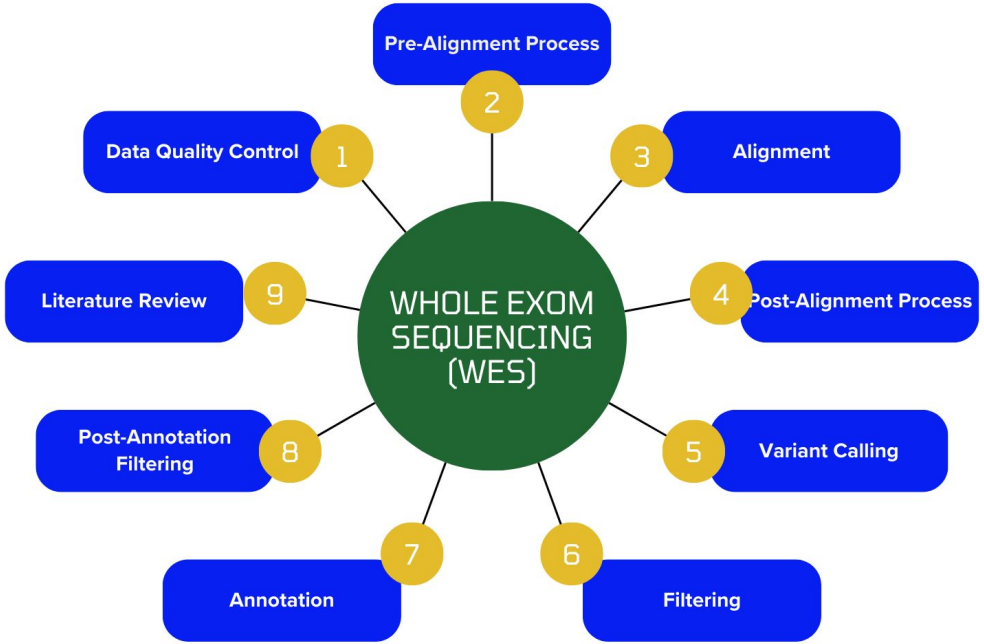
EVOLUTIONARY HISTORY OF TMEM216 PROTEIN

Samet Mert -23511
Dila Karataş -28852
Nil Özde Özgen -30802
Türkan Doğa Gizer - 29072
Mehmet Barış Tekdemir -29068

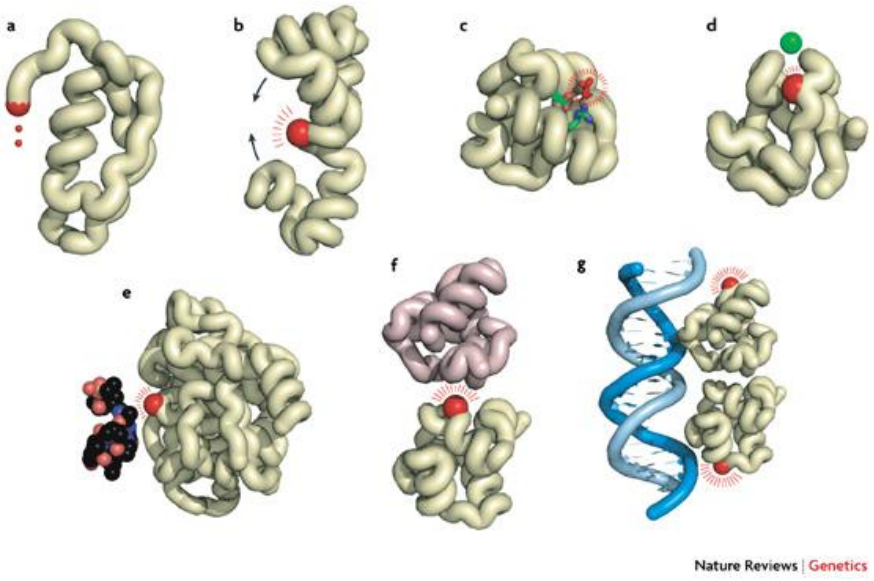
Outline

- Recap
- Introduction
- Tools & Methods
- Results
- Discussion
- Issues
- References

Recap


WHOLE EXOM SEQUENCING (WES)
1
Data Quality Control
2
Pre-Alignment Process
3
Alignment
4
Post-Alignment Process
5
Variant Calling
6
Filtering
7
Annotation
8
Post-Annotation Filtering
9
Literature Review

Mutations in Proteins



– Errors in the Binding Process.

