SOP TABLE OF CONTENTS:

Any word marked with an asterisk ‘*’ has a definition or explanation provided in the glossary section of this document.

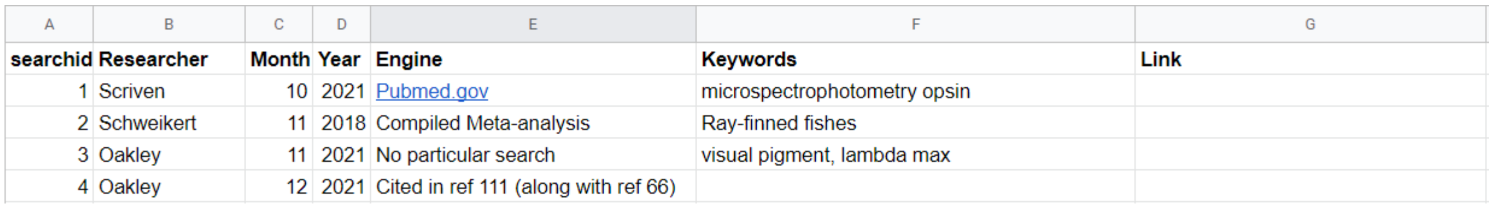
Any word marked with an asterisk ‘*’ has a definition or explanation provided in the glossary section of this document.

ENTERING DATA INTO VIZPHIZ DATABASE:

STEP 1 - LITERATURE SEARCH: [`litsearch`]

[Description] The first step to building a database of this type is to find publications with data of interest. We do want to keep track of how we find these papers (like search terms), so we need to account for and enter the search terms used to find the papers. Sometimes, we find papers in arbitrary ways, which is fine; just specify that in the `litsearch` sheet under the “*Engine*” and/or “*Keywords*” data fields.

Here are a few examples.



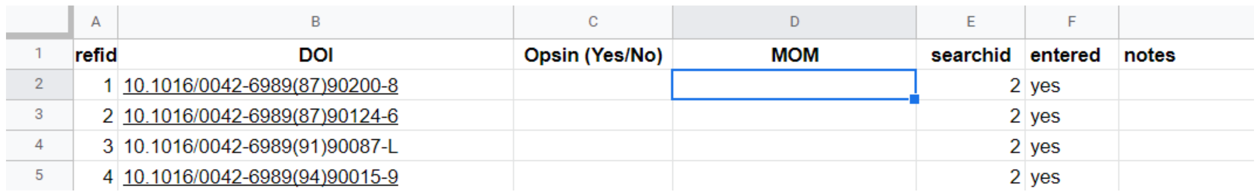
searchid	Researcher	Month	Year	Engine	Keywords	Link
1	Scriven	10	2021	Pubmed.gov	microspectrophotometry opsin	
2	Schweikert	11	2018	Compiled Meta-analysis	Ray-finned fishes	
3	Oakley	11	2021	No particular search	visual pigment, lambda max	
4	Oakley	12	2021	Cited in ref 111 (along with ref 66)		

<u>Order of Operations For Entry</u>

I. **searchid** - Used to link to particular papers in the `references` sheet; so we can keep track of all the papers found with a particular search. Enter the number that comes sequentially after the previous ‘***searchid***’ number.
II. **Researcher** - Enter your last name.
III. **Month** - Enter the current month as number, 1-12.
IV. **Year** - Enter the current year including the first 2 digits (i.e. 2022 not 22).
V. **Engine** - Enter the name of the search engine you used to conduct your literature search (i.e. Pubmed, Google Scholar, ect,). **IF** you found papers as references in another paper or in some similar fashion put ***“Cited in refid #x”*** or “***No particular search”.***
VI. **Keywords** - Enter the keywords used as search terms while pursuing the contents of the search engine. [**IF** no engines were used, simply add keywords that best reflect the scientific scope and focus of the papers you found.]
VII. **Link** - Add a link to the database/website where this search can be replicated in case someone seeks to do so.

STEP 2 - ENTER INDIVIDUAL DOI REFERENCE DATA: [`references`]

[Description]



refid	DOI	Opsin (Yes/No)	MOM	searchid	entered	notes
1	10.1016/0042-6989(87)90200-8			2	yes	
2	10.1016/0042-6989(87)90124-6			2	yes	
3	10.1016/0042-6989(91)90087-L			2	yes	
4	10.1016/0042-6989(94)90015-9			2	yes	

Order of Operations For Entry

I. **refid** - Used to link a numerical value to a particular paper's **DOI**. Thus, we can keep track of all the data associated with a **DOI** by using it as **foreign key** in another sheet. Enter the number that comes sequentially after the previous **'*refid*'** number.

II. **DOI** - Enter the DOI of the paper you are pulling data from, this will allow anyone to return to this reference at any time. Plus, because DOIs are unique, you can avoid a duplicate entry by using the '**CTRL + F'** function to check if the DOI is already present. [**Note** - I wrote a cell formula that can also check this and tell you whether or not the data for a DOI has already been entered.]

III. **Opsin** - For accounting purposes just enter **'*yes'*** if the paper contains information regarding the ***lambda max*** values for at least one opsin gene, and enter **'*no*'** if the paper contains no information regarding the lambda max value of at least one opsin gene. [Note - there are going to be papers present that contain information on other visual proteins, like those that control visual acuity, thus you should enter ***'no'*** for these DOIs.

IV. **MOM** - This refers to the **"Method of Measurement"** used to obtain the ***lambda max*** value of the opsin genes being studied in a paper.

  A. Enter **'*heterologous*'** if the lambda max values of the opsins were measured using ***heterologous expression**** **[**see ***DEFINITIONS*** section for a description].

  B. Enter '***msp'*** if the lambda max value of the opsins were measured using ***micro-spectrophotometry**** [see ***DEFINITIONS*** section]

  C. **IF** more than one ***MOM*** was used in a paper, list them separated by a comma.

V. **searchid** - Enter the corresponding '*searchid'* from the *litsearch* sheet that you used to find this particular DOI.

VI. **entered** -

  A. Enter **'*no'*** if you haven't any of the opsin gene data into the *heterologous, msp,* or *opsins* sheets.

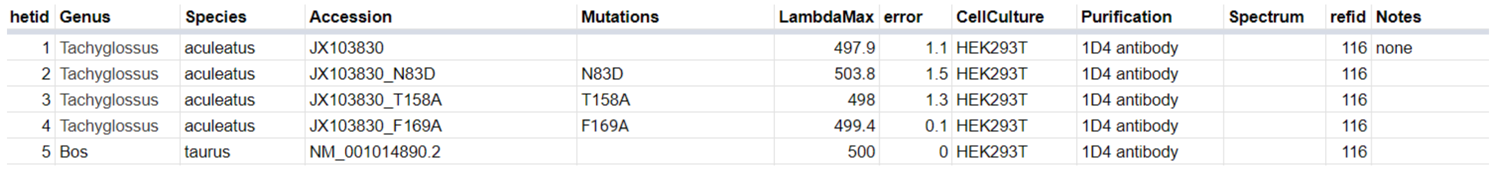
  B. Enter **'**partial**'** if you have entered at least one piece of opsin gene data into the *heterologous, msp,* or *opsins* sheets. [Make sure to add a description to the '*notes'* datafield explaining the reason why the data was only partially entered]

  C. Enter **'yes'** if all the opsin gene data from if you have entered **ALL** relevant opsin gene data from a paper into the *heterologous, msp,* or *opsins* sheets.

VII. **notes** - Enter any information that you deem necessary/pertinent for me to see. This may include a description of why the data was only partially entered; such as their being ***mutant* or *chimeric* opsins** [the conditions for what do do in these situations is discussed in the ***CONTINGENT STEPS*** section].

STEP 3a - ENTER HETEROLOGOUS EXPRESSION DATA: [`heterologous`]

[Description]



hetid	Genus	Species	Accession	Mutations	LambdaMax	error	CellCulture	Purification	Spectrum	refid	Notes
1	Tachyglossus	aculeatus	JX103830		497.9	1.1	HEK293T	1D4 antibody		116	none
2	Tachyglossus	aculeatus	JX103830_N83D	N83D	503.8	1.5	HEK293T	1D4 antibody		116	
3	Tachyglossus	aculeatus	JX103830_T158A	T158A	498	1.3	HEK293T	1D4 antibody		116	
4	Tachyglossus	aculeatus	JX103830_F169A	F169A	499.4	0.1	HEK293T	1D4 antibody		116	
5	Bos	taurus	NM_001014890.2		500	0	HEK293T	1D4 antibody		116	

`=

__**Order of Operations For Entry**__

I. **hetid** - Used to link a numerical value to each data entry in the ***`heterologous`*** sheet. Enter the number that comes sequentially after the previous **‘*hetid*’** number.

II. **Genus** - Enter the genus name for the species that the opsin gene belongs to//is derived from.

III. **Species** - Enter the species name from which the opsin gene belongs to//is derived from.

IV. **Accession** - Enter the ***accession number**** [See ***DEFINITIONS***] for an opsin gene that was tested on in the paper. The accession number for an opsin protein is usually explicitly stated somewhere in the methods section or in the legend of some figure.
   A. For information on entering the accession number of a ***mutant opsin*** reference the “***CONTINGENT STEPS***” section.

