PREVENTING INBREEDING

NEW MEXICO ADVENTURES IN

SUPERCOMPUTING CHALLENGE

FINAL REPORT

APRIL 5, 2006

TEAM 6

ALBUQUERQUE ACADEMY

Project Members:

Franklin Orndorff-Plunkett

Punit Shah

Karalyn Baca

Teacher:

Jim Mims

Mentor:

Kevin Fowler

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Inbreeding is a problem experienced by many organizations attempting to revive endangered species. Inbreeding is when a small, isolated population of animals breeds with highly genetically similar members of the species. In species revival programs, small gene pools that result in inbreeding magnify species' genetic problems, which then hurt the animals' populations further. With a larger population, these detrimental traits can instead be suppressed.

This project aims to find the minimum size for a gene pool that would prevent excessive inbreeding. Such information would assist the previously described breeding programs form more successful populations in facilities with limited room, like zoos, where these programs often take place.

We created a computer model of a population defined by a number of parameters, such as the initial number of males and females and the number of organisms with a certain genetically defined trait. Each animal had an ID number and were grouped by gender. The model then created the animals and allowed them to mate at random with the other gender. The program successfully showed that larger gene pools resulted in more successful populations because they prevented inbreeding. Basically, any population where the number of matings between two closely related individuals is the healthiest for a population of organisms.

This project successfully modeled a population using an innovative approach. This method of creating individual animals with specific traits for each animal was innovative compared to the traditional approaches to simulation populations. Additionally, the approach allows for further expanding to widen the areas of use.

INTRODUCTION

To understand the development process we've taken in our project we must define the terms and variables of our problem. Following are definitions of all the terms we use in genetic science that one must understand to evaluate the project.

- Genotype- the total set of genes (DNA sequences) that control an organism's chemistry and physiology, as distinguished from the physical appearance, of an organism or a group of organisms.
- Phenotype- the set of physical characteristics of an organism that are passed down to offspring.
- Allele- one constituent part of a pair or series of genes that occupy a specific position on a specific chromosome.
- Heterozygous- having different alleles at one or more corresponding chromosomal loci.
- Homozygous Recessive- having two recessive alleles (generally a trait is only activated with homozygous dominant and homozygous recessive generally makes the animal lose some physical feature)
- Homozygous Dominant- having to dominant alleles (generally
- Heterozygosity- the state of an individual of being heterozygous or the relative number of heterozygous organisms in a population
- Inbreeding- mating between genetically distant populations that leads to a lesser heterozygosity and more homozygosity within and therefore, if the trait has a negative impact on a species, that population will become less and less genetically health

- Identical By Descent- two alleles are identical by descent if they are identical copies of the same allele in an earlier generation
- Inbreeding Coefficient- this is the calculable value by which scientists measure inbreeding.
- Outbreeding- mating between two individuals that are very distantly related that causes an increase in heterozygosity, generally lessening the prevalence of a negative trait within a population to make it more healthy
- Non-random Mating- mating in which individuals are selected for a certain phenotype
- Assortative Mating- positive mating between individuals of a certain phenotype (positive- same phenotype; negative- different phenotypes)
- Sexual Selection vs. Natural Selection- sexual selection- based on phenotypes ; natural selection- nature excludes weak individuals from mating either with environmental stresses or sterility

Our program is based on Mendelian genetics, which can be succinctly summarized in a Punnet Square (Fig.1), while this diagram shows both parents as heterozygotes, a parent may also be a heterozygote and the same principles would apply. In the simplest case we will focus on one gene with only two alleles, recessive and dominant. Each parent has two alleles in their genotype and can pass only one of these alleles on to their offspring and there is an equal chance that each will be passed along. Each parent contributes one allele and these two alleles make up the two allele genotype of the offspring. This process can become much more complicated if the gene has more alleles, resulting in successively larger Punnet Squares.

While researching for this project, we came over many different case studies that showed the effects of inbreeding on a population. One, for instance, involved the New Mexico’s Rio Grande silvery minnow. Inbreeding was preventing its population from increasing because the recessive traits were coming out in the highly endangered species. This was further contributing to the demise of the species along with habitat loss. This demonstrated the problems inbreeding creates.