– Rare Diseases.

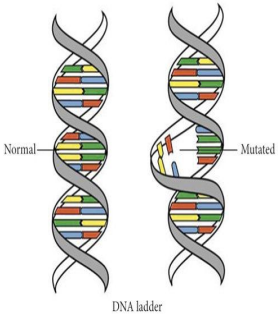
Why TMEM 216 ?

About 150 mutations in proteins:

Genes with mutations → Fatal Diseases.

Genes with mutations → Not Fatal Diseases.

Genes with mutations → Not Fatal Disease, but Fatal Complications.



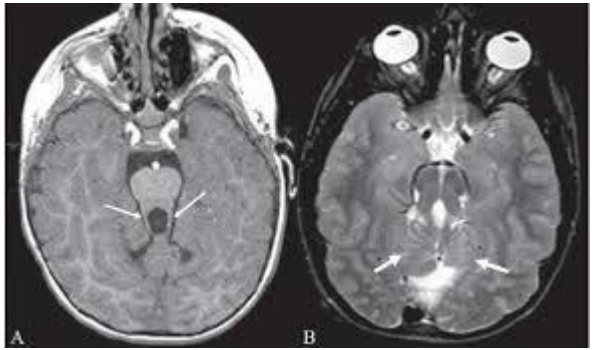
According to Pathogenicity:

→ Elimination and Analysis.

→ NPHP4 → TMEM 216 → Joubert Syndrome 2.

Joubert Syndrome 2

- Rare, neurological, genetic disease.
- Autosomal recessive.



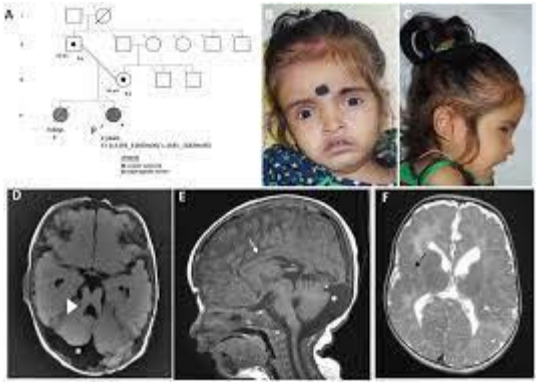
-MTS (Mesial Temporal Sclerosis).

-Retinal dystrophy.

-Muscle weakness.

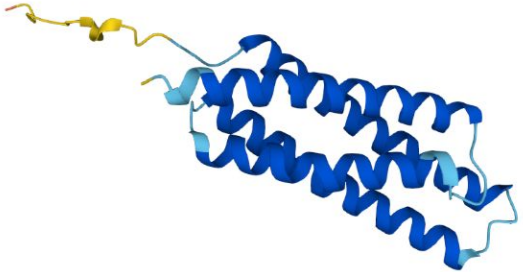
-Renal diseases.

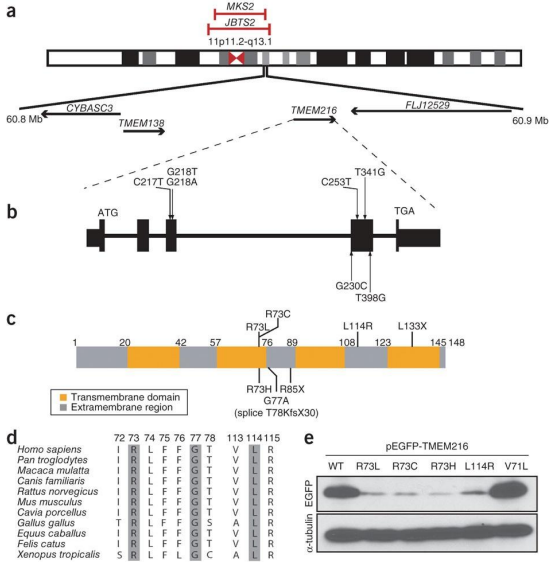
-Endocrine abnormalities.