V. **Mutations** - Only matters if you are entering a **non-wild-type//mutant opsin**. If you are entering a wild-type opsin then you can ignore this datafield.
   A. Enter any **mutations** made to the opsin in the format ***‘XaaY’***, where ‘X’ is the original amino acid, and ‘Y’ is the mutant amino acid, and ‘aa’ is the corresponding amino acid position/number when aligned to the bovine reference (this is done for you in almost all papers).

VI. **LambdaMax** - Enter the ***lambda max*** value for the opsin protein tested, it may also be referenced as λ_{max} when read in a paper.

VII. **error** - This is in reference to the statistical error range for the lambda max value entered. It is not always guaranteed that a paper will include this information, so don’t stress about it too much. Enter the error value if given.

VIII. **CellCulture** - Enter the **cell culture type** used to express the opsin protein for testing.
   A. The most common ones I’ve seen are ***‘COS1’, ‘COS7’, ‘HEK293S’, ‘HEK292T’***.

IX. **Purification** - Refers to the method used to isolate the opsin protein for testing; this is usually explicitly stated in the methods section or discussion.
   A. The most common ones I’ve seen are ***‘cell membranes’*** or some type of ***‘antibody’***.

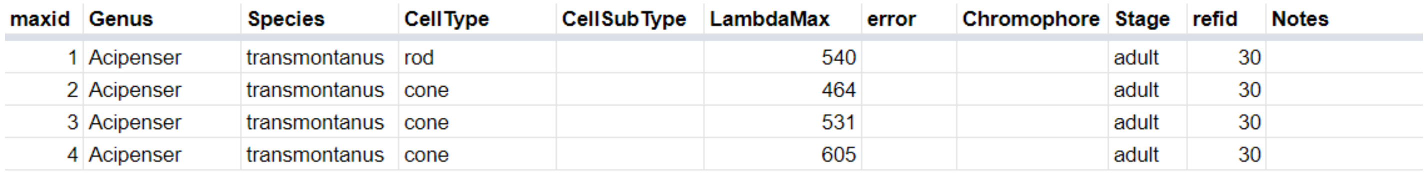
X. **Spectrum** - Refers to the light condition in which the purified opsins were tested.
   A. ***Dark*** - Enter if the purified opsin was tested with a spectrophotometer in the dark.
   B. ***Difference*** - Reference the ***Photobleaching Difference Spectra*** technique for finding the lambda max value of an opsin by creating a difference peak of the absorbance spectra taken before and after photobleaching. Enter if the purified opsin was tested using this technique.

XI. **refid** - Enter the corresponding ***refid*** for the paper that you obtained the opsin gene data from.

XII. **Notes** - Enter any information that you deem necessary/pertinent for me; including anything that you were confused about.

STEP 3b - ENTER MICRO-SPECTROPHOTOMETRY (MSP) DATA: [*msp*]

[Description]



maxid	Genus	Species	CellType	CellSubType	LambdaMax	error	Chromophore	Stage	refid	Notes
1	Acipenser	transmontanus	rod		540			adult	30	
2	Acipenser	transmontanus	cone		464			adult	30	
3	Acipenser	transmontanus	cone		531			adult	30	
4	Acipenser	transmontanus	cone		605			adult	30	

Order of Operations For Entry

I. **maxid** - Used to link a numerical value to each data entry in the ***msp*** sheet. Enter the number that comes sequentially after the previous **'*maxid*'** number.

II. **Genus** - Enter the genus name for the species that the opsin gene belongs to//is derived from.

III. **Species** - Enter the species name from which the opsin gene belongs to//is derived from.

IV. **CellType** - This should be explicitly stated in the abstract, intro, or methods. There are only three types that you should encounter, ***Rods***, ***Cones***, and ***Rhabdomeres*** (rare).

V. **CellSubType** - This should be explicitly stated as well.
   A. Enter ***single*** if the describes a single ***cone cell***, not joined together with another member of the same or different cell type.
   B. Enter ***double*** if the paper describes two ***cone cells*** joined together that may also be coupled optically/electrically.
      1. Enter ***twin*** if the two members are of the same cell type.

VI. **LambdaMax** - Enter the ***lambda max*** value for the opsin protein tested, it may also be referenced as λ_{max} when read in a paper.

VII. **error** - This is in reference to the statistical error range for the lambda max value entered. It is not always guaranteed that a paper will include this information, so don't stress about it too much. Enter the error value if given.

VIII. **Chromophore** - No Entries Yet - Ignore For Now.

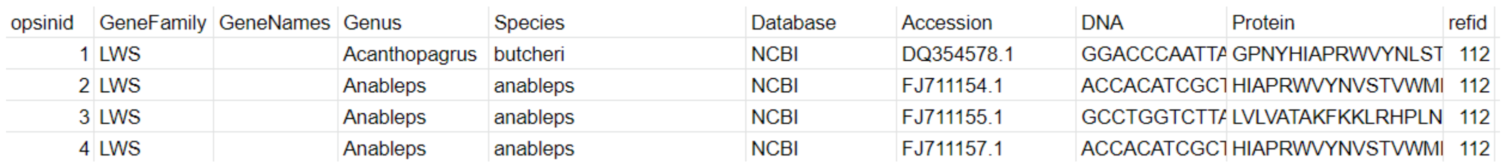
IX. **Stage** - Not every MSP test is done on adult photoreceptor cells; they may be described as ***Juvenile***, ***Small Juvenile***, ***Large Juvenile***, ***Adult***, or you can enter ***Unknown*** if not explicitly stated.

X. **refid** - Enter the corresponding ***refid*** for the paper that you obtained the opsin gene data from.

XI. **Notes** - Enter any information that you deem necessary/pertinent for me; including anything that you were confused about.

STEP 4 - ENTER OPSIN GENE DATA: [*`opsins`*]

[Description]



opsinid	GeneFamily	GeneNames	Genus	Species	Database	Accession	DNA	Protein	refid
1	LWS		Acanthopagrus	butcheri	NCBI	DQ354578.1	GGACCCAATTA	GPNYHIAPRWVYNLST	112
2	LWS		Anableps	anableps	NCBI	FJ711154.1	ACCACATCGCT	HIAPRWVYNVSTVWMI	112
3	LWS		Anableps	anableps	NCBI	FJ711155.1	GCCTGGTCTTA	LVLVATAKFKKLRHPLN	112
4	LWS		Anableps	anableps	NCBI	FJ711157.1	ACCACATCGCT	HIAPRWVYNVSTVWMI	112

<u>Order of Operations For Entry</u>

I. **opsinid** - Used to link a numerical value to each data entry in the ***`opsin`*** sheet. Enter the number that comes sequentially after the previous **'*opsinid*'** number.

II. **GeneFamily** - There are five relevant gene families for opsins, including **LWS, SWS1, SWS2, Rh1, and Rh2*.** The identity of an opsin's gene family should be explicitly stated in the abstract, intro, methods, or discussion sections.
   A. [Refer to the wikipedia link under the definition for ***Opsins*** in the ***DEFINITIONS*** section for more information about each of these subtypes.]

III. **GeneNames** - Sometimes genes are given an explicit name to be referenced by, however this is not too common in the literature, so it's not the most important thing to get hung up on.

IV. **Genus** - Enter the genus name for the species that the opsin gene belongs to//is derived from.

V. **Species** - Enter the species name from which the opsin gene belongs to//is derived from.

VI. **Database** - This refers to the database in which the genetic information for the opsin gene is stored in. Most likely you'll use ***NCBI*** to obtain the genetic information in ***GenBank***.
   A. In fact, here's a link to NCBI GenBank for whenever you need it. → https://www.ncbi.nlm.nih.gov/genbank/

VII. **Accession** - Enter the ***accession number**** [See ***DEFINITIONS***] for an opsin gene that was tested on in the paper. The accession number for an opsin protein is usually explicitly stated somewhere in the methods section or in the legend of some figure.
   A. For information on entering the accession number of a ***mutant opsin*** reference the "***CONTINGENT STEPS***" section.
   B. REMEMBER - YOU WILL NOT ENTER THE MUTANT DNA OR PROTEIN SEQUENCES.

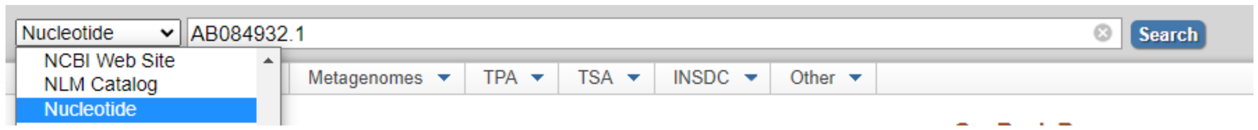
VIII. **DNA** - Enter the DNA sequence obtained from GenBank using the wild-type accession number. [Note - You should be able to simply copy and paste the sequence, however you may need to format it when you paste it into the ***`opsins`*** sheet.]
   A. Reference ***RETRIEVING DNA & PROTEIN SEQUENCES FROM NCBI*** for a brief guide on how to do this.

IX. **Protein** - Enter the protein amino acid sequence obtained from GenBank using the wild-type accession number. [Note - You should be able to simply copy and paste the sequence, however you may need to format it when you paste it into the ***`opsins`*** sheet.]

X. **refid** - Enter the corresponding ***refid*** for the paper that you obtained the opsin gene data from.

RETRIEVING SEQUENCE DATA FROM NCBI GENBANK

1. Navigate to the NCBI GenBank website.
   a. https://www.ncbi.nlm.nih.gov/genbank/
2. Enter/Paste the accession number of interest into the search bar at the top of the page.
3. Check that you have ***'Nucleotide'*** selected from the database selection drop-down window.



