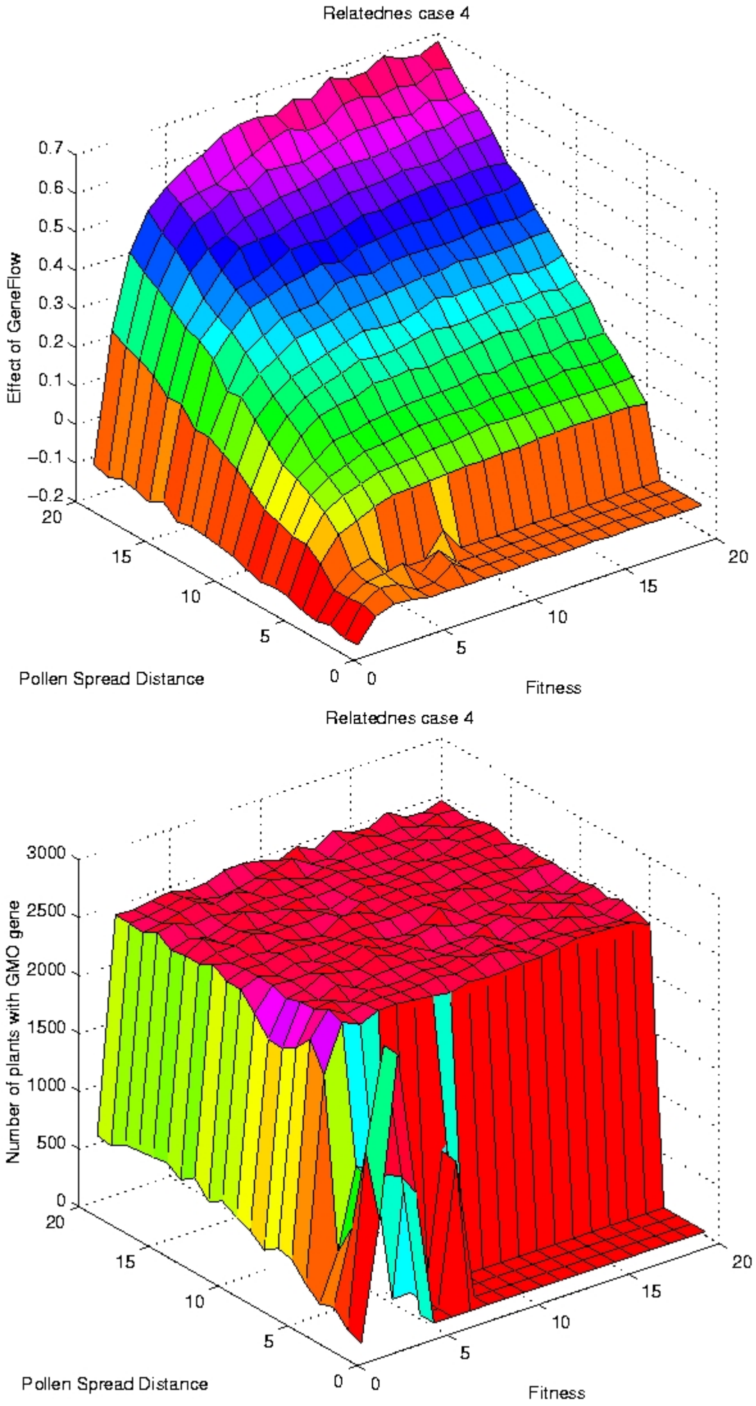
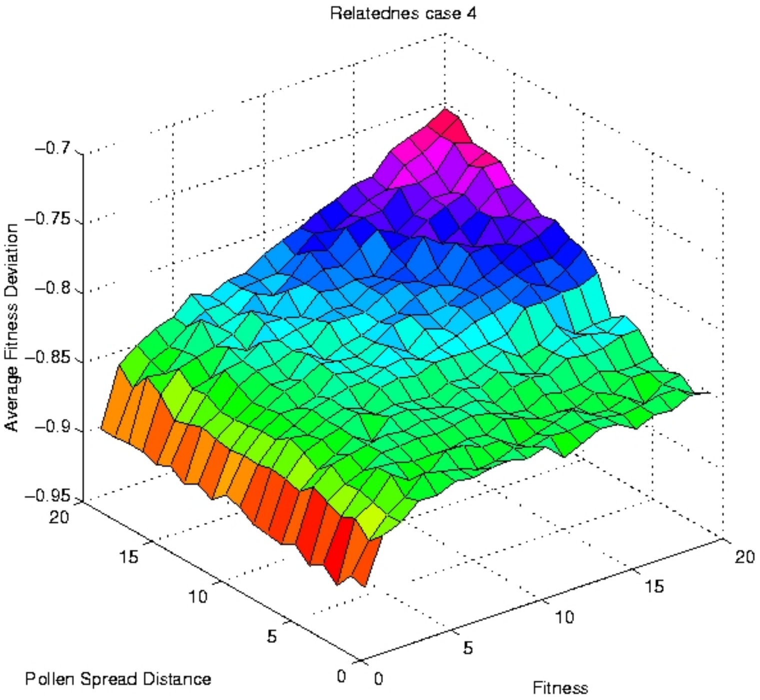


Figure G.7: Parameter sweep results, Relatedness level IV, 20 generations.



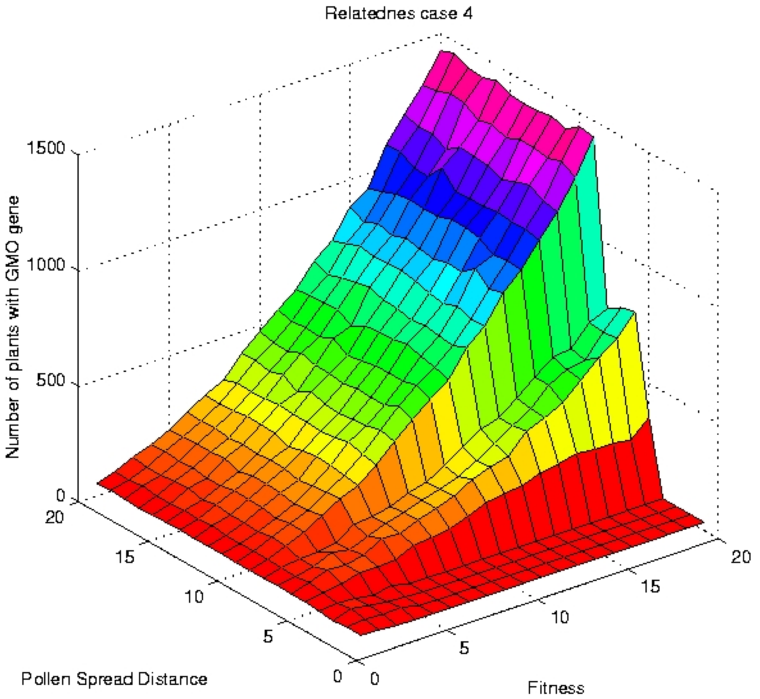


Figure G.8: Parameter sweep results, Relatedness level IV, 2 generations.

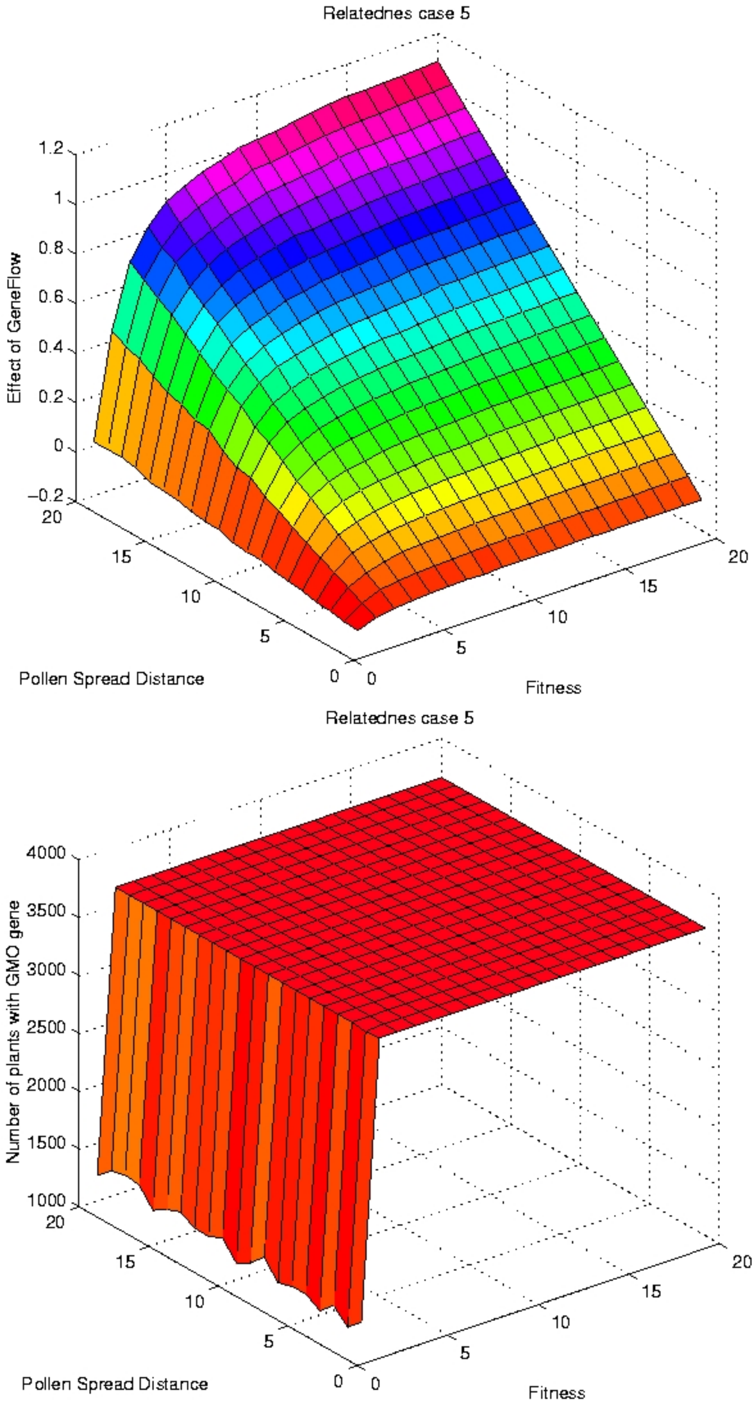


Figure G.9: Parameter sweep results, Relatedness level V, 20 generations.

