Comparing the Genes for Eye Development Across Various Organisms

Sameer Saxena
Advisor: Kelley Bethoney, PhD.
Episcopal Academy
1785 Bishop White Drive, Newtown Square, PA 19073
PJAS Regional Competition: February 25, 2017

Introduction

Question: If I analyze and discover the DNA sequences of the genes for both human eye development and other organisms' eye developments, will I find repeated patterns and be able to predict abnormalities of the human eye based off of the data from other species?

Question for Drosophila Eye Color: Will finding similarities between various genes and proteins of mutated Drosophila eye colors give us further insight into how the proteins coded by these genes function and work together to create the organism's eye color?

Big Picture: As development of Drosophila relate to and are very similar to human eye development, the observations we find in this experiment could be applied to the human genome. Knowing minor changes in the gene sequences lead to major changes in the eye, this small scale project could lead to great information. Exploring further in eye development will yield information of eye development for other organisms as well.

Manipulated Variable: DNA sequence input into the program to compare to the WT, and then the DNA sequence of humans

Responding Variable: Percentage of similarity of DNA sequence compared to the baseline

Control: The baseline DNA sequence was that of the Wild-Type Drosophila Melanogaster

Constants: Method of comparing both the protein sequences and the gene sequences as to ensure percentages are comparable

Question: If I analyze and discover the DNA sequences of the genes for both human eye development and other organisms' eye developments, will I find repeated patterns and be able to predict abnormalities of the human eye based off of the data from other species?

Big Picture: As development of Drosophila relate to and are very similar to human eye development, the observations we find in this experiment could be applied to the human genome. Knowing minor changes in the gene sequences lead to major changes in the eye, this small scale project could lead to great information. Exploring further in eye development will yield information of eye development for other organisms as well.

Investigating Genes for Eye Color within *Drosophila melanogaster*

Question: Will finding similarities between various genes and proteins of mutated Drosophila eye colors give us further insight into how the proteins coded by these genes function and work together to create the organism's eye color?

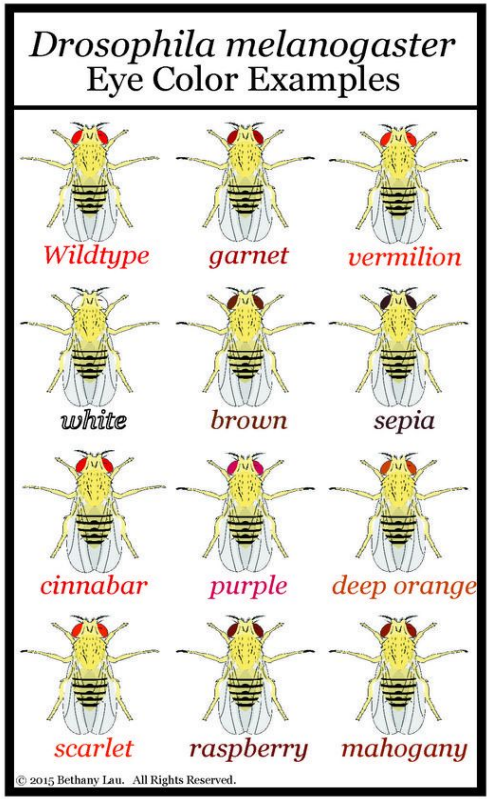
Future: Can we then use this information to deduce where the similar genes are in other organisms? By comparing the similar genes between Drosophila and another organism (i.e. humans) we could gain insight on abnormalities, mutations, and variations of eye color development in both organisms.

The Wild-Type Exhibits a Red Eye Color

- Red eye color: the wild type “normal”
  - Genes transport enough red and brown pigment for the eye
  - No mutations are coded to mess with the transport and activation of pigments to the eye

An example of a normal (WT) Drosophila melanogaster

Mutations in the *Drosophila melanogaster*



Illustrates the several types of mutations regarding eye color in the Drosophila melanogaster

How the Mutations Can Affect the Process

- Color of eye is sum of all pigments built and transported
- Many steps in this process:
  - Creating precursors
  - Transporting precursors
  - Activating precursors into pigment
  - Transporting pigment to eye
- Upcoming mutations affect different parts of these processes
- The name of each gene is the color of the eye with the presence of a mutation

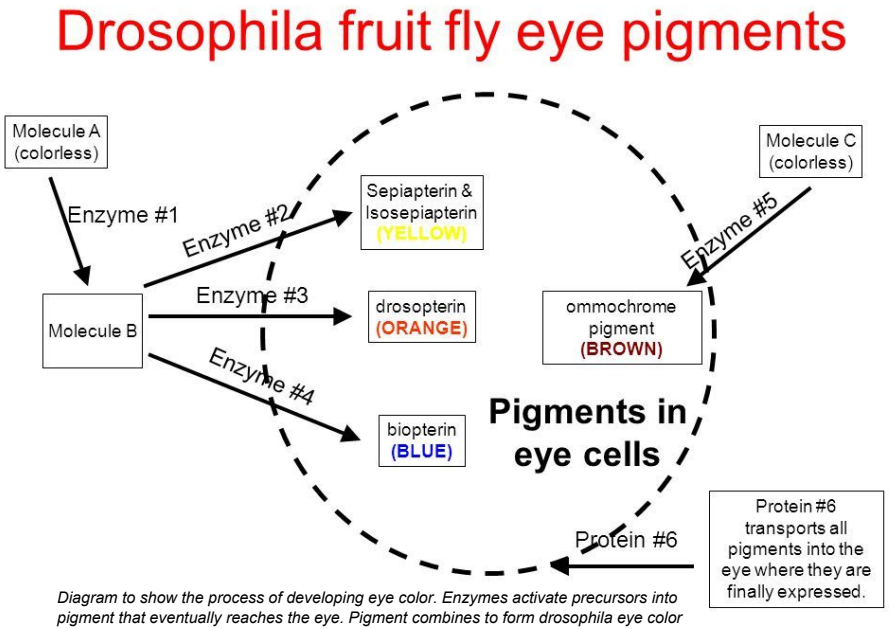


Diagram to show the process of developing eye color. Enzymes activate precursors into pigment that eventually reaches the eye. Pigment combines to form drosophila eye color

Background Information

- Red eye color: the wild type “normal”
  - Genes transport enough red and brown pigment for the eye
  - No mutations are coded to mess with the transport and activation of pigments to the eye

- Color of eye is sum of all pigments built and transported
- Many steps in this process:
  - Creating precursors
  - Transporting precursors
  - Activating precursors into pigment
  - Transporting pigment to eye
- Upcoming mutations affect different parts of these processes
- The name of each gene is the color of the eye with the presence of a mutation

An example of a normal (WT) Drosophila melanogaster

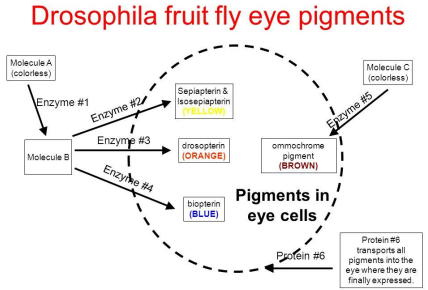
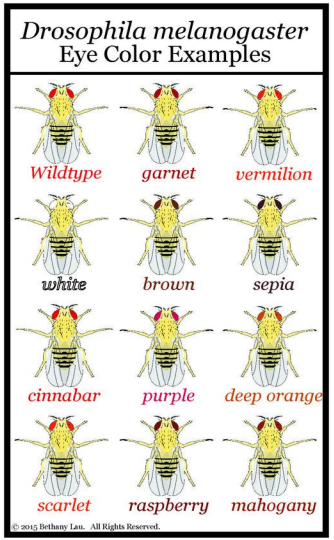


Diagram to show the process of developing eye color. Enzymes activate precursors into pigment that eventually reaches the eye. Pigment combines to form drosophila eye color



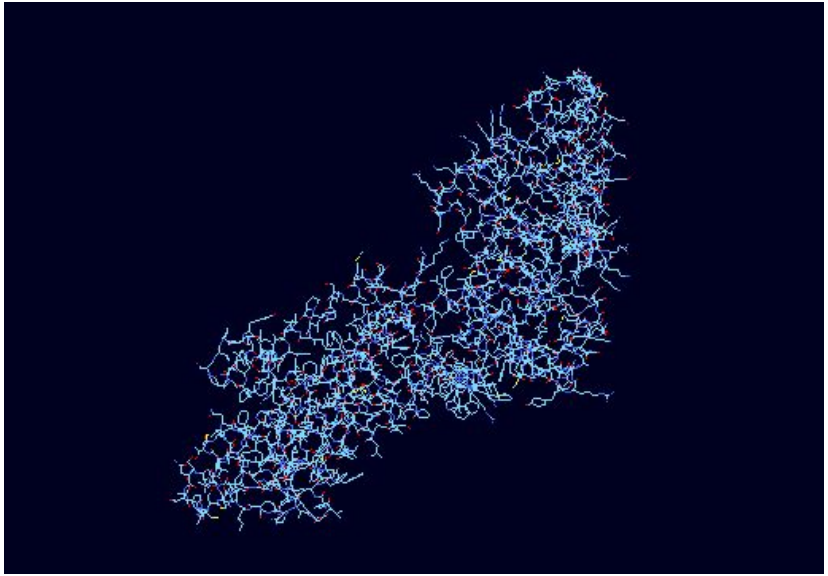
Illustrates the several types of mutations regarding eye color in the Drosophila melanogaster