4. Press '***enter***' or click the '***search***' button.
5. Click on the '***FASTA***' hyperlink to be taken to the nucleotide sequence for the opsin gene.
   a. Copy and paste that sequence into the text-box at the following link.
      → Remove All Whitespace - Delete Spaces, Tabs, Newlines - Online - Browserling Web Developer Tools
   b. Click the '***Remove All Spaces***' button.
   c. Copy the newly formatted DNA sequence to your clipboard.
6. Paste the DNA sequence into the '***DNA***' datafield on the ***`opsins`*** sheet.
7. Navigate back to the NCBI page from earlier and select the browser's back arrow to be taken to the previous page.
8. Scroll down to the section titled '***FEATURES***'.
   a. You should find the protein's amino acid sequence following the text ***'/translation='***
   b. Copy that sequence and repeat the steps for removing the spaces from the sequence.
9. Insert the formatted sequence into the '***Protein***' datafield on the ***`opsins`*** sheet.

***NOTE** - IF THERE IS NO ACCESSION # PROVIDED BY THE PAPER YOU WILL LOOK UP THE SPECIES NAME AND OPSIN FAMILY INSTEAD.
Or, if it is too confusing you can simply make note of it in the ***`references`*** sheet and fill in the cell as **PURPLE**.

CONTINGENT STEPS:

I. **Mutant Opsins -**

   A. **Make note of any mutant opsin genes found in papers listed on the *references* sheet under the 'notes' data field.**

      1. Ex. - If I am reviewing a paper for heterologously expressed opsins that runs tests on to find the lambda max values for X# of mutant opsins, I would enter "X# of mutant opsins present and tested for lambda max". **Then change the color of that cell to green.**

   B. **Best practice to enter all *wild type* data pertaining to an opsin BEFORE entering the mutant opsin data.**

   C. **Enter the data for the mutant opsin on the heterologous sheet as normal, with the addition of filling out the 'Mutations' data field in the manner that follows "Xaa#Y".**

      1. Ex. - If I am given a mutant opsin that has the 97th amino acid changed from 'T' to 'S', I would then enter that as 'T97S' in the '*Mutations*' data field.
      2. Ex.2 - If given a mutant opsin with two mutations, list them separated by commas. → 'T97S, E122I'

   D. **Additionally, when entering the '*Accession'* number for a mutant you should do so in the manner that follows "WTAcc#_XaaY"**

      1. Ex. - If the mutant opsin's wild type accession number is "NM_001014890" and the mutation is 'T97S', then the mutant accession number would be entered as "NM_001014890_T97S"
      2. Ex.2 - If the mutant opsin has >1 mutation, say 'T97S' and 'E122I' with a wild-type accession of "NM_001014890", then the resulting mutant accession number would be ""NM_001014890_T97S_E122I".

   E. **Enter the rest of the data for the mutant opsin as you normally would. EXCEPT, YOU WILL NOT ENTER THE DNA OR PROTEIN SEQUENCES ON THE *'OPSINS' SHEET*! I will take care of those with python code if you are not trained how to use it.**

II. **Mutating Sequences - 'mutagenesis.py'**

   A. **What does it do?**

      1. Takes a target wild-type opsin amino acid sequence, aligns it to a reference sequence and makes desired amino acid substitutions at target positions on sequence.

   B. **How do I use this script?**

      1. **Packages -** Ensure the following packages are downloaded on the python environment used to run this script.

         a) *argparse*
         b) *Biopython*
         c) *Skbio*

2. **Parameters//Arguments -**
    a) **Mutations [-m]** = the actual mutations performed on the sequence in the format of 'amino acid X at some position on the sequence to amino acid Y'
        (1) 'aaX_position_aaY'
        (2) **Ex.** '-m A120S'
        [amino acid substitution at position 120 from A → to → S]
        (3) For multiple mutations on one sequence, use ',' to separate each mutation.
            (a) **Ex.** -m N83D, A120S, …
        (4) IF the first amino acid letter of a mutation set does not match the amino acid actually present on the sequence at that position (so in N83D -> If 'N' != the amino acid present at site 83) then the script will abort and raise the following exception - "
            (a) This usually occurs when the incorrect reference sequence and/or substitution matrix is used for alignment and is typically a game of guess and check at that point. 🙂
    b) **Reference Accession [-ra]**: The sequence which acts as a reference for alignment and mutation of the target sequence (the sequence being mutated).
        (1) Input - Requires an **accession number*** for the desired reference sequence.
            (a) **Ex.** -ra NM_001014890
        (2) For vertebrate opsins, it's standard that the **Bovine Rhodopsin (BRhO)*** acts as the reference sequence.
        (3) In fact, the BRhO accession is set as the **default** for this parameter, so this is **technically optional** and the **program will still run if nothing is entered.**
        (4) Alternatives - For invertebrate opsins, there is no technical standard for reference sequence alignment BUT the two most commonly used sequences are…
            (a) *Loligo forbesii* - X56788.1
            (b) *Todarodes pacificus* - X70498.1
        (5) **IF** your reference accession is **NOT IN NCBI** then enter '**manual'** for this parameter and you will be given the option to enter/paste the amino acid sequence into a line when prompted '**Enter Reference Sequence: '**
            (a) **Make sure there are no spaces in your sequence!**
    c) **Target Accession [-ta]**: The sequence that is actually being aligned relative to some reference sequence and modified/mutated by the python script.

(1) Input - Requires an accession number for the desired target sequence.
   (a) Ex. -ta JX103830
(2) **IF** your target accession is **NOT IN NCBI** then enter ‘**manual’** for this parameter and you will be given the option to enter/paste the amino acid sequence into a line when prompted ‘**Enter Target Sequence: ’**
   (a) **Make sure there are no spaces in your sequence!**

d) **Email [-em]**: Your institution/university email that will be used to prompt the NCBI database for the desired sequences.

3. **Use -** One liner format.
   a) mutagenesis.py [-h] -m MUT [-ra REFERENCE_ACCESSION] -ta TARGET_ACCESSION [-v VERBOSE] -em EMAIL
      (1) **Ex**. python mutagenesis.py -

III. Chimeric Opsins -

A. Make note of any chimeric opsin genes found in papers listed on the *references* sheet under the ‘notes’ data field.

1. Ex. - If I am reviewing a paper for heterologously expressed opsins that runs tests on to find the lambda max values for #X of chimeric opsin proteins, I would enter “X# of chimeric opsins present and tested for lambda max”. **Then change the color of that cell to red.**
2. Additionally, make note of any figure/table #’s that will be helpful for data entry.

B. After that, do nothing else unless you are trained on chimera entry. It can get quite complicated. :)

IV. Making Chimeric Sequences - ‘chimera_mut.py’

A. What does it do?

1. **Takes multiple opsin amino acid sequences and splices segments together in whatever fashion designated.**
   a) **Ex. -** I want a chimeric opsin between the Bovine Rhodopsin and Human Blue Opsin. I can designate that this chimera contains the amino acids of the Bovine from positions 1-100, the human opsin from positions 101-200, and the bovine again from 201-end of sequence.
   b) **Ex. 2 -** In more specific use cases, I can designate that I want to specifically swap one or more of the **transmembrane domains*** from the bovine into the human blue opsin. All that is required is the amino acid range in which those transmembrane domains are found.

B. How do you use this script?

1. **Packages -** Ensure the following packages are downloaded on the python environment used to run this script.

   a) *Argparse*
   b) *Posixpath*
   c) *Biopython*
   d) *skbio*

2. **Parameters//Arguments -**
   a) **Chimeric Opsin [-co]:** The chimeric opsin accession tag which codes for all the sequence modifications.
      (1) **Input -** Requires a continuous string of accession numbers and range of amino acids to take from each sequence.
      (2) Naming Convention - Acc#1_AARange-Acc#2_AARange-Acc#x_AA_Range
         (a) Technically, this script can take any number of sequences and their ranges, it is not limited as long as the naming convention is properly formatted.
         (b) Note - '_' is used to separate an accession from its target amino acid range, while '-' is used to separate sequence 'chunks' (accession + range = chunk) from each other.
         (c) Ex. -co AB458130.1_1_100-AB084945.1_101_300
      (3) **IF** one or more of your target accessions are **NOT IN NCBI** then enter '**manual'** for that chunk and you will be given the option to enter/paste the amino acid sequence into a line when prompted '**Enter Target Sequence: '**
         (a) Ex. -co manual_1_100-AB084945.1_101_300
         (b) **Make sure there are no spaces in your sequence!**
      (4) IF an accession starts with two letters followed by an '_' then you will be prompted to re-enter that accession during the script run-time. This is an unfortunate exception that arose due to the way I parse the input chimeric sequence names.
         (a) Ex. NM_001014890 - This would prompt an exception response and you would simply re-enter the accession number.
   b) **Mutations [-m]** = the actual mutations performed on the sequence in the format of 'amino acid X at some position on the sequence to amino acid Y'
      (1) 'aaX_position_aaY'
      (2) **Ex.** '-m A120S'
      [amino acid substitution at position 120 from A → to → S]
      (3) For multiple mutations on one sequence, use ',' to separate each mutation.
         (a) **Ex.** -m N83D, A120S, …

(4) **IF** the first amino acid letter of a mutation set does not match the amino acid actually present on the sequence at that position (so in N83D -> If ‘N’ != the amino acid present at site 83) then the script will abort and raise the following exception - “
   (a) This usually occurs when the incorrect reference sequence and/or substitution matrix is used for alignment and is typically a game of guess and check at that point. 🙂

c) **Reference Accession [-ra]:** The sequence which acts as a reference for alignment and mutation of the target sequence (the sequence being mutated).
   (1) Input - Requires an **accession number*** for the desired reference sequence.
      (a) **Ex.** -ra NM_001014890
   (2) Refer to the **Mutagenesis** section above for more about reference accessions.

d) **Email [-em]**: Your institution/university email that will be used to prompt the NCBI database for the desired sequences.