We are in the process of implementing equations for finding frequencies of genotypes using the frequencies of alleles and then using the genotypic frequencies obtain the inbreeding coefficient. The following are, in order, the equations for determining the genotypic frequencies, the algebra to find the inbreeding coefficient and the already established and accepted equation to calculate the effective population size based on the traits of the organisms:

- P(Homozygous dominant set) = $p^2 + fpq$ (the first term is the allelic frequency of the dominant or capital allele; the second term is the factor by which the dominant homozygosity increases or decreases as a result of inbreeding or outbreeding)
- P(heterozygous set) = H = $2pq+ 2fpq$ (the first term stands for the aA allele set and the second for the Aa allele set because either of the two can be expressed in an offspring)
- P(homozygous recessive set) = $q^2 + fpq$ (the first term represents the allelic frequency of the recessive allele; the second term is the factor by which the recessive homozygosity will increase or decrease as a result of inbreeding or outbreeding)
- $H= 2pq + 2fpq$ can be rearranged to form this equation: $H= 2pq(1-f)$ which can be algebraically manipulated into $f= 1 - (H / 2pq)$ which is how in the end we solve for the inbreeding coefficient.

We have not included factors like epistasis, which is when two genes interact in such a way that the dominant of one affects the outcome trait of the other. This is not entirely too complex but in the meantime we will perfect the process of simulating a closed mating population and then calculating the homozygosity of both the dominant, recessive, and heterozygous allele sets and from those calculating the inbreeding coefficient. We have not found a way to calculate the lower limit of a population to prevent "dangerous" inbreeding but we will have a foundation to work from and actual data to evaluate and further research will show us what the healthy value of the inbreeding coefficient for a population would be.

DESCRIPTION AND METHOD

Our program is designed to simulate a breeding population of organism and observe the success of two characteristics. This characteristic is present in the population only when the organism posses two recessive alleles, we assume it is a detrimental one because these characteristics pose the greatest threat to endangered breeding populations in the wild and in controlled environments.

To simulate a breeding population, we at first give the user the option to create a number of organisms, male and female, to begin the population. The program also allows the user to determine the number of organism that will have one of the three genotypes. A higher frequency of homozygous recessive individuals will be entered if the detrimental characteristic is especially prevalent among the species. Then the animals are allowed to mate at random with a member of the opposite sex. Two arrays, arrayfemale and arraymale, contain elements that are objects of the class organism. By choosing a random number between one and the total number of females or males present (this number corresponds with the elements number in the array), the parents of the offspring are determined. The genotype of each of the parents will determine the genotype of the offspring. Each allele from a parent has an equal chance of being expressed, in other words each allele from one parent has a one in two chance of being expressed. Choosing a random number between zero and one, being one or the other and assigning the value to the homozygous of heterozygous allele, achieves this random chance in the program. Zero represents a recessive allele while one represents the dominant allele. However if the parent is homozygous the parent must pass on the dominant allele if the parent is homozygous dominant and the recessive allele if the parent is homozygous recessive. The program will inform the user of the success rate of the allele by showing which organisms present have homozygous recessive, homozygous dominant, or heterozygous genotypes.

RESULTS

Our results indicate that the larger a population is, the less likely a detrimental characteristic is to be present. Large populations ensure that organisms have a wide array of mates to choose from, presumably the fitter ones will be more attractive to the opposite sex. This large population also ensures there is competition for resources such as food and territory, or mates so that only the most fit animal, the one that does not posses the detrimental characteristic will reproduce. When there are fewer organisms present in a population, there are fewer mates to choose from and the organism cannot choose the most fit of the population. This also ensures that closely related organisms are not forced to mate with each other. A larger population, again, provides more choice for the organism seeking a mate thus reducing the chance that they will mate with a parent or sibling.