Comparing Genes of Different Mutations and Observing Data

White Gene Allows Red Pigment into the Eye

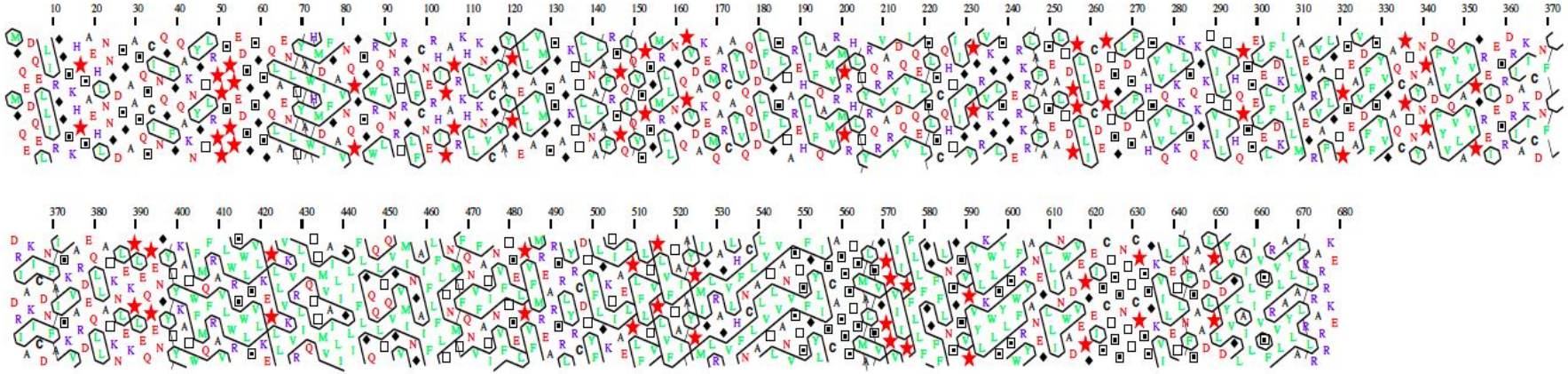
- Malfunctioning/mutated white gene:
  - Will block red and brown pigment resulting in completely white eyes
- Protein Sequence:
  - 39% similar to the scarlet gene
  - 26% similar to the brown gene
- Gene Sequence:

Brown	**28.49%**
Scarlet	**25.82%**
Purple	6.69%
Cinnabar	18.70%
Garnet	**22.7%**
Vermillion	15.46%
Sepia	9.84%
Mahogany	15.84%



Swiss PDB Viewer showing the White Gene in a 3D view

White Protein Clusters - Secondary Structure



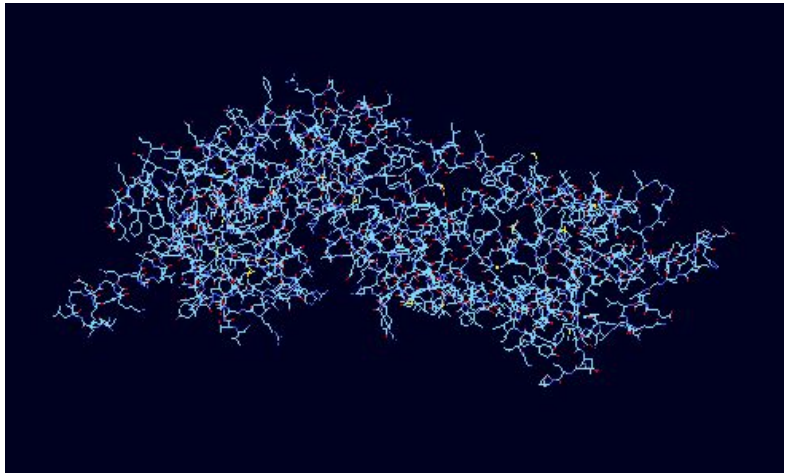
HCA (Hydrophobic Cluster Analysis) job which shows where the protein coded by the white gene combines and clusters

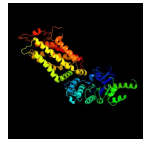
These figures are useful to further compare these proteins at different levels. The sequences are the proteins' primary structures, and the clusters are the proteins' secondary structures.

Brown Gene Allows for Red Pigment As Well

- A malfunction with this gene leads to brown eyes just as a malfunction in white gene leads to white eyes
  - Mutations will lead to blocked red pigment (drosopterin) and leave ommochromes in the eye
- 26% protein sequence to scarlet gene
- Gene sequences:

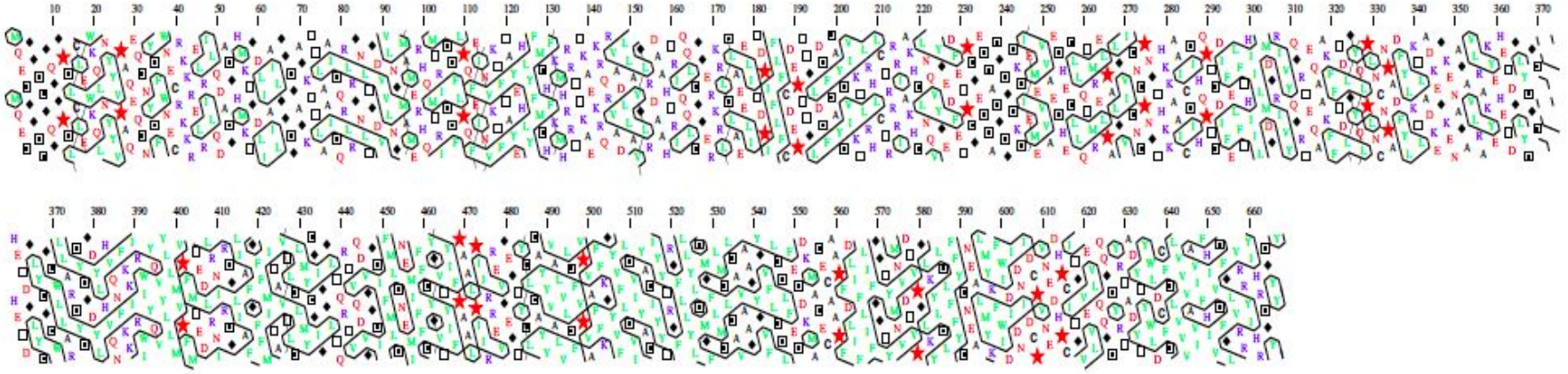
White	**28.49%**
Scarlet	**26.23%**
Purple	7.45%
Cinnabar	17.21%
Garnet	**21.54%**
Vermillion	15.63%
Sepia	8.48%
Mahogany	17.16%





3D model of the protein generated by Phyre 2. Model is observed and analyzed using Swiss-PDBViewer

Brown Protein Clusters - Secondary Structure

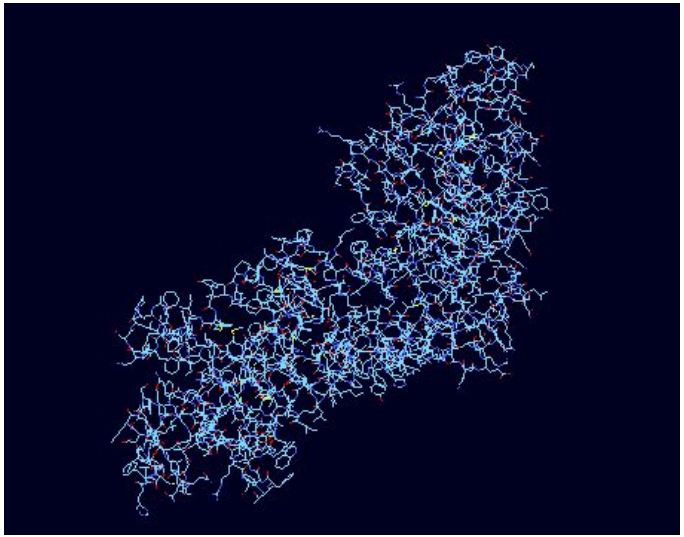


HCA (Hydrophobic Cluster Analysis) job which shows where Brown Protein combines and clusters

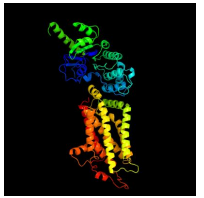
Scarlet Gene Allows for Brown Pigment in the Eye

- Elimination of it/most mutations leading to dysfunctionality mean scarlet eyes
  - Opposite of brown gene - mutations block ommochromes and therefore leave drosopterin in eye (bright red)
- Gene sequences:

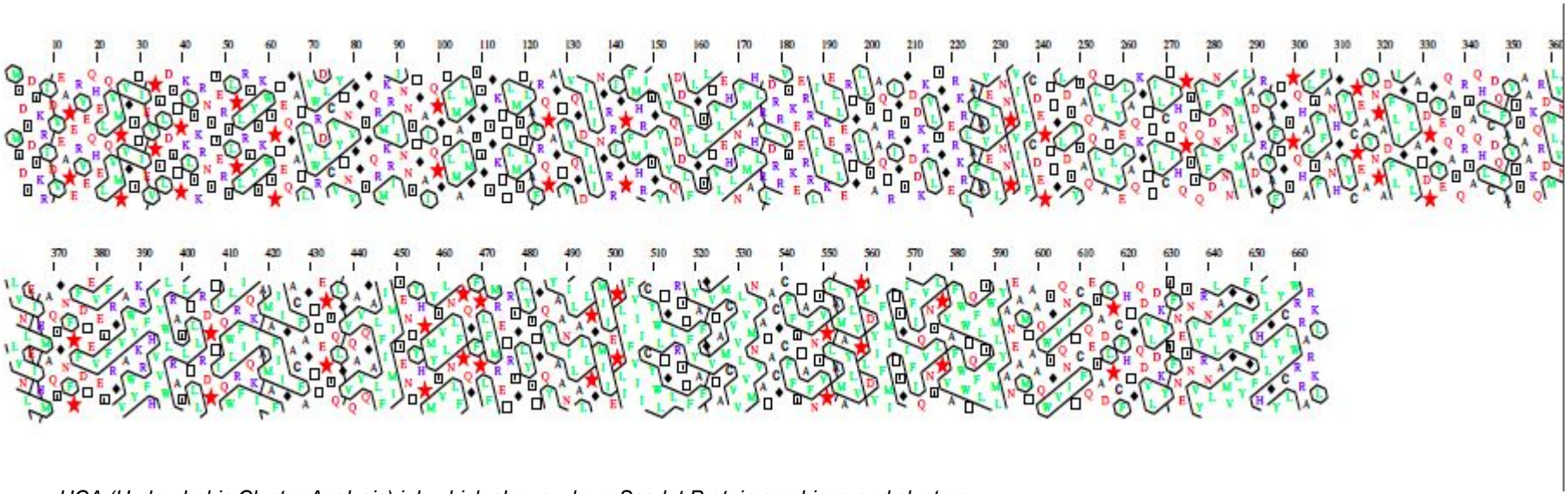
White	**25.82%**
Brown	**26.23%**
Purple	7.00%
Cinnabar	19.09%
Garnet	**20.84%**
Vermillion	14.84%
Sepia	9.90%
Mahogany	17.59%