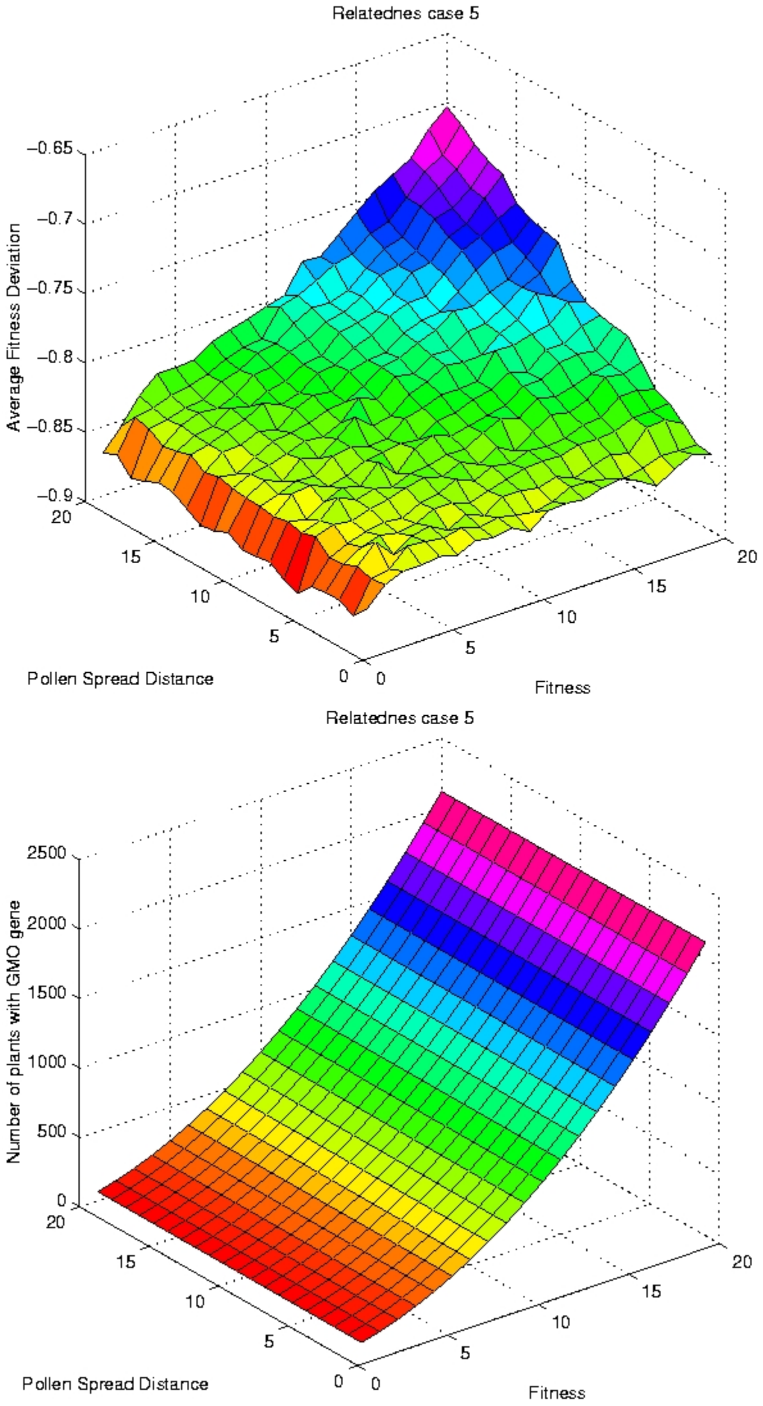


Figure G.10: Parameter sweep results, Relatedness level V, 2 generations.

Appendix H

GeneScape Pseudo Code

In this Appendix I will present the GeneScape Pseudo code. It is meant as a explanation of the software structures for the reader that is not fammiliar with Java terminology.

Notation Referring to the type of an Agent or Scape is done with a capitalized name, like in Plant, but when referring to a specific Plant object, lowercase is used, like in a plant. So, a plant is an instance of the Plant object. A variable is written with lowercase letter, with all the other words capitalized, i.e. thisIsAVariable, methods (functions) are denoted by () behind their name, i.e. thisIsAMethod().

The names of the corresponding variables and methods are placed in the brackets in sans serf font behind the pseudo code so that the reader can refer to the source code given in the Appendix I.

GeneScape pseudo-code

- Set GeneScape as an Agent or Scape that can contain other agents in an ordered manner (class GeneScape extends ScapeVector)
- Set what each position of the genome means (genomeStringExplanation)
- Set the genome of the GMO plant (genomeStringGmo)
- Set what is the fitness increasing effect of each gene (fitnessOfGenes)
- Set the size of the "Nature" (fieldWidth,fieldHeight)
- Set the size of the GMO field (gmoFieldWidth,gmoFieldHeight)
- Set the mating preferences (gmoAndGmoMatingPreference,...)

- Set the type and create the Field agent that will contain the Plant Agents. Field Agent is a collection of agents with a Moore neighbourhood that has "sqyare" surrounding. (ScapeArray2DMoore field = new ScapeArray2DMoore())

- Create a new plant, of the type Plant, that will be defined later (Plant plant = new Plant())

- Clear the default built-in rules of the Field (field.getRules().clear())

- Set the size of the Field, depending on the sizes provided by the GeneScape (field.setExtent(new Coordinate2DDiscrete(fieldWidth, fieldHeight)))

- Set the name of the geometry Scape/Agent (field.setName("Field"))

- Setting the type of "Inhabitants" the Field contains and populating the geometry with the agents (field.setPrototypeAgent(plant))

- Set the order by which the agents in the geometry will be iterated (field.setExecutionOrder(Scape.RULE _ ORDER))

- Set the periodicity (continuousness) of the Field Scape to non-continuous (field.getGeometry().setPeriodic(false))

- Adding the Field to GeneScape, and establish the hierarchical links (addAgent(field);)

- Define the Rules that the Plants must follow

  - Planting GMOs rule: Request a Plant agent to "Plant" itself if it is a GMO. See the Plant code how to be "Planted". (field.addRule(new Rule("Plant GMO Crop"))
  - Growth rule : Request a Plant to "Grow". See Plant code how to "Grow" (field.addRule(new Rule("Plant Growth"))
  - Pollen and Egg production rule: Request a Plant to produce the Pollen and Egg. See Plant code on how to produce an Egg and Pollen (field.addRule(new Rule("Pollen and Egg production"))
  - Mating Rule : Request a Plant to Mate (field.addRule(new Rule("Mating"))

- Rest of GeneScape code sets Statistics, Views, Charts and communication protocols between the Plant, Field and GeneScape

Plant pseudo-code

- Create a Plant agent that is of type CellOccupant (class Plant extends CellOccupant)
- Set the states/information content of a plant (fitness,numPollen,attractiveness,isWild...)
- Initialize the plant that has just been created: (initialize())
    - Choose a random species (switch (randomToLimit(...)
    - Assign a random Genome (randomFitnessGenome())
    - Set the plant states depending on the Genome (setPlantState())
- Method for setting of Plant states (setPlantState())
    - Determine on which position the genes are that determine the species (speciesGeneCounter())
    - Set the species flag for this plant (this.setWild(),this.setRemote() etc...)
    - Determine which species the pollen of this plant will have (this.pollenIsSpecies =...)
    - Determine whether the GMO gene is present, and set a flag if so (fitnessGeneCounter())
- Method for setting of Egg states (setEggState())
    - Same as setPlantState, except is applies to the Egg. Might be useful is future versions if Egg needs to be treated differently.
- Method for setting of Pollen states (setPollenState())
    - Same as setPlantState, except it is applies to Pollen. Might be useful is future versions if Pollen need to be treated differently
- Rule implementation: Planting of GMOs (plantGMO())
    - If this plant is located within the boundaries of the GMO field as defined by GeneScape (if(location[0] >=...)
        - Replace its genome with that of a GMO crop and set the plant flags (this.genome = GeneScape.genomeStringGmo)
    - Otherwise do nothing.
- Rule implementation: Plant growth (grow())
    - Set the state of the plant (setPlantState())

  - If mutation is turned on, evaluate wether this plant should experience a mutation
    - If yes, call the mutate method on the genome of this plant (mutate())
  - Evaluate the fitness of the plant from its genome (fitnessOf())
  - Execute the determineNumPollen method to relate the Fitness to the number of Pollen (determineNumPollen())

- Determine the Number of Pollen produced (determineNumPollen())
  - If the PollenDeterminationModelUsed flag is set off in GeneScape
    - The number of Pollen plant produces is equal to the Fitness
  - If it is on, multiply the number of pollen by the fitnessToPollenFactor set by GeneScape.

- Mutation (mutate())
  - Take in a string supplied
  - Choose a random position in the fitness determining part of the genome to mutate.
  - Turn the 1 at that position to 0, and vice versa

- Fitness evaluation (fitnessOf())
  - Check every position in the fitness determining part of the genome.
  - If 1 is present on the location, read the fitness increasing value from GeneScape fitness definition, and add it to the overall fitness.

- Rule implementation: Pollen and Egg production
  - Pollen production
    - Execute the mendeleanSplitOfGenome method on the genome of this plant (mendeleanSplitOfGenome())
    - Set the Pollen State of the plant (setPollenState())
  - Egg production
    - Execute the mendeleanSplitOfGenome method on the genome of this plant (mendeleanSplitOfGenome())
    - Set the Egg State of the plant (setEggState())

- Mendelean split of the genome (mendeleanSplitOfGenome())
  - Take the two positions comprising the gene, choose one in random, and add its contents it to the new haploid string (splitGenome[positionHolder = stringToSplit[...)
  - Erase the 1's in the species determination part of the haploid string, and add a 1 on the position of the species that the parent was.
- Rule implementation: Mating (mating())
  - Create the genome of the new plant by fusing the Egg of this plant with the Pollen of the best mate (this.genome = fusionOfGenomes(this.egg, bestMate().pollen))
- Finding the best mating partner (bestMate())
  - Ask the Field for all Plants in the neighbourhood at a distance set in GeneScape by pollenSpreadDistance (plantsNear = getCellsNear(...)
  - If pollenSpreadModelUsed is turned off
    - Evaluate the Attractiveness of each neighbour by multiplying the number of Pollen of that neighbour by the Mating Preference of this plant for the species of the neighbour (matingPreference())
  - If pollenSpreadModelUsed is turned on
    - Attractiveness is calculated as above, and additionally divided by the distance to the neighbour
  - The neighbour with the highest Attractiveness is chosen for mating
  - If all Attractivenesses are equal to 0
    - Return yourself as the best mate (self-pollination)
  - If there is more than one neighbour is having the maximum attractiveness
    - Choose one of the maximally attractive plants at random to be the mate
- Mating preference determination (matingPreference())
  - If the species of this Egg is Non
    - And the species of the other is GMO, than take the Non-GMO mating preference form GeneScape

    - And the species is Wild, than ....
  - This is repeated for all species

- Fusion of Genomes (`fusionOfGenomes(...)`)
  - Take the two supplied genomes
  - Take the first letter in the first genome and place it on the first place of the new genome.
  - Take the first letter of the second genome and place it on the second place of the new genome.
  - Repeat until end

Rest of Plant code sets the communication protocols between the Plant, Field and GeneScape, and describes which species should be coloured in which colour, and handles trivial issues like random genetic code creation, counting of the number of species and fitness genes present and such.

