01_find_rhodopsins_blast

September 2, 2025

0.0.1 Step 1: Retrieve homologs of known Proton pumping microbial rhodopsins

Gene name	Species	max (nm)	GenBank protein accession	Source reference
Y_CyRII_P7104R	Nodosilinea nodulosa PCC 7104	570	WP_017301364.1	Hasegawa-Takano et al. 2024 (ISME J)
BR (Bacteriorhodopsin)	Halobacterium salinarum	570	WP_136361479.1	Oesterhelt & Stoeckenius 1971; Lozier et al. 1975
N2098R	Calothrix sp. NIES-2098	550	BAY09002.1	Hasegawa et al. 2020 (Sci Rep)
GR (Gloeobacter rhodopsin)	Gloeobacter violaceus PCC 7421	544	WP_011140202.1	Choi et al. 2014 (PLoS ONE)
PR (GPR, green proteorhodopsin)	Uncultivated marine -proteobacterium (SAR86; EBAC31A08)	520	AAG10475.1	Béjà et al. 2000 (Science)
PR (BPR, blue proteorhodopsin)	Marine bacterioplankton (HOT75m4 variant)	490	Q9AFF7.2	Béjà et al. 2001 (Nature)

Create a table from data above from Hasegawa-Takano et al 2024. Find similar sequences to those, totaling 90 sequences.

[1]:
```python
from pathlib import Path

#Define paths for current project
# --- Centralized paths ---
ROOT = Path("..")
DATA = ROOT / "data"
LOGS = ROOT / "logs"
SCRIPTS = ROOT / "scripts"
RESULTS = ROOT / "results"
ALIGN_DIR = RESULTS / "align"
TREE_DIR = RESULTS / "trees"
FIGURES = RESULTS / "figures"
```

```
[ ]: #Using table above as bait, search Uniref and download clost blast hits of␣
      ↪Uniref90

     #This takes about 50 minutes for a run from scratch

     #job_id = "ncbiblast-I20250901-173114-0788-19601246-p1m"
     #!python {SCRIPTS / "blast_uniref90.py"} dummy dummy --max_hits 33 --out {DATA /
      ↪ "WP_011140202.1_top33_uniref90.fasta"} --log {LOGS / "WP_011140202.
      ↪1_blast_runs.log"} --jobid {job_id} --force

     from pathlib import Path

     # Table of accessions and max_hits. Aiming for 15,15,15,15,30 .. but need to␣
      ↪add a few to account for
     # later deletions of incomplete sequences

     bait_table = [
         ("YCyR2hit", "WP_017301364.1", 15),
         ("BRhit", "WP_136361479.1", 16),
         ("GCyR2hit", "BAY09002.1", 18),
         ("GRhit", "WP_011140202.1", 32),
         #("GPRhit", "AAG10475.1", 10),
         #("BPRhit", "Q9AFF7.2", 10), #BPR and GPR mostly hitting the same genes,␣
      ↪just do PR instead
         ("PRhit", "AAG10475.1", 15),
     ]

     #run_table = [
     #    ("BRhit", "WP_136361479.1",␣
      ↪"ncbiblast-R20250902-012715-0963-64586178-p1m", 17),
     #    ("GRhit", "WP_011140202.1",␣
      ↪"ncbiblast-R20250902-014018-0359-36391454-p1m", 32),
     #    ("GCyR2hit", "BAY09002.1", "ncbiblast-R20250902-013359-0049-40227659-p1m",␣
      ↪18),
     #]

     email = "oakley@ucsb.edu"

     #for prefix, accession, job_id, max_hits in run_table:
     #    out_fasta = DATA / f"{accession}_top{max_hits}_uniref90.fasta"
     #    log_file = LOGS / f"{accession}_blast_runs.log"
     #    print(f"Fetching BLAST results for {prefix} (job_id={job_id},␣
      ↪max_hits={max_hits})")
     #    !python {SCRIPTS / "blast_uniref90.py"} --max_hits {max_hits} --out␣
      ↪{out_fasta} --log {log_file} --jobid {job_id}
```

```
for prefix, accession, max_hits in bait_table:
    out_fasta = DATA / f"{accession}_top{max_hits}_uniref90.fasta"
    log_file = LOGS / f"{accession}_blast_runs.log"
    print(f"Running BLAST for {accession} (max_hits={max_hits})")
    !python {SCRIPTS / "blast_uniref90.py"} {accession} {email} --max_hits␣
 ↪{max_hits} --out {out_fasta} --log {log_file}
```

[3]:
```
# Rename the sequences with a common prefix to keep track of them during␣
 ↪analyses

#Need bait_table from above
renamed_files = []

for prefix, accession, max_hits in bait_table:
    blast_out = DATA / f"{accession}_top{max_hits}_uniref90.fasta"
    renamed_fasta = DATA / f"{accession}_top{max_hits}_renamed.fasta"
    renamed_files.append(renamed_fasta)
    print(f"Renaming {blast_out} -> {renamed_fasta} with prefix {prefix}")
    !python {SCRIPTS / "rename_fasta_headers.py"} {blast_out} {renamed_fasta}␣
 ↪--prefix {prefix} --force

# Concatenate all renamed FASTA files into one
combined_fasta = DATA / "pumphits.fasta"
with open(combined_fasta, "w") as outfile:
    for fname in renamed_files:
        with open(fname) as infile:
            outfile.write(infile.read())
print(f"Combined all renamed FASTA files into {combined_fasta}")
```

```
Renaming ../data/WP_017301364.1_top15_uniref90.fasta ->
../data/WP_017301364.1_top15_renamed.fasta with prefix YCyR2hit
Reformatted 15 headers → ../data/WP_017301364.1_top15_renamed.fasta
Renaming ../data/WP_136361479.1_top16_uniref90.fasta ->
../data/WP_136361479.1_top16_renamed.fasta with prefix BRhit
Reformatted 15 headers → ../data/WP_136361479.1_top16_renamed.fasta
Renaming ../data/BAY09002.1_top18_uniref90.fasta ->
../data/BAY09002.1_top18_renamed.fasta with prefix GCyR2hit
Reformatted 16 headers → ../data/BAY09002.1_top18_renamed.fasta
Renaming ../data/WP_011140202.1_top32_uniref90.fasta ->
../data/WP_011140202.1_top32_renamed.fasta with prefix GRhit
Reformatted 32 headers → ../data/WP_011140202.1_top32_renamed.fasta
Renaming ../data/AAG10475.1_top15_uniref90.fasta ->
../data/AAG10475.1_top15_renamed.fasta with prefix PRhit
Reformatted 15 headers → ../data/AAG10475.1_top15_renamed.fasta
Combined all renamed FASTA files into ../data/pumphits.fasta
```