TMEM 216 Protein

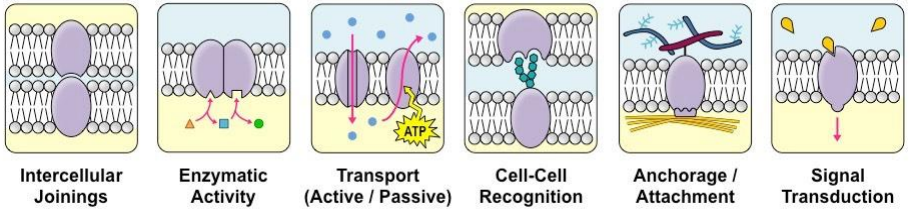
- TMEM 216 gene.
- Cell communication and Signaling
- 87 amino acid & 2 transmembrane domain.

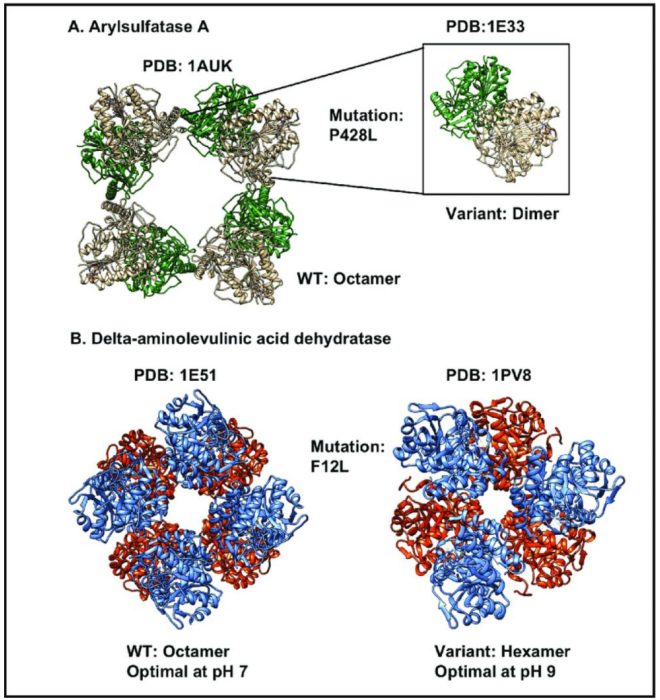




Transmembrane Proteins

- Transportation, signaling, enzyme activity.