Appendix I

GeneScape code

```
import java.awt.*;
import java.util.*;
import java.io.*;
import edu.brook.ascape.model.*;
import edu.brook.ascape.rule.*;
import edu.brook.ascape.util.*;
import edu.brook.ascape.view.*;

/**
* GeneScape.java
*
* Created on March 26, 2001, 9:32 AM
*
*
* @author  Nikolic
* version 1.4.2
* Species conservation through pollen and egg is implemented.
*
* This version implements an improved mating code that does not prefer the first quadrant if all the
* mates are the same
* Further, it implements mutation of the fitness part of the genome.
* and the field is set to "non-periodic", interface , no wrapping. it is no longer a torus, but a square.
* This way edge effects can be observed
*
* Field is initialized with clones of the different species. They have exactly the same
fitness genes.
 * This causes behaviour to be
* very uniform (ie,  a more remotely related species will neve be considered a mating pattner,because
* the preference sees exactly the same fitnesses
* In this version, the fitness genes will be determined at random during initialisation , just as the species
* This should create far more 'interesting"behaviour , which is more life like, since not all plants in a
* population are copies of each other
* Done.
*
* Another posibility is to let plants plant the seed at a different location than the mother plant was.
* Question is wether this is necessary
*
* The ability to turn off distance added in 1.4.1
*
* Version 1.4.2 carries the data saving and parameter sweep parts
*
*/

public class GeneScape extends ScapeVector {


// protected boolean turnOnCharts = false;
      protected boolean turnOnCharts = false;

// Should we do a parameter sweep ?
      protected boolean doTheSweep = false;

// Should we save data as a time series or not ?
      protected boolean saveRunEndDataOnly = false;

// Wether searching for a mate should use the pollen spread model
      protected int pollenSpreadModelUsed = 0;
```

```
//Wether the numOfPollen should use the pollen number detrmination model of not
     protected int pollenDeterminationModelUsed = 0;

/**
 * Genome definition: array of integers [SpeciesG SpeciesG SpeciesW SpeciesW SpeciesR SpeciesR
 * SpeciesN SpeciesN GeneG GeneG GeneA GeneA GeneB GeneB ]
 * The number of gene coding bits MUST be 2 ! (ie, 11, or 00)
 * If the number, or the  order of species is changed, the Plant.setState() must be adapted !
 */
    public static String[] genomeStringExplanation = {"GMO", "GMO", "W", "W", "R", "R", "N", "N",
    "G", "G", "A", "A", "B", "B"};
        //public static final int[] genomeStringGmo = {1,1,0,0,0,0,0,0, 1,1,0,0,0,0};
        public static final int[] genomeStringW =   {0,0,1,1,0,0,0,0, 0,0,1,1,1,1};
        public static final int[] genomeStringR =   {0,0,0,0,1,1,0,0, 0,0,1,1,1,1};
        public static final int[] genomeStringN =   {0,0,0,0,0,0,1,1, 0,0,1,1,1,1};
       // public final int[] genomeStringTest =   {1,0,0,0,0,0,0,1, 1,0,0,0,0,1};

/**
* The GMO plant is as fit as natural and it has an extra gene for fitness
*/
        public static final int[] genomeStringGmo = {1,1,0,0,0,0,0,0, 1,1,1,1,1,1};
        public static final int numberOfSpeciesGenesInGenome = 8;
        public static final int numberOfFitnessGenesInGenome = 6;

/**
 * Mutation rate is set as a fraction of crossings where mutation takes place
 */
        protected float mutationRate = 0;

/**
 * Fitness of a plant is a cumulative of the dominant genes. Genes are in order: Gene Gmo, Gene A, Gene B
 * If fitnessOfGeneGMO is declared static protected, the weeper can not acess it, and also sends no error
 * message
 * this makes the sweep impossible.
 */

        // the default set
        //static final int[] fitnessOfGenes = {0, 0, 15, 15, 5, 5};

        // the set for the Case
        static final int[] fitnessOfGenes = {2, 2, 15, 15, 5, 5};

/**
* The number of polen is the product of the fitness and the fitnessToPollenFactor
* At the moment it is a linear  function, but it can be made nonlinear.
* In that case the determinePollen method needs to be adapted
*/
        protected int fitnessToPollenFactor = 1;

/**
* The pollenSpreadDistance is expressed in the number of plants ,ie. how many plants far
* can the pollen get. It should be related to actual distance in the worked out case
* Performance is greatly influenced by this size !
*/

         protected int pollenSpreadDistance = 5 ;




/**
* Random number object generator
*/

        public static Random rand = new Random();

/**
 *The sizes of the fields must be even numbers!
 */
        protected int fieldWidth = 160;
        protected int fieldHeight = 160;
        protected int gmoFieldWidth = 6;
        protected int gmoFieldHeight = 6;
        protected int fieldLayout = 0;


/**
* Mating preferences
* it is a relative scape that should be derived from the literature
* the first species is the species of the Egg. There could be theorethically a difference in G&W and W&G
* depending on the recieving egg, or on the pollen
*
*For the testing, scale from 1 to 10 is used
*
* Preference is meant to contain the fact that different species  pollen
* has different 'tunneling'speeds through the egg organs of different species
* Each species with itself give the highest score
* GMO and Wild differ only very slighly ( GMO is a domesticated Wild)
* GMO and Remote and far related, and can still (but hardly) mate
* Non is a non-related species that only mates with itself
* Wild and Remote are also weekly related, but more than GMO and Remote
*/
        protected int gmoAndGmoMatingPreference = 10;
        protected int gmoAndWildMatingPreference = 8;
        protected int gmoAndRemoteMatingPreference = 1;
        protected int gmoAndNonMatingPreference = 0;

        protected int wildAndGmoMatingPreference = 8;
        protected int wildAndWildMatingPreference = 10;
        protected int wildAndRemoteMatingPreference = 2;
        protected int wildAndNonMatingPreference = 0;
```

```
        protected int remoteAndGmoMatingPreference = 1;
        protected int remoteAndWildMatingPreference = 2;
        protected int remoteAndRemoteMatingPreference = 10;
        protected int remoteAndNonMatingPreference = 0;

        protected int nonAndGmoMatingPreference = 0;
        protected int nonAndWildMatingPreference = 0;
        protected int nonAndRemoteMatingPreference = 0;
        protected int nonAndNonMatingPreference = 10;

/* Relatedness level. This is a set of cases for different levels of relatedness of the GMO to its environment
 * These cases are only used during the parameter sweeps. It is also possible to do it parameterwise. In that
 * case the parameter space is 16D.  Only by executing the setRelatednessCase are the different relatedness
 * parameters set.
 */

        protected int relatednessCase = 4;


 /* Population stucture cases
  */
        protected int populationStructureCase = 0;

/*
* Field and Plant initialization
*/

/* Choice for  ScapeArray2DMoore is because it is the ”fullest” surrounding,
* resembling a ”circular” surrounding that a pollen might experience.
*/

        ScapeArray2DMoore field = new  ScapeArray2DMoore();

        Plant plant = new Plant();



/**
 * Scape creation
 * Choice for  ScapeArray2DMoore is because it is the ”fullest” surrounding,
 * resembling a ”circular” surrounding that a pollen might experience.
 */

        public void createScape() {
            super.createScape();

// for evaluationg the case

            startOnOpen = false;
            setRelatednessCase(6);
            setPollenSpreadDistance(3);
            setFitnessOfGeneGMO(5);
            setFieldLayout(1);


/*
* clearing of the default rules
*/

        field.getRules().clear();

/* Dimension determination of the lattice
*/

        field.setExtent(new Coordinate2DDiscrete(fieldWidth, fieldHeight));

/* Setting the name of the geometry scape
*/
        field.setName(”Field”);

/* Setting what does the geometry contain
* Populating the geometry with the agents
*/

        field.setPrototypeAgent(plant);

/* Setting the order bt which the agents inthe geometry will be iterated
*/

        field.setExecutionOrder(Scape.RULE_ORDER);

/* Setting the periodicity (continuesness) of the scape.
         */

        field.getGeometry().setPeriodic(false);

/* Adding the geometry to the ”World”
*/

        addAgent(field);


/**
* Rules definitions
*
* Planting rule; the rule determining the agent placement GMO agent placement */

        field.addRule(new  Rule(”Plant_GMO_Crop”) {
     public void execute(Agent agent) {
                ((Plant) agent).plantGMO();
```

```
                            }
                       }
                       );


/* Growth rule :
*Here the fitness of each plant is evaluated, and assigned to it.
*Consequently the number of of pollen made is determined
*/
            field.addRule(new Rule("Plant_Growth") {
             public void execute(Agent agent) {
                        ((Plant) agent).grow();
             }
            });

/**
* Pollen and Egg production rule:
* here the genome is split following the Mendelean rules of reproduction
* The produced genomes have the Mendelean chance of appearing, and conwaying information.
*/
            field.addRule(new Rule("Pollen_and_Egg_production") {
             public void execute(Agent agent) {
                         ((Plant) agent).pollenProduction();
                         ((Plant) agent).eggProduction();

             }


            });


/**
* Mating Rule :
* In this rule throws plant searches for the maximaly attractive mate in its surroundings.
* The "suroundings" is the maximal pollen dispersion distance that is set in the model
* The most attractive mate is the plant whose distance * number of pollen * preference_for_species
* is maximal.
* The Genome is replaced by a fusion of the Egg with the selected Pollen
*/
            field.addRule(new Rule("Mating") {
             public void execute(Agent agent) {
                        ((Plant) agent).mating();
                        }
            });
    }


/* Statistics, View and Chart setup */


    Overhead2DView view = new Overhead2DView();



    public void createViews() {
        super.createViews();
        final StatCollector[] statistics = {
                new StatCollectorCond("Is_GMO") {
                        public boolean meetsCondition(Object object) {
                    return (((Plant) object).getEggIsGMO());
                }
            },

                new StatCollectorCond("Is_Wild") {
                    public boolean meetsCondition(Object object) {
                  return (((Plant) object).getEggIsWild());
                }
            },

            new StatCollectorCond("Is_Remote") {
                public boolean meetsCondition(Object object) {
                    return (((Plant) object).getEggIsRemote());
                }
            },
            new StatCollectorCond("Is_Non") {
                public boolean meetsCondition(Object object) {
                    return (((Plant) object).getEggIsNon());
                }
            },
            new StatCollectorCond("Has_GMO_Gene") {