[4]:
```
# A well known red-shifted BR is called D85A as far as I can tell, that doesn't␣
↪occur in the 85th
# aa but rather at about 97. Changing that here. See for example Karayuma␣
↪database for ML, the
# last sequence is this one

from Bio import Entrez, SeqIO, SeqRecord, Seq

Entrez.email = "oakley@ucsb.edu"
accession = "WP_136361479.1"

# Fetch sequence from GenBank
handle = Entrez.efetch(db="protein", id=accession, rettype="fasta",␣
↪retmode="text")
record = SeqIO.read(handle, "fasta")
handle.close()

seq_list = list(str(record.seq))
motif = "YADWL"
motif_pos = str(record.seq).find(motif)
if motif_pos != -1:
    d_index = motif_pos + 2
    if seq_list[d_index] == "D":
        seq_list[d_index] = "A"
        print(f"Mutated D at position {d_index+1} (YADWL motif) to A.")
    else:
        print(f"Warning: Central residue of YADWL is not D, found␣
↪{seq_list[d_index]}")
else:
    print("YADWL motif not found in sequence.")

mutated_seq = "".join(seq_list)
mutated_id = "BRmutant_" + accession + "_D85A"
mutated_record = SeqRecord.SeqRecord(Seq.Seq(mutated_seq), id=mutated_id,␣
↪description="")

output_path = DATA / f"{mutated_id}.fasta"
with open(output_path, "w") as out_handle:
    SeqIO.write(mutated_record, out_handle, "fasta")
print(f"Mutated sequence written to {output_path}")

# Concatenate mutated sequence to pumphits.fasta and write to pumphitsM.fasta
pumphits_path = DATA / "pumphits.fasta"
mutant_path = DATA / "pumphitsM.fasta"

with open(mutant_path, "w") as out_handle:
    with open(pumphits_path) as in_handle:
        out_handle.write(in_handle.read())
    SeqIO.write(mutated_record, out_handle, "fasta")

print(f"Mutant FASTA written to {mutant_path}")
```

```
Mutated D at position 99 (YADWL motif) to A.
Mutated sequence written to ../data/BRmutant_WP_136361479.1_D85A.fasta
Mutant FASTA written to ../data/pumphitsM.fasta
```

[5]:
```
#Delete some sequences that are partial

from pathlib import Path
from Bio import SeqIO

# Input FASTA to filter (update this as needed)
input_fasta = DATA / "pumphitsM.fasta"  # <-- This is your starting file
CULLED_FASTA = DATA / "pumphits_culled.fasta"  # Output after removing sequences

# List of sequence IDs to remove (exact matches) because they are truncated and␣
 ↪probably incomplete
remove_ids = [
    "BRhit__Halobacterium_salinarum__UniRef90_UPI0000110B77",
    "GRhit__Chamaesiphon_sp__UniRef90_UPI0035935AE6",
   ␣
 ↪"GCyR2hit__Pseudanabaenaceae_cyanobacterium_LEGE_13415__UniRef90_A0A928YZN9",
    "GRhit__Chamaesiphon_sp__UniRef90_UPI0035946169",
]

def normalize_id(s):
    # Remove quotes, whitespace, tabs, and collapse all spaces
    return "".join(s.strip().strip('"').strip("'").split())

remove_ids_normalized = set(normalize_id(x) for x in remove_ids)

# Remove listed sequences and write culled FASTA
records = []
removed_count = 0
with open(input_fasta) as in_handle:
    for record in SeqIO.parse(in_handle, "fasta"):
        rec_id_norm = normalize_id(record.id)
        if rec_id_norm not in remove_ids_normalized:
            records.append(record)
        else:
            print(f"Removed: {record.id!r} (normalized: {rec_id_norm!r})")
            removed_count += 1

with open(CULLED_FASTA, "w") as out_handle:
    SeqIO.write(records, out_handle, "fasta")

print(f"Filtered FASTA written to {CULLED_FASTA}")
print(f"Total sequences removed: {removed_count}")
print(f"Total unique sequences retained: {len(records)}")

# Align with MAFFT
ALN_FASTA = ALIGN_DIR / "pumphits_ALN.fasta"
!mafft --auto --thread -1 --quiet "{CULLED_FASTA}" > "{ALN_FASTA}"
print("Aligned ->", ALN_FASTA)

# Build a maximum likelihood tree with IQ-TREE2
IQ_PREFIX = TREE_DIR / "pumphits_iqtree"
#!iqtree2 -s "{ALN_FASTA}" -m LG+F+R4 -nt AUTO -keep-ident -pre "{IQ_PREFIX}"␣
 ↪-quiet
```

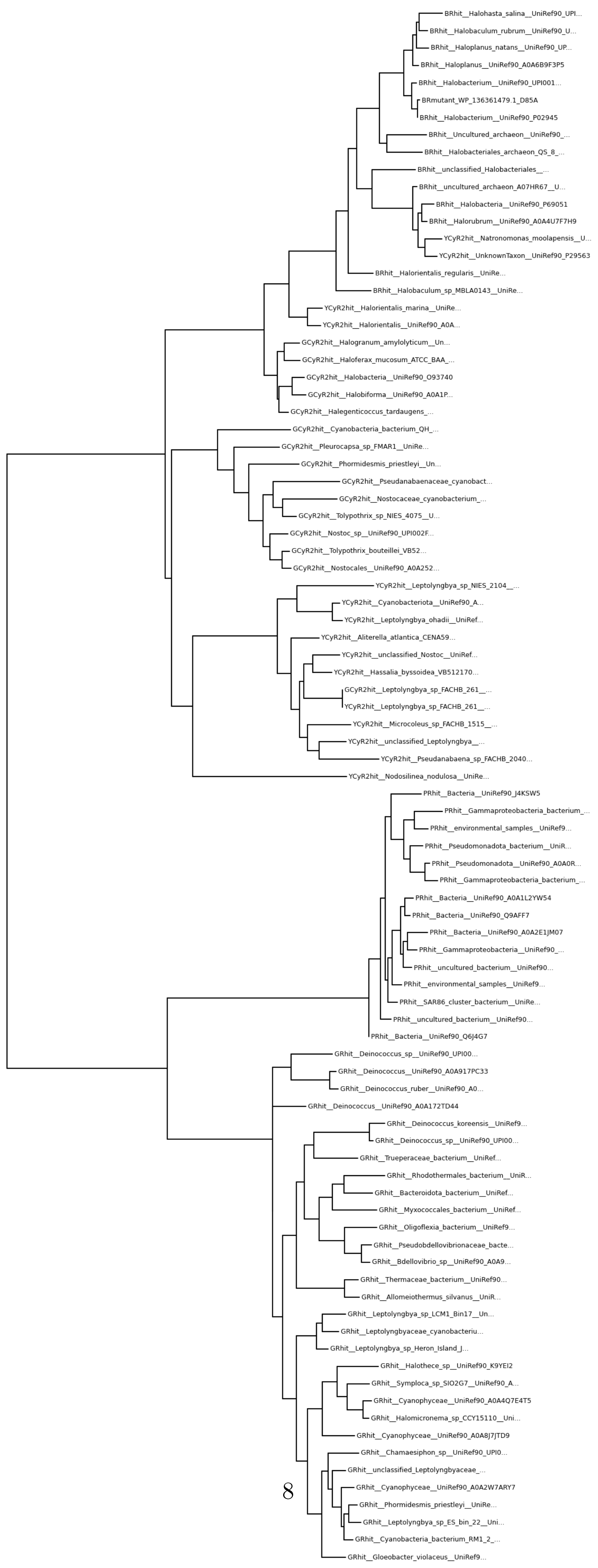
```
Removed: 'BRhit__Halobacterium_salinarum__UniRef90_UPI0000110B77' (normalized:
'BRhit__Halobacterium_salinarum__UniRef90_UPI0000110B77')
Removed:
'GCyR2hit__Pseudanabaenaceae_cyanobacterium_LEGE_13415__UniRef90_A0A928YZN9'
(normalized:
'GCyR2hit__Pseudanabaenaceae_cyanobacterium_LEGE_13415__UniRef90_A0A928YZN9')
Removed: 'GRhit__Chamaesiphon_sp__UniRef90_UPI0035946169' (normalized:
'GRhit__Chamaesiphon_sp__UniRef90_UPI0035946169')
Removed: 'GRhit__Chamaesiphon_sp__UniRef90_UPI0035935AE6' (normalized:
'GRhit__Chamaesiphon_sp__UniRef90_UPI0035935AE6')
Filtered FASTA written to ../data/pumphits_culled.fasta
Total sequences removed: 4
Total unique sequences retained: 90
Aligned -> ../results/align/pumphits_ALN.fasta
```