ANALYSIS

A larger population is essentially a healthier one. The more organisms there are present, the less successful a detrimental characteristic is. This information is especially pertinent to endangered species that are only present in small numbers in the wild. These organisms are at the greatest risk for inbreeding, or the success of a detrimental characteristic, as opposed to their fellow animals which are present in larger amounts.

Any small populations of animals who are isolated are in danger of serious inbreeding and thus extinction. Detrimental characteristics are more likely to become pronounced in these small groups and these characteristics are termed detrimental because they decrease the fitness of the animal and are thus a danger to their livelihood. We need to be aware of the risks that endangered animals in the wild are facing and strengthen captive breeding programs and ensure that animals in the wild are not isolated from each other and have access to a large gene pool to ensure the success of their species.

CONCLUSIONS

We can conclude that whenever possible, it is best to ensure that a large number of animals are present in an area where they might otherwise be isolated from others of their kind. It is also important to ensure that the animals present represent a diverse gene pool, which minimizes the impact of a detrimental characteristic by introducing stronger characteristics in organisms that will eventually reproduce. For captive breeding programs it is most important to ensure that the animals are not closely related, which is the easiest way to prevent inbreeding in a controlled environment.

The fact that small populations of animals are more prone to weakening of their species by the prevalence of genetic defects reminds us of the importance of thought in bringing endangered species back to healthy, large populations. Small, isolated populations, especially those isolated by man-made constructions such as highways or cities are by no means exempt from the process of natural selection, which generally favors a large, diverse gene pool. Wild populations of species should be given the largest possible habitats that include a number of the same species.

ACHIEVEMENTS OF THE PROJECT

The project successfully modeled the populations. This means that we can use this project for further study into inbreeding and population progressions, both in endangered species revival programs as well as in nature. This would help scientists gain additional insight when attempting to protect animals about the necessities genetically for internal population reproduction. Additionally, the project tells us more about the size of populations that breeding programs should strive to remain.

Though this problem has been approached in different ways previously, this project takes on an innovative approach that uses individual animals as objects, which allows them to carry certain individual traits. This approach has proven to be effective and is an achievement of the project because this sort of approach may be able to be used in other problems with involved interactions between different entities in a specific population.

The project is also very expandable, which gives it many more applications outside of the areas it may have originally been designed for. For example, as animals encounter harder and harder times due to stressed environments, the program's design allows us to consider this by possibly adding in calculated probability for reproduction due to environmental factors.

RECOMMENDATIONS

Object-oriented programming is very useful when dealing with virtual individual organisms of one class within which there are arrays with the individual's information, in our case genotype and thus place on the continuum of sexual selection and natural selection. Some of the mathematics behind the genetics is rather complex and formulating the algorithms to calculate values like the inbreeding coefficient has been a challenge. We would hope to add this higher level of mathematics if we continued our project.

ACKNOWLEDGEMENTS AND CITATIONS

ACKNOWLEDGEMENTS

We would like to thank our teacher, Mr. Jim Mims. He has helped guide us and push us to succeed. Additionally, he ensured our project stayed on schedule. We also extend our thanks to our mentor, Mr. Kevin Fowler. We would especially like to thank Franklin's mother, Mrs. Mable Orndorff, for providing suggestions for our interim presentation.

CITATIONS

WEBSITES

Buchanan, David S. *Swine Genetics.* Fact Sheet from Purdue University.
http://www.osuextra.okstate.edu/pdfs/F-3606web.pdf

Non-Random Mating and Inbreeding
http://bio.classes.ucsc.edu/bio107/Class%20pdfs/W05_lecture7.pdf

Hardy Weinberg Equation of Population Genetics
http://www.people.virginia/edu/~rjh9u/hwequa.html

BIOL2007-Inbreeding and Neutral Evolution
http://www.ucl.ac.uk/~ucbhdjm/courses/b242/InbrDrift/InbrDrift.html