         public boolean meetsCondition(Object object) {
                    return (((Plant) object).getHasGmoGene());
                    }

            },
        new StatCollectorCSAMM("Fitness") {
                public double getValue(Object object) {
                return ((Plant) object).getFitness(); }
            }
        };

            field.addStatCollectors(statistics);
            view.setCellSize(4);
            field.addView(view);


        if (!(ViewFrameBridge.isInApplet())) {
            /*
                ChartView speciesDivisionChart = new ChartView(ChartView.PIE);
         field.addView(speciesDivisionChart);
```

```
          speciesDivisionChart .addSeries("Count Is GMO", Color.red);
          speciesDivisionChart .addSeries("Count Is Wild", Color.green);
                  speciesDivisionChart .addSeries("Count Is Remote", Color.yellow);
                  speciesDivisionChart .addSeries("Count Is Non", Color.blue);
                  // chart.addSeries("Has GMO Gene", Color.yellow);
                  // chart.addSeries("Fitness", Color.yellow);

                */
              if(turnOnCharts){
                  ChartView averageFitnessChart = new ChartView();
          field.addView(averageFitnessChart);
          averageFitnessChart .addSeries("Average␣Fitness", Color.red);

                  ChartView numberOfGMOs = new ChartView();
          field.addView(numberOfGMOs);
          numberOfGMOs.addSeries("Count␣Has␣GMO␣Gene", Color.red);

              }


/* parameter sweep
*/

              if(doTheSweep == true){
               /*
               BatchView batchView = new BatchView();
               addView(batchView);
                */

              SweepControlView sweeper = new SweepControlView();
              field.addView(sweeper);


              SweepDimension relatednessDim = new SweepDimension(((GeneScape) getModel()), "relatednessCase",
              1, 5, 1);
              SweepDimension spreadDim = new SweepDimension(((GeneScape) getModel()), "pollenSpreadDistance",
              1, 20, 4);
              SweepDimension fitnessDim = new SweepDimension(((GeneScape) getModel()), "fitnessOfGeneGMO",
              1, 20 , 4);

               SweepGroup toSweep = new SweepGroup();
               toSweep.addMember(relatednessDim);
               toSweep.addMember(spreadDim);
               toSweep.addMember(fitnessDim);
               sweeper.setSweepGroup(toSweep);
               toSweep.setRunsPer(1);
               } //sweep on




/*Stats to collect and save
 */
              final DataSeries noOfGMOs = ((StatCollectorCond) statistics[4]).getDataSeries("Count");
              final DataSeries populationFitness = ((StatCollectorCSAMM) statistics[5]).getDataSeries("Average");


                  /*
              DataOutputView dataView = new DataOutputView() {
                public void writeRunHeader() throws IOException {
     super.writeRunHeader();
        runDataStream.writeBytes("NumberOfGMOs");
        runDataStream.writeBytes("\t");
        runDataStream.writeBytes("AveragePopulationFitness");
        runDataStream.writeBytes("\t");
        }
    public void writeRunData() throws IOException {
     super.writeRunData();
                              runDataStream.writeBytes("\t");
        runDataStream.writeBytes(Double.toString(noOfGMOs.getMax()));
        runDataStream.writeBytes("\t");
        runDataStream.writeBytes(Double.toString(populationFitness.getAvg()));
        runDataStream.writeBytes("\t");
        runDataStream.writeBytes(Double.toString(populationFitness.getMax()));
        runDataStream.writeBytes("\t");
        runDataStream.writeBytes(Double.toString(populationFitness.getMin()));
        runDataStream.writeBytes("\t");
    }
                    };


              */

                    DataOutputView dataView = new DataOutputView(){
                        public void writeRunHeader() throws IOException {
     super.writeRunHeader();
        runDataStream.writeBytes("NumberOfGMOs");
        runDataStream.writeBytes("\t");
        runDataStream.writeBytes("AveragePopulationFitness");
        runDataStream.writeBytes("\t");
        }
                      public void writePeriodHeader() throws IOException {
     super.writePeriodHeader();
        periodDataStream.writeBytes("NumberOfGMOs");
        periodDataStream.writeBytes("\t");
        periodDataStream.writeBytes("AveragePopulationFitness");
        periodDataStream.writeBytes("\t");
        }
    public void writePeriodData() throws IOException {
     super.writePeriodData();
                              periodDataStream.writeBytes("\t");
```

```
       periodDataStream.writeBytes(Double.toString(noOfGMOs.getMax()));
       periodDataStream.writeBytes(”\t”);
       periodDataStream.writeBytes(Double.toString(populationFitness.getAvg()));
       periodDataStream.writeBytes(”\t”);
       periodDataStream.writeBytes(Double.toString(populationFitness.getMax()));
       periodDataStream.writeBytes(”\t”);
       periodDataStream.writeBytes(Double.toString(populationFitness.getMin()));
       periodDataStream.writeBytes(”\t”);
   }
                };


            //DataOutputView dataView = new DataOutputView() ;

           addView(dataView);

/*
            try {
      setStartPeriod(0);
// and here !
      setStopPeriod(100);
     }



     catch (SpatialTemporalException e) {
         System.out.println(”Bad start/stop periods: ” + e);
     }

 */
        try {
            dataView.setRunFile(new File(”Output.txt”));
        }
        catch (IOException e) {
            throw new RuntimeException(e.toString());
        }




        try {
            dataView.setPeriodFile(new File(”Period.txt”));
        }
        catch (IOException e) {
            throw new RuntimeException(e.toString());
        }


     }// applet on
}//create views

    public int getPopulationStructureCase(){

        return populationStructureCase;
    }

    public void setPopulationStructureCase(int _populationStructureCase){
        populationStructureCase = _populationStructureCase;
    }



    public int getFieldLayout(){
        return fieldLayout;
    }


    public void setFieldLayout(int _fieldLayout){
        fieldLayout = _fieldLayout;
    }


    public int getRelatednessCase(){
        return relatednessCase;
    }

    public void setRelatednessCase(int _relatednessCase){
        relatednessCase = _relatednessCase;
        switch(_relatednessCase){
            case 1 :
/* This is the least related istuation. the three populations live together but can not cross-pollinate
 */
                setPreferenceGmoAndGmoMating(10);
                setPreferenceGmoAndWildMating(0);
                setPreferenceGmoAndRemoteMating(0);
                setPreferenceGmoAndNonMating(0);
                setPreferenceWildAndGmoMating(0);
                setPreferenceWildAndWildMating(10);
                setPreferenceWildAndRemoteMating(0);
                setPreferenceWildAndNonMating(0);
                setPreferenceRemoteAndGmoMating(0);
                setPreferenceRemoteAndWildMating(0);
                setPreferenceRemoteAndRemoteMating(10);
                setPreferenceRemoteAndNonMating(0);
                setPreferenceNonAndGmoMating(0);
                setPreferenceNonAndWildMating(0);
                setPreferenceNonAndRemoteMating(0);
                setPreferenceNonAndNonMating(10);
                break;

            case 2 :
```

```
/* This is the case with low level of relatedness between the plants. Only extreme differences in fitness
 *should be able to create flowing fo the gene.
 */
                setPreferenceGmoAndGmoMating(10);
                setPreferenceGmoAndWildMating(3);
                setPreferenceGmoAndRemoteMating(1);
                setPreferenceGmoAndNonMating(0);
                setPreferenceWildAndGmoMating(3);
                setPreferenceWildAndWildMating(10);
                setPreferenceWildAndRemoteMating(3);
                setPreferenceWildAndNonMating(0);
                setPreferenceRemoteAndGmoMating(1);
                setPreferenceRemoteAndWildMating(3);
                setPreferenceRemoteAndRemoteMating(10);
                setPreferenceRemoteAndNonMating(0);
                setPreferenceNonAndGmoMating(0);
                setPreferenceNonAndWildMating(0);
                setPreferenceNonAndRemoteMating(0);
                setPreferenceNonAndNonMating(10);
                break;

/* Intermediate relatedness case. Plants have a preference ofr each other of 5
 */
           case 3 :
                setPreferenceGmoAndGmoMating(10);
                setPreferenceGmoAndWildMating(5);
                setPreferenceGmoAndRemoteMating(5);
                setPreferenceGmoAndNonMating(0);
                setPreferenceWildAndGmoMating(5);
                setPreferenceWildAndWildMating(10); //GRRR !!!!! 5 instead of 10
                setPreferenceWildAndRemoteMating(5);
                setPreferenceWildAndNonMating(0);
                setPreferenceRemoteAndGmoMating(5);
                setPreferenceRemoteAndWildMating(5);
                setPreferenceRemoteAndRemoteMating(10);
                setPreferenceRemoteAndNonMating(0);
                setPreferenceNonAndGmoMating(0);
                setPreferenceNonAndWildMating(0);
                setPreferenceNonAndRemoteMating(0);
                setPreferenceNonAndNonMating(10);
                break;

/* The high level relatedness case
 */
           case 4 :
                setPreferenceGmoAndGmoMating(10);
                setPreferenceGmoAndWildMating(8);
                setPreferenceGmoAndRemoteMating(7);
                setPreferenceGmoAndNonMating(0);
                setPreferenceWildAndGmoMating(8);
                setPreferenceWildAndWildMating(10);
                setPreferenceWildAndRemoteMating(8);
                setPreferenceWildAndNonMating(0);
                setPreferenceRemoteAndGmoMating(7);
                setPreferenceRemoteAndWildMating(8);
                setPreferenceRemoteAndRemoteMating(10);
                setPreferenceRemoteAndNonMating(0);
                setPreferenceNonAndGmoMating(0);
                setPreferenceNonAndWildMating(0);
                setPreferenceNonAndRemoteMating(0);
                setPreferenceNonAndNonMating(10);
                break;



            case 5 :
/* The extreme relatedness case, all the plants are related and can freely interbreed
 *This will be the extreme value in the full test. This is basically a case with
one big populaiton.
 */
                setPreferenceGmoAndGmoMating(10);
                setPreferenceGmoAndWildMating(10);
                setPreferenceGmoAndRemoteMating(10);
                setPreferenceGmoAndNonMating(10);
                setPreferenceWildAndGmoMating(10);
                setPreferenceWildAndWildMating(10);
                setPreferenceWildAndRemoteMating(10);
                setPreferenceWildAndNonMating(10);
                setPreferenceRemoteAndGmoMating(10);
                setPreferenceRemoteAndWildMating(10);
                setPreferenceRemoteAndRemoteMating(10);