[8]:
```
# Count the number of times each bait prefix appears in the aligned FASTA
# Just checking final numbers
from Bio import SeqIO

bait_prefixes = [x[0] for x in bait_table] + ["BRmutant"]
counts = {prefix: 0 for prefix in bait_prefixes}

with open(ALN_FASTA) as handle:
    for record in SeqIO.parse(handle, "fasta"):
        for prefix in bait_prefixes:
            if record.id.startswith(prefix):
                counts[prefix] += 1
total = sum(counts.values())
for prefix, count in counts.items():
    print(f"{prefix}: {count}")
print("Total:", total)
```

```
YCyR2hit: 15
BRhit: 14
GCyR2hit: 15
GRhit: 30
PRhit: 15
BRmutant: 1
Total: 90
```

[25]:
```
#Check the phylogeny for proton pump hits
import sys
sys.path.append(str(SCRIPTS))
from plot_tree import plot_tree

# Set IQ_PREFIX so output files go into TREE_DIR
IQ_PREFIX = TREE_DIR / "pumphits_iqtree"

# IQ-TREE2 command (in shell, use -pre "{IQ_PREFIX}" to ensure output goes to␣
 ↪TREE_DIR)
!iqtree2 -s "{ALN_FASTA}" -m LG+F+R4 -nt AUTO -keep-ident -pre "{IQ_PREFIX}"␣
 ↪-quiet -redo

# In Python, use IQ_PREFIX for downstream file paths
tree_file = IQ_PREFIX.with_suffix(".treefile")
fig_path = FIGURES / f"{IQ_PREFIX.name}_tree.pdf"

plot_tree(tree_file, save_path=fig_path, figsize=(12, 30))
```

```
Saved tree figure to: ../results/figures/pumphits_iqtree_tree.pdf
```


BRhit__Halohasta_salina__UniRef90_UPI...
BRhit__Halobaculum_rubrum__UniRef90_U...
BRhit__Haloplanus_natans__UniRef90_UP...
BRhit__Haloplanus__UniRef90_A0A6B9F3P5
BRhit__Halobacterium__UniRef90_UPI001...
BRmutant_WP_136361479.1_D85A
BRhit__Halobacterium__UniRef90_P02945
BRhit__Uncultured_archaeon__UniRef90_...
BRhit__Halobacteriales_archaeon_QS_8_...
BRhit__unclassified_Halobacteriales__...
BRhit__uncultured_archaeon_A07HR67__U...
BRhit__Halobacteria__UniRef90_P69051
BRhit__Halorubrum__UniRef90_A0A4U7F7H9
YCyR2hit__Natronomonas_moolapensis__U...
YCyR2hit__UnknownTaxon__UniRef90_P29563
BRhit__Halorientalis_regularis__UniRe...
BRhit__Halobaculum_sp_MBLA0143__UniRe...
YCyR2hit__Halorientalis_marina__UniRe...
YCyR2hit__Halorientalis__UniRef90_A0A...
GCyR2hit__Halogranum_amylolyticum__Un...
GCyR2hit__Haloferax_mucosum_ATCC_BAA_...
GCyR2hit__Halobacteria__UniRef90_O93740
GCyR2hit__Halobiforma__UniRef90_A0A1P...
GCyR2hit__Halegenticoccus_tardaugens_...
GCyR2hit__Cyanobacteria_bacterium_QH_...
GCyR2hit__Pleurocapsa_sp_FMAR1__UniRe...
GCyR2hit__Phormidesmis_priestleyi__Un...
GCyR2hit__Pseudanabaenaceae_cyanobact...
GCyR2hit__Nostocaceae_cyanobacterium_...
GCyR2hit__Tolypothrix_sp_NIES_4075__U...
GCyR2hit__Nostoc_sp__UniRef90_UPI002F...
GCyR2hit__Tolypothrix_bouteillei_VB52...
GCyR2hit__Nostocales__UniRef90_A0A252...
YCyR2hit__Leptolyngbya_sp_NIES_2104__...
YCyR2hit__Cyanobacteriota__UniRef90_A...
YCyR2hit__Leptolyngbya_ohadii__UniRef...
YCyR2hit__Aliterella_atlantica_CENA59...
YCyR2hit__unclassified_Nostoc__UniRef...
YCyR2hit__Hassalia_byssoidea_VB512170...
GCyR2hit__Leptolyngbya_sp_FACHB_261__...
YCyR2hit__Leptolyngbya_sp_FACHB_261__...
YCyR2hit__Microcoleus_sp_FACHB_1515__...
YCyR2hit__unclassified_Leptolyngbya__...
YCyR2hit__Pseudanabaena_sp_FACHB_2040...
YCyR2hit__Nodosilinea_nodulosa__UniRe...
PRhit__Bacteria__UniRef90_J4KSW5
PRhit__Gammaproteobacteria_bacterium_...
PRhit__environmental_samples__UniRef9...
PRhit__Pseudomonadota_bacterium__UniR...
PRhit__Pseudomonadota__UniRef90_A0A0R...
PRhit__Gammaproteobacteria_bacterium_...
PRhit__Bacteria__UniRef90_A0A1L2YW54
PRhit__Bacteria__UniRef90_Q9AFF7
PRhit__Bacteria__UniRef90_A0A2E1JM07
PRhit__Gammaproteobacteria__UniRef90_...
PRhit__uncultured_bacterium__UniRef90...
PRhit__environmental_samples__UniRef9...
PRhit__SAR86_cluster_bacterium__UniRe...
PRhit__uncultured_bacterium__UniRef90...
PRhit__Bacteria__UniRef90_Q6J4G7
GRhit__Deinococcus_sp__UniRef90_UPI00...
GRhit__Deinococcus__UniRef90_A0A917PC33
GRhit__Deinococcus_ruber__UniRef90_A0...
GRhit__Deinococcus__UniRef90_A0A172TD44
GRhit__Deinococcus_koreensis__UniRef9...
GRhit__Deinococcus_sp__UniRef90_UPI00...
GRhit__Trueperaceae_bacterium__UniRef...
GRhit__Rhodothermales_bacterium__UniR...
GRhit__Bacteroidota_bacterium__UniRef...
GRhit__Myxococcales_bacterium__UniRef...
GRhit__Oligoflexia_bacterium__UniRef9...
GRhit__Pseudobdellovibrionaceae_bacte...
GRhit__Bdellovibrio_sp__UniRef90_A0A9...
GRhit__Thermaceae_bacterium__UniRef90...
GRhit__Allomeiothermus_silvanus__UniR...
GRhit__Leptolyngbya_sp_LCM1_Bin17__Un...
GRhit__Leptolyngbyaceae_cyanobacteriu...
GRhit__Leptolyngbya_sp_Heron_Island_J...
GRhit__Halothece_sp__UniRef90_K9YEI2
GRhit__Symploca_sp_SIO2G7__UniRef90_A...
GRhit__Cyanophyceae__UniRef90_A0A4Q7E4T5
GRhit__Halomicronema_sp_CCY15110__Uni...
GRhit__Cyanophyceae__UniRef90_A0A8J7JTD9
GRhit__Chamaesiphon_sp__UniRef90_UPI0...
GRhit__unclassified_Leptolyngbyaceae_...
GRhit__Cyanophyceae__UniRef90_A0A2W7ARY7
GRhit__Phormidesmis_priestleyi__UniRe...
GRhit__Leptolyngbya_sp_ES_bin_22__Uni...
GRhit__Cyanobacteria_bacterium_RM1_2_...
GRhit__Gloeobacter_violaceus__UniRef9...