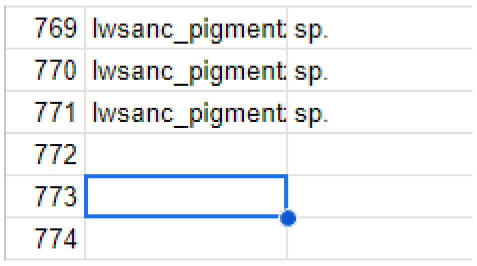
3. Use - One liner format
   a) **Ex**. -co AB458130.1_1_100-AB084945.1_101_300 -m N83D -ra NM_001014890 -em seth_is_awesome@aol.com

Setting up mySQL Database:

‘Visual Physiology Database’ → Public Copy of VisualPhysiologyDB

Formatting

I. **Data Clean-up Tips**
   A. **No trailing white spaces for any entries** in the *heterologous* and *opsins* sheets. This is especially important because if something like an accession number on one sheet has no trailing white space and the same accession on a connected sheet does, there will be errors when extracting data from mySQL.
   B. **No unpopulated ‘id’ entries**, these cause errors when importing data into mySQL.



769	lwsanc_pigment	sp.
770	lwsanc_pigment	sp.
771	lwsanc_pigment	sp.
772		
773		
774		

C. **No blank entries for the '*error'* column in the *heterologous* sheet -** this will also cause errors when importing data into mySQL.

II. **Downloading Sheets as a TSV**
   A. **For both the *heterologous* and *opsins* sheets select and download them as '.tsv' files. This is the format that our python script uses to import the data properly.**

Schema Structure

I. **'Opsins' Table**
   A. **Primary Key* - '*opsinid***
   B. **Foreign Key* - '*accession*'**

II. **'Heterologous' Table**
   A. **Foreign Key - '*accession'***
   B. **Primary Key - '*hetid'***

Scripts

I. '**setupVPD.sql' - SQL text file which sets up 'Vizphiz' database**
   A. Run this file in the actual mySQL workbench.
   B. If otherwise unfamiliar with mySQL I highly recommend reading up on it in general before trying all this. [Linking a *linkedin learning* intro about mySQL below]
      1. https://www.linkedin.com/learning/mysql-installation-and-configuration?trk=learning-serp_learning-search-card_search-card&upsellOrderOrigin=default_guest_learning

II. **'Hetero2sql.py' - Import data from the 'heterolgous.tsv' file into DB.**
   A. **Packages -** Ensure the following packages are downloaded on the python environment used to run this script.
      1. mySQL
      2. mySQL.connector
   B. Run this file in the provided jupyter notebook if possible '*condensed_wf.jynb'.*
   C. Otherwise, use this as a python script, and you will be prompted for your database name and password.
      1. The password will be up to you, but the name of the database should be '*vizphiz'* if you used our code to set it up.
   D. **Use** - python hetero2sql.py
   E. **Make sure that the *heterologous.tsv* file is in your working directory!**

III. **'Ops2sql.py' - Import data from the 'opsinsdb.tsv' file into DB.**
   A. **Packages -** Ensure the following packages are downloaded on the python environment used to run this script.
      1. mySQL
      2. mySQL.connector
   B. Run this file in the provided jupyter notebook if possible '*condensed_wf.jynb'.*
   C. Otherwise, use this as a python script, and you will be prompted for your database name and password.

1. The password will be up to you, but the name of the database should be '*vizphiz'* if you used our code to set it up.

D. **Use** - python ops2sql.py
E. **Make sure that the *opsindb.tsv* file is in your working directory!**

deepBreaks ML Pipeline:

Two versions of the pipeline are covered - one for the creation of a linear regression model [lin-reg], and the other for a classification model [classifier].

Download the respective jupyter notebooks, '*opsin_wf_linreg.ipynb'* and 'opsin_wf_classifier.ipynb' for the full pipeline.

Packages

This is a relatively complex pipeline so there are many package contingencies, if you are not familiar with creating, downloading packages to, and working in python environments then I would recommend reading up before doing this next section.

- *deepBreaks*
  - Actually a collection of several packages and scripts put together by collaborators at '*George Washington University*'.
  - Follow the directions for downloading the deepBreaks packages present on the deepBreaks gitHub documentation package.
    - Link to GitHub here → https://github.com/omicsEye/deepbreaks
- *Argparse*
- *mySQL*
- *mySQL.connector*
- *Pycaret*
- *Scipy*
- *Matplotlib*
- *Sklearn*
- *Numpy*
- *Pandas*
- *seaborn*

Extraction

Purpose - **Extract data from the VizPhiz mySQL database and create several 'cuts' of the larger dataset for running sub-tests//analyzes.**

Lin-Reg -

- **Input:** You will be prompted for your database name and password just as you were when importing data. Other than that just click the '*run-cell'* tab and relax.
- **Output:**
  - 6 different 'versions' or 'splits' of the data, in a **FASTA*** ready format [.txt format].
    - Whole dataset (WDS)

        - Whole dataset, minus all invertebrate opsins (WDS_NI)
        - Only Rod and ‘Rod-like’ opsins (ROD)
        - Only SWS & UV Opsins (SWD)
        - Only LWS & MWS Opsins (MWD)
        - Only Wild-type Opsins - No mutants or Chimeras. (NMOC)
    - Equal number of meta-data files for each raw-data-file above [.tsv format]
        - Meta-data files contain the **lambda max, species, opsin family, and accession** for each sequence, which is linked to the meta-data file via a sequence ID.
        - Ex. ‘S1’ = the first sequence in our raw-data file and is also be found in the meta-data file with all values listed above

Classifier

- **Input:** You will be prompted for your database name and password just as you were when importing data.
- **Output:**
    - 2 ‘versions’ or ‘splits’ in a **FASTA*** ready format [.txt format].
        - Whole dataset (WDS)
        - Whole dataset, minus all invertebrate opsins (WDS_NI)
    - 4 meta-data files, 2 are identical to the lin-reg outputs, but are none-the-less necessary for this pipeline.
        - The classifier specific metadata files simply connect a sequence tag, ‘S1’, to its **Opsin Class**, ‘7’**.**
        - Opsin classes are designated as ‘buckets’ of wavelengths corresponding to ranges of lambda max values which encapsulate some ‘class’ of opsin. Each opsin will fit into at least one of these class ranges.
        - There are 10 total classes, with class assignments starting at ‘0’.
        - . Ex. Class 0 = any opsin with a lambda max between 340 nm-360 nm.

Alignment

[Same for Lin-reg & Classifier Models // External process, requires MAFFT program]

Purpose - **Align all sequences in a raw data-file so they are of standardized length, and account for all potential ‘gaps’*.**

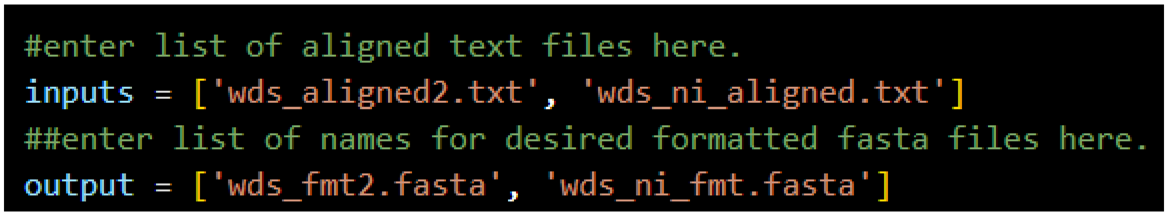
- **Input:** Any of the output data splits from the section above.
    - For proper documentation and tutorials on how to use MAFFT visit this link to the documentation → https://mafft.cbrc.jp/alignment/software/
    - Suggested parameters for MAFFT alignment are…
        - Fasta Format (Sorted)
        - Strategy = FFT-NS-2 or G-INS-1
- **Output:** A text file containing aligned opsin sequences, in FASTA block format (this will be fixed in the next step).

Formatting

[Same for Lin-reg & Classifier Models]

Purpose - Taking the aligned opsin sequences and formatting them specifically for deepBreaks processing. This essentially eliminates all white spaces and makes the sequence fit on one line instead of block form.