Are our variants pathogenic?

```
chr11   61391696        .       G       GT      53.7    .       AC=2;AF=1;AN=2;DP=2;ExcessHet=3.0103;FS=0;M
LEAC=2;MLEAF=1;MQ=60;QD=26.85;SOR=0.693;EFF=upstream_gene_variant(MODIFIER||696||84|TMEM216|protein_coding|
CODING|ENST00000398979||GT),upstream_gene_variant(MODIFIER||890||148|TMEM216|protein_coding|CODING|ENST0000
0334888||GT),upstream_gene_variant(MODIFIER||890||145|TMEM216|protein_coding|CODING|ENST00000515837||GT),up
stream_gene_variant(MODIFIER||890|||TMEM216|retained_intron|CODING|ENST00000544795||GT),upstream_gene_varia
nt(MODIFIER||935|||TMEM216|retained_intron|CODING|ENST00000541473||GT),upstream_gene_variant(MODIFIER||890|
||TMEM216|nonsense_mediated_decay|CODING|ENST00000690736||GT),upstream_gene_variant(MODIFIER||916||59|TMEM2
16|protein_coding|CODING|ENST00000688959||GT),upstream_gene_variant(MODIFIER||933|||TMEM216|processed_trans
cript|CODING|ENST00000684926||GT),intergenic_region(MODIFIER|||||||||GT)GT:AD:DP:GQ:PL    1/1:0,2:2:6:90,6,
0
```

```
chr11   61391719        .       A       G       62.74   .       AC=2;AF=1;AN=2;DP=2;ExcessHet=3.0103;FS=0;M
LEAC=2;MLEAF=1;MQ=60;QD=31.37;SOR=0.693;EFF=upstream_gene_variant(MODIFIER||674||84|TMEM216|protein_coding|
CODING|ENST00000398979||G),upstream_gene_variant(MODIFIER||868||148|TMEM216|protein_coding|CODING|ENST00000
334888||G),upstream_gene_variant(MODIFIER||868||145|TMEM216|protein_coding|CODING|ENST00000515837||G),upstr
eam_gene_variant(MODIFIER||868|||TMEM216|retained_intron|CODING|ENST00000544795||G),upstream_gene_variant(M
ODIFIER||913|||TMEM216|retained_intron|CODING|ENST00000541473||G),upstream_gene_variant(MODIFIER||868|||TME
M216|nonsense_mediated_decay|CODING|ENST00000690736||G),upstream_gene_variant(MODIFIER||894||59|TMEM216|pro
tein_coding|CODING|ENST00000688959||G),upstream_gene_variant(MODIFIER||911|||TMEM216|processed_transcript|C
ODING|ENST00000684926||G),intergenic_region(MODIFIER|||||||||G)       GT:AD:DP:GQ:PL      1/1:0,2:2:6:90,6,
0
```

```
chr11   61402023        .       C       CT      49.25   .       AC=2;AF=1;AN=2;DP=3;ExcessHet=3.0103;FS=0;M
LEAC=2;MLEAF=1;MQ=60;QD=16.42;SOR=1.179;EFF=downstream_gene_variant(MODIFIER||617||514|CPSF7|protein_coding
|CODING|ENST00000340437||CT),downstream_gene_variant(MODIFIER||3531||84|TMEM216|protein_coding|CODING|ENST0
0000398979||CT),downstream_gene_variant(MODIFIER||705||462|CPSF7|protein_coding|CODING|ENST00000439958||CT)
,downstream_gene_variant(MODIFIER||3178||148|TMEM216|protein_coding|CODING|ENST00000334888||CT),downstream_
gene_variant(MODIFIER||3178||145|TMEM216|protein_coding|CODING|ENST00000515837||CT),downstream_gene_variant
(MODIFIER||625||471|CPSF7|protein_coding|CODING|ENST00000394888||CT),downstream_gene_variant(MODIFIER||2264
|||CPSF7|processed_transcript|CODING|ENST00000494016||CT),downstream_gene_variant(MODIFIER||3158|||TMEM216|
retained_intron|CODING|ENST00000544795||CT),downstream_gene_variant(MODIFIER||2395||462|CPSF7|protein_codin
g|CODING|ENST00000448745||CT),downstream_gene_variant(MODIFIER||3178|||TMEM216|nonsense_mediated_decay|CODI
NG|ENST00000690736||CT),downstream_gene_variant(MODIFIER||3280||59|TMEM216|protein_coding|CODING|ENST000006
88959||CT),downstream_gene_variant(MODIFIER||3158|||TMEM216|processed_transcript|CODING|ENST00000684926||CT
),intergenic_region(MODIFIER|||||||||CT)    GT:AD:DP:GQ:PL  1/1:0,3:3:9:86,9,0
```

```
chr11   61397808        .       G       A       2492.77 .       AC=2;AF=1;AN=2;BaseQRankSum=2.237;ClippingR
ankSum=0;DP=91;ExcessHet=3.0103;FS=3.261;MLEAC=2;MLEAF=1;MQ=60;MQRankSum=0;QD=28.01;ReadPosRankSum=1.986;SO
R=0.442;EFF=synonymous_variant(LOW|SILENT|ccG/ccA|P27|84|TMEM216|protein_coding|CODING|ENST00000398979|4|A)
,synonymous_variant(LOW|SILENT|ccG/ccA|P88|148|TMEM216|protein_coding|CODING|ENST00000334888|4|A),synonymou
s_variant(LOW|SILENT|ccG/ccA|P88|145|TMEM216|protein_coding|CODING|ENST00000515837|4|A),synonymous_variant(
LOW|SILENT|ccG/ccA|P2|59|TMEM216|protein_coding|CODING|ENST00000688959|4|A),downstream_gene_variant(MODIFIE
R||4833||514|CPSF7|protein_coding|CODING|ENST00000340437||A),downstream_gene_variant(MODIFIER||4921||462|CP
SF7|protein_coding|CODING|ENST00000439958||A),downstream_gene_variant(MODIFIER||4841||471|CPSF7|protein_cod
ing|CODING|ENST00000394888||A),downstream_gene_variant(MODIFIER||3507|||TMEM216|retained_intron|CODING|ENST
00000541473||A),non_coding_transcript_exon_variant(MODIFIER|||||TMEM216|retained_intron|CODING|ENST00000544
795|4|A),non_coding_transcript_exon_variant(MODIFIER|||||TMEM216|nonsense_mediated_decay|CODING|ENST0000069
0736|4|A),non_coding_transcript_exon_variant(MODIFIER|||||TMEM216|processed_transcript|CODING|ENST000006849
26|4|A)     GT:AD:DP:GQ:PL  1/1:1,88:89:99:2521,257,0
```