3D model of the protein generated by Phyre 2. Model is observed and analyzed using Swiss-PDBViewer



Scarlet Protein Clusters - Secondary Structure



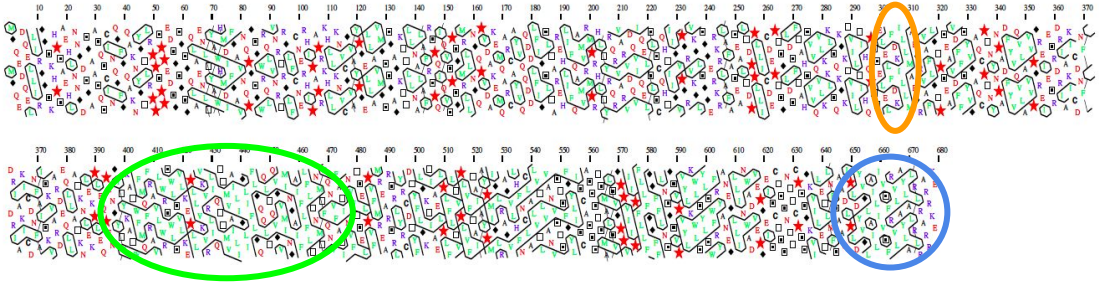
HCA (Hydrophobic Cluster Analysis) job which shows where Scarlet Protein combines and clusters

Scarlet, Brown, White Genes All Had Many Similarities

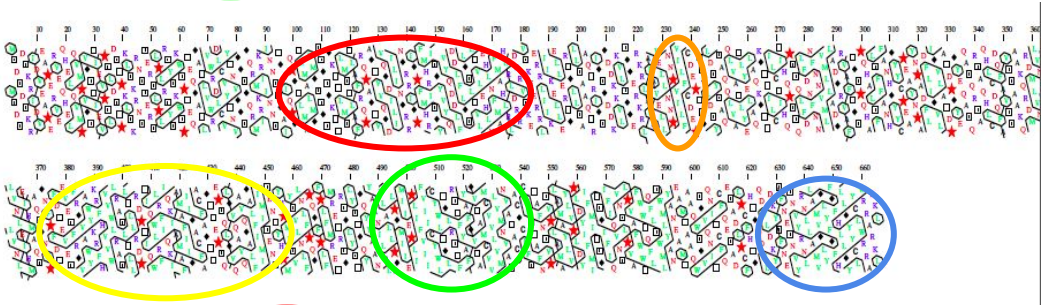
- Protein and gene sequences found many high percentages of similarity
- Protein sequences:
  - White 39% similar to the scarlet gene
  - White 26% similar to the brown gene
  - Brown 26% similar to the scarlet gene
- Gene sequences:
  - scarlet - brown: 26.23%
  - scarlet - white: 25.82%
  - brown - white: 28.49%
- Clusters had formed in similar places on each of their secondary structures
- 3D models seemed very similar and shape of them seemed to be the same
- **Big Picture:** Grouping these together as genes coding for transport proteins taking pigments to the eye - we now know the functions and similarities of these “Big 3” genes

Scarlet, Brown, and White Genes are Very Similar

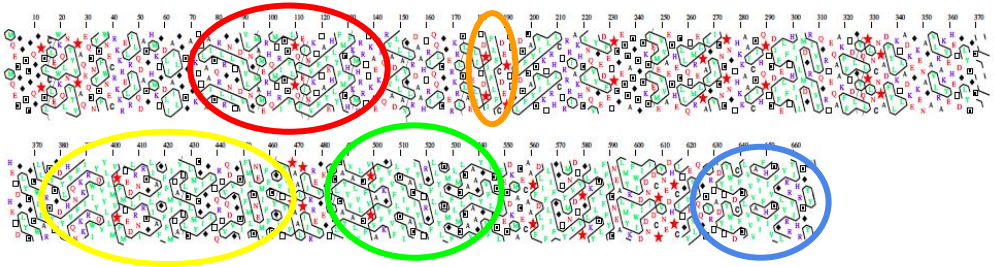
White:

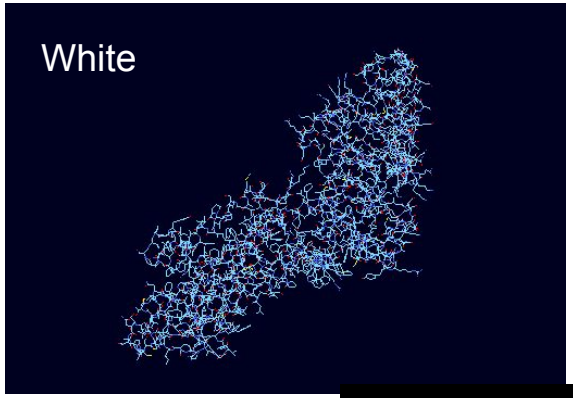


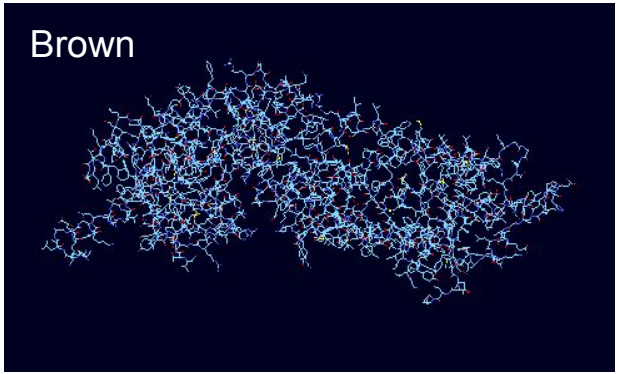
Scarlet:

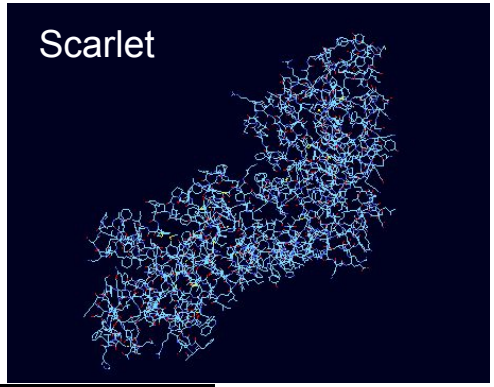


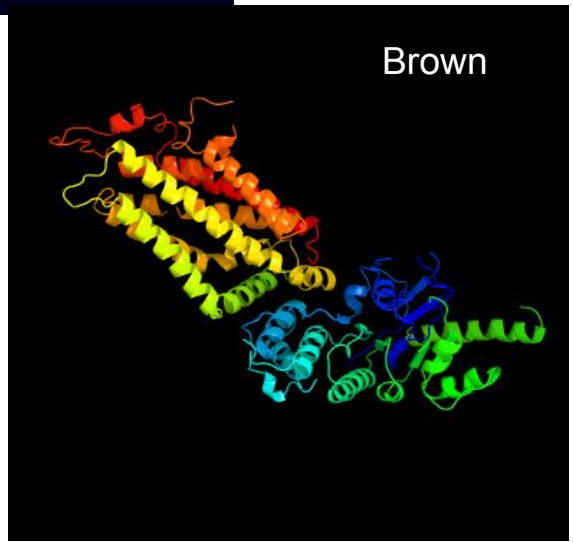
Brown:

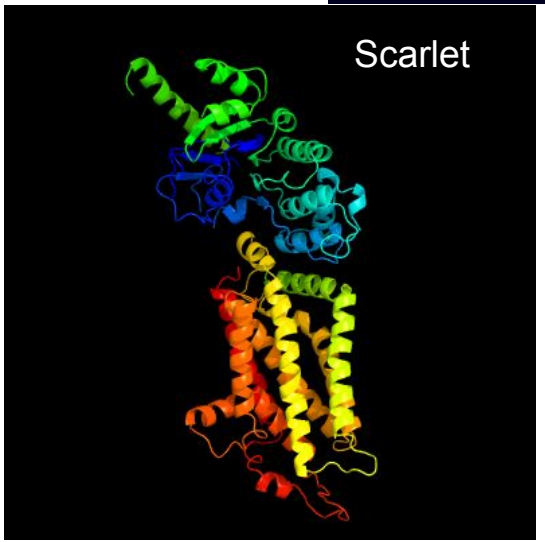



White


Brown


Scarlet

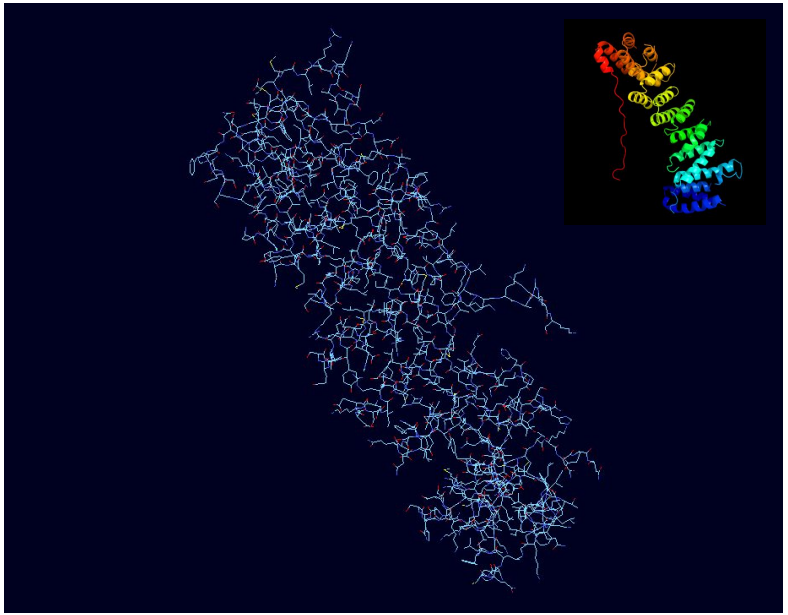

Brown


Scarlet

Garnet Gene Had Nothing Significant in Protein Sequences but Was Similar to the “Big Three”

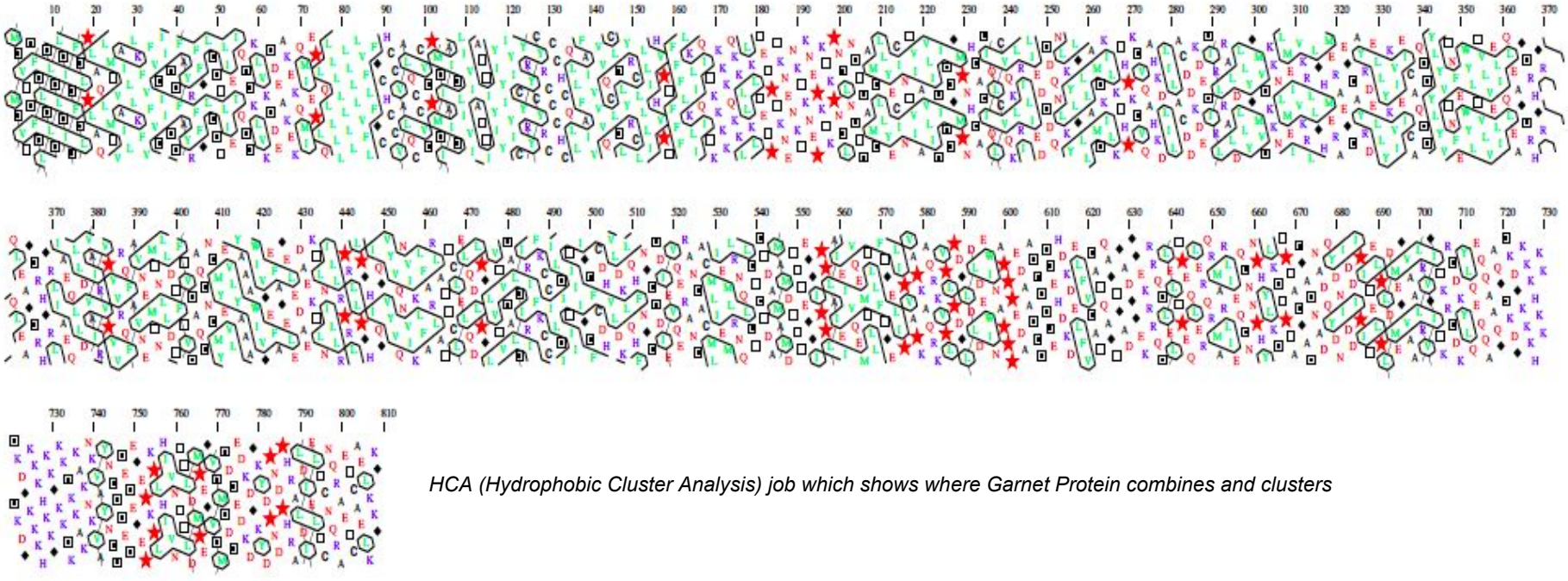
- Gene sequence:
  - 99% similar to chromosome X on *Drosophila melanogaster*

White	**22.77%**
Brown	**21.54%**
Scarlet	**20.84%**
Purple	5.01%
Cinnabar	14.22%
Vermillion	12.49%
Sepia	7.44%
Mahogany	13.11%



3D model of the protein generated by Phyre 2. Model is observed and analyzed using Swiss-PDBViewer

Garnet Protein Clusters - Secondary Structure



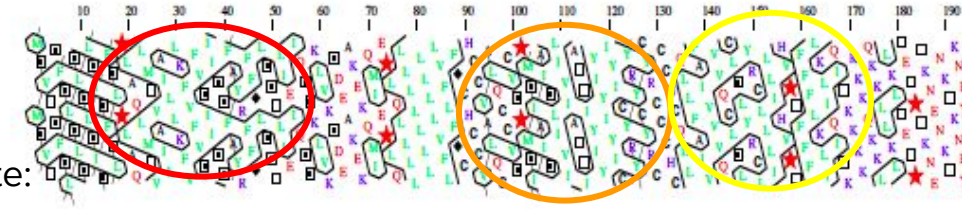
HCA (Hydrophobic Cluster Analysis) job which shows where Garnet Protein combines and clusters

Garnet Gene Was Incredibly Similar to Big Three

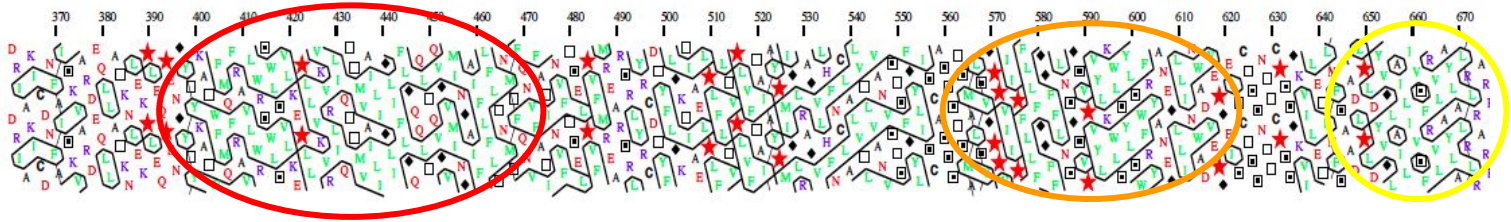
- Gene sequence:
  - 99% similar to chromosome X on *Drosophila melanogaster* - Where white gene is located and most likely where it is located
  - **garnet - scarlet: 20.84%**
  - **garnet - brown: 21.54%**
  - **garnet - white: 22.77%**

- **Big Picture:** Garnet gene is heavily related to transport, as it is very similar to each of the other three genes which are as well
  - Found Garnet gene codes for coat protein which pertains to intracellular vesicle transport

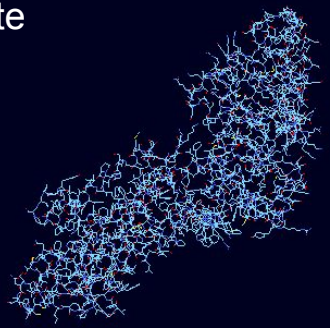
White:



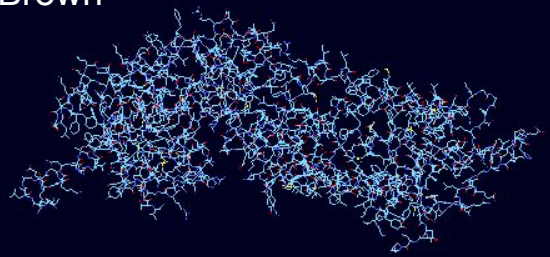
Garnet:



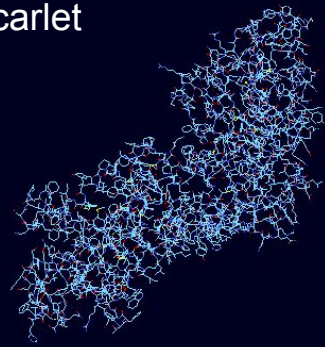
White



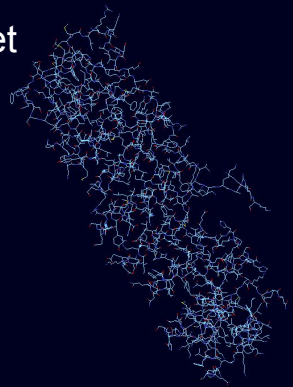
Brown



Scarlet



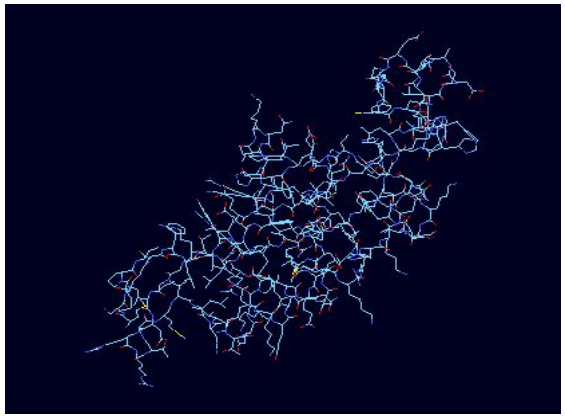
Garnet



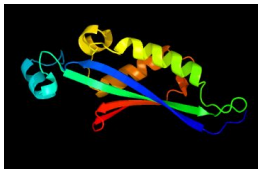
Purple Gene Had No Similarities to Any Other Genes