- **Input:** Any of the output aligned data splits from MAFFT processing, you will need to simply modify line 2 of this code block to contain the file names for all alignments you want formatted. **[See screenshot below]**



```
#enter list of aligned text files here.
inputs = ['wds_aligned2.txt', 'wds_ni_aligned.txt']
##enter list of names for desired formatted fasta files here.
output = ['wds_fmt2.fasta', 'wds_ni_fmt.fasta']
```

- **Output:** A FASTA file containing aligned opsin sequences formatted for deepBreaks processing.
    - Note, you must modify line 4 of this code block to contain the corresponding list of output file names for your formatted sequences. **Make sure that the file name ends in '.fasta'. [See screenshot above]**

Running deepBreaks // Interpreting Results

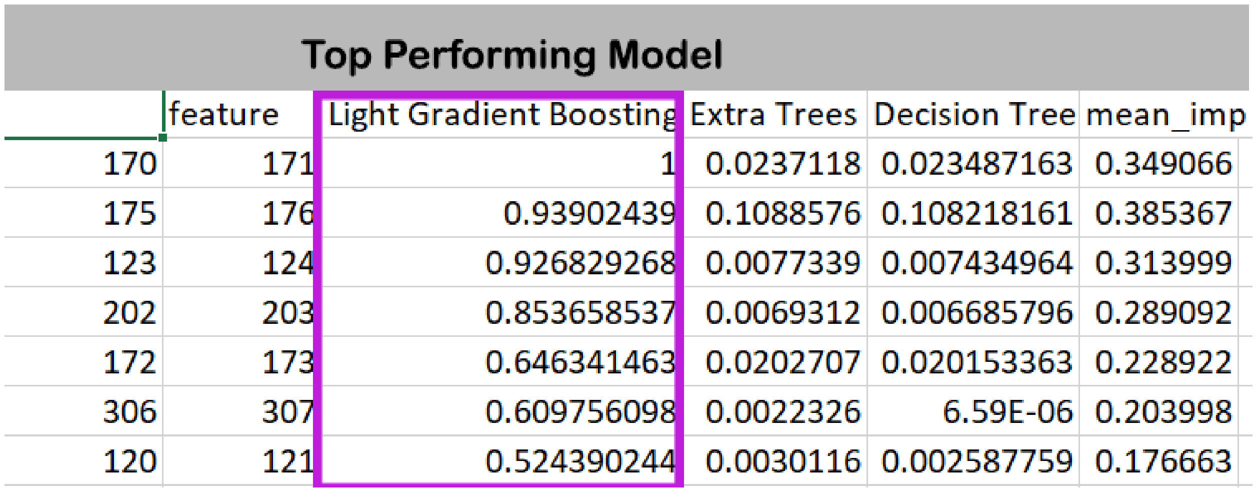
Purpose - Train classical ML models* to either predict 'candidate spectral tuning sites'* (CSTS) on the opsin sequence (lin-reg) or predict opsin class (spectral range//lambda max) from the amino acid sequence alone.

Lin-Reg - [Candidate Spectral Tuning Sites]

- **Input:**
    - An aligned and formatted data file which contains sequence IDs and their corresponding sequences.
        - **Ex.** 'wds_ni_fmt.fasta'
        - Set Line 4 of the first code block in this section to the file name//directory for the desired formatted sequence file
            - Variable = 'seqFileName'.
    - The corresponding metadata file
        - **Ex.** 'wds_meta.tsv'
        - Set Line 5 of the first code block in this section to the file name//directory for the desired/corresponding classifier specific metadata file.
            - Variable = 'meta_data'
- **Output:** An analysis of CSTS and a ton of graphs. I will highlight the most important files below.
    - ***'Models_summary.csv'***
        - Provides a metrics summary for each of the model's tested. Most will only be concerned with the R^2 value, which is used to rank

the performance of each model relative to one another. The model with the highest R^2 (thus ranked the highest) will be at the top of the file.

- ***'Avg_top_models_feature_importance.csv'***
  - This file contains and compares the relative 'feature importance' (each 'feature' translating to an aa position) for each of the top performing models. The model name in the furthest left-hand column (should be the 3rd column overall) is your top performing model. All features are ranked by their relative importance from highest to lowest, but are sorted with respect to the top performing model.
  - Ex. The top performing model in this case, highlighted by the purple square below, is the '*Light Gradient Boosting Machine*'.

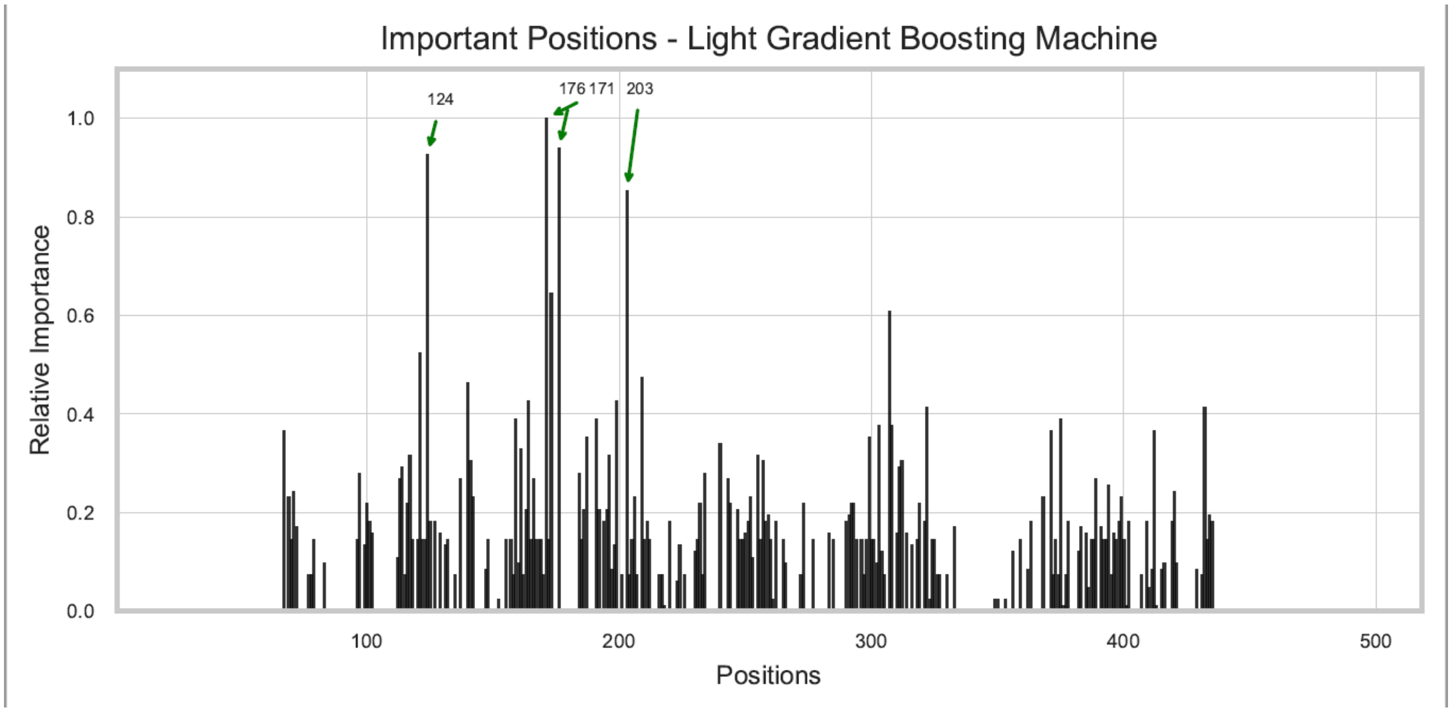


Top Performing Model

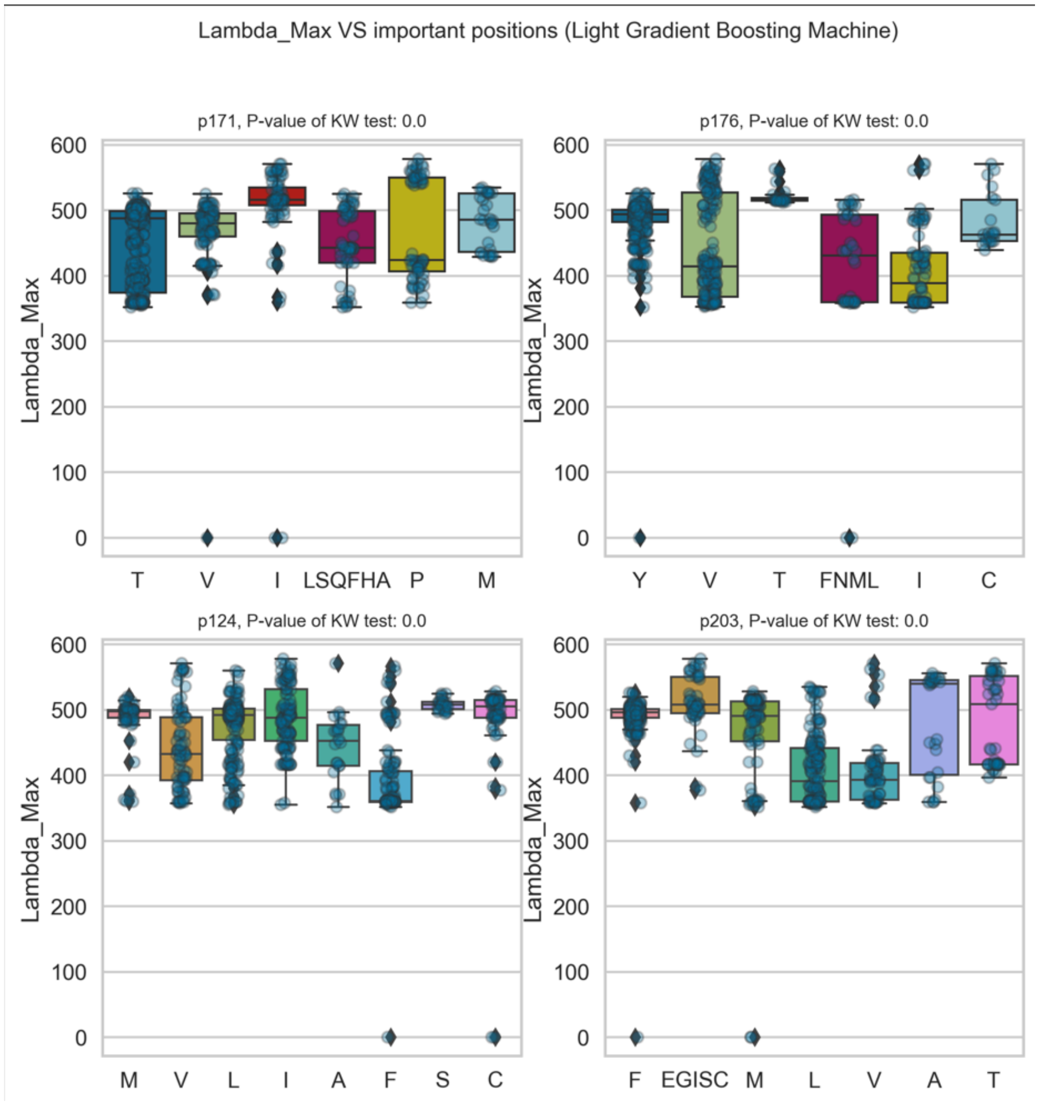
	feature	Light Gradient Boosting	Extra Trees	Decision Tree	mean_imp
170	171	1	0.0237118	0.023487163	0.349066
175	176	0.93902439	0.1088576	0.108218161	0.385367
123	124	0.926829268	0.0077339	0.007434964	0.313999
202	203	0.853658537	0.0069312	0.006685796	0.289092
172	173	0.646341463	0.0202707	0.020153363	0.228922
306	307	0.609756098	0.0022326	6.59E-06	0.203998
120	121	0.524390244	0.0030116	0.002587759	0.176663

