Software Development
Athens, Georgia
2017

Table of Contents

Research

Background

Mendelian genetics are a simplified way to look at real-life genetics. Usually they only deal with one or two traits, with each trait being assigned a letter. Capital letters signify the dominant allele, and lowercase letters are the recessive alleles. All organisms have two copies of each alleles, and each allele can be recessive or dominant.

AA: Homozygous Dominant
Aa: Heterozygous
aa: Homozygous Recessive

We usually compute punnett squares using one or two genes, though a greater number is possible, if uncommon. We call these Dihybrid crosses.

Punnett Squares

Are 2x2 or 4x4 grids, with the parent alleles along the top. The inside is filled in by adding the alleles vertically up and horizontally left.

Monohybrid Crosses

These are 2x2 grids, made of one trait for each parent. There are at most four possible outcomes from this cross, with the child inheriting one allele from each parent.

Dihybrid Crosses

These are 4x4 grids, with each parent having two genes. There are at most sixteen possible outcomes from this cross, with the child inheriting two alleles for each parent (one allele per gene).

Unlinked/Independent

The chance of each allele being passed on is independent of its partner allele.

Linked

The genes are located near each other on the chromosome, increasing the chances that the genes will be passed on together.
They can either be homozygous linked, meaning the dominant versions are on the same chromosome, and the recessive versions are on the same chromosome. They can also be heterozygous linked, with each chromosome having a dominant copy of one gene, and a recessive copy of the other.

Sex-Linked

These are alleles that are carried on the X chromosome. Females must be homozygous dominant to display the effects of such a gene, whereas males with only one X chromosome must be heterozygous.

Blood Types

There are four different blood types, and six different genotypes.
- Type A: AA, AO
- Type B: BB, BO
- Type AB: AB
- Type O: OO

O is a recessive type, only being displayed if the organism is homozygous. Otherwise, the organism can be type A or B and carry the O allele. AB as well can only be displayed if the organism is exactly AB.

Project

Problem

Students need a way to experiment and learn about genes without using an internet connection. Additionally, they need a way to check the answers to their punnett squares. There are sources that do this, but they either do not cover every different type of Punnett square, or do not have the option to hide the parents. This allows for only a finite amount of practice.

Solution

I've created a program that computes every type of punnett square necessary for entry level biology students, as well as a few more complex ones. While it can be used as a calculator to find the answers when the parents are known, the student can also work backwards from a completed punnett square. This, along with the randomize parents button, provides an almost endless amount of practice. The entire program is very pointed towards its goal; essentially, it does what it says it will.

Educational Value

What the project lacks in polish, it makes up for in ease of use and simplicity. This simplicity allows any level of student to use the program. Additionally, it provides a valuable educational tool for teachers who may not have the time to give students additional practice problems.