- 99% protein sequence similarity to this protein: **uncharacterized protein Dyak_GE12609 [Drosophila yakuba]**
- 97% protein sequence similarity to this protein: **PREDICTED: 6-pyruvoyl tetrahydrobiopterin synthase [Drosophila elegans]**
- Mutations such as purple block certain steps of the path to activate and transport pigment
- Genetic sequences:

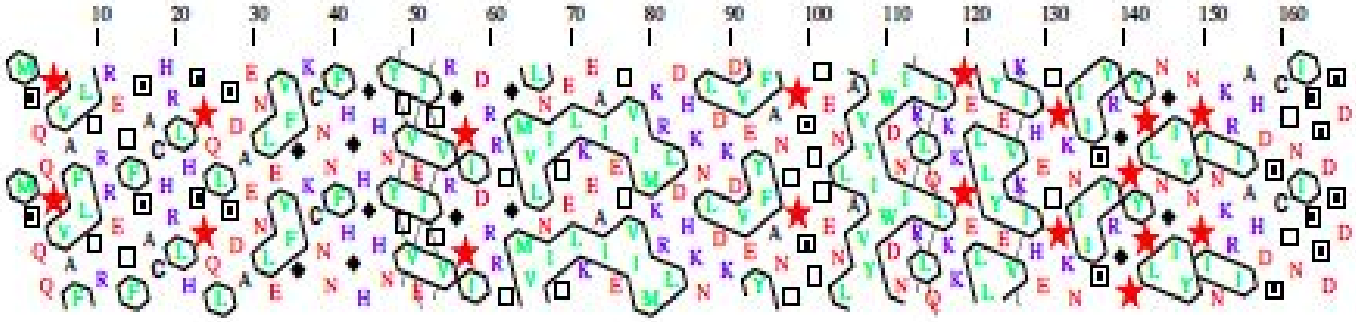
White	6.69%
Brown	7.46%
Scarlet	7.00%
Cinnabar	8.71%
Garnet	5.01%
Vermillion	12.98%
Sepia	**19.67%**
Mahogany	11.47%



3D model of the protein generated by Phyre 2. Model is observed and analyzed using Swiss-PDBViewer



Purple Protein Clusters - Secondary Structure

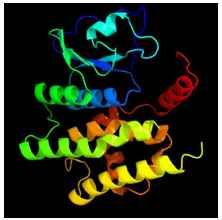


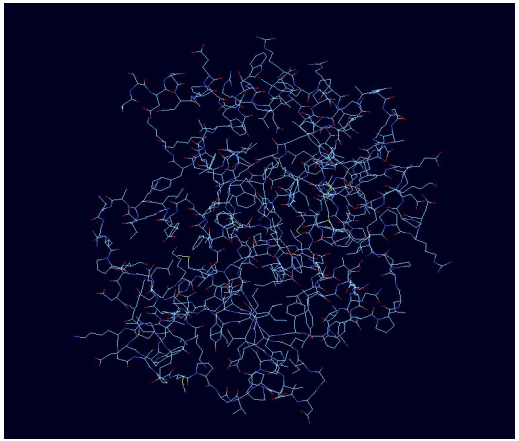
HCA (Hydrophobic Cluster Analysis) job which shows where Purple Protein combines and clusters

Sepia Gene Found Many Similarities to Uncharacterized Proteins

- 93% - 95% similar to pyrimidodiazepine synthase for Drosophila melanogaster
- 96% similar to uncharacterized protein Dere_GG14321 [Drosophila erecta]
- 96% similar to uncharacterized protein Dyak_GE20749 [Drosophila yakuba]
- 98% similar to uncharacterized protein Dsimw501_GD14100 [Drosophila simulans]
- Gene sequence:

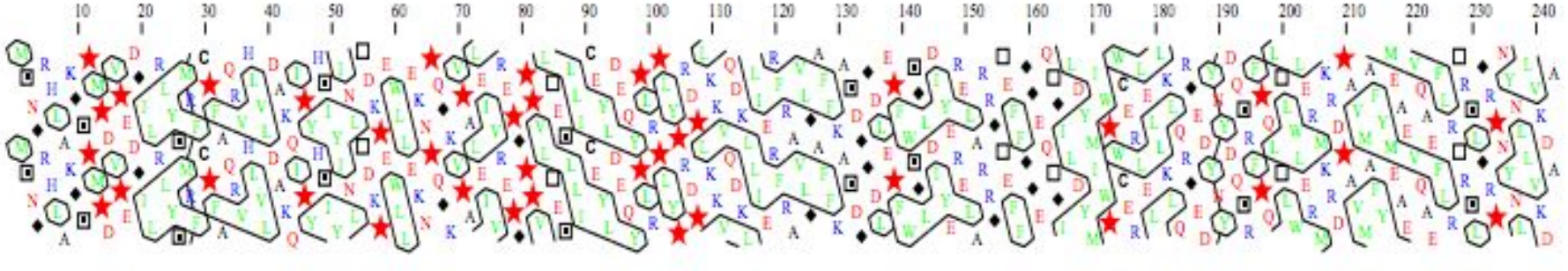
White	9.84%
Brown	8.48%
Scarlet	9,90%
Purple	**19.67%**
Cinnabar	13.47%
Garnet	7.44%
Vermillion	16.40%
Mahogany	16.33%





3D model of the protein generated by Phyre 2. Model is observed and analyzed using Swiss-PDBViewer

Sepia Protein Clusters - Secondary Structures

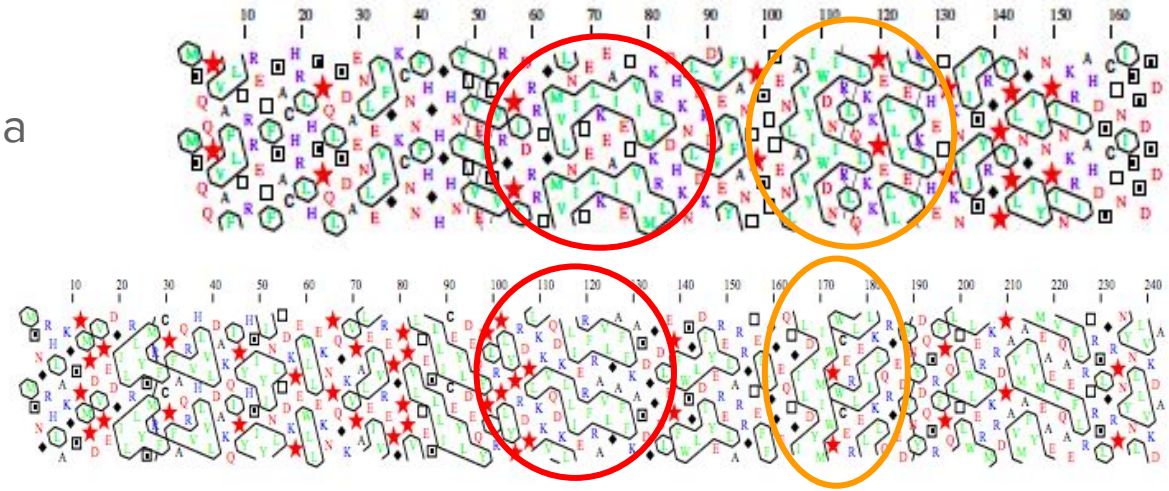


HCA (Hydrophobic Cluster Analysis) job which shows where Sepia Protein combines and clusters

Purple and Sepia Genes Had Very Little Similarities to Other Genes

- Gene sequences: Purple
  - purple - mahogany: 11.47%
  - **purple - sepia: 19.67%**
  - purple - vermillion: 12.98%
  - purple - garnet: 5.01%
  - purple - cinnabar: 8.71%
  - purple - scarlet: 7.00%
  - purple - brown: 7.46%
  - purple - white: 6.69%
- Gene sequences: Sepia
  - sepia - mahogany: 16.33%
  - sepia - vermillion: 16.40%
  - sepia - garnet: 7.44%
  - sepia - cinnabar: 13.47%
  - **sepia - purple: 19.67%**
  - sepia - scarlet: 9.90%
  - sepia - brown: 8.48%
  - sepia - white: 9.84%

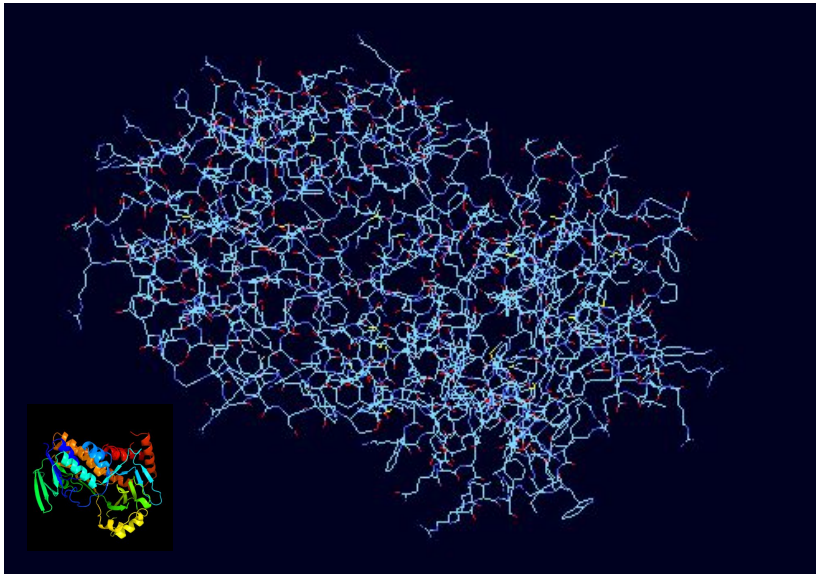
- **Big Picture:** Both coded for a synthase enzyme which aided the process of transporting and activating pigments
  - *purple* coded for PTP synthase
  - *sepia* coded for PDA synthase