  - The highest ranked positions from the top performing model can be recorded and back translated to the bovine standard. There is a code block and separate script provided that do this for you by inputting the aligned bovine sequence and the sites you wish to translate [See ***Post Processing*** section for more details].

- **Significant Positions Plots**
  - **Barplots -** Provide visual comparison of positions vs. their relative importance. You could technically get all the data of the sheet discussed above, but this serves as a useful visualization nonetheless.
    Ex. '*Light Gradient Boosting Machine_350.pdf'*



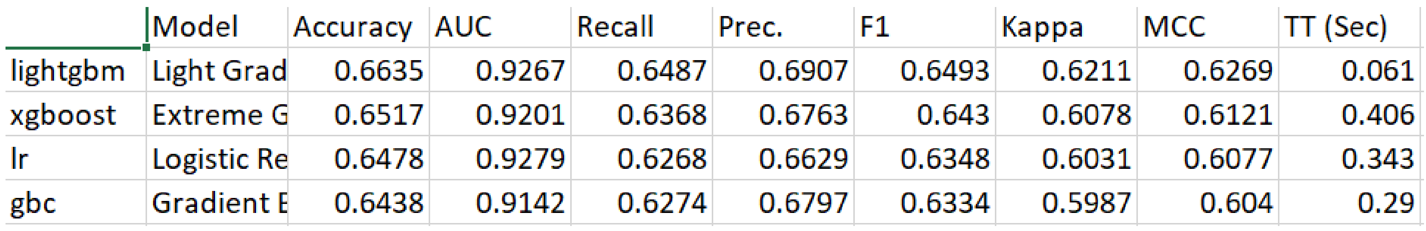
■ **Box Plots -** Provide a visualization for which amino acid residues are associated with different ranges of lambda max at an 'important position' or CSTS, ranked from most frequent to least frequent amino acid (left to right).
Ex. 'Light Gradient Boosting Machine_box_position_350.png'



This example specifically highlights the box plots for the four positions ranked with the highest relative importance. A similar individual box plot can be found for each position on the aligned sequence under the '***significant_positions_plots'*** sub-folder.

Classifier - [Opsin Spectral Sensitivity Classifier]

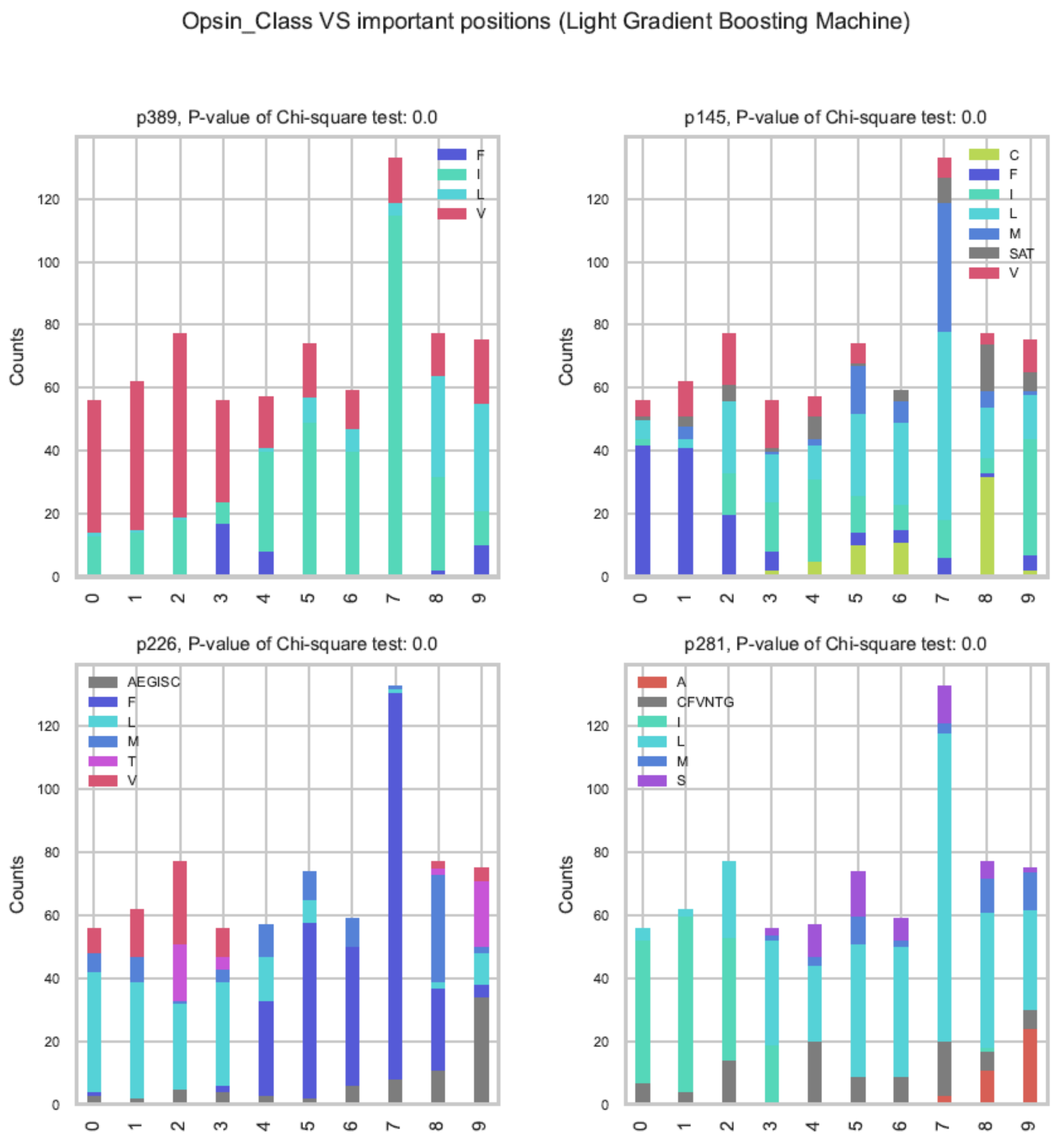
- **Input:**
    - An aligned and formatted data file which contains sequence IDs and their corresponding sequences.
        - **Ex.** 'wds_ni_fmt.fasta'
        - Set Line 4 of the first code block in this section to the file name//directory for the desired formatted sequence file
            - Variable = 'seqFileName'.
    - 2 corresponding metadata files.
        - 1 - Classifier specific metadata file.
            - **Ex.** 'wds_classifier_meta.tsv'
        - 1 - Metadata file identical to that of the linear regression metadata files.
            - **Ex.** 'wds_meta.tsv'
        - Set Line 5 of the first code block in this section to the file name//directory for the desired/corresponding classifier specific metadata file.
            - Variable = 'meta_data'.
- **Output:**
    - **'*Models_summary.csv'* - similar to lin-reg, but with classifier specific statistics.**
        - **Just as before, model performance is ranked top-to-bottom.**
        - **Most useful statistics are Accuracy*,AUC*, Recall*, Prec. and F1*.**