                setPreferenceRemoteAndNonMating(10);
                setPreferenceNonAndGmoMating(10);
                setPreferenceNonAndWildMating(10);
                setPreferenceNonAndRemoteMating(10);
                setPreferenceNonAndNonMating(10);
                break;
// Case study evaluation case

            case 6 :
                setPreferenceGmoAndGmoMating(10);
                setPreferenceGmoAndWildMating(9);
                setPreferenceGmoAndRemoteMating(8);
                setPreferenceGmoAndNonMating(0);
                setPreferenceWildAndGmoMating(9);
                setPreferenceWildAndWildMating(10);
                setPreferenceWildAndRemoteMating(9);
                setPreferenceWildAndNonMating(0);
                setPreferenceRemoteAndGmoMating(8);
                setPreferenceRemoteAndWildMating(9);
                setPreferenceRemoteAndRemoteMating(10);
```

```
                setPreferenceRemoteAndNonMating(0);
                setPreferenceNonAndGmoMating(0);
                setPreferenceNonAndWildMating(0);
                setPreferenceNonAndRemoteMating(0);
                setPreferenceNonAndNonMating(10);
                break;


    }
    }

/*  THE COPY-PASTE SOURCE FOR EXTRA CASES
                case 0 :
                setPreferenceGmoAndGmoMating();
                setPreferenceGmoAndWildMating();
                setPreferenceGmoAndRemoteMating();
                setPreferenceGmoAndNonMating();
                setPreferenceWildAndGmoMating();
                setPreferenceWildAndWildMating();
                setPreferenceWildAndRemoteMating();
                setPreferenceWildAndNonMating();
                setPreferenceRemoteAndGmoMating();
                setPreferenceRemoteAndWildMating();
                setPreferenceRemoteAndRemoteMating();
                setPreferenceRemoteAndNonMating();
                setPreferenceNonAndGmoMating();
                setPreferenceNonAndWildMating();
                setPreferenceNonAndRemoteMating();
                setPreferenceNonAndNonMating();
                break;

 */






    /* Getters and Setters for model parameters
     */


    public float getMutationRate(){
        return mutationRate;
    }

    public void setMutationRate(float _mutationRate){
        this.mutationRate = _mutationRate;
    }


    public int getFitnessOfGeneGMO() {
        return fitnessOfGenes[0];
    }

    public void setFitnessOfGeneGMO(int _FitnessOfGeneGMO){
        this.fitnessOfGenes[0] = _FitnessOfGeneGMO;
        this.fitnessOfGenes[1] = _FitnessOfGeneGMO;

    }

    public int getFitnessOfGeneA() {
        return fitnessOfGenes[2];
    }

    public void setFitnessOfGeneA(int _FitnessOfGeneA){
        this.fitnessOfGenes[2] = _FitnessOfGeneA;
        this.fitnessOfGenes[3] = _FitnessOfGeneA;
    }
    public int getFitnessOfGeneB() {
        return fitnessOfGenes[4];
    }

    public void setFitnessOfGeneB(int _FitnessOfGeneB){
        this.fitnessOfGenes[4] = _FitnessOfGeneB;
        this.fitnessOfGenes[5] = _FitnessOfGeneB;
    }

    public  int getFieldWidth() {
        return fieldWidth;
    }

    public  void setFieldWidth(int _FieldWidth){
        fieldWidth = _FieldWidth;
    }

    public  int getFieldHeight() {
        return fieldHeight;
    }

    public  void setFieldHeight(int _FieldHeight){
        fieldHeight = _FieldHeight;
    }

    public  int getGMOFieldWidth() {
        return gmoFieldWidth;
    }

    public  void setGMOFieldWidth(int _GMOFieldWidth){
        gmoFieldWidth = _GMOFieldWidth;
    }
```

```
public  int getGMOFieldHeight() {
    return gmoFieldHeight;
}

public  void setGMOFieldHeight(int _GMOFieldHeight){
    gmoFieldHeight = _GMOFieldHeight;
}

public int getFitnessToPollenFactor() {
    return fitnessToPollenFactor;
}

public  void setFitnessToPollenFactor(int _fitnessToPollenFactor){
    fitnessToPollenFactor = _fitnessToPollenFactor;
}

public int getPollenSpreadDistance() {
    return pollenSpreadDistance;
}

public  void setPollenSpreadDistance(int _pollenSpreadDistance){
    pollenSpreadDistance = _pollenSpreadDistance;
}

/* Mating preference Getter and setters
 */



public int getPreferenceGmoAndGmoMating(){
    return gmoAndGmoMatingPreference;

}

public void setPreferenceGmoAndGmoMating(int _gmoAndGmoMatingPreference){
    gmoAndGmoMatingPreference = _gmoAndGmoMatingPreference;

}

public int getPreferenceGmoAndWildMating(){
    return gmoAndWildMatingPreference;

}

public void setPreferenceGmoAndWildMating(int _gmoAndWildMatingPreference){
    gmoAndWildMatingPreference = _gmoAndWildMatingPreference;

}

public int getPreferenceGmoAndRemoteMating(){
    return gmoAndRemoteMatingPreference;

}

public void setPreferenceGmoAndRemoteMating(int _gmoAndRemoteMatingPreference){
   gmoAndRemoteMatingPreference  = _gmoAndRemoteMatingPreference;

}

public int getPreferenceGmoAndNonMating(){
    return gmoAndNonMatingPreference;

}

public void setPreferenceGmoAndNonMating(int _gmoAndNonMatingPreference){
     gmoAndNonMatingPreference = _gmoAndNonMatingPreference;

}

public int getPreferenceWildAndGmoMating(){
    return wildAndGmoMatingPreference;

}

public void setPreferenceWildAndGmoMating(int _wildAndGmoMatingPreference){
    wildAndGmoMatingPreference = _wildAndGmoMatingPreference;

}

public int getPreferenceWildAndWildMating(){
    return wildAndWildMatingPreference;

}

public void setPreferenceWildAndWildMating(int _wildAndWildMatingPreference){
    wildAndWildMatingPreference = _wildAndWildMatingPreference;

}

public int getPreferenceWildAndRemoteMating(){
    return wildAndRemoteMatingPreference;

}

public void setPreferenceWildAndRemoteMating(int _wildAndRemoteMatingPreference){
    wildAndRemoteMatingPreference = _wildAndRemoteMatingPreference;

}

public int getPreferenceWildAndNonMating(){
```

```
        return wildAndNonMatingPreference;

    }

    public void setPreferenceWildAndNonMating(int _wildAndNonMatingPreference){
       wildAndNonMatingPreference  = _wildAndNonMatingPreference;

    }

    public int getPreferenceRemoteAndGmoMating(){
        return remoteAndGmoMatingPreference;

    }

    public void setPreferenceRemoteAndGmoMating(int _remoteAndGmoMatingPreference){
        remoteAndGmoMatingPreference = _remoteAndGmoMatingPreference;

    }

    public int getPreferenceRemoteAndWildMating(){
        return remoteAndWildMatingPreference;

    }

    public void setPreferenceRemoteAndWildMating(int _remoteAndWildMatingPreference){
        remoteAndWildMatingPreference = _remoteAndWildMatingPreference;

    }

    public int getPreferenceRemoteAndRemoteMating(){
        return remoteAndRemoteMatingPreference;

    }

    public void setPreferenceRemoteAndRemoteMating(int _remoteAndRemoteMatingPreference){
        remoteAndRemoteMatingPreference = _remoteAndRemoteMatingPreference;

    }

    public int getPreferenceRemoteAndNonMating(){
        return remoteAndNonMatingPreference;

    }

    public void setPreferenceRemoteAndNonMating(int _remoteAndNonMatingPreference){
        remoteAndNonMatingPreference = _remoteAndNonMatingPreference;

    }

    public int getPreferenceNonAndGmoMating(){
        return nonAndGmoMatingPreference;

    }

    public void setPreferenceNonAndGmoMating(int _nonAndGmoMatingPreference){
         nonAndGmoMatingPreference = _nonAndGmoMatingPreference;

    }

    public int getPreferenceNonAndWildMating(){
        return nonAndWildMatingPreference;

    }

    public void setPreferenceNonAndWildMating(int _nonAndWildMatingPreference){
         nonAndWildMatingPreference = _nonAndWildMatingPreference;

    }

    public int getPreferenceNonAndRemoteMating(){
        return nonAndRemoteMatingPreference;

    }

    public void setPreferenceNonAndRemoteMating(int _nonAndRemoteMatingPreference){
         nonAndRemoteMatingPreference = _nonAndRemoteMatingPreference;

    }

    public int getPreferenceNonAndNonMating(){
        return nonAndNonMatingPreference;

    }

    public void setPreferenceNonAndNonMating(int _nonAndNonMatingPreference){
        nonAndNonMatingPreference = _nonAndNonMatingPreference;

    }

    public void setPollenSpreadModelUsed(int _pollenSpreadModelUsed){
        pollenSpreadModelUsed= _pollenSpreadModelUsed;
    }

    public int getPollenSpreadModelUsed(){
        return pollenSpreadModelUsed;
    }

    public int getPollenDeterminationModelUsed(){
        return pollenDeterminationModelUsed;
    }
```

```
    public void setPollenDeterminationModelUsed(int _pollenDeterminationModelUsed){
        pollenDeterminationModelUsed = _pollenDeterminationModelUsed;
    }


}


/*
 * Plant.java
 *
 * Created on March 26, 2001, 9:32 AM
*
*
* @author  Nikolic
*
*/


import java.awt.*;
import java.util.*;

import edu.brook.ascape.model.*;
import edu.brook.ascape.rule.*;

import edu.brook.ascape.util.*;
import edu.brook.ascape.view.*;


public class Plant extends CellOccupant {



    /**
     * Plant states
     */
        protected int[] genome = new int[GeneScape.numberOfSpeciesGenesInGenome +
        GeneScape.numberOfFitnessGenesInGenome];
        protected int[] pollen = new int[(GeneScape.numberOfSpeciesGenesInGenome +
        GeneScape.numberOfFitnessGenesInGenome)/2];
        protected int[] egg = new int[(GeneScape.numberOfSpeciesGenesInGenome +
        GeneScape.numberOfFitnessGenesInGenome)/2];

        protected int fitness = 0;
        protected int numPollen = 0;
        protected float attractiveness = 0;
        protected int gmoGenePresent = 0;

        // Plant states
        protected boolean isGMO = false;
        protected boolean isWild = false;
        protected boolean isRemote = false;
        protected boolean isNon = false;
        protected boolean hasGmoGene= false;

        // Pollen states
        protected boolean pollenIsGMO = false;
        protected boolean pollenIsWild = false;
        protected boolean pollenIsRemote = false;
        protected boolean pollenIsNon = false;
        protected boolean pollenHasGmoGene= false;
        protected int pollenIsSpecies ;

        // Egg states
        protected boolean eggIsGMO = false;
        protected boolean eggIsWild = false;
        protected boolean eggIsRemote = false;
        protected boolean eggIsNon = false;
        protected boolean eggHasGmoGene= false;
        protected int eggIsSpecies ;

 /**
  * Begining population values. Genome is random draw, coordinate placement to fill the lattice.
         * New switch statements must be added if new species are introduced in the model !!!
         */
 public void initialize() {
  super.initialize();

                switch ((((GeneScape) getModel()).getPopulationStructureCase())){

                 case 0:

                    switch (randomToLimit((GeneScape.numberOfSpeciesGenesInGenome/2) - 1)){
                    /*there are no GMO's in the wild, they will be planted after the
"nature" is set up.
                     * That is the reason for (GeneScape.numberOfSpeciesGenesInGenome/2) - 1
                     * In this case, all three species are equaly represented in the population
                     *
                     */