Cinnabar Gene Had Very Few Similarities

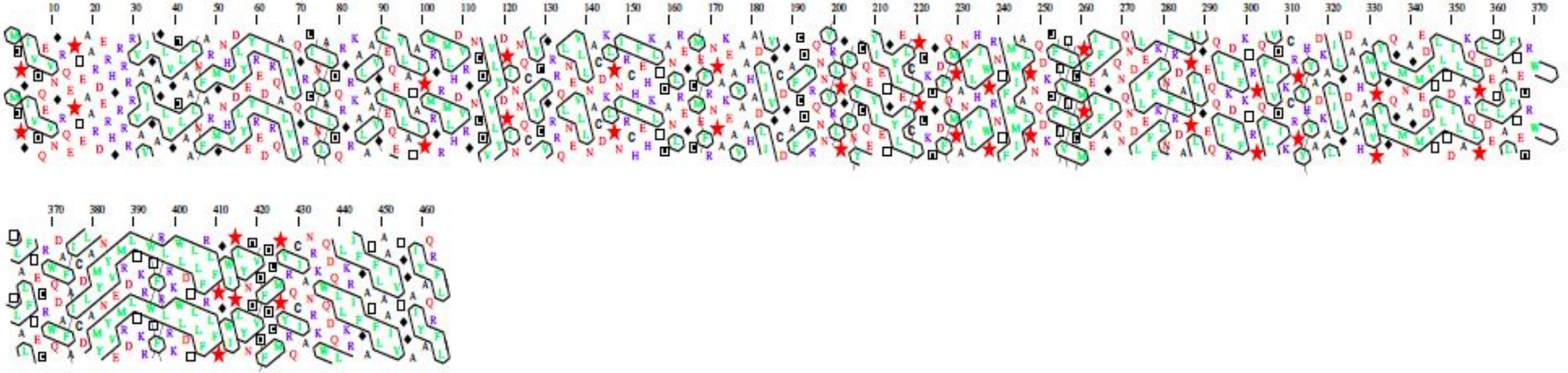
- 99% to **kynurenine 3-monooxygenase (Drosophila melanogaster)** but also similar to this gene in many other different organisms
- Gene sequences

White	18.70%
Brown	17.21%
Scarlet	19.09%
Purple	8.71%
Sepia	13.47%
Garnet	14.22%
Vermillion	**23.21%**
Mahogany	**22.02%**



3D model of the protein generated by Phyre 2. Model is observed and analyzed using Swiss-PDBViewer

Cinnabar Protein Clusters - Secondary Structure

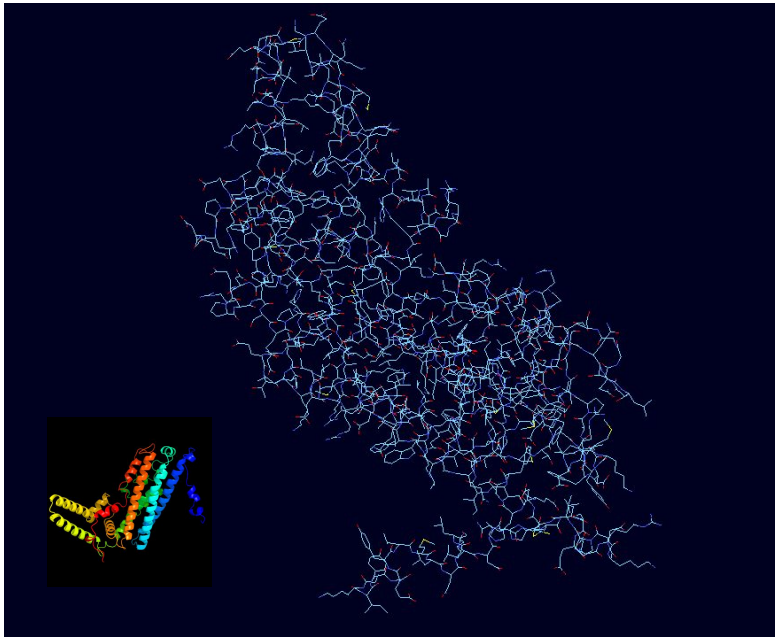


HCA (Hydrophobic Cluster Analysis) job which shows where Cinnabar Protein combines and clusters

Vermillion Gene Had Very High Similarity to Mahogany

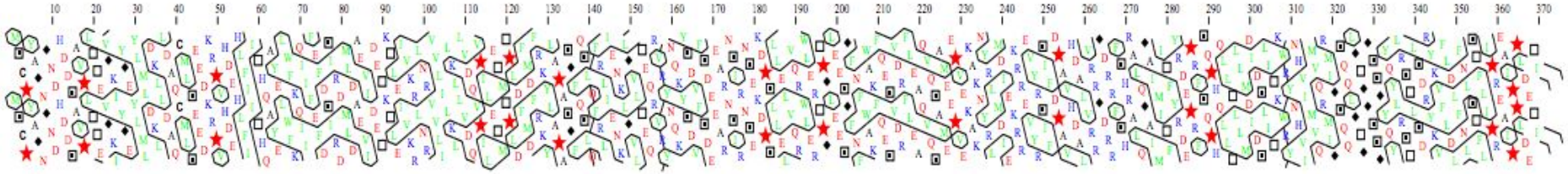
- 98% - 99% similar to tryptophan 2,3 dioxygenase of other Drosophila
- Gene Sequence:

White	15.46%
Brown	15.63%
Scarlet	14.84%
Purple	12.98%
Sepia	16.40%
Garnet	12.49%
Cinnabar	**23.21%**
Mahogany	**28.81%**



3D model of the protein generated by Phyre 2. Model is observed and analyzed using Swiss-PDBViewer

Vermilion Protein Clusters - Secondary Structure

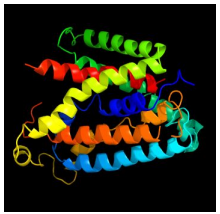


HCA (Hydrophobic Cluster Analysis) job which shows where Vermillion Protein combines and clusters

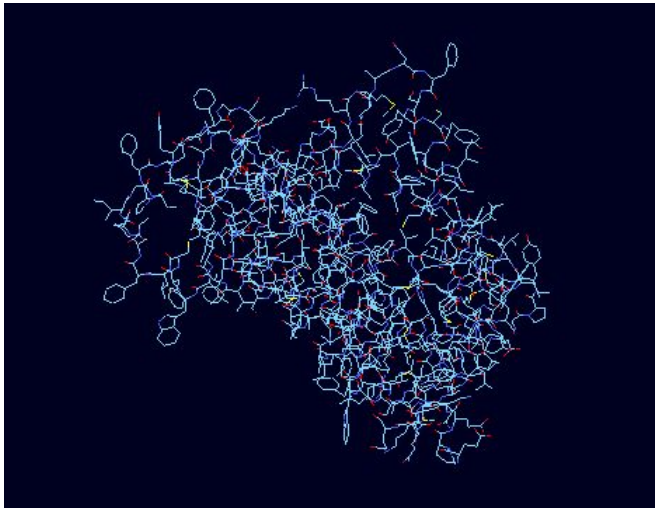
Mahogany Gene Was Very Similar to Vermillion and Cinnabar

- 99% similar to uncharacterized protein Dere_GG11342 [Drosophila erecta]
- 100% similar to uncharacterized protein Dmel_CG13646 [Drosophila melanogaster]
- 99% similar to uncharacterized protein Dsimw501_GD21153 [Drosophila simulans]
- 99% similar to uncharacterized protein Dyak_GE23537 [Drosophila yakuba]
- Gene Sequences:

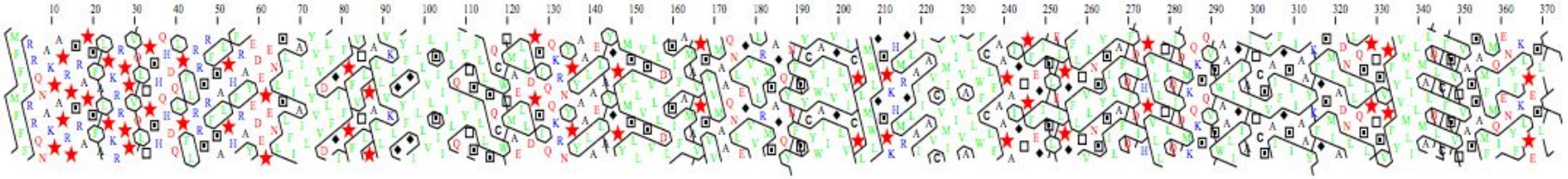
White	15.48%
Brown	17.16%
Scarlet	17.59%
Purple	11.47%
Sepia	16.33%
Garnet	13.11%
Cinnabar	**22.02%**
Vermillion	**28.81%**

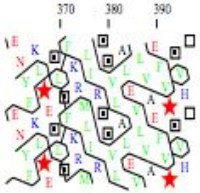


3D model of the protein generated by Phyre 2. Model is observed and analyzed using Swiss-PDBViewer