☑ 142.	NM_001173990.3(**TMEM216**):c.264_265delinsAG (p.Leu89Val) *GRCh37*: Chr11:61165280-61165281 *GRCh38*: Chr11:61397808-61397809	**TMEM216**	L28V, L89V	Familial aplasia of the vermis	Uncertain significance (Mar 10, 2022)	criteria provided, single submitter
☑ 143.	NM_001173990.3(**TMEM216**):c.264G>C (p.Pro88=) *GRCh37*: Chr11:61165280 *GRCh38*: Chr11:61397808	**TMEM216**		Familial aplasia of the vermis	Likely benign (Nov 30, 2019)	criteria provided, single submitter
☑ 144.	NM_001173990.3(**TMEM216**):c.264_265delinsAT (p.Leu89Phe) *GRCh37*: Chr11:61165280-61165281 *GRCh38*: Chr11:61397808-61397809	**TMEM216**	L28F, L89F	Familial aplasia of the vermis	Uncertain significance (Sep 27, 2022)	criteria provided, single submitter
☑ 145.	NM_001173990.3(**TMEM216**):c.264G>A (p.Pro88=) *GRCh37*: Chr11:61165280 *GRCh38*: Chr11:61397808	**TMEM216**		not specified, Meckel syndrome, type 2, Familial aplasia of the vermis, not provided, Joubert syndrome 2	Benign (Oct 7, 2022)	criteria provided, multiple submitters, no conflicts

```
chr11   61398259        .       C       CA      2802.73 .       AC=2;AF=1;AN=2;DP=66;ExcessHet=3.0103;FS=0;
MLEAC=2;MLEAF=1;MQ=60;QD=34.61;SOR=0.723;EFF=splice_acceptor_variant+intron_variant(HIGH||||148|TMEM216|pro
tein_coding|CODING|ENST00000334888|4|CA),downstream_gene_variant(MODIFIER||4381||514|CPSF7|protein_coding|C
ODING|ENST00000340437||CA),downstream_gene_variant(MODIFIER||4469||462|CPSF7|protein_coding|CODING|ENST0000
0439958||CA),downstream_gene_variant(MODIFIER||4389||471|CPSF7|protein_coding|CODING|ENST00000394888||CA),d
ownstream_gene_variant(MODIFIER||3959|||TMEM216|retained_intron|CODING|ENST00000541473||CA),intron_variant(
MODIFIER||||84|TMEM216|protein_coding|CODING|ENST00000398979|4|CA),intron_variant(MODIFIER||||145|TMEM216|p
rotein_coding|CODING|ENST00000515837|4|CA),intron_variant(MODIFIER|||||TMEM216|retained_intron|CODING|ENST0
0000544795|4|CA),intron_variant(MODIFIER|||||TMEM216|nonsense_mediated_decay|CODING|ENST00000690736|4|CA),i
ntron_variant(MODIFIER||||59|TMEM216|protein_coding|CODING|ENST00000688959|4|CA),intron_variant(MODIFIER|||
||TMEM216|processed_transcript|CODING|ENST00000684926|4|CA);LOF=(TMEM216|ENSG00000187049|8|0.13)      GT:A
D:DP:GQ:PL  1/1:0,65:65:99:2840,196,0
```