Social Value

This tool could be used to spread awareness of genetics and how they work, better educating the general public. It's an easy way to lessen the ignorance around blood typing, paternity, eye color, and many other traits that are easily illustrated using punnett squares.
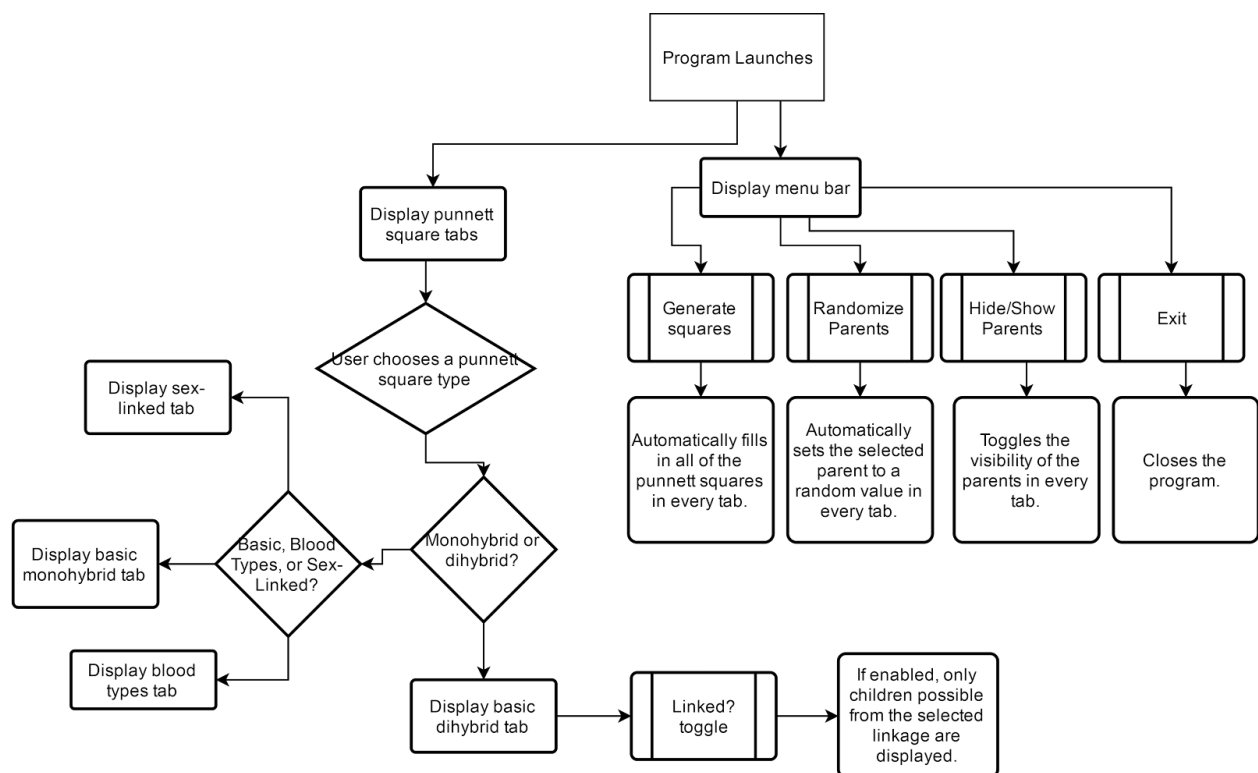
Plan of Work

(Print and fill out PDF separately)

Process

Project requirements

The project can run on any computer with java installed, which includes the vast majority of PCs. Also, since it runs as an app and does not require an install, it can be run on computers with restrictions.

High-level software design

Testing

The bottom section of the program contains debugging information, which looks like a great deal of A's and B's to the user.

For me though, they detail that how the strings are being divided and makes sure that they're all being placed correctly. The picture above is from a run of the dihybrid punnett square between the parents pictured to the left.

The console code is as follows:

```
AABb | AB Ab AB Ab A A B b
AaBB | AB AB aB aB A a B B
AB, L: AB, L1: AB, L2: ab
```

Mother

| Gene 1 | Homozygous Dominant ▼ |
| Gene 2 | Heterozygous ▼ |

Father

| Gene 1 | Heterozygous ▼ |
| Gene 2 | Homozygous Dominant ▼ |

◯ Linked? Linked Genes: | AB ▼ |

```
//console
OF.setText(dpn.getMom()[0]+" | "+dpn.getMom()[1]+" "+dpn.getMom()[02]+" "+dpn.getMom()[03]+" "+dpn.getMom()[04]+" "+dpn.getMom()[05]+" "+dpn.getMom()[06]+" "+dpn.getMom()[07]+" "+dpn.getMom()[8]+
    "\n"+dpn.getDad()[0]+" | "+dpn.getDad()[1]+" "+dpn.getDad()[02]+" "+dpn.getDad()[03]+" "+dpn.getDad()[04]+" "+dpn.getDad()[05]+" "+dpn.getDad()[06]+" "+dpn.getDad()[07]+" "+dpn.getDad()[8]+
    "\n"+linkagebox.getSelectedItem().toString()+", "+dpn.getLinkages()+
```