                    case 0 : this.genome = randomFitnessGenome(2);
                    this.pollenIsSpecies = 2;
                    this.eggIsSpecies = 2;
                    this.setPlantState();
                                                            break;
                    case 1 : this.genome =   randomFitnessGenome(4);
                    this.pollenIsSpecies = 4;
                    this.eggIsSpecies = 4;
                    this.setPlantState();
                                                            break;
```

```
                    case 2 : this.genome =   randomFitnessGenome(6);
                    this.pollenIsSpecies = 6;
                    this.eggIsSpecies = 6;
                    this.setPlantState();
                                                              break;

                }
                break;

                case 1:
                        /* In this case there are 1/6th Wild 2/6ths Remote and 2/3 Non
                         */
                    switch (case1Species())

                    {


                    case 0 : this.genome = randomFitnessGenome(2);
                    this.pollenIsSpecies = 2;
                    this.eggIsSpecies = 2;
                    this.setPlantState();
                                                              break;
                    case 1 : this.genome =   randomFitnessGenome(4);
                    this.pollenIsSpecies = 4;
                    this.eggIsSpecies = 4;
                    this.setPlantState();
                                                              break;

                    case 2 : this.genome =   randomFitnessGenome(6);
                    this.pollenIsSpecies = 6;
                    this.eggIsSpecies = 6;
                    this.setPlantState();
                                                              break;

                }

}
        }





        /**
         * After the genome has been initialized, the Plant states are set.
         * Species flags are set, and the presence of a GMO gene is determined
         */
        public void setPlantState(){

            ArrayList speciesHolder = speciesGeneCounter(this.genome);
            this.setGMO(false);
            this.setWild(false);
            this.setRemote(false);
            this.setNon(false);
            this.setHasGmoGene(false);

            for(int i = 1; i < speciesHolder.size(); i++){
                if( ((Integer) speciesHolder.get(i)).intValue() == 0  | ((Integer)
                speciesHolder.get(i)).intValue() == 1){
                    this.setGMO(true);
                    this.pollenIsSpecies = 0;
                }

                if( ((Integer) speciesHolder.get(i)).intValue() == 2 | ((Integer)
                speciesHolder.get(i)).intValue() == 3){
                    this.setWild(true);
                    this.pollenIsSpecies = 2;
                }

                if( ((Integer) speciesHolder.get(i)).intValue() == 4 | ((Integer)
                speciesHolder.get(i)).intValue() == 5){
                    this.setRemote(true);
                    this.pollenIsSpecies = 4;
                }

                if( (((Integer) speciesHolder.get(i)).intValue() == 6) | (((Integer)
                speciesHolder.get(i)).intValue() == 7)){
                    this.setNon(true);
                    this.pollenIsSpecies = 6;
                }
            }

            ArrayList fitnessHolder = fitnessGeneCounter(this.genome);

            for(int i = 0; i < fitnessHolder.size() ; i++){
                if( (((Integer) fitnessHolder.get(i)).intValue() == 8) | (((Integer)
                fitnessHolder.get(i)).intValue() == 9)){
                    this.setHasGmoGene(true);
                }
            }
        }


        /**
         * Same as setPlantState, except is applies to Egg
         * Might be useful is future versions if eggs need to be treated differently
         */

        public void setEggState(){

            ArrayList speciesHolder = speciesGeneCounter(this.egg);
```

```
    this.setEggIsGMO(false);
    this.setEggIsWild(false);
    this.setEggIsRemote(false);
    this.setEggIsNon(false);
    this.setEggHasGmoGene(false);
    for(int i = 0; i < speciesHolder.size(); i++){
        if( ((Integer) speciesHolder.get(i)).intValue() == 0){
            this.setEggIsGMO(true);
            eggIsSpecies = 0;
        }
        if( ((Integer) speciesHolder.get(i)).intValue() == 2 ){
            this.setEggIsWild(true);
            eggIsSpecies = 2;
        }
        if( ((Integer) speciesHolder.get(i)).intValue() == 4 ){
            this.setEggIsRemote(true);
            eggIsSpecies = 4;
        }
        if( ((Integer) speciesHolder.get(i)).intValue() == 6 ){
            this.setEggIsNon(true);
            eggIsSpecies = 6;
        }

    }
    ArrayList fitnessHolder = fitnessGeneCounter(this.egg);
    for(int i = 0; i < fitnessHolder.size() ; i++){
        if( ((Integer) fitnessHolder.get(i)).intValue() == 8 | (((Integer)
        fitnessHolder.get(i)).intValue() == 9)){
            this.setEggHasGmoGene(true);
        }
    }
}

/** Same as setPlantState, except is applies to Pollen
 * Might be useful is future versions if pollen needs to be treated differently
 */
public void setPollenState(){

    ArrayList speciesHolder = speciesGeneCounter(this.pollen);

    this.setPollenIsNon(false);
    this.setPollenIsRemote(false);
    this.setPollenIsWild(false);
    this.setPollenIsGMO(false);
    this.setPollenHasGmoGene(false);

    for(int i = 0; i < speciesHolder.size() ; i++){
        if( ((Integer) speciesHolder.get(i)).intValue() == 0 ){
            this.setPollenIsGMO(true);
        }
        if( ((Integer) speciesHolder.get(i)).intValue() == 2 ){
            this.setPollenIsWild(true);
        }
        if( ((Integer) speciesHolder.get(i)).intValue() == 4 ){
            this.setPollenIsRemote(true);
        }

        if( ((Integer) speciesHolder.get(i)).intValue() == 6 ){
            this.setPollenIsNon(true);
        }

    }
    ArrayList fitnessHolder = fitnessGeneCounter(this.pollen);
    for(int i = 0; i < fitnessHolder.size(); i++){
        if( (((Integer) fitnessHolder.get(i)).intValue() == 8) | (((Integer)
        fitnessHolder.get(i)).intValue() == 9)){
            this.setPollenHasGmoGene(true);
        }
    }
}



 /**
  * GMO Planting rule implementation
  * The method for determining wether the Plant is located in the  GMO field,
  * and if so, the Plants genome is set to that of an  GMO plant.
  * ((GeneScape) getModel()).XXX is necesary because it is an anonymous inner method
  * In version 1.4.2 different field layouts are implemented
  */
public void plantGMO() {

     int[]  location = (this.getCoordinate()).getValues();
            int midXField = ((GeneScape) getModel()).getFieldWidth()/2 ;
            int widthPlot =  ((GeneScape) getModel()).getGMOFieldWidth()/2;

            int midYField = ((GeneScape) getModel()).getFieldHeight()/2 ;
            int heightPlot =  ((GeneScape) getModel()).getGMOFieldHeight()/2;


    switch((((GeneScape) getModel()).getFieldLayout())){

        case 0 :
    /* Default field layout: GMOs in the middle, other three species randomly distributed
     * across the field
     */

    if(location[0] >= midXField-widthPlot && location[0] <= midXField+widthPlot) {
        if (location[1] >= midYField-heightPlot && location[1] <= midYField+heightPlot) {
            this.genome = GeneScape.genomeStringGmo;
            this.pollenIsSpecies = 0;
```

```
            this.eggIsSpecies = 0;
            this.setPlantState();
        }
    }
        break;

        case 1 :

            /* Case with a pollen spread limiting measure implemented : Barren zone/non
             * related species of 4 m
             */

            if(location[0] >= midXField-widthPlot-40 && location[0] <= midXField+widthPlot+40){
                if (location[1] >= midYField-heightPlot-40 && location[1] <= midYField+heightPlot+40){
                    this.genome = GeneScape.genomeStringN;
                    this.pollenIsSpecies = 0;
                    this.eggIsSpecies = 0;
                    this.setPlantState();

                    if(location[0] >= midXField-widthPlot && location[0] <= midXField+widthPlot){
                        if (location[1] >= midYField-heightPlot && location[1] <= midYField+heightPlot){
                            this.genome = GeneScape.genomeStringGmo;
                            this.pollenIsSpecies = 0;
                            this.eggIsSpecies = 0;
                            this.setPlantState();
                }
            }
                }
            }


             break;



         // case 2 :

            /* Case with a pollen spread limiting measure implemented : Barren zone of 8 m
             */
    }
}



/** Plant growth rule implementation
 * In this method the fintess of the plant is evaluated, and the corresponding
 * number of pollen is determined.
 * During the growth of the polant, mutations can take place inthe genome
 */

public void grow() {

    this.setPlantState();

    /* During the growth of the polant, mutations can take place inthe genome
     */

    if(GeneScape.rand.nextFloat() <= ((GeneScape) getModel()).getMutationRate()){
        int[] mutatedGenome = this.mutate(this.genome);
        this.genome = mutatedGenome;
    }

    this.fitness = fitnessOf(this.genome);
    this.numPollen = determineNumPollen();
    requestUpdate();
 }


/** This method handles the pollen number determination
 * PollenDeterminationModelUsed is checked to see whetehr the model determination model
 * should be used. otherwise, the No. of pollen is equal to the fitness
 * At the moment the pollen model is is only dependant on the fitness and the factor.
 * It can be adapted to make it non-linear if necessary
 */

 public int determineNumPollen() {
     if(((GeneScape) getModel()).getPollenDeterminationModelUsed() == 1){
     return  ((GeneScape) getModel()).getFitnessToPollenFactor() * this.getFitness();
     }
     else{
         return this.getFitness();
     }
 }


 public void pollenProduction() {
     this.pollen = mendeleanSplitOfGenome(this.genome, this.pollenIsSpecies);
     this.setPollenState();
     requestUpdate();
 }



public void eggProduction() {
    this.egg = mendeleanSplitOfGenome(this.genome, this.eggIsSpecies);
    this.setEggState();
    requestUpdate();
}
```

```
     /**
     * this.XXX is considered to be the Egg, the other is the pollen.
     * So Egg has preferences for different types of pollen.
     * These are not necessarily reciprocive eg. W&R does not have to be the same as R&W
     */

    public int matingPreference(Plant them) {


        int preference = 0;

        if(this.getEggIsNon()){
            if(them.getPollenIsGMO()){preference = ((GeneScape) getModel()).getPreferenceNonAndGmoMating();}
            if(them.getPollenIsWild()){preference = ((GeneScape) getModel()).getPreferenceNonAndWildMating();}
            if(them.getPollenIsRemote()){preference = ((GeneScape) getModel()).getPreferenceNonAndRemoteMating();}