0.0.2 Step 2: Retrieve diverse microbial opsins from study by Hasegawa-Katano 24

[27]:
```python
import pandas as pd
from pathlib import Path
from Bio import Entrez

Entrez.email = "oakley@ucsb.edu"

csv_path = DATA / "hasegawa24" / "Supporting_Data_3.csv"
df = pd.read_csv(csv_path)
accession_col = "rhodopsin_accessions"  # Adjust if needed

output_fasta = DATA / "hasegawa24" / "rhodopsins_from_accessions.fasta"

def parse_accession(entry):
    entry = entry.strip().strip('"').strip("'")
    # Example: "BAC88139.1 (XLR)"
    if "(" in entry and ")" in entry:
        acc = entry.split("(")[0].strip()
        clade = entry.split("(")[1].split(")")[0].strip()
        return acc, clade
    else:
        return entry, ""

# Flatten and parse accessions
all_entries = (
    df[accession_col]
    .dropna()
    .apply(lambda x: [a.strip().strip('"').strip("'") for a in str(x).
 ↪split(",")])
    .explode()
    .dropna()
)

parsed = [parse_accession(e) for e in all_entries if e]

with open(output_fasta, "w") as out_handle:
    for acc, clade in parsed:
        try:
            handle = Entrez.efetch(db="protein", id=acc, rettype="fasta",␣
 ↪retmode="text")
            seq = handle.read()
            if seq.strip() and seq.startswith(">"):
                # Add clade to FASTA header with underscore instead of space
                lines = seq.splitlines()
                if clade and lines:
                    lines[0] = f">{clade}_{lines[0][1:]}"
                    seq = "\n".join(lines)
                out_handle.write(seq + "\n")
            else:
                print(f"Did not find {acc}")
        except Exception:
            print(f"Did not find {acc}")

print(f"Sequences written to {output_fasta}")
```

```
Did not find SRR6869043_N0001541_6
Did not find SRR6869043_N0010062_2
Did not find SRR6869040_N0001326_5
Did not find SRR6869040_N0001714_12
Sequences written to ../data/hasegawa24/rhodopsins_from_accessions.fasta
```

[28]:
```
#Combine with proton pump hits from above and remove a few truncated sequences

from pathlib import Path
from Bio import SeqIO

# Define input files
fasta1 = DATA / "pumphits_culled.fasta"  # Output after removing sequences
fasta2 = DATA / "hasegawa24" / "rhodopsins_from_accessions.fasta"
combined_fasta = DATA / "combined_plus_hasegawa24.fasta"

# Combine the two FASTA files
with open(combined_fasta, "w") as outfile:
    for fname in [fasta1, fasta2]:
        with open(fname) as infile:
            outfile.write(infile.read())

print(f"Combined FASTA written to {combined_fasta}")

# File to filter
input_fasta = DATA / "combined_plus_hasegawa24.fasta"
output_fasta = DATA / "combined_plus_hasegawa24_culled.fasta"

# List of sequence IDs to remove (exact matches)
remove_ids = [
    "BRhit__Halobacterium_salinarum__UniRef90_UPI0000110B77",
    "GRhit__Chamaesiphon_sp__UniRef90_UPI0035935AE6",
   ␣
 ↪"GCyR2hit__Pseudanabaenaceae_cyanobacterium_LEGE_13415__UniRef90_A0A928YZN9",
    "GRhit__Chamaesiphon_sp__UniRef90_UPI0035946169",
```

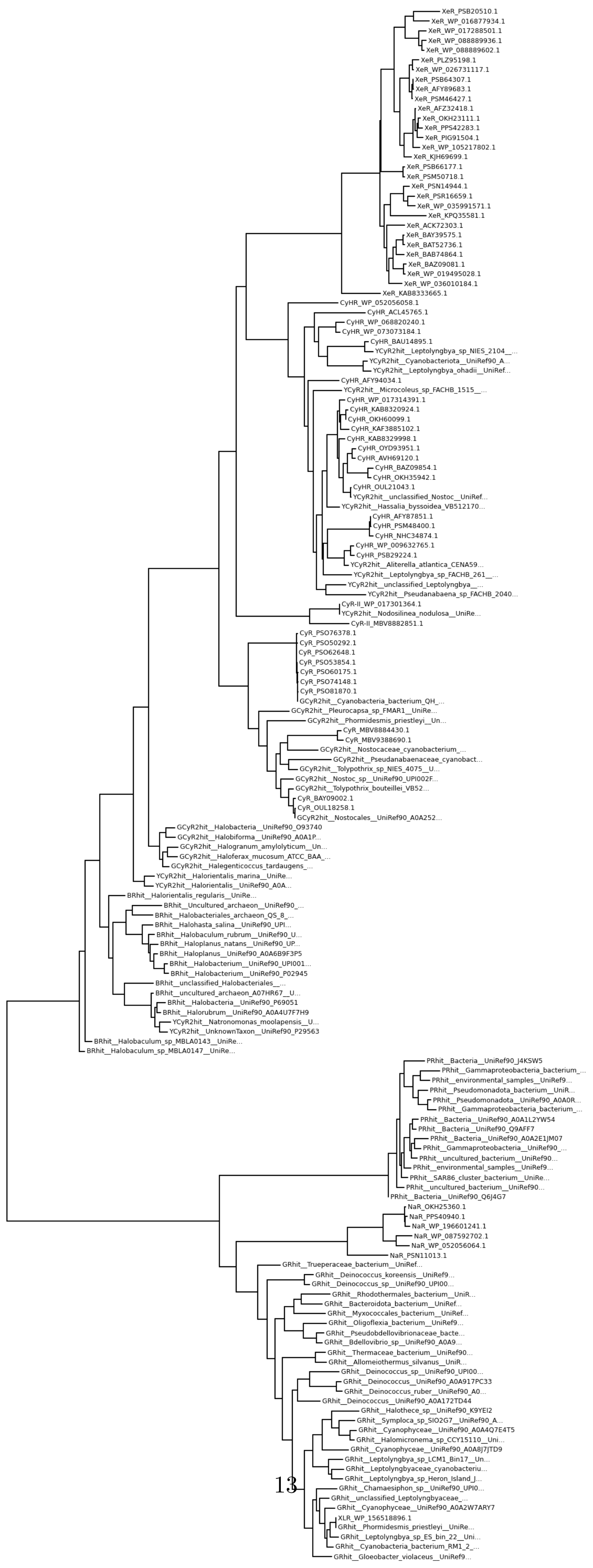
```python
    "XeR_AFY92621.1",
    "CyR-II_MBV9385464.1",
    #Next is missing conserved lysine
    "XeR_ACL43260.1",
]

def normalize_id(s):
    # Remove quotes, whitespace, tabs, and collapse all spaces
    return "".join(s.strip().strip('"').strip("'").split())

# Normalize remove_ids for robust matching
remove_ids_normalized = set(normalize_id(x) for x in remove_ids)

# Remove listed sequences and deduplicate by sequence
records = []
removed_count = 0
with open(input_fasta) as in_handle:
    for record in SeqIO.parse(in_handle, "fasta"):
        rec_id_norm = normalize_id(record.id)
        if rec_id_norm not in remove_ids_normalized:
            records.append(record)
        else:
            print(f"Removed: {record.id!r} (normalized: {rec_id_norm!r})")
            removed_count += 1

# Deduplicate by sequence (keep first occurrence)
seq_seen = set()
deduped_records = []
for rec in records:
    seq_str = str(rec.seq)
    if seq_str not in seq_seen:
        deduped_records.append(rec)
        seq_seen.add(seq_str)

with open(output_fasta, "w") as out_handle:
    SeqIO.write(deduped_records, out_handle, "fasta")

print(f"Filtered and deduplicated FASTA written to {output_fasta}")
print(f"Total sequences removed: {removed_count}")
print(f"Total unique sequences retained: {len(deduped_records)}")

# Align with MAFFT
ALN_FASTA = ALIGN_DIR / "combined_plus_hasegawa24_ALN.fasta"
!mafft --auto --thread -1 --quiet "{output_fasta}" > "{ALN_FASTA}"
print("Aligned ->", ALN_FASTA)
```

```
Combined FASTA written to ../data/combined_plus_hasegawa24.fasta
```

```
Removed: 'XeR_ACL43260.1' (normalized: 'XeR_ACL43260.1')
Removed: 'XeR_AFY92621.1' (normalized: 'XeR_AFY92621.1')
Removed: 'CyR-II_MBV9385464.1' (normalized: 'CyR-II_MBV9385464.1')
Filtered and deduplicated FASTA written to
../data/combined_plus_hasegawa24_culled.fasta
Total sequences removed: 3
Total unique sequences retained: 160
Aligned -> ../results/align/combined_plus_hasegawa24_ALN.fasta
```