This also ensures that the punnett squares are being calculated correctly.

```
public String calculate(int a, int b){
    String x = a+""; String y = b+"";
    //String ans = momgenes[x]+dadgenes[y];
    String ans = momgenes[Integer.parseInt(rel1.get(x))]+dadgenes[Integer.parseInt(rel1.get(y))];
    if (ans.equals("aA")) { ans = "Aa"; }
    ans += momgenes[Integer.parseInt(rel2.get(x))]+dadgenes[Integer.parseInt(rel2.get(y))];
    if (ans.substring(2,4).equals("bB")) {ans = ans.substring(0,2) + "Bb"; }

    if(isLinked){
    if(!((ans.substring(0,1)+ans.substring(2,3)+"").equals(linkage1))){
        if(!((ans.substring(0,1)+ans.substring(2,3)+"").equals(linkage2))){
            ans = "";
        }
    }
    for(int i =0; i< unlinkX.size(); i++) {
        if(x.equals(unlinkX.get(i).toString())){
            ans = "";
        }
    }
    for(int i =0; i< unlinkY.size(); i++) {
        if(y.equals(unlinkY.get(i).toString())){
            ans = "";
        }
    }
}
public String[] calculate() {
    chigenes[0] = momgenes[1]+dadgenes[1];
    chigenes[1] = momgenes[2]+dadgenes[1];
    chigenes[2] = momgenes[1]+dadgenes[2];
    chigenes[3] = momgenes[2]+dadgenes[2];

    for(int i = 0; i<4; i++){
        if(chigenes[i].equals("aA")) {
            chigenes[i] = "Aa";
        }

    }

    return chigenes;
}
```

The code to calculate the dihybrid squares is pictured to the right, and it essentially concatenates two strings together, checks to make sure the capital letter is first, and then outputs the string.

It then checks to see if the user wishes to only output linked genes, and removes the child if it violates the linkage rule.

The other squares were tested in a similar method, but monohybrid squares are significantly less complex than dihybrid squares and required fewer lines. Like calculating real-life punnett squares. Due to the simplicity of calculation, I didn't really encounter any errors.

End-user product documentation

Welcome to Simple Punnetts! The program has been designed with simplicity in mind, but here's some advice on getting started.

When you launch the program, you should see this screen:
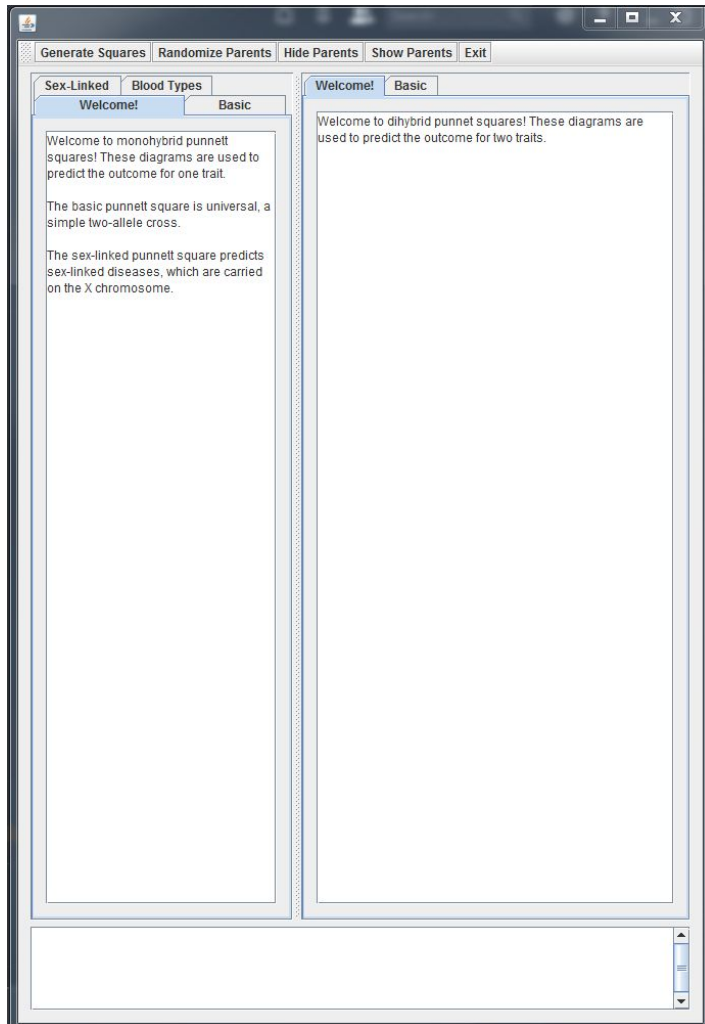
As you can see, there's a menu bar along the top that can be used to control a number of features within the app:

- **Generate Squares** is the primary one you'll be using, and clicking it causes the program to fill in all of the punnett squares according to the parents you've selected.
- **Randomize Parents** sets all of the parents in the program to a random value, which is especially useful if you just want to experiment with working backwards.
- **Hide Parents** can be used to hide all of the values for the parents. It's useful if you'd like to try and work backwards from a filled in punnett square.
- **Show Parents** is used to toggle the parents back to being visible after you hide them.
- **Exit** closes out of the program.