Mahogany Protein Clusters - Secondary Structures

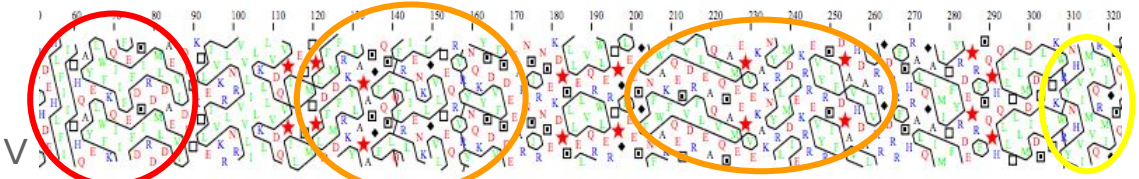


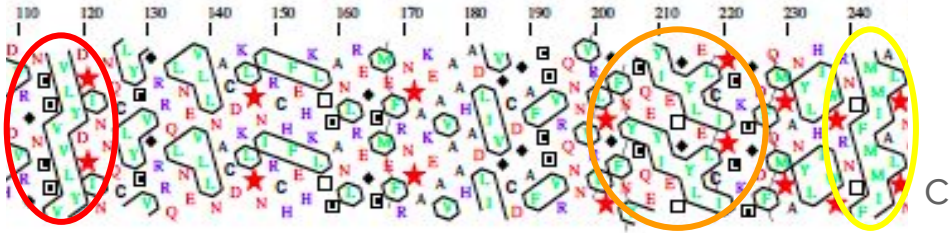


HCA (Hydrophobic Cluster Analysis) job which shows where Mahogany Protein combines and clusters

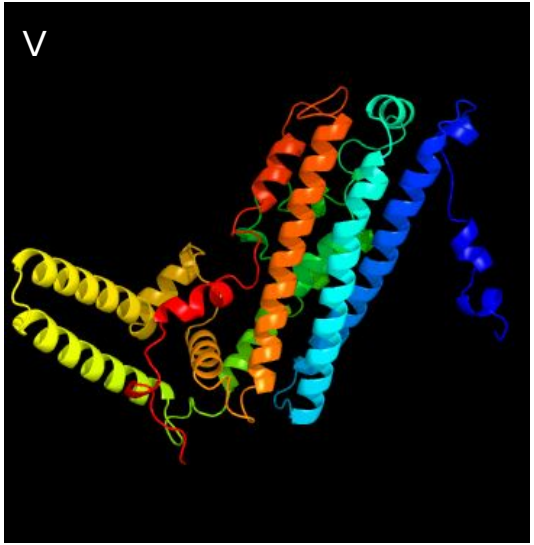
Vermillion, Mahogany, and Cinnabar Genes All Had Many Similarities

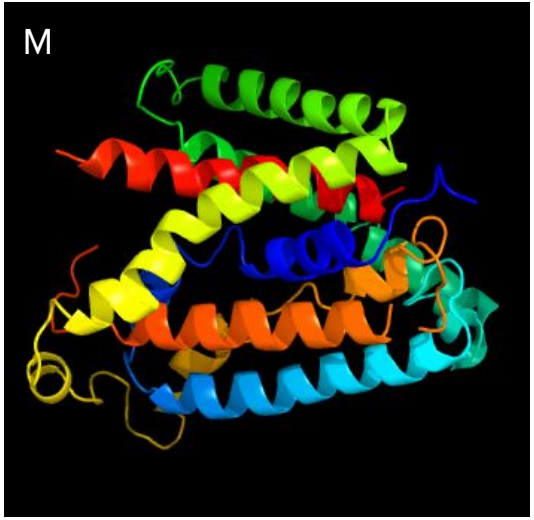
- Primary Structures:
  - mahogany - vermillion: 28.81%
  - mahogany - cinnabar: 22.02%
  - vermillion - cinnabar: 23.21%
- Secondary Structures:

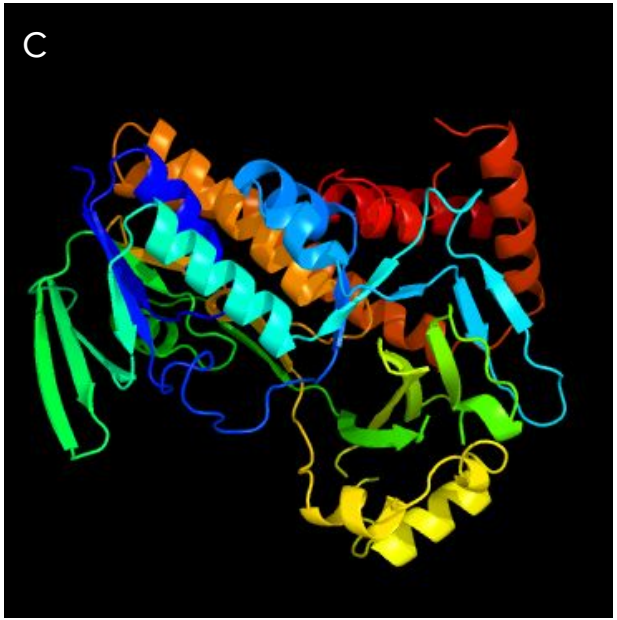




- **Big Picture:** Vermillion, Cinnabar, Mahogany genes must have similar functions
- Further research gave more insight
  - Vermillion: tryptophan dioxygenase
  - Cinnabar: kynurenine monooxygenase
    - Mahogany: transport protein (arginine antiporter)


V


M


C

Overall Conclusions

Overall Conclusions

- **White, Scarlet, Brown** genes all grouped together as coding for main transport proteins for pigments
  - Important genes when regarding color of the eye
  - Main contributors to how eye receives red pigment (drosopterin) and brown pigment (ommochromes)
- **Garnet** also a transport protein and was similar to White, Brown, Scarlet genes
  - Codes for intracellular vesicular transport
- **Purple** and **Sepia** genes were far off in terms of similarity but were grouped together
  - Both coded for synthase enzymes which contributed in process (PTA and PDA)
- **Vermillion**, **Mahogany**, and **Cinnabar** had similarities and were grouped together
  - All dealt with different types of amino acid transport or generation (precursors to pigments)
  - Vermillion - tryptophan
  - Cinnabar - kynurenine
  - Mahogany - arginine

How Will The Conclusions Help?

- Looking at structure and therefore function of the proteins
- Find those same proteins in the humans and other organisms
- Observations found through comparing various genes of the Drosophila could lead to information about these similar proteins in a different organism
- With these comparisons -
    - Could we predict mutations of the human genome?
    - Could we discover information helping to understand human eye development?
    - Could we observe genomic sequences that may lead to different eye colors?
    - Could we find new mutations and new variations of genes across organisms?
    - Will we find that organisms' eye development are similar enough to experience the same abnormalities and diseases?
- **Knowing how each of these genes function and grouping them together to simplify similar structures and sequences allows us to easily find the same genes in other organisms in the next step of the project**

Further Research

- Knowing certain mutations in the eye color of Drosophila it would be useful to find these mutations in other organisms
- Humans will have many similar genes which code for eye color - continuation of project would open up opportunities to research there and compare those to Drosophila
- Using further information after comparing organisms - **will it be possible to predict mutations if sequences are similar enough?**
- Later on could use lab for gene-editing: Manipulate a gene, hypothesize, and discover what the manipulation affected

Acknowledgements

PJAS

PJAS Judges

PJAS Audience

Dominguez Lab

The Episcopal Academy

Kelley Bethoney, Ph.D.

EA Chapter of PJAS

Matthew Memmo

Christie Rheam

Citations

- Phyre 2: http://www.sbg.bio.ic.ac.uk/phyre2/html/page.cgi?id=index
- NCBI: https://www.ncbi.nlm.nih.gov/
- PDB: http://www.rcsb.org/pdb/home/home.do
- HCA: http://mobyle.rpbs.univ-paris-diderot.fr/cgi-bin/portal.py?form=HCA#forms::HCA
- http://insects.eugenes.org/species/about/species-gallery/Drosophila_melanogaster/
- https://en.wikipedia.org/wiki/ATP-binding_cassette_transporter#/media/File:1l7v_opm.png
- https://i1.wp.com/www.jargonwall.com/wp-content/uploads/2014/08/pax6.jpg
- http://www.yourgenome.org/sites/default/files/styles/banner/public/banners/stories/fruit-flies-in-the-laboratory/single-fruit-fly-drosophila-melanogaster-on-white-background-cropped.jpg?itok=3pTTh8c5
- http://insects.eugenes.org/species/about/species-gallery/Drosophila_melanogaster/Drosophila_melanogaster.jpg
- http://www.scienceandmathwithmrslau.com/wp-content/uploads/2015/03/fruit-fly-graphic1-622x1024.jpg
- http://slideplayer.com/slide/9022835/

Methods of Comparing the Sequences

Comparing the Genes Involved Many Steps

- Wanted to analyze the gene sequences and the proteins which the genes code for
- Proteins have a **primary, secondary, tertiary, and quaternary structure**
  - Primary structure: the sequence (MQERKANR...)
  - Secondary structure: how the amino acids in the sequence fold and cluster together
  - Tertiary structure: three-dimensional shape of the protein after it further folds in on itself
- Each of the structures **HEAVILY** affect the functions of the protein
- Therefore - must analyze all the way from primary to tertiary structure of each protein
- Gives me understanding on how they function - the overarching goal

Method of Retrieving and Analyzing the Data

- Gathered protein and gene sequences from NCBI
- Protein Sequences: Ran an NCBI blast
  - Found any similarity the protein sequence may have to any other proteins in the data bank
  - Highlights where the similarities and differences were between the baseline and MV
- Gene Sequences: Wrote my own program using Python and Javascript
  - NCBI's method was inefficient and inconclusive
  - Wrote algorithm to compare two sequences and return percentage (Python)
  - Accompanied by visual depiction of where the sequences were similar and where they were not (Javascript)