☐ 205.	NM_001173990.3(**TMEM216**):c.432-11_432-10insA *GRCh37*: Chr11:61165731-61165732 *GRCh38*: Chr11:61398259-61398260	TMEM216		not specified, not provided, Familial aplasia of the vermis, Joubert syndrome 1, Meckel-Gruber syndrome	Benign (Oct 7, 2022)

```
chr11   61398269        .       G       C       2690.77 .       AC=2;AF=1;AN=2;DP=60;ExcessHet=3.0103;FS=0;
MLEAC=2;MLEAF=1;MQ=60;QD=28.06;SOR=0.831;EFF=splice_acceptor_variant+intron_variant(HIGH||||84|TMEM216|prot
ein_coding|CODING|ENST00000398979|4|C),splice_acceptor_variant+intron_variant(HIGH||||145|TMEM216|protein_c
oding|CODING|ENST00000515837|4|C),splice_acceptor_variant+intron_variant(HIGH|||||TMEM216|retained_intron|C
ODING|ENST00000544795|4|C),splice_acceptor_variant+intron_variant(HIGH|||||TMEM216|nonsense_mediated_decay|
CODING|ENST00000690736|4|C),splice_acceptor_variant+intron_variant(HIGH||||59|TMEM216|protein_coding|CODING
|ENST00000688959|4|C),splice_acceptor_variant+intron_variant(HIGH|||||TMEM216|processed_transcript|CODING|E
NST00000684926|4|C),missense_variant(MODERATE|MISSENSE|aGg/aCg|R147T|148|TMEM216|protein_coding|CODING|ENST
00000334888|5|C),downstream_gene_variant(MODIFIER||4372||514|CPSF7|protein_coding|CODING|ENST00000340437||C
),downstream_gene_variant(MODIFIER||4460||462|CPSF7|protein_coding|CODING|ENST00000439958||C),downstream_ge
ne_variant(MODIFIER||4380||471|CPSF7|protein_coding|CODING|ENST00000394888||C),downstream_gene_variant(MODI
FIER||3968|||TMEM216|retained_intron|CODING|ENST00000541473||C);LOF=(TMEM216|ENSG00000187049|8|0.38)   GT:A
D:DP:GQ:PL  1/1:0,60:60:99:2719,187,0
```

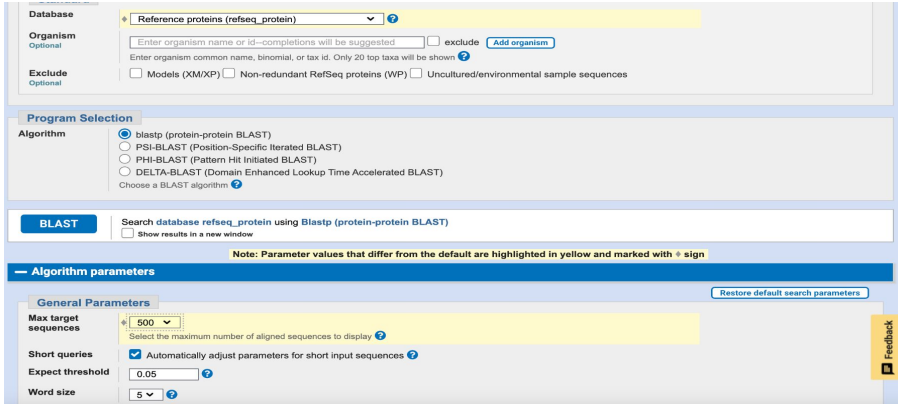
☐ 211.	NM_001173990.3(**TMEM216**):c.432-1G>C *GRCh37*: Chr11:61165741 *GRCh38*: Chr11:61398269	**TMEM216**	R147T, R86T	not specified, Joubert syndrome 1, Meckel syndrome, type 2, Joubert syndrome 2, not provided, Familial aplasia of the vermis	Benign (Oct 7, 2022)