[ ]:
```python
#Check the phylogeny for proton pump hits
import sys
sys.path.append(str(SCRIPTS))
from plot_tree import plot_tree

# Set IQ_PREFIX so output files go into TREE_DIR
IQ_PREFIX = TREE_DIR / "combined_plus_hasegawa24_iqtree"

#mafft run in previous cell
ALN_FASTA = ALIGN_DIR / "combined_plus_hasegawa24_ALN.fasta"

# IQ-TREE2 command (in shell, use -pre "{IQ_PREFIX}" to ensure output goes to␣
↪TREE_DIR)
!iqtree2 -s "{ALN_FASTA}" -m LG+F+R4 -nt AUTO -keep-ident -pre "{IQ_PREFIX}"␣
↪-quiet

# In Python, use IQ_PREFIX for downstream file paths
tree_file = IQ_PREFIX.with_suffix(".treefile")
fig_path = FIGURES / f"{IQ_PREFIX.name}_tree.pdf"

plot_tree(tree_file, save_path=fig_path, figsize=(12, 30))
```

```
Checkpoint (../results/trees/combined_plus_hasegawa24_iqtree.ckp.gz) indicates
that a previous run successfully finished
Use `-redo` option if you really want to redo the analysis and overwrite all
output files.
Use `--redo-tree` option if you want to restore ModelFinder and only redo tree
search.
Use `--undo` option if you want to continue previous run when changing/adding
options.
Saved tree figure to:
../results/figures/combined_plus_hasegawa24_iqtree_tree.pdf
```