Methods of Comparing the Sequences

- Proteins have a **primary, secondary, tertiary, and quaternary structure**
  - Primary structure: the sequence (MQERKANR...)
  - Secondary structure: how the amino acids in the sequence fold and cluster together
  - Tertiary structure: three-dimensional shape of the protein after it further folds in on itself
- Analyzed all the way from primary to tertiary structure of each protein to give me an understanding on how they function - the overarching goal
- Gathered protein and gene sequences from NCBI
- Protein Sequences: Ran an NCBI blast
  - Found any similarity the protein sequence may have to any other proteins in the data bank
  - Highlights where the similarities and differences were between the baseline and MV
- Gene Sequences: Wrote my own program using Python and Javascript
  - NCBI's method was inefficient and inconclusive
  - Wrote algorithm to compare two sequences and return percentage (Python)
  - Accompanied by visual depiction of where the sequences were similar and where they were not (Javascript)

```
white = "ATGGGCCAAGAGGATCAGGAGCTATTAATTCGCGGAGGCAGCAAACACCCATCTGCCGAGCATCTGAACAATGGTGACAGCGGAGCGGCTTCGCAGAGCTGCATTAACCAGGGCTTCGGGCAGGCCAAAAACTACGGCACGCTCCGGCCACCCAGTCCGCCGGAGGACTCCGGTTCAGGGAGCGGCCAACTAC
brown = "ATGCAAGAATCCGGCGGCTCGTCGGGCCAGGGCGGTCCATCGCTGTGCCTCGAGTGGAAGCAGCTGAACTACTATGTGCCCGCCCAGGAGCAGAGCAACTACAGTTTCTGGAATGAGTGCCGCAAGAAGCGCGAGCTGCGGATCCTCCAAGACGCCAGCGGGCACATGAAGACCGGCGACCTCATCGCCATAC
scarlet = "ATGTCGGATTCAGATAGCAAGCGGATCGATGTGGAGGCCCCGGAGCGAGTGGAGCAGCACGAATTGCAGGTGATGCCGGTGGGCAGCACCATTGAGGTGCCCAGCTTGGACAGCACACCCAAGCTATCGAAACGGAACAGTTCGGAGAGAAGTCTACCGCTCAGGAGCTACAGCAAATGGTCGCCCACGGA
purple = "ATGTCGCAGCAACCTGTTGCCTTTTTAACCCGTCGCGAAACTTTCAGCGCCTGCCATCGTCTCCATAGTCCCCAATTGAGCGACGCCGAGAATCTGGAAGTCTTCGGCAAGTGCAACAATTTCCACGGCCACGGACACAACTATACAGTTGAGATAACCGTCCGTGGTCCCATCGATCGGCGAACCGGAATC
cinnabar = "ATGAGCCCAGGAATCGTTAGCCAGGAGGTAAACGGCCGCCAGGAGCCAACAGAAGCTGCGAGGGACGAGAGACATGGACGACGGCGCCGGGTGGCGGTTATTGGAGCAGGACTTGTGGGTTCTCTGGCAGCCTTGAACTTTGCCCGCATGGGCAACCACGTGGATTTGTACGAGTACCGGGAGGATATCC
garnet = "ATGTCCGTTACCTCTTTCTCTCTCTCTCTCTCTCTGTCTATCTCTTTCTCTCTCCCACTCGCTCAACTTATGACTGTCGCCATACTGCTTAAGTTTGTATTTGTTTCTATAATTTCATCGTTTGCCTATCGATTTTTTGGCTTATCGAGCACACTCATTGAGATCCAAAGTTCTAAACTAGTCAAATCGGAT
vermillion = "ATGAGCTGTCCCTATGCAGGAAACGGAAACGATCACGATGATTCGGCGGTGCCATTAACCACGGAAGTGGGCAAAATCTATGGAGAGTATCTGATGCTGGACAAACTGCTGGATGCCCAGTGTATGCTCTCCGAGGAGGACAAGCGACCCGTGCACGATGAGCATCTGTTCATCATCACGCACCAGGC
sepia = "ATGAGTAACGGCAGGCATTTGGCAAAAGGCTCACCCATGCCGGATGTTCCCGAAGATGGTATCCTTCGCCTGTACTCGATGCGCTTCTGCCCATTCTGGTCCTGGACGCCAAGCAGATCCCGTATCACAGCATCTACATTAATCTCACAGACAAGCCGGAGTGGCTGCTGGAGAAGA
mahogany = "ATGTTCTTCTTCCGTCGCCAGAATGCTAAACCCGCGCCACGCCCTTCGCGCGCTCCGTCCTTTGCGCTTCCAAAGCGTCGGCCTCCTGTCCAGCTAACGCCCTCGCATCAGCACGACCAGTTGCCCTTCCTGCGAAGATCGCGGTTCAGCGCCCTTCCCCATTTCCGCGCCTACGAGGACGAACCGGAGA

nameArray = ["mahogany", "sepia", "vermillion", "garnet", "cinnabar", "purple", "scarlet", "brown", "white"]
geneArray = [mahogany, sepia, vermillion, garnet, cinnabar, purple, scarlet, brown, white]

def compare(str1, str2):
    array1 = []
    array2 = []
    first = 0
    second = 1

    for i in range(len(str1)):
        array1.append(str1[first:second])
        first = first + 1
        second = second + 1
    first = 0
    second = 1
    for i in range(len(str2)):
        array2.append(str2[first:second])
        first = first + 1
        second = second + 1
    if len(str1) > len(str2):
        for i in range((len(str1)-len(str2))):
            array2.append("")
    else:
        for i in range((len(str2)-len(str1))):
            array1.append("")
    #print(len(array1))
    #print(len(array2))
    for i in range(len(array1)):
        if array1[i] != array2[i]:
            array1[i] = "-"
    string = ''.join(array1)
    return string

print(compare(mahogany,vermillion))
```

```
sepia - purple
19.672131147540984%
sepia - scarlet
9.89505247376312%
sepia - brown
8.481262327416173%
sepia - white
9.835271317829458%
vermillion - garnet
12.494862309905466%
vermillion - cinnabar
23.214285714285715%
vermillion - purple
12.982456140350877%
vermillion - scarlet
14.842578710644677%
vermillion - brown
15.631163708086785%
vermillion - white
15.455426356589147%
garnet - cinnabar
14.221126181668723%
garnet - purple
5.01438553226469 35%
garnet - scarlet
20.838471023427868%
garnet - brown
21.537196876284423%
garnet - white
22.77024249897246%
cinnabar - purple
8.705357142857142%
cinnabar - scarlet
19.090454772613693%
cinnabar - brown
17.209072978303748%
cinnabar - white
18.7015503875969%
purple - scarlet
6.9965017491254375%
```

```
def percent(string):
    first = 0
    second = 1
    count = 0
    full = len(string)
    for i in range(len(string)):
        if string[first:second] != "-":
            count += 1
        first += 1
        second += 1
    return count/full * 100

for gene in geneArray:
    indexed = int(geneArray.index(gene) + 1)
    #print(indexed)
    #print(gene)
    for i in range(indexed, len(geneArray)):
        print("" + nameArray[indexed - 1] + " - " + nameArray[i])
        cstr = compare(geneArray[indexed - 1], geneArray[i])
        #print(cstr)
        perc = str(percent(cstr)) + "%"
        print(perc)

while True:
    desiredGene = input("Enter the gene you would like the result for: ")
    ind = nameArray.index(desiredGene)
    dGene = geneArray[ind]

    for gene in geneArray:
        print("" + desiredGene + " - " + nameArray[geneArray.index(gene)])
        cstr = compare(dGene, gene)
        perc = str(percent(cstr)) + "%"
        print(perc)
```

```
<!DOCTYPE html>
<html>
    <head>
   <style>
        canvas{
        </style
    </head>
<body>

    <script>
    function hi
        inputTe
        var inn
        var ind
        if ( in
            inn
            inn
            inp
        }

    function st
        inputTe
        var inn
        var ind
        if ( in
            inn
            inn
            inp
        }
    }

    var white =
    "ATGGGCCAAG
    CCGGCCACCCA
    CGGACTATTCT
    CTTTCGATCGC
    ACACCTGATTT
```

```
var nameArray = ["mahogany", "sepia", "vermillion", "garnet", "cinnabar", "purple", "scarlet", "brown", "white"]
var geneArray = [mahogany, sepia, vermillion, garnet, cinnabar, purple, scarlet, brown, white]

var highlightPositions = [];
var strikePositions = [];

function runCompare() {
    var firstSequence = document.getElementById("firstSeq");
    var secondSequence = document.getElementById("secondSeq");
    compare(white, brown);
}

function compare(str1, str2) {
    var array1 = []
    var array2 = []
    var first = 0
    var second = 1

    for (var i=0; i < str1.length; i++) {
        array1.push(str1.substring(first, second));
        first = first + 1;
        second = second + 1;
    }

    for (var i=0; i < str1.length; i++) {
        array1.push(str1.substring(first, second));
        first = first + 1;
        second = second + 1;
    }

    first = 0;
    second = 1;

    for (var i=0; i < str2.length; i++) {
        array2.push(str2.substring(first, second));
        first = first + 1;
        second = second + 1;
    }

    if (str1.length > str2.length) {
        for (var i = 0; i < (str1.length - str2.length); i++) {
            array2.push("");
        }
    } else {
        for (var i = 0; i < (str2.length - str1.length); i++) {
            array1.push("");
```

```
ngth) + "</span>" +
```

```
ATGGGCCAAGAGGATCAGGAGCTATTAATTCGCGGAGGCAGCAAACACCCATCTGCCGAGCATCTGAACAATGGTGACAGCGGAGCGGCTTCGCAGACTCATTACCAGGGCTTCGGCCAGGCC
```

```
) + "</span>" +
```

```
GCAGGCCAAAAACTACGGCACG
GCGGCAGCTGGTCAACCGGACA
AGACGACCCTGCTGAATGCCCTT
TTATCGGCTCCCTAACGGCCAGG
TCGGTGTGCCCGGCAGGGTGAAA
```