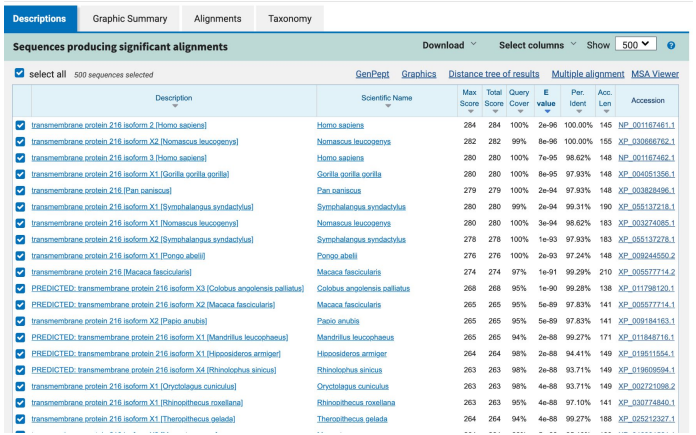
```
chr11   61398834        .       G       A       61.28   .       AC=2;AF=1;AN=2;DP=3;ExcessHet=3.0103;FS=0;M
LEAC=2;MLEAF=1;MQ=60;QD=20.43;SOR=1.179;EFF=3_prime_UTR_variant(MODIFIER||558||148|TMEM216|protein_coding|C
ODING|ENST00000334888|5|A),3_prime_UTR_variant(MODIFIER||558||145|TMEM216|protein_coding|CODING|ENST0000051
5837|5|A),3_prime_UTR_variant(MODIFIER||1015|||TMEM216|nonsense_mediated_decay|CODING|ENST00000690736|5|A),
downstream_gene_variant(MODIFIER||3807||514|CPSF7|protein_coding|CODING|ENST00000340437||A),downstream_gene
_variant(MODIFIER||341||84|TMEM216|protein_coding|CODING|ENST00000398979||A),downstream_gene_variant(MODIFI
ER||3895||462|CPSF7|protein_coding|CODING|ENST00000439958||A),downstream_gene_variant(MODIFIER||3815||471|C
PSF7|protein_coding|CODING|ENST00000394888||A),downstream_gene_variant(MODIFIER||4533|||TMEM216|retained_in
tron|CODING|ENST00000541473||A),downstream_gene_variant(MODIFIER||90||59|TMEM216|protein_coding|CODING|ENST
00000688959||A),non_coding_transcript_exon_variant(MODIFIER|||||TMEM216|retained_intron|CODING|ENST00000544
795|5|A),non_coding_transcript_exon_variant(MODIFIER|||||TMEM216|nonsense_mediated_decay|CODING|ENST0000069
0736|5|A),non_coding_transcript_exon_variant(MODIFIER|||||TMEM216|processed_transcript|CODING|ENST000006849
26|5|A)      GT:AD:DP:GQ:PL  1/1:0,3:3:9:89,9,0
```

☐ 229.	NM_001173990.3(**TMEM216**):c.*558G>A *GRCh37*: Chr11:61166306 *GRCh38*: Chr11:61398834	TMEM216		Joubert syndrome 2, Meckel syndrome, type 2	Benign (Jan 12, 2018)