XeR_PSB20510.1
XeR_WP_016877934.1
XeR_WP_017288501.1
XeR_WP_088889936.1
XeR_WP_088889602.1
XeR_PLZ95198.1
XeR_WP_026731117.1
XeR_PSB64307.1
XeR_AFY89683.1
XeR_PSM46427.1
XeR_AFZ32418.1
XeR_OKH23111.1
XeR_PPS42283.1
XeR_PIG91504.1
XeR_WP_105217802.1
XeR_KJH69699.1
XeR_PSB66177.1
XeR_PSM50718.1
XeR_PSN14944.1
XeR_PSR16659.1
XeR_WP_035991571.1
XeR_KPQ35581.1
XeR_ACK72303.1
XeR_BAY39575.1
XeR_BAT52736.1
XeR_BAB74864.1
XeR_BAZ09081.1
XeR_WP_019495028.1
XeR_WP_036010184.1
XeR_KAB8333665.1
CyHR_WP_052056058.1
CyHR_ACL45765.1
CyHR_WP_068820240.1
CyHR_WP_073073184.1
CyHR_BAU14895.1
YCyR2hit__Leptolyngbya_sp_NIES_2104__...
YCyR2hit__Cyanobacteriota__UniRef90_A...
YCyR2hit__Leptolyngbya_ohadii__UniRef...
CyHR_AFY94034.1
YCyR2hit__Microcoleus_sp_FACHB_1515__...
CyHR_WP_017314391.1
CyHR_KAB8320924.1
CyHR_OKH60099.1
CyHR_KAF3885102.1
CyHR_KAB8329998.1
CyHR_OYD93951.1
CyHR_AVH69120.1
CyHR_BAZ09854.1
CyHR_OKH35942.1
CyHR_OUL21043.1
YCyR2hit__unclassified_Nostoc__UniRef...
YCyR2hit__Hassalia_byssoidea_VB512170...
CyHR_AFY87851.1
CyHR_PSM48400.1
CyHR_NHC34874.1
CyHR_WP_009632765.1
CyHR_PSB29224.1
YCyR2hit__Aliterella_atlantica_CENA59...
YCyR2hit__Leptolyngbya_sp_FACHB_261__...
YCyR2hit__unclassified_Leptolyngbya__...
YCyR2hit__Pseudanabaena_sp_FACHB_2040...
CyR-II_WP_017301364.1
YCyR2hit__Nodosilinea_nodulosa__UniRe...
CyR-II_MBV8882851.1
CyR_PSO76378.1
CyR_PSO50292.1
CyR_PSO62648.1
CyR_PSO53854.1
CyR_PSO60175.1
CyR_PSO74148.1
CyR_PSO81870.1
GCyR2hit__Cyanobacteria_bacterium_QH_...
GCyR2hit__Pleurocapsa_sp_FMAR1__UniRe...
GCyR2hit__Phormidesmis_priestleyi__Un...
CyR_MBV8884430.1
CyR_MBV9388690.1
GCyR2hit__Nostocaceae_cyanobacterium_...
GCyR2hit__Pseudanabaenaceae_cyanobact...
GCyR2hit__Tolypothrix_sp_NIES_4075__U...
GCyR2hit__Nostoc_sp__UniRef90_UPI002F...
GCyR2hit__Tolypothrix_bouteillei_VB52...
CyR_BAY09002.1
CyR_OUL18258.1
GCyR2hit__Nostocales__UniRef90_A0A252...
GCyR2hit__Halobacteria__UniRef90_O93740
GCyR2hit__Halobiforma__UniRef90_A0A1P...
GCyR2hit__Halogranum_amylolyticum__Un...
GCyR2hit__Haloferax_mucosum_ATCC_BAA_...
GCyR2hit__Halegenticoccus_tardaugens_...
YCyR2hit__Halorientalis_marina__UniRe...
YCyR2hit__Halorientalis__UniRef90_A0A...
BRhit__Halorientalis_regularis__UniRe...
BRhit__Uncultured_archaeon__UniRef90_...
BRhit__Halobacteriales_archaeon_QS_8_...
BRhit__Halohasta_salina__UniRef90_UPI...
BRhit__Halobaculum_rubrum__UniRef90_U...
BRhit__Haloplanus_natans__UniRef90_UP...
BRhit__Haloplanus__UniRef90_A0A6B9F3P5
BRhit__Halobacterium__UniRef90_UPI001...
BRhit__Halobacterium__UniRef90_P02945
BRhit__unclassified_Halobacteriales__...
BRhit__uncultured_archaeon_A07HR67__U...
BRhit__Halobacteria__UniRef90_P69051
BRhit__Halorubrum__UniRef90_A0A4U7F7H9
YCyR2hit__Natronomonas_moolapensis__U...
YCyR2hit__UnknownTaxon__UniRef90_P29563
BRhit__Halobaculum_sp_MBLA0143__UniRe...
BRhit__Halobaculum_sp_MBLA0147__UniRe...
PRhit__Bacteria__UniRef90_J4KSW5
PRhit__Gammaproteobacteria_bacterium_...
PRhit__environmental_samples__UniRef9...
PRhit__Pseudomonadota_bacterium__UniR...
PRhit__Pseudomonadota__UniRef90_A0A0R...
PRhit__Gammaproteobacteria_bacterium_...
PRhit__Bacteria__UniRef90_A0A1L2YW54
PRhit__Bacteria__UniRef90_Q9AFF7
PRhit__Bacteria__UniRef90_A0A2E1JM07
PRhit__Gammaproteobacteria__UniRef90_...
PRhit__uncultured_bacterium__UniRef90...
PRhit__environmental_samples__UniRef9...
PRhit__SAR86_cluster_bacterium__UniRe...
PRhit__uncultured_bacterium__UniRef90...
PRhit__Bacteria__UniRef90_Q6J4G7
NaR_OKH25360.1
NaR_PPS40940.1
NaR_WP_196601241.1
NaR_WP_087592702.1
NaR_WP_052056064.1
NaR_PSN11013.1
GRhit__Trueperaceae_bacterium__UniRef...
GRhit__Deinococcus_koreensis__UniRef9...
GRhit__Deinococcus_sp__UniRef90_UPI00...
GRhit__Rhodothermales_bacterium__UniR...
GRhit__Bacteroidota_bacterium__UniRef...
GRhit__Myxococcales_bacterium__UniRef...
GRhit__Oligoflexia_bacterium__UniRef9...
GRhit__Pseudobdellovibrionaceae_bacte...
GRhit__Bdellovibrio_sp__UniRef90_A0A9...
GRhit__Thermaceae_bacterium__UniRef90...
GRhit__Allomeiothermus_silvanus__UniR...
GRhit__Deinococcus_sp__UniRef90_UPI00...
GRhit__Deinococcus__UniRef90_A0A917PC33
GRhit__Deinococcus_ruber__UniRef90_A0...
GRhit__Deinococcus__UniRef90_A0A172TD44
GRhit__Halothece_sp__UniRef90_K9YEI2
GRhit__Symploca_sp_SIO2G7__UniRef90_A...
GRhit__Cyanophyceae__UniRef90_A0A4Q7E4T5
GRhit__Halomicronema_sp_CCY15110__Uni...
GRhit__Cyanophyceae__UniRef90_A0A8J7JTD9
GRhit__Leptolyngbya_sp_LCM1_Bin17__Un...
GRhit__Leptolyngbyaceae_cyanobacteriu...
GRhit__Leptolyngbya_sp_Heron_Island_J...
GRhit__Chamaesiphon_sp__UniRef90_UPI0...
GRhit__unclassified_Leptolyngbyaceae_...
GRhit__Cyanophyceae__UniRef90_A0A2W7ARY7
XLR_WP_156518896.1
GRhit__Phormidesmis_priestleyi__UniRe...
GRhit__Leptolyngbya_sp_ES_bin_22__Uni...
GRhit__Cyanobacteria_bacterium_RM1_2_...
GRhit__Gloeobacter_violaceus__UniRef9...