Carlson, Peter C., William S. LaHaye, and Alan B. Franklin. "Incestuous Behavior in Spotted Owl." The Wilson Bulletin 110.4 (Dec. 1998): 562-4. Searchable Ornithological Research Archive (SORA). Ed. Blair O. Wolf, SORA Coordinator. 2004. The University of New Mexico, New Mexico. 23 Oct. 2005 http://elibrary.unm.edu/sora/Wilson/v110n04/p0562-p0564.pdf

"China's Highway Cut Panda Habitat Into Pieces." AFP, Yahoo News 5 Dec. 2005. 17 Oct. 2005
http://news.yahoo.com/s/afp/20051205/sc_afp/chinaanimalspanda_051205172511

Hartwell, Sarah. "Pros and Cons of Inbreeding." 2003. 14 Oct. 2005
http://www.messybeast.com/inbreed.htm

“Tiger Handbook- Captive Breeding Criteria.” Tiger Information Center. 2005. Save The Tiger Fund. 6 Nov. 2005
http://www.savethetigerfund.org/AllAboutTigers/Adventures/handbook/d2a.htm#inbreeding

DATABASES

Frankam, Richard. *Genetics and Extinction.* March 2005. “Biological Conservation 126 (2005)

131-140. http://www.sciencedirect.com. Last accessed April 1, 2006.

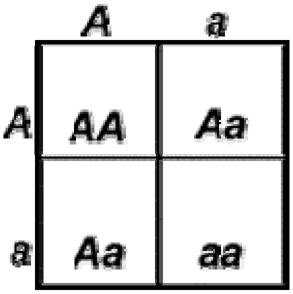
BOOKS

Campbell, Neil A., Dr. “Mendel and the Gene Idea, The Chromosomal Basis of Inheritance, and The Evolution of Populations.” Biology. 4th ed. Menlo Park, CA: Benjamin/Cummings Publishing Company, 1996. 238-61, 262-78, and 416-33.

General use of :
Deitel, H.M., et al., “C++ How to Program.” 4th ed. Upper Saddle River, NJ: Pearson Education Inc., 2003

APPENDICES

APPENDIX A:



	A	a
A	AA	Aa
a	Aa	aa

FIGURE 1

APPENDIX B:

MAIN:

```
/*
AiS Challenge 05-06
Team 6
Karalyn Baca
Franklin Orndorff-Plunkett
Punit Shah
*/
//Please note any text following '//' or between '/*' and '*/' is a comment
#include <iostream.h>
#include <stdlib.h>
#include "organism_h.h"
```

```
int totalfemale;

int totalmale;

int totalcreatures;

int matingprogram ();

int main ()

{

int initialnumberfemale;

int initialnumbermale;

int total;

cout<< "Prevention of Inbreeding AiS 2005-06"<<endl;

cout<<""<<endl;


//females

cout<<"Enter the initial number of females"<<endl;

cin>>initialnumberfemale;

cout<<initialnumberfemale<<" Females"<<endl;

//males

cout<<"Enter the initial number of males"<<endl;

cin>>initialnumbermale;

cout<<initialnumbermale<<" Males"<<endl;

//max number of organisms

cout<<"What is the maximum number of the population you wish to simulate?"<<endl;

cin>>total;
```

```
//arrays to hold males and females

organism arrayfemale [total];

organism arraymale [total];

//genotypes of organisms

int hdomfirst;//homozygous dominant

int hrecfirst;//homozygous recessive

int hetfirst;//heterozygous

int hdomsecond;

int hrecsecond;

int hetsecond;


do{

//first trait

cout<<"First trait:"<<endl;

cout<<"How many organisms will be homozygous dominant?"<<endl;

cin>>hdomfirst;

cout<<"How many will be homozygous recessive?"<<endl;

cin>>hrecfirst;

cout<<"How many will be heterozygous?"<<endl;

cin>>hetfirst;

//second trait

cout<<"Second trait:"<<endl;

cout<<"How many organisms will be homozygous dominant?"<<endl;

cin>>hdomsecond;

cout<<"How many will be homozygous recessive?"<<endl;

cin>>hrecsecond;
}

cout<<"How many will be heterozygous?"<<endl;

cin>>hetsecond;

} while ((hdomfirst+ hrecfirst+hetfirst) !=total)

//note, this checks numbers entered

totalfemale = initialnumberfemale;

totalmale = initialnumbermale;


//assigning genotypes

for (int a=0;  a++; a>=(hdomfirst/2))

{

        for (int b= 0;  b++, b>=(hetsecond/2))

        {

        organism creature(1,1,0,1, 1);//female

        totalfemale = totalfemale +1;

        arrayfemale[b]= creature;

        }

}


for (int a=0;  a++; a>=(hdomfirst/2))

{

        for (int b= 0; b++; b>=(hetsecond/2))

        {

        organism creature(1,1,0,1,0);//male

        totalmale = totalmale +1;

        arraymale[b]=creature;

        }
```

```
}

///////////

for (int a1=0; a1++; a1>=(hrecfirst/2))

{



        for (int b= 0; b++; b>=(hdomsecond/2))

        {

        organism creature(1,1,0,1, 1);//female

        totalfemale = totalfemale +1;

        arrayfemale[(hetsecond/2)+b]=creature;

        }

}


for (int a=0; a++; a>=(hrecfirst/2))

{



        for (int b=0; b++; b>=(hdomsecond/2))

        {

        organism creature(1,1,0,1,0);//male

        totalmale = totalmale +1;

        arraymale[(hetsecond/2) +b] = creature;

        }

}


////////////////
```

```
for (int a=0;a++; a>=(hetfirst/2))
{


        for (int b=0;  b++; b>=(hrecsecond/2))
        {
        organism creature(1,1,0,1,1);//female
        totalfemale = totalfemale +1;
        arrayfemale[(hdomsecond/2)+ (hetsecond/2) +b]= creature;
        }
}

for (int a2=0; a2++; a2>=(hetfirst/2))
{


        for (int b=0; b++; b>=(hrecsecond/2))
        {
        organism creature(1,1,0,1,0);//male
        totalmale = totalmale +1;
        arraymale[(hdomsecond/2)+ (hetsecond/2) +b]= creature;
        }
}




do
{
```

```
matingprogram();
}
while (totalcreatures<=total);

return 0;
}




//Functions////
///
///

int matingprogram ()
{
//mating
//randomly choose a male and female

int y;
        y = ( rand()% (totalfemale+1));

int k;
        k = ( rand()% (totalmale+1));

//mother= arrayfemale[y]
//father= arraymale[k]
```

```
int pf;

int qf;

int rf;

int sf;


arrayfemale[y].oarray[0]=pf;

arrayfemale[y].oarray[1]=qf;

arrayfemale[y].oarray[2]=rf;

(arrayfemale[y]).oarray[3]=sf;


int pm;

int qm;

int rm;

int sm;


arraymale[k].oarray[0]=pm;

arraymale[k].oarray[1]=qm;

arraymale[k].oarray[2]=rm;

arraymale[k].oarray[3]=sm;


//gender

int q;

        q= rand () % 2;//either 0 or 1

//another random number to determine which allele is passed from each parent

// 1 allele from each parent for each characteristic
```

```
int a;
        a= rand () % 2;//either 0 or 1



int m,n,o,p;
if (a==0)
{
m=pm;
n=qf;
o=rm;
p=sf;
}
else
{
m=pf;
n=qm;
o=rf;
p=sm;
}
creature.mother = ((arrayfemale[y]).idnumber);
creature.father = ((arraymale[k]).idnumber);
if (q==1)
{
arrayfemale[totalfemale] = (organism creature(m,n,o,p,q));
totalfemale = totalfemale +1;
}
```

```
    else
    {
    arraymale[totalmale]= = (organism creature(m,n,o,p,q));
    totalmale = totalmale+1;
    }

    int totalcreatures;
    totalcreatures = (totalfemale+totalmale);
    return 0;
}
```