Homologous Sequence Search

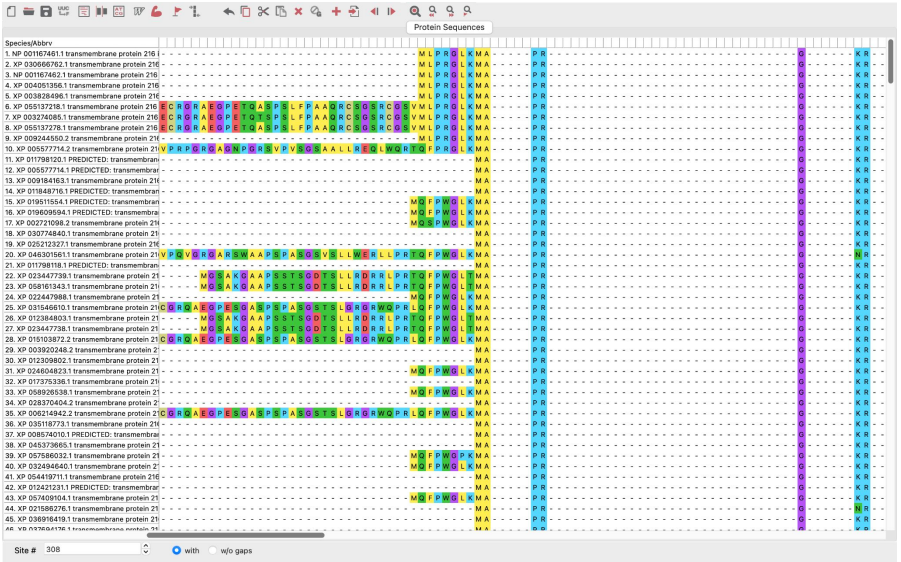
- BLASTp





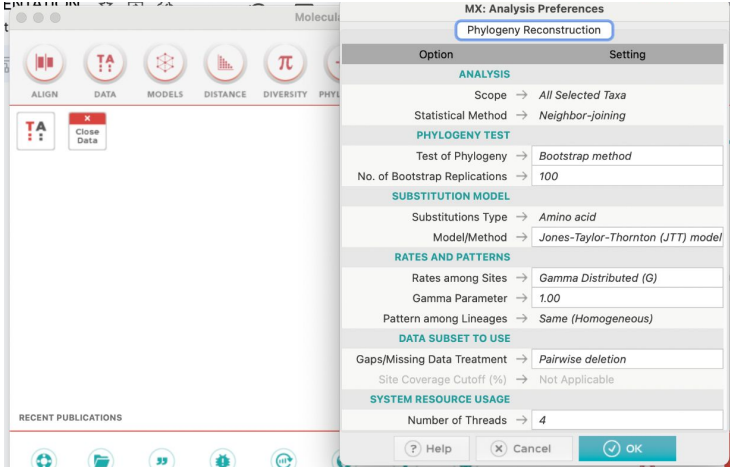
Multiple Sequence Alignment (MSA)

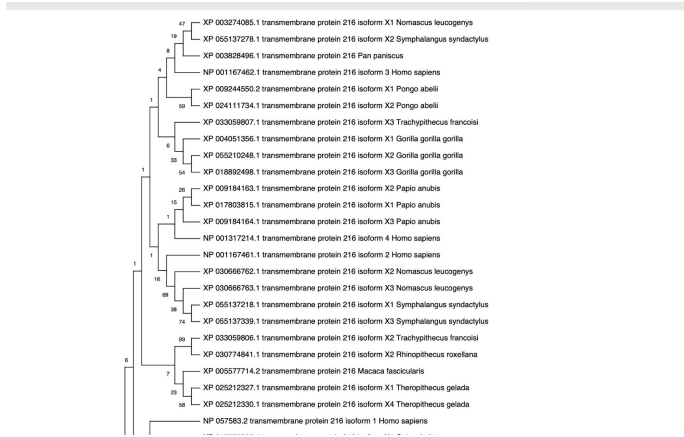
- MEGAX
- MUSCLE algorithm



Phylogenetic Tree Construction