02_OPTICS_predict_lmax

September 2, 2025

[1]:
```python
import sys
from pathlib import Path

#Define paths for current project
# --- Centralized paths ---
ROOT = Path("..")
DATA = ROOT / "data"
LOGS = ROOT / "logs"
SCRIPTS = ROOT / "scripts"
RESULTS = ROOT / "results"
ALIGN_DIR = RESULTS / "align"
TREE_DIR = RESULTS / "trees"
FIGURES = RESULTS / "figures"


# Set the path to your local optics codebase
optics_path = str(Path.home() / "labdata/users/Oakley/GitHub/optics")
if optics_path not in sys.path:
    sys.path.append(optics_path)

from optics_predictions import run_optics_predictions

# Input FASTA: deduplicated, combined rhodopsin sequences
fasta_file = DATA / "pumphits_culled.fasta"
results_dir = "../results/optics"

# Run OPTICS predictions with options matching the CLI help
optics_df, optics_pred_file = run_optics_predictions(
    input_sequence=fasta_file,
    pred_dir=results_dir,                              # --output_dir
    output="optics_predictions",                       # --prediction_prefix
    model="type-one",                                  # --model
    encoding_method="aa_prop",                         # --encoding
    blastp=False,                                      # --blastp
    iden_report="blastp_report.txt",                   # --blastp_report
    refseq="bovine",                                   # --refseq
    bootstrap=False                                    # --bootstrap
)

print(f"OPTICS predictions saved to: {optics_pred_file}")
optics_df.head()
```

```
Processing Sequences: 100%|             |
90/90 [00:27<00:00,  3.27seqs/s]

Error: Cached prediction file can't be saved…

YCyR2hit__Nodosilinea_nodulosa__UniRef90_A0AAJ6N6B2     544.6   -       256

YCyR2hit__Pseudanabaena_sp_FACHB_2040__UniRef90_A0A926ZYX4      532.5   -
235

YCyR2hit__Leptolyngbya_sp_FACHB_261__UniRef90_A0A926UF78        528.7   -
229

YCyR2hit__Hassalia_byssoidea_VB512170__UniRef90_A0A846HFU5      537.3   -
233

YCyR2hit__Halorientalis__UniRef90_A0A1G7PUC9    552.4   -       245

YCyR2hit__UnknownTaxon__UniRef90_P29563 546.8   -       259

YCyR2hit__Leptolyngbya_ohadii__UniRef90_UPI000B5A185C   568.0   -       236

YCyR2hit__Aliterella_atlantica_CENA595__UniRef90_A0A0D8ZW42     542.8   -
236

YCyR2hit__Cyanobacteriota__UniRef90_A0A926PVG5  554.1   -       232

YCyR2hit__Microcoleus_sp_FACHB_1515__UniRef90_A0A926U182        534.7   -
233

YCyR2hit__Halorientalis_marina__UniRef90_UPI001FF6C9D7  539.8   -       243

YCyR2hit__unclassified_Leptolyngbya__UniRef90_UPI0016884D19     556.9   -
235

YCyR2hit__Leptolyngbya_sp_NIES_2104__UniRef90_A0A0P4V0S4        557.0   -
236

YCyR2hit__unclassified_Nostoc__UniRef90_A0A252DUV9      538.5   -       234

YCyR2hit__Natronomonas_moolapensis__UniRef90_M1Y5I7     546.8   -       258

BRhit__Halobacterium__UniRef90_P02945   561.7   -       262
```

```
BRhit__Halobacterium__UniRef90_UPI001F011AB0    562.5   -       262

BRhit__Haloplanus__UniRef90_A0A6B9F3P5  558.1   -       256

BRhit__Haloplanus_natans__UniRef90_UPI000A5CE34D        562.1   -       256

BRhit__Halobaculum_rubrum__UniRef90_UPI001CA389E1       556.6   -       248

BRhit__Halohasta_salina__UniRef90_UPI002110EEC8 551.6   -       263

BRhit__Halobacteriales_archaeon_QS_8_69_26__UniRef90_A0A2R6JS63 548.4   -
265

BRhit__Uncultured_archaeon__UniRef90_I1X958     545.3   -       259

BRhit__Halorientalis_regularis__UniRef90_A0A1G7RA78     554.0   -       258

BRhit__unclassified_Halobacteriales__UniRef90_A0A2R6LDY7        545.8   -
264

BRhit__uncultured_archaeon_A07HR67__UniRef90_V4ZQH6     553.3   -       260

BRhit__Halobaculum_sp_MBLA0143__UniRef90_UPI00352426DC  556.7   -       260

BRhit__Halorubrum__UniRef90_A0A4U7F7H9  553.2   -       258

BRhit__Halobacteria__UniRef90_P69051    558.0   -       260

GCyR2hit__Nostocales__UniRef90_A0A252D874       543.6   -       247

GCyR2hit__Tolypothrix_bouteillei_VB521301__UniRef90_A0A8S9SVI8  543.3   -
247

GCyR2hit__Nostoc_sp__UniRef90_UPI002FFB8FFD     542.3   -       248

GCyR2hit__Tolypothrix_sp_NIES_4075__UniRef90_A0A218QMM7 539.6   -       252

GCyR2hit__Pleurocapsa_sp_FMAR1__UniRef90_UPI0029C671DC  541.8   -       246

GCyR2hit__Nostocaceae_cyanobacterium__UniRef90_A0A838VMV4       516.6   -
244

GCyR2hit__Phormidesmis_priestleyi__UniRef90_UPI0009442622       542.9   -
248

GCyR2hit__Pseudanabaenaceae_cyanobacterium_LEGE_13415__UniRef90_A0A928V9X3
542.6   -       248
```

```
GCyR2hit__Cyanobacteria_bacterium_QH_9_48_43__UniRef90_A0A2T2RN81       542.0
-       234

GCyR2hit__Halegenticoccus_tardaugens__UniRef90_UPI00100C2973    542.0   -
241

GCyR2hit__Haloferax_mucosum_ATCC_BAA_1512__UniRef90_M0IDG5      547.7   -
243

GCyR2hit__Leptolyngbya_sp_FACHB_261__UniRef90_A0A926UF78        528.7   -
229

GCyR2hit__Halobiforma__UniRef90_A0A1P8LVS0      545.0   -       240

GCyR2hit__Halogranum_amylolyticum__UniRef90_A0A1H8RSA7  546.2   -       241

GCyR2hit__Halobacteria__UniRef90_O93740 547.0   -       250

GRhit__Gloeobacter_violaceus__UniRef90_Q7NP59   539.0   -       298

GRhit__Cyanobacteria_bacterium_RM1_2_2__UniRef90_A0A969T0G4     538.5   -
299

GRhit__Cyanophyceae__UniRef90_A0A2W7ARY7        538.9   -       297

GRhit__unclassified_Leptolyngbyaceae__UniRef90_A0A969FEC7       538.5   -
297

GRhit__Leptolyngbya_sp_ES_bin_22__UniRef90_A0A925M2S9   540.3   -       297

GRhit__Phormidesmis_priestleyi__UniRef90_UPI000A475A3D  539.6   -       298

GRhit__Chamaesiphon_sp__UniRef90_UPI003592F9D2  538.1   -       304

GRhit__Leptolyngbya_sp_Heron_Island_J__UniRef90_U9W0I1  541.7   -       282

GRhit__Cyanophyceae__UniRef90_A0A8J7JTD9        544.0   -       269

GRhit__Halomicronema_sp_CCY15110__UniRef90_UPI0021030F7B        543.2   -
281

GRhit__Cyanophyceae__UniRef90_A0A4Q7E4T5        543.4   -       281

GRhit__Leptolyngbyaceae_cyanobacterium_JSC_12__UniRef90_K8GTY7  538.7   -
262

GRhit__Halothece_sp__UniRef90_K9YEI2    543.6   -       281
```

```
GRhit__Leptolyngbya_sp_LCM1_Bin17__UniRef90_A0A6H2NHV9  540.0   -       276

GRhit__Symploca_sp_SIO2G7__UniRef90_A0A845YHD9  538.7   -       265

GRhit__Deinococcus__UniRef90_A0A172TD44 545.0   -       285

GRhit__Deinococcus_ruber__UniRef90_A0A918CFF3   543.7   -       293

GRhit__Deinococcus__UniRef90_A0A917PC33 541.0   -       295

GRhit__Deinococcus_sp__UniRef90_UPI0025F8D0E5   534.1   -       259

GRhit__Allomeiothermus_silvanus__UniRef90_UPI0023F38943 541.4   -       258

GRhit__Bdellovibrio_sp__UniRef90_A0A924B300     544.2   -       253

GRhit__Trueperaceae_bacterium__UniRef90_A0A5Q4G3V2      543.2   -       264

GRhit__Myxococcales_bacterium__UniRef90_A0A2E7ZR97      537.5   -       261

GRhit__Thermaceae_bacterium__UniRef90_A0A931EK09        534.2   -       262

GRhit__Oligoflexia_bacterium__UniRef90_A0A923U6H7       550.1   -       255

GRhit__Deinococcus_sp__UniRef90_UPI002869BBA0   542.5   -       299

GRhit__Bacteroidota_bacterium__UniRef90_A0A3M2FMH9      542.2   -       260

GRhit__Rhodothermales_bacterium__UniRef90_A0A9D8YSH6    538.4   -       261

GRhit__Pseudobdellovibrionaceae_bacterium__UniRef90_A0A924ZXQ0  548.8   -
253

GRhit__Deinococcus_koreensis__UniRef90_A0A2K3UTP4       537.1   -       285

PRhit__Bacteria__UniRef90_Q6J4G7        520.4   -       250

PRhit__uncultured_bacterium__UniRef90_Q84C21    517.5   -       258

PRhit__SAR86_cluster_bacterium__UniRef90_A0A937M2P1     519.3   -       250

PRhit__environmental_samples__UniRef90_Q6PL05   520.8   -       251

PRhit__uncultured_bacterium__UniRef90_A0A1L2YW20        500.0   -       251

PRhit__Bacteria__UniRef90_Q9AFF7        495.9   -       251
```

```
PRhit__Bacteria__UniRef90_A0A1L2YW54    495.4   -       249

PRhit__Gammaproteobacteria__UniRef90_A0A368C6X1 506.3   -       250

PRhit__Gammaproteobacteria_bacterium_TMED104__UniRef90_A0A3S5JK93       526.2
-       247

PRhit__Bacteria__UniRef90_J4KSW5        521.3   -       251

PRhit__Bacteria__UniRef90_A0A2E1JM07    504.6   -       250

PRhit__environmental_samples__UniRef90_A0A1L2YWB0       520.2   -       250

PRhit__Pseudomonadota_bacterium__UniRef90_A0A944UJK0    520.1   -       249

PRhit__Pseudomonadota__UniRef90_A0A0R2UEA1      525.2   -       248

PRhit__Gammaproteobacteria_bacterium__UniRef90_A0A2E8KSB6       506.1   -
251

BRmutant_WP_136361479.1_D85A    610.2   -       263


Model Used:     type-one
Encoding Method:        aa_prop

Predictions Complete!
OPTICS predictions saved to: ../results/optics/optics_on_optics_predictions_2025
-09-01_21-43-10/optics_predictions_predictions.tsv
```