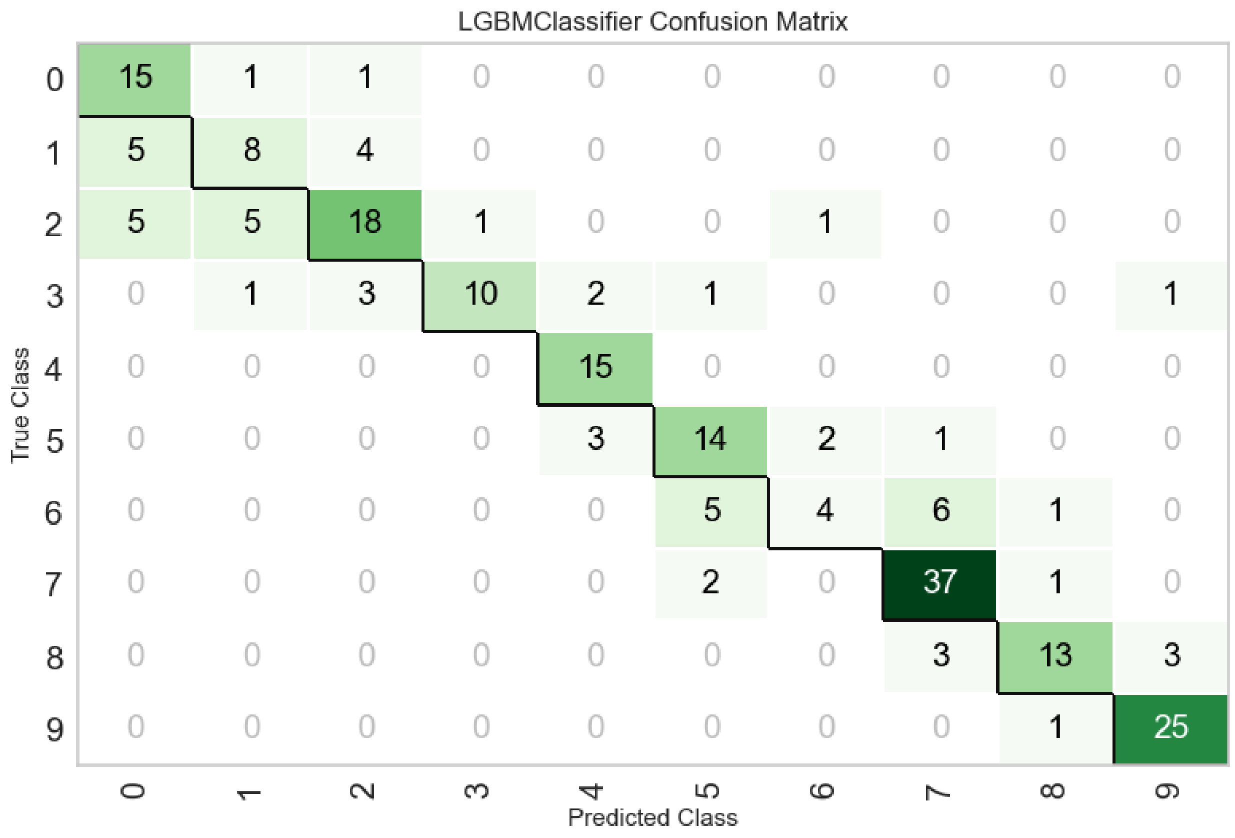
	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lightgbm	Light Grad	0.6635	0.9267	0.6487	0.6907	0.6493	0.6211	0.6269	0.061
xgboost	Extreme G	0.6517	0.9201	0.6368	0.6763	0.643	0.6078	0.6121	0.406
lr	Logistic Re	0.6478	0.9279	0.6268	0.6629	0.6348	0.6031	0.6077	0.343
gbc	Gradient E	0.6438	0.9142	0.6274	0.6797	0.6334	0.5987	0.604	0.29

-
    - ***'Avg_top_models_feature_importance'***
        - Similarly to the lin-reg output file, this file contains and compares the relative 'feature importance' (each 'feature' translating to an aa position) for each of the top performing models.
        - **However,** note that in the case of the linear regression model, these significant positions were considered CSTS. Here we consider these positions important to the classification of an opsin (in terms of its spectral range/lambda max).
    - **Significant Position Plots**

- **Position Importance Barplot -** Identical to the lin-reg output, but the positions highlighted are important to classification, not to spectral tuning (although realistically they may not be mutually exclusive).
- **Position AA Frequency Classification Stacked Barplot -** Probably one of the most interesting outputs from the pipeline in my opinion. These graphs break down the amino acid residue frequency by class at positions deemed important to classification. This way it's quite easy to visualize how amino acid residues frequency changes across opsin classes/spectral ranges. Potential use in mutagenesis experiments in the future?
  Ex. '*Light Gradient Boosting Machinetop_positions350.pdf*'
  Note - Just like the lin-reg output, there is a bar-graph for each significant position in the sub-folder '*significant_positions_plots*'.



- ***'Confusion_matrix.png'***
  - This output file visually summarizes the performance of the top model on its test set from the training pipeline. With this we can see what classes of opsins the model had the hardest time classifying, and see patterns in misclassification (i.e - if class 6 was consistently misclassified as class 5) which may indicate problems in the training data/bin sizes.
  - For more about confusion matrices* see the glossary.
  - **Ex.**



	0	1	2	3	4	5	6	7	8	9
0	15	1	1	0	0	0	0	0	0	0
1	5	8	4	0	0	0	0	0	0	0
2	5	5	18	1	0	0	1	0	0	0
3	0	1	3	10	2	1	0	0	0	1
4	0	0	0	0	15	0	0	0	0	0
5	0	0	0	0	3	14	2	1	0	0
6	0	0	0	0	0	5	4	6	1	0
7	0	0	0	0	0	2	0	37	1	0
8	0	0	0	0	0	0	0	3	13	3
9	0	0	0	0	0	0	0	0	1	25

- **Queryable classification model -** one feature that we've manually added to this pipeline is the ability to save and interact with a trained classification model. For simplicity's sake, it's automatically set such that you only query the top-performing model. Instructions for interacting with the model are provided in the next section.

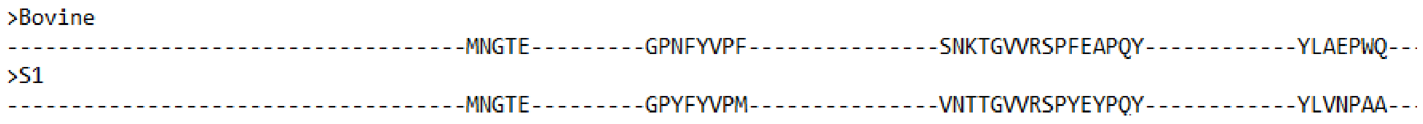
Post Processing

Note - These functions are present in the same respective Jupyter notebooks used for the rest of this pipeline.

Lin-Reg

- **CSTS Translator Script**

  - **Purpose -** Translate a list of aligned CSTS to the standardized bovine equivalent.
  - **Input - [**Enter when prompted**]**
    - **List of top CSTSs from the '*Avg_top_models_feature_importance'* file separated by commas.**
    - **Desired output file name.**
    - **Aligned Bovine Sequence -** This should be at the top of whatever respective formatted sequence file you used to train the model.



```
>Bovine
------------------------------------MNGTE---------GPNFYVPF---------------SNKTGVVRSPFEAPQY------------YLAEPWQ--
>S1
------------------------------------MNGTE---------GPYFYVPM---------------VNTTGVVRSPYEYPQY------------YLVNPAA--
```

  - **Output -** A text file which contains the list of translated CSTS, including whether or not those positions are in any of the opsin's seven transmembrane domains.
    - The translated positions are what we compare to previous literature, to see if those positions have been validated as spectral tuning sites or if they are untested. For the purposes of our experiments we plan to take the top performing CSTS (so untested) and perform mutagenesis on the bovine rhodopsin.

Classifier

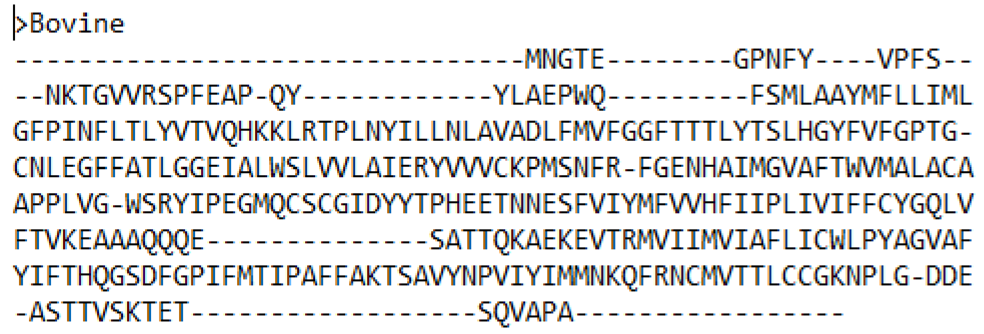
- **Querying the Classification Model**
  - Use of this feature requires preparation of untested opsin sequences that are aligned with the rest of the dataset during the MAFFT alignment step.
  - Remove the target opsins following the deepBreaks formatting step and place them in a separate text-file named 'unseendata.fasta'
  - Navigate to '*Step 3.5'* on the jupyter notebook provided and run the next two cells.
  - The output from these cells will be a file named '*classification_results.csv*', which provides the model's classification of those opsins and the confidence of each prediction.
- **Earth Mover's Distance (EMD)* -** I will eventually update this, but for now it's not entirely relevant to the larger pipeline. I would still recommend reading the glossary to understand what EMD is and how we can apply it to compare model performance.