ORGANISM. CPP:

```
    #include "organism_h.h"
    #include <iostream.h>
    #include <stdlib.h>


    organism::organism(int a, int b, int c, int d, int e ) //genotype, last is gender
    {

    idnumber = (idgenerator());

    //organism's array, contains alleles
```

```
oarray[0] = a;

oarray[1] = b;

oarray[2]= c;

oarray[3] = d;


gender= e;


}




long organism::idgenerator()

{

int id;

id= (10+ rand() %4000);


        //need to check to ensure id number is unique


        // if it isn't, assign new number


        for (int a= 0; a++; a>=(totalmale))

                if (id==arraymale[a].idnumber)

                {

                        id = (20+ rand()% 5032);

                }


        for (int b = 0; b++; b>=(totalfemale))
```

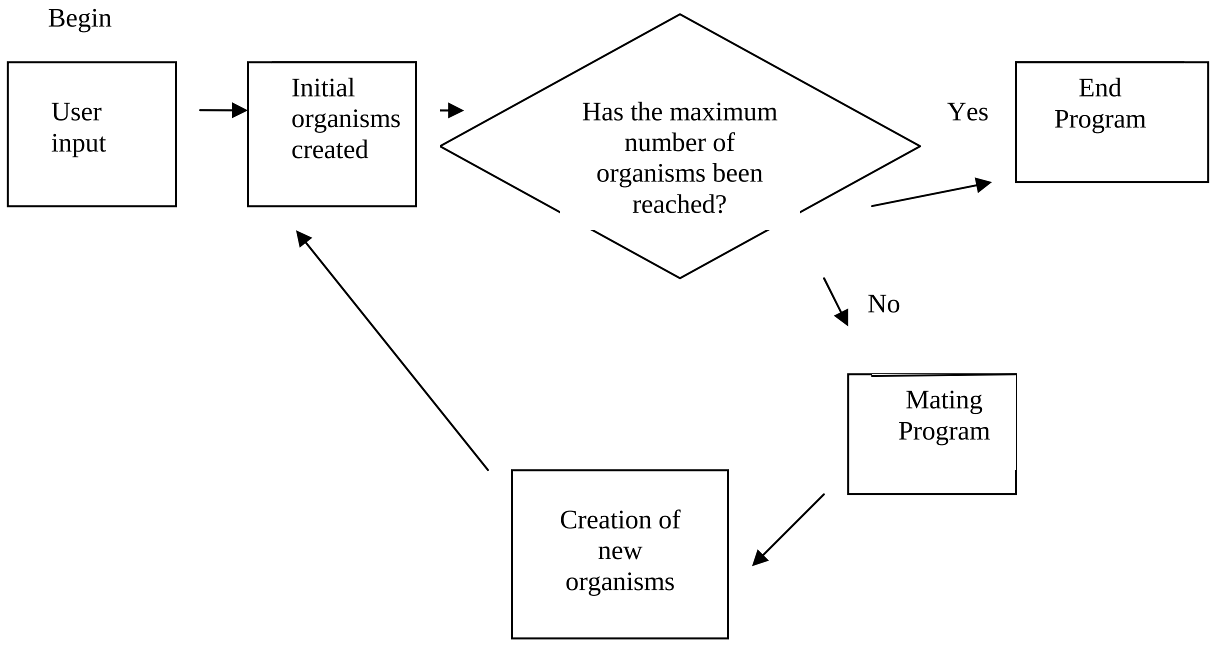
```
            if (id ==arrayfemale[b].idnumber)

                id = (15 + rand()% 3033);

            }


    return id;

}
```

ORGANISM.H:

```
    #ifndef ORGANISM_H

    #define ORGANISM_H


    class organism

    {

    public: //functions


         organism(int, int, int, int, int);


        long idgenerator();//generates random id number


        int oarray[4]; //contains alleles, 0 is recessive, 1 is dominant

                        //first two are for first trait, second two for second


        int gender; // 0 is male, 1 is female


        long idnumber;
```

```
                long mother;//id number of each parent

                long father;



        };

#endif
```

APPENDIX C


Begin
User input
Initial organisms created
Has the maximum number of organisms been reached?
Yes
End Program
No
Mating Program
Creation of new organisms