            if(them.getPollenIsNon()){preference = ((GeneScape) getModel()).getPreferenceNonAndNonMating();}
        }

        if(this.getEggIsGMO()){
            if(them.getPollenIsGMO()) {preference =  ((GeneScape) getModel()).getPreferenceGmoAndGmoMating();}
            if(them.getPollenIsWild()) {preference = ((GeneScape) getModel()).getPreferenceGmoAndWildMating();}
            if(them.getPollenIsRemote()) {preference = ((GeneScape) getModel()).getPreferenceGmoAndRemoteMating();}
            if(them.getPollenIsNon()) {preference =   ((GeneScape) getModel()).getPreferenceGmoAndNonMating();}
            }

        if(this.getEggIsWild()){
            if(them.getPollenIsGMO()) {preference = ((GeneScape) getModel()).getPreferenceWildAndGmoMating();}
            if(them.getPollenIsWild()) {preference = ((GeneScape) getModel()).getPreferenceWildAndWildMating();}
            if(them.getPollenIsRemote()) {preference = ((GeneScape) getModel()).getPreferenceWildAndRemoteMating();}
            if(them.getPollenIsNon()) {preference = ((GeneScape) getModel()).getPreferenceWildAndNonMating();}
        }

        if(this.getEggIsRemote()){
            if(them.getPollenIsGMO()) {preference = ((GeneScape) getModel()).getPreferenceRemoteAndGmoMating();}
            if(them.getPollenIsWild()) {preference = ((GeneScape) getModel()).getPreferenceRemoteAndWildMating();}
            if(them.getPollenIsRemote()) {preference = ((GeneScape) getModel()).getPreferenceRemoteAndRemoteMating();}
            if(them.getPollenIsNon()) {preference = ((GeneScape) getModel()).getPreferenceRemoteAndNonMating();}
        }
        return preference;
    }


     /**
      * Method for findig the best mate.
      * Array of Cells that are witin a disteance is returned. The atractiveness of each cell is calculated,
      * and the most attractive found. that cell is returned.
      * Attractiveness calculation is the number of pollen of that cell times the preference of this
      * plant egg for the other Pollen divided by the distance
      */

    public Plant bestMate(){

        int maximalRegister = 0;
        float lastBiggest = Float.NEGATIVE_INFINITY;
        boolean allAttractivenessesEqual  = false;

        Cell[] plantsNear = getCellsNear(((GeneScape) getModel()).getPollenSpreadDistance(), false);

        for(int i = 0; i < plantsNear.length; i++){

            /* if GeneScape.pollenSpreadModelUsed is turned to 0, the Pollen spread model will not be used.
             *GeneScape assuemes that in that case the pollen is equally distributed within the
             * Pollen spread distance.
            * The Pollen spread mo0del is a simple 1/distance reduction in actractiveness. Any
             * other model could be placed in it.
             */
            if(((GeneScape) getModel()).getPollenSpreadModelUsed() == 0){
                ((Plant) plantsNear[i]).setAttractiveness((((Plant) plantsNear[i]).getNumPollen() *
                matingPreference(((Plant) plantsNear[i]))));
            }
            else{
            ((Plant) plantsNear[i]).setAttractiveness((((Plant) plantsNear[i]).getNumPollen() *
            matingPreference(((Plant) plantsNear[i])))/( getDistance(((Plant) plantsNear[i]))));
            }
            if(((Plant) plantsNear[i]).getAttractiveness() > lastBiggest){
                maximalRegister = i;
                lastBiggest = ((Plant) plantsNear[i]).getAttractiveness();
            }
        }

        /*
         *There is a problem here :
         * If all the surrounding cells have an attractiveness of 0, than the first one will be chosen
         * to mate anyway. This causes the GMO gene to ultimately conquer the whole population.
         * The question is, should a Plant die if all its neighbours have an attractiveness of 0
(ie, no
         * prtner to mate with) or should the plant self pollinate then ?
         * If all neighbours have an atractivernes of 0, plant will  sellf polinate
         *
         * 0 attractiveness can be caused by either 0 fitness of 0 preference.
         * In both cases that plant will never be taken as a mating partner.
         * However, fitness of 0 does not cause the plant to die.
         */

        if(((Plant) plantsNear[maximalRegister]).getAttractiveness() == 0){
            return this;
        }
```

```
        /*
         * We know which is the highes attractiveness value. If there are more than one plants having it,
         * find all cells which have the highest attractiveness and pick one of them in random !
         */


        ArrayList cellsWithHighestEqualAtractiveness = new ArrayList();

        for(int i = 0; i < plantsNear.length; i++){

            /* Finding of the cells that contain the maximal atractivenesses
             */

            if(((Plant) plantsNear[i]).getAttractiveness() == lastBiggest){
                cellsWithHighestEqualAtractiveness.add(((Plant) plantsNear[i]));
            }
        }

        if(cellsWithHighestEqualAtractiveness.size() == 1){
            /*If there is onyl one, take that one
             */
            return ((Plant) plantsNear[maximalRegister]);
        }
        else{
            /*If there are more, take a random
             */
            return ((Plant) cellsWithHighestEqualAtractiveness.get(GeneScape.rand.nextInt(
            cellsWithHighestEqualAtractiveness.size())));
        }
    }




        /**
         * Mating is means combining the Egg of this plant with the pollen of the most attractive mate
         */

    public void mating(){
        this.genome = fusionOfGenomes(this.egg, bestMate().pollen);
    }


/*
 * The 'Getters"and "Setters"
 *
 * Plant section
 */

public boolean getGMO() {
    return this.isGMO;
}

public void setGMO(boolean _isGMO) {
    this.isGMO = _isGMO;
}


public boolean getWild() {
    return this.isWild;
}

public void setWild(boolean _isWild) {
    this.isWild = _isWild;
}

public boolean getRemote() {
    return this.isRemote;
}

public void setRemote(boolean _isRemote) {
    this.isRemote = _isRemote;
}

public boolean getNon() {
    return this.isNon;
}

public void setNon(boolean _isNon) {
    this.isNon = _isNon;
}

public boolean getHasGmoGene() {
    return this.hasGmoGene;
}

public void setHasGmoGene(boolean _hasGmoGene) {
    this.hasGmoGene = _hasGmoGene;
}


/*
 * Pollen section
 */

public boolean getPollenIsGMO() {
    return this.pollenIsGMO;
}

public void setPollenIsGMO(boolean _pollenIsGMO) {
    this.pollenIsGMO = _pollenIsGMO;
```

```
}


public boolean getPollenIsWild() {
    return this.pollenIsWild;
}

public void setPollenIsWild(boolean _pollenIsWild) {
    this.pollenIsWild = _pollenIsWild;
}

public boolean getPollenIsRemote() {
    return this.pollenIsRemote;
}

public void setPollenIsRemote(boolean _pollenIsRemote) {
    this.pollenIsRemote = _pollenIsRemote;
}

public boolean getPollenIsNon() {
    return this.pollenIsNon;
}

public void setPollenIsNon(boolean _pollenIsNon) {
    this.pollenIsNon = _pollenIsNon;
}

public boolean getPollenHasGmoGene() {
    return this.pollenHasGmoGene;
}

public void setPollenHasGmoGene(boolean _pollenHasGmoGene) {
    this.pollenHasGmoGene = _pollenHasGmoGene;
}

/*
 * Egg section
 */

public boolean getEggIsGMO() {
    return this.eggIsGMO;
}

public void setEggIsGMO(boolean _eggIsGMO) {
    this.eggIsGMO = _eggIsGMO;
}


public boolean getEggIsWild() {
    return this.eggIsWild;
}

public void setEggIsWild(boolean _eggIsWild) {
    this.eggIsWild = _eggIsWild;
}

public boolean getEggIsRemote() {
    return this.eggIsRemote;
}

public void setEggIsRemote(boolean _eggIsRemote) {
    this.eggIsRemote = _eggIsRemote;
}

public boolean getEggIsNon() {
    return this.eggIsNon;
}

public void setEggIsNon(boolean _eggIsNon) {
    this.eggIsNon = _eggIsNon;
}

public boolean getEggHasGmoGene() {
    return this.eggHasGmoGene;
}

public void setEggHasGmoGene(boolean _eggHasGmoGene) {
    this.eggHasGmoGene = _eggHasGmoGene;
}

/*
 * Plant Section
 */

public int getFitness() {
    return fitness;
}

public void setFitness(int _fitness) {
    this.fitness = _fitness;
}

public int getNumPollen() {
    return numPollen;
}

public void setNumPollen(int _numPollen) {
    this.numPollen = _numPollen;
}

public void  setAttractiveness(float _attractiveness){
```

```
      attractiveness = _attractiveness;
  }

  public float getAttractiveness(){
      return attractiveness;
  }



  public void setGmoGenePresent(){

  }

  public int getGmoGenePresent(){
      if(hasGmoGene){
          return 1;
      }
      else{
          return 0;
      }
  }


  /** Coloring :
   * The color to paint this agent;
   * Red if in contains the GMO gene,
   * Green if W
   * Blue if  R
   * Yellow if N
   *
   *NB colouring for gmo is based on wether the gmo gene is present, not on the species
   */


  public Color getColor() {
              if (getEggHasGmoGene()) {
                  return Color.red;
}
if (getEggIsWild()) {
 return Color.green;
}
       if (getEggIsRemote()) {
 return Color.yellow;
}
              if (getEggIsNon()) {
 return Color.blue;
}
              else {
                    return Color.white;
              }


  }

  public String getName () {

              if (getEggIsGMO()) {
                  return "GMO";
}
if (getEggIsWild()) {
 return "Wild";
}
       if (getEggIsRemote()) {
 return "Remote";
}
              if (getEggIsNon()) {
 return "Non";
}
              else {
                  return "Unknown";
              }
  }








  /*******************************************************
   * Utilities
   *******************************************************/

  /* printing of the genome
   */

  public void genomePrint(int[] genomeToPrint) {

      System.out.println("The␣genome␣is␣");
      int counter = 0 ;

      if(genomeToPrint.length == (GeneScape.numberOfSpeciesGenesInGenome +
      GeneScape.numberOfFitnessGenesInGenome)){
          for(int i=0 ; i < genomeToPrint.length; i++){
              System.out.println(GeneScape.genomeStringExplanation[i] + "␣=␣" + genomeToPrint[i]);
          }
      }
      else {
          for(int i=0 ; i < genomeToPrint.length; i++){
```

```
             System.out.println(GeneScape.genomeStringExplanation[counter] + ”␣=␣” + genomeToPrint[i]);
             counter = counter + 2;
         }
     }
 }


/*
 *Fitness evaluation
 */


 public int fitnessOf(int[] stringToEvaluate) {
    /* Calculates the fitness of the genome according to the  information
     * in fitnessofGenes varable. It works only on a diploid genome
     **/

    int fitnessToEvaluate = 0;
         for(int i = GeneScape.numberOfSpeciesGenesInGenome ; i <
         (GeneScape.numberOfSpeciesGenesInGenome+GeneScape.numberOfFitnessGenesInGenome); i++) {
             if (stringToEvaluate[i] == 1)  {
                 fitnessToEvaluate += GeneScape.fitnessOfGenes[i-GeneScape.numberOfSpeciesGenesInGenome];
             }
         }
         return fitnessToEvaluate;
     }

     public ArrayList speciesGeneCounter(int[] stringToEvaluate) {
     /* Returns the position in the GeneScape.genomeStringExplanation of the
      * genes that determine the species of the genome. It can handle both diploid and haploid genomes