Now, there are also four different types of punnett squares this program is capable of computing; basic monohybrid punnett, basic dihybrid punnett, sex-linked punnett, and blood types punnett. If you have a specific task in mind, click on that tab. If you're just looking to experiment with the program, click on the "basic" tab on the left side.

That bring you to this image; you have the choice to either set the parents yourself, or use the randomize button at the top of the screen. Once you do, you can click "generate squares" and the boxes will be filled in.

We chose to have a heterozygous mother, which means her genes are Aa. The mother's genes are displayed across. The father is Homozygous dominant, which means his genes are AA. The four bolded squares within the center represent all possible children of the two parents; in this case, 50% of the children will be homozygous dominant and 50% will be heterozygous.

The sex-linked tab isn't elaborated upon here, because it functions identically to this tab, with the only exception being the letters used for the genes.

If you want to compute a dihybrid cross, it's slightly more complicated. As listed above, a dihybrid cross is between a mother and father who have two genes.

These genes can also be linked, which means they're only passed on together.

As with the monohybrid punnett square, you set the genes of the parents and then click generate square to get your results.

In this case, we crossed a mother who was homozygous dominant for gene one (AA) and heterozygous for gene two (Bb) with a father who had the same genes. The proportions of the offspring are displayed to the left.

We also have the option to select for only linked offspring, which would make our punnett square look slightly different.

In this case, we are selecting for homozygous linkage; only the alleles that are dominant (indicated by capital letters) are passed on together. This significantly reduces the number of possible offspring; now we have only one possible type.

The last tab calculates blood type between two parents. Blood type is slightly different from normal punnett squares, as it deals with multiple alleles. There are four different types of blood: AB, A, B, and O. However, there are six different genotypes: AA, AO, BB, BO, AB, and OO.

As we may not know whether our type A parent is homozygous (AA/BB) or heterozygous (AO/BO), this section of the app contains four different punnett squares.

The first square is assuming both the mother and father are homozygous.

The second treats the mother as hetero, and the father as homo.

The third does the reverse, treating the mother as hom and the father as hetero.

The fourth treats both parents as hetero.

This difference in homozygous vs heterozygous is only relevant for A and B blood types, as O and AB only have one possible genotype (OO and AB, respectively).

Self-evaluation & Future Prospects

The program, as it is right now, is unfortunately bland-looking. It also lacks the ability to let the student type in their own answers, and check them. Currently, it only gives the option to generate a variety of punnett squares. While that is valuable, it lacks a flexibility that is necessary for a classroom setting.

It could easily be distributed to teachers and schools as-is though, because running as a .jar means it is compatible with any machine than can run java.

Possible improvements:
- In the future, this project could be expanded to generate pedigree trees as well, but this would require a higher level of internal logic than is currently present in the program.
- The randomize, show/hide, and generate buttons should only affect the currently selected tab, to reduce memory usage.
- The show parents and hide parents should be a single radio button, not two separate buttons.
- When parents are hidden, the UI gets messed up and difficult to read. Ideally, the parents would simply hold a null value rather than the component being removed entirely.
- There's a lot of empty space at the bottom of the basic mono, sex linked, and basic di punnett squares that makes the program look awkward. Ideally I'd find something to add there, or redistribute the UI to remove the emp

References

Background

https://www.ndsu.edu/pubweb/~mcclean/plsc431/mendel/mendel1.htm

Punnett Squares

http://www.science.oregonstate.edu/genbio/otherresources/punnett%20squares.htm

Monohybrid Crosses

http://www.siskiyous.edu/class/bio1/genetics/monohybrid_v2.swf

Dihybrid Crosses

Unlinked/Independent

https://online.science.psu.edu/biol011_sandbox_7239/node/7336

Linked

http://ibbiologyhelp.com/Genetics/dihybridcrosslinkage.html

Student Copyright Checklist