Glossary

- **Opsins:** Opsins are the universal photoreceptor molecules of all visual systems in the animal kingdom. They can change their conformation from a resting state to a signaling

state upon light absorption, which activates the G protein, thereby resulting in a signaling cascade that produces physiological responses.
    - https://en.wikipedia.org/wiki/Opsin#Type_I_opsins
- **Lambda Max (λ_{max}):** The wavelength at which a substance has its strongest photon absorption; in this case it's with regards to the wavelength of light that a photoreceptive opsin protein displays the highest sensitivity to//photon absorption.
    - [This is done in a number of ways depending on the manner in which you are testing the gene, i.e. ***heterologous expression vs. MSP vs. ERG.***}
- **Heterologous Expression:** Refers to gene products that are obtained using a heterologous host, one that lacks the gene until it is artificially inserted. In this case a heterologous host will produce an opsin so that it can be studied.
- **Micro-spectrophotometry (MSP):** Absorbance is measured using a single photoreceptor cell (e.g. rod, cone, or rhabdome) by using spectrophotometry on a micron level scale.
- **Spectrophotometer:** An instrument that measures the amount of photons (the intensity of light) absorbed after it passes through sample solution; in this case it will measure the absorbance of an opsin protein as it is exposed to different wavelengths of light by a monochromator.
- **Monochromator:** The job of a monochromator is to produce a single spectral line from a broadband (multi-wavelength) source and is an optical instrument which measures the light spectrum. Light is focused in the input slit and diffracted by a grating. In this way, only one color is transmitted through the output slit at a given time. Spectra are then recorded wavelength by wavelength, rotating the grating.
- **Chromophore:** A chromophore is the part of a molecule responsible for its color.The color that is seen by our eyes is the one not absorbed by the reflecting object within a certain wavelength spectrum of visible light.
    - https://en.wikipedia.org/wiki/Chromophore
- **Accession Number:** This is a unique sequence of characters (Letters & Numbers) assigned by a particular database as an additional means of locating a specific article. Here it is in reference to the unique sequence assigned to a particular opsin gene when it is entered into the NCBI//GenBank database.
- **Foreign Key:** A field in a relational table//database that matches the primary key column of another table.
    - **Primary Key:** A key in a relational database that is unique for each record. It is a unique identifier, such as a driver license number, telephone number (including area code), or vehicle identification number (VIN).
    - For the purposes of this database, the ***refid*** is the primary key of the ***references*** sheet, and is used as a ***foreign key*** in the ***`heterologous`, `msp`, and `opsins`*** sheets.
- **Bovine (Cow) Rhodopsin:** The cow rhodopsin is used as the 'standard' comparative sequence when naming the positions of mutations during mutagenesis. This is why we align any sequence we mutate to the bovine sequence, such that this numbering convention remains consistent. It's also considered to be the 'basal' root in any genetic analysis of vertebrate opsin sequences?

- **Spectral Tuning Sites:** Positions on the amino acid sequence which are known to cause shifts in the spectral sensitivity (thus lambda max) of an opsin when mutated.
    - Candidate Spectral Tuning Sites (CSTS) = Those sites which are predicted to be spectral tuning sites by the deepBreaks ML models but have not yet been functionally tested to validate.
- **FASTA (format):** In bioinformatics and biochemistry, the FASTA format is a text-based format for representing either nucleotide sequences or amino acid sequences, in which nucleotides or amino acids are represented using single-letter codes. The format allows for sequence names and comments to precede the sequences.
    - Credit: https://en.wikipedia.org/wiki/FASTA_format
    - Ex. Aligned Bovine Rhodopsin



```
>Bovine
--------------------------------MNGTE--------GPNFY----VPFS--
--NKTGVVRSPFEAP-QY------------YLAEPWQ---------FSMLAAYMFLLIML
GFPINFLTLYVTVQHKKLRTPLNYILLNLAVADLFMVFGGFTTTLYTSLHGYFVFGPTG-
CNLEGFFATLGGEIALWSLVVLAIERYVVVCKPMSNFR-FGENHAIMGVAFTWVMALACA
APPLVG-WSRYIPEGMQCSCGIDYYTPHEETNNESFVIYMFVVHFIIPLIVIFFCYGQLV
FTVKEAAAQQQE--------------SATTQKAEKEVTRMVIIMVIAFLICWLPYAGVAF
YIFTHQGSDFGPIFMTIPAFFAKTSAVYNPVIYIMMNKQFRNCMVTTLCCGKNPLG-DDE
-ASTTVSKTET------------------SQVAPA-----------------
```

- **Classical Machine Learning Models:** the process of building algorithms that can learn from existing observations (or data sets), and leverage that learning to predict new observations, or determine the output of new input. Machine learning uses algorithms to parse data, learn from that data, and make informed decisions based on what it has learned. Deep learning structures algorithms in layers to create an "artificial neural network" that can learn and make intelligent decisions on its own. Deep learning is a subset of machine learning.
    - To learn more about ML in general and the difference between deep learning and classical machine learning algorithms visit these website links.
      →https://towardsdatascience.com/deep-learning-vs-classical-machine-learning-9a42c6d48aa
      →https://www.zendesk.com/blog/machine-learning-and-deep-learning/#:~:text=Machine%20learning%20uses%20algorithms%20to,a%20subset%20of%20machine%20learning.
      →https://www.simplilearn.com/10-algorithms-machine-learning-engineers-need-to-know-article
- **Confusion Matrix:** A table that is used to define the performance of a classification algorithm and shows how many predictions are correct and incorrect per class. It helps in understanding what classes that the model is confusing as other classes. All the diagonal elements denote correctly classified outcomes. The misclassified outcomes are represented on the off diagonals of the confusion matrix. Hence, the best classifier will have a confusion matrix with only diagonal elements and the rest of the elements set to

zero. A confusion matrix generates actual values and predicted values after the classification process.

- Here's an assortment of reading's that pertain to confusion matrices and their role in ML.
  - https://www-sciencedirect-com.proxy.library.ucsb.edu:9443/topics/engineering/confusion-matrix#:~:text=A%20confusion%20matrix%20is%20a,performance%20of%20a%20classification%20algorithm.

- This one specifically cover's the metrics that can be calculated from a confusion matrix, including **F1, precision, recall, and accuracy*.**
  - https://www-sciencedirect-com.proxy.library.ucsb.edu:9443/science/article/pii/B9780128212295000033
- **Earth Mover's Distance (EMD)**: a measure of the distance between two probability distributions over a region, proportional to the minimum amount of work required to change one distribution into the other. Here one unit of work is defined as the amount of work necessary to move one unit of weight by one unit of distance. We can compare the performance of two models by the difference of their EMD values (so lower EMD = better performance), which is especially useful when working with a system in which class boundaries are relatively arbitrary. [Credit: https://en.wikipedia.org/wiki/Earth_mover%27s_distance]
  - **I.e** - The sequence difference between an opsin with a lambda max of 418, in class 0 and an opsin with a lambda max of 422 in class 1 are miniscule; especially when considering the boundary between these classes is arbitrary. EMD can account for this fact by assigning a lower weight to the misclassification of an opsin in a closely related class, while increasing the 'weight' and 'work' of more egregious misclassifications (i.e - misclassifying an opsin from class 0 as class 9).
  - Here's a link to read more about EMD → https://www.ibm.com/blogs/research/2019/07/earth-movers-distance/
- **Area Under the Curve (AUC)* -** Provides an aggregate measure of performance across all possible classification thresholds and represents the probability that a random positive example is positioned to the right of a random negative example. AUC ranges in value from 0 to 1. A model whose predictions are 100% wrong has an AUC of 0.0; one whose predictions are 100% correct has an AUC of 1.0.
  - Credit + Additional reading on this ML "crash-course" website. → https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc#:~:text=AUC%20represents%20the%20probability%20that,has%20an%20AUC%20of%201.0.

The next 8 definitions are credited to…
https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative

- **True Positive [TP]:** an outcome where the model *correctly* predicts the *positive* class.
- **True Negative [TN]:** an outcome where the model *correctly* predicts the *negative* class.
- **False Positive [FP]:** an outcome where the model *incorrectly* predicts the *positive* class.
- **False Negative [FN]:** an outcome where the model *incorrectly* predicts the *negative* class.
- **Accuracy:** The total number of **correct predictions / Total Number of Predictions**
    - OR **(TP+TN / TP + TN + FP + FN)**
- **Precision:** Proportion of positive identifications that were actually correct.
    - **Precision = TP / TP + FP**
- **Recall:** Proportion of actual positives (so ground truth positives) were actually predicted correctly.
    - **Recall = TP / TP + FN**
- **F1 [Score]:** states the equilibrium between the precision and the recall.
    - **F1 = 2*precision*recall/precision + recall**

IMPORTANT LINKS

Since manual data entry is currently done using a Google Spreadsheet, the following link is important to have!
→ Public Copy of VisualPhysiologyDB

Here's a link to a visual flowchart that can accompany this document -
Copy of VisualPhysiologyDB_flowchart

And another link for breaking down the *"Anatomy of an Opsin Research Paper"*
→ Anatomy of a resarch paper (1).pdf

Plus a PowerPoint presentation w/ an accompanying video that really breaks down the step-by-step process for you…
→ **Collecting Opsin Data from a Journal.pptx**
→https://drive.google.com/file/d/1wEZu6pV0HqEDidv8qe2Jqpot25eyeNZg/view?usp=sharing

IGNORE EVERYTHING BELOW FOR NOW:

Key Phrases: [For assisting in literature searches]

Heterologous

- General
- Mutants
- Chimeras

MSP

- General