      */

         ArrayList speciesGenes = new ArrayList();

         if(stringToEvaluate.length > ((GeneScape.numberOfSpeciesGenesInGenome +
         GeneScape.numberOfFitnessGenesInGenome)/2)){
             for(int i=0; i < GeneScape.numberOfSpeciesGenesInGenome; i++){
                 if(stringToEvaluate[i] == 1){
                     speciesGenes.add(new Integer(i));
                 }
             }
             return speciesGenes;
         }
         else{

             for(int i=0; i < (GeneScape.numberOfSpeciesGenesInGenome/2) ; i++){
                 if(stringToEvaluate[i] == 1){
                     speciesGenes.add(new Integer(2*i));

                 }
             }
             return speciesGenes;
         }
     }

     public  ArrayList fitnessGeneCounter(int[] stringToEvaluate) {
    /* Returns the position in the GeneScape.genomeStringExplanation of the
     * genes that determine the fitness of the genome
     * It can handle both diploid and haploid genomes
     **/
         ArrayList fitnessGenes = new ArrayList();

         if(stringToEvaluate.length > ((GeneScape.numberOfSpeciesGenesInGenome +
         GeneScape.numberOfFitnessGenesInGenome)/2)){
             for(int i = GeneScape.numberOfSpeciesGenesInGenome; i <
             (GeneScape.numberOfFitnessGenesInGenome+GeneScape.numberOfSpeciesGenesInGenome); i++){
                 if(stringToEvaluate[i] == 1){
                     fitnessGenes.add(new Integer(i));
                 }
             }

             return fitnessGenes;
         }
         else {
             for(int i = (GeneScape.numberOfSpeciesGenesInGenome/2); i <
             (GeneScape.numberOfFitnessGenesInGenome+GeneScape.numberOfSpeciesGenesInGenome)/2; i++){
                 if(stringToEvaluate[i] == 1){
                     fitnessGenes.add(new Integer(2*i));
                 }
             }
             return fitnessGenes;
         }
     }

 /*
  *It is important to ensure that the species of the pollen is determined by the mother plant
  * At the current setup the plant is a product of two species, and has two species genes,and the species
  * plants can chage their species by making a new egg inthe second generation.
  *This should be resttricted.
  */

 public  int[] mendeleanSplitOfGenome(int[] stringToSplit, int species) {
     /*
      *splits the incoming string into the mendelean genome depending , conserving the species
      *
      *of each pair of genes, one is randomly picked, and added to the new string. This way,
      * mendel proportions are conserved.
      * The species of the split String is conserved
      * this has been fixed
      *However, it can happen that if there are two species genes, both with single haplod, that
      *the funciton produces a plant without species genes. This one cannot mate, since no
```

```
          * preference can be calculated for it
          *(unless the preference function has a default preferenc not equal to 0)
          */

        int positionHolder = 0;
        int numberOfSpeciesGenesInSplitGenome = 0;
        int[] splitGenome = new int[stringToSplit.length/2];
        int[] splitGenomeSpeciesGenePlaces = new int[GeneScape.numberOfSpeciesGenesInGenome/2];


            for(int i = 0; i < stringToSplit.length; i=i+2){
             splitGenome[positionHolder] = stringToSplit[i + GeneScape.rand.nextInt(2)];

            if((splitGenome[positionHolder] == 1) & (positionHolder <
            (GeneScape.numberOfSpeciesGenesInGenome/2))){
               splitGenome[positionHolder] = 0;
                splitGenomeSpeciesGenePlaces[numberOfSpeciesGenesInSplitGenome] = positionHolder;
                numberOfSpeciesGenesInSplitGenome++;
            }
            positionHolder++;
            splitGenome[species/2] = 1;
        }

        /*
        }
        else{
        for(int i = 0; i < stringToSplit.length; i=i+2){
            splitGenome[positionHolder] = stringToSplit[i + GeneScape.rand.nextInt(2)];

            if((splitGenome[positionHolder] == 1) & (positionHolder <
          (GeneScape.numberOfSpeciesGenesInGenome/2))){
                splitGenome[positionHolder] = 0;
                splitGenomeSpeciesGenePlaces[numberOfSpeciesGenesInSplitGenome] = positionHolder;
                numberOfSpeciesGenesInSplitGenome++;
            }
            positionHolder++;
            splitGenome[species/2] = 1;
        }
        }
         */

     /* the chek wether not too many species determination genes are being passed on
       * if too few are passed on, no-species plants can be created.
       *
       */
        /*
        if(numberOfSpeciesGenesInSplitGenome > 1){
            //splitGenome[splitGenomeSpeciesGenePlaces[rand.nextInt(numberOfSpeciesGenesInSplitGenome)]]=0;
            System.out.println("Too many species genes in egg/pollen !! There are " +
          numberOfSpeciesGenesInSplitGenome );
        }
        if(numberOfSpeciesGenesInSplitGenome == 0){
            System.out.println("No species genes in egg/pollen !!");
        }
         */

        return splitGenome;

    }

    public  int[] fusionOfGenomes(int[] stringToFuseOne, int[] stringToFuseTwo) {
        /* father genes go first
         */

        int placeHolderOne = 0;
        int placeHolderTwo = 1;
        int[] fusedString = new int[stringToFuseOne.length + stringToFuseTwo.length];

        if(stringToFuseOne.length == stringToFuseTwo.length){
            for(int i = 0; i < stringToFuseOne.length; i++){
                fusedString[placeHolderOne] = stringToFuseOne[i];
                fusedString[placeHolderTwo] = stringToFuseTwo[i];
                placeHolderOne = placeHolderOne + 2;
                placeHolderTwo = placeHolderTwo + 2;
            }
        }
        else { System.out.println("Lengths␣of␣the␣strings␣to␣fuse␣must␣be␣equal␣!!");}
      return fusedString;
    }

    public static void stringPring(int[] string){
        System.out.println("String␣is:␣");
    for(int i=0; i < string.length; i++){
        System.out.println(string[i]);
    }
    }

    public  int[] randomFitnessGenome(int species) {
         /* randomFitnessGenome takes in an int describing the begining of the species determination
          * gene sequence, and returns
          * genome with the correct species determination sequence and a randomly created
fitness
          * genome ( without GMO genes)
          */

        int[] newRandomGenome = new int[GeneScape.numberOfSpeciesGenesInGenome +
        GeneScape.numberOfFitnessGenesInGenome];
        for(int i = 0; i < newRandomGenome.length; i++){
            if(i < GeneScape.numberOfSpeciesGenesInGenome){
                if(i == species | i == species+1 ){
                    newRandomGenome[i] = 1;
```

```
               }
               else{
                   newRandomGenome[i] = 0;
               }
           }

           if(i >= GeneScape.numberOfSpeciesGenesInGenome ){
               newRandomGenome[i] = GeneScape.rand.nextInt(2);
           }
       }
       /* making sure that no GMO genes  are initialized
        */
       newRandomGenome[8] = 0;
       newRandomGenome[9] = 0;
       //, genomePrint(newRandomGenome);
       return newRandomGenome;
   }

   public int[] mutate(int[] stringToMutate) {
       /* mutate() takes in a string, chooses a random position in the genome to mutate.
        * than it checks wether the value is a species determination gene.
        * If not, the 1 at that position is turned to 0, and vice versa
        * If so,  a check is performed wether the position to mutate is allready 1.
        * If not,  it is turned to 1, and one of the 1's in the original position is randomly turned to 0
        * If the position to mutate is allready on 1, than that position is turned to zero, and a position
        * two places up (or down, depending on which end the mutation is) is turned to 1
        *
        * None of the above.....
        *
        * Mutation takes place wherever it can, and the world determines if certain mutations "survive"
        * so it there are three species genes, or none, or one, the  polen is not viable...
        *
        *(07/05/01)
        * This can all be turned on, but in
        * the GeneScape implementation 1.3 the mutation only takes place in the fitness part of the genome
        */

        int positionToMutate = GeneScape.rand.nextInt(GeneScape.numberOfFitnessGenesInGenome);
        int[] mutatedString = stringToMutate ;

     /* boolean mutationLocationConflictFlag = false;
        int mutationLocationConflictPlace = 0;
        int[] conflictLocation = Genome.speciesGeneCounter(stringToMutate);

        System.out.println("Mutation at " + positionToMutate);
        for(int i=0; i < Genome.speciesGeneCounter(stringToMutate).length; i++){
      System.out.println("ConflictLocation " + conflictLocation [i] );}

           if(positionToMutate < numberOfSpeciesGenesInGenome){
                            for(int i=0; i < conflictLocation.length; i++){
                            if(conflictLocation[i] == positionToMutate){
                                mutationLocationConflictFlag = true;
                                mutationLocationConflictPlace = i ;
                                // System.out.println("mutationLocationConflictPlace
" +
      mutationLocationConflictPlace);
                            }
                        }
                        if(mutationLocationConflictFlag) {
                            if(conflictLocation[mutationLocationConflictPlace]+2 >=
      numberOfSpeciesGenesInGenome){
                            stringToMutate[conflictLocation[mutationLocationConflictPlace]-2] = 1;
                            // System.out.println(" -2 chosen");
                            }
                            else{stringToMutate[conflictLocation[mutationLocationConflictPlace]+2] = 1;
                            // System.out.println(" +2 chosen");
                            }
                            stringToMutate[conflictLocation[mutationLocationConflictPlace]] = 0;
                        }
                        else{
                        stringToMutate[conflictLocation[rand.nextInt(conflictLocation.length)]]=0;
                        stringToMutate[positionToMutate]=1;
                        }
        }


          if(positionToMutate >= numberOfSpeciesGenesInGenome){

         */
            if(stringToMutate[GeneScape.numberOfSpeciesGenesInGenome+positionToMutate] == 1) {
    mutatedString[GeneScape.numberOfSpeciesGenesInGenome+positionToMutate]=0;
    }
            else {
    mutatedString[GeneScape.numberOfSpeciesGenesInGenome+positionToMutate]=1;
    }
        //}

        return mutatedString;
   }

   public int case1Species(){
        int species = 2;
        float x = 6;

                    if(GeneScape.rand.nextFloat() <= 1/x){species = 0;}
                    if(GeneScape.rand.nextFloat() > 1/x && GeneScape.rand.nextFloat() <= 3/x){species = 1;}
                    if(GeneScape.rand.nextFloat() > 3/x && GeneScape.rand.nextFloat() <= 6/x){ species = 2;}
                  return species;
   }

}
```