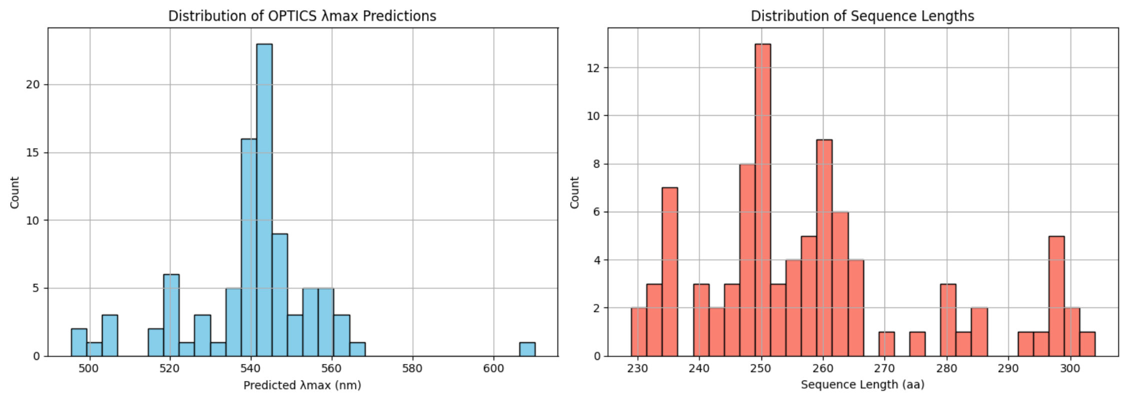
[1]:

```
                                              Names  Single_Prediction  \
0  YCyR2hit__Nodosilinea_nodulosa__UniRef90_A0AAJ…              544.6
1  YCyR2hit__Pseudanabaena_sp_FACHB_2040__UniRef9…              532.5
2  YCyR2hit__Leptolyngbya_sp_FACHB_261__UniRef90_…              528.7
3  YCyR2hit__Hassalia_byssoidea_VB512170__UniRef9…              537.3
4      YCyR2hit__Halorientalis__UniRef90_A0A1G7PUC9              552.4

  %Identity_Nearest_VPOD_Sequence  Sequence_Length Lmax_Hex_Color
0                               -              256        #91ff00
1                               -              235        #67ff00
2                               -              229        #59ff00
3                               -              233        #78ff00
4                               -              245        #abff00
```

[2]:

```python
#Plot histograms for sequence length and OPTICS predictions of lambda-max

%matplotlib inline
```

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the OPTICS results TSV
optics_tsv = RESULTS / "optics/optics_on_optics_predictions_2025-09-01_21-43-10/
 ↪optics_predictions_predictions.tsv"
df = pd.read_csv(optics_tsv, sep="\t")

# Plot histograms of lambda-max predictions and sequence length
fig, axes = plt.subplots(1, 2, figsize=(14, 5))

# Histogram for lambda-max predictions
df["Single_Prediction"].hist(bins=30, color="skyblue", edgecolor="black",␣
 ↪ax=axes[0])
axes[0].set_xlabel("Predicted  max (nm)")
axes[0].set_ylabel("Count")
axes[0].set_title("Distribution of OPTICS  max Predictions")

# Histogram for sequence length
df["Sequence_Length"].hist(bins=30, color="salmon", edgecolor="black",␣
 ↪ax=axes[1])
axes[1].set_xlabel("Sequence Length (aa)")
axes[1].set_ylabel("Count")
axes[1].set_title("Distribution of Sequence Lengths")

plt.tight_layout()
plt.show()
```



03_ancestral_states

September 2, 2025

[10]:
```
#Need ancseq kernel

from pathlib import Path

#Define paths for current project
# --- Centralized paths ---
ROOT = Path("..")
DATA = ROOT / "data"
LOGS = ROOT / "logs"
SCRIPTS = ROOT / "scripts"
RESULTS = ROOT / "results"
ALIGN_DIR = RESULTS / "align"
TREE_DIR = RESULTS / "trees"
FIGURES = RESULTS / "figures"
ANCESTORS = RESULTS / "ancestors"
```

Note:
For long-running jobs (like `ancseq`), do **not** run them in a Jupyter notebook if you are worried about losing your connection.
Instead, use a terminal multiplexer such as `tmux` or `screen` in a regular terminal:

1. Open a terminal.
2. Start a tmux session: `tmux`
3. Activate your environment and run your command:

   ```
   conda activate ancseq
   ancseq -s ../results/align/combined_plus_hasegawa24_ALN.fasta -m AA -o ../results/ancestor
   ```
4. Detach from tmux with `Ctrl+b` then `d`.

This ensures your job continues running even if you disconnect.

[11]:
```
from Bio import SeqIO

# Paths
HITS = ALIGN_DIR / "pumphits_ALN.fasta"
ANCESTOR_FASTA = ANCESTORS / "cph24/30_result/ancestral_state_result.fasta"
FINAL = DATA / "final_with_ancestors.fasta"
```

```
# Ancestor node names
Ancestors = ("Node36", "Node73", "Node37", "Node6",
             "Node87", "Node4", "Node117", "Node38",
             "Node40", "Node44")

# Read original alignment
aln_records = list(SeqIO.parse(HITS, "fasta"))

# Read ancestor sequences
ancestor_records = [rec for rec in SeqIO.parse(ANCESTOR_FASTA, "fasta") if rec.
 ↪id in Ancestors]

# Concatenate and write to FINAL
with open(FINAL, "w") as out_handle:
    SeqIO.write(aln_records + ancestor_records, out_handle, "fasta")

print(f"Combined alignment and ancestor sequences written to {FINAL}")

# Align with MAFFT
ALN_FASTA = ALIGN_DIR / "final_with_ancestors_ALN.fasta"
!mafft --auto --thread -1 --quiet "{FINAL}" > "{ALN_FASTA}"
print("Aligned ->", ALN_FASTA)
```

```
Combined alignment and ancestor sequences written to
../data/final_with_ancestors.fasta
Aligned -> ../results/align/final_with_ancestors_ALN.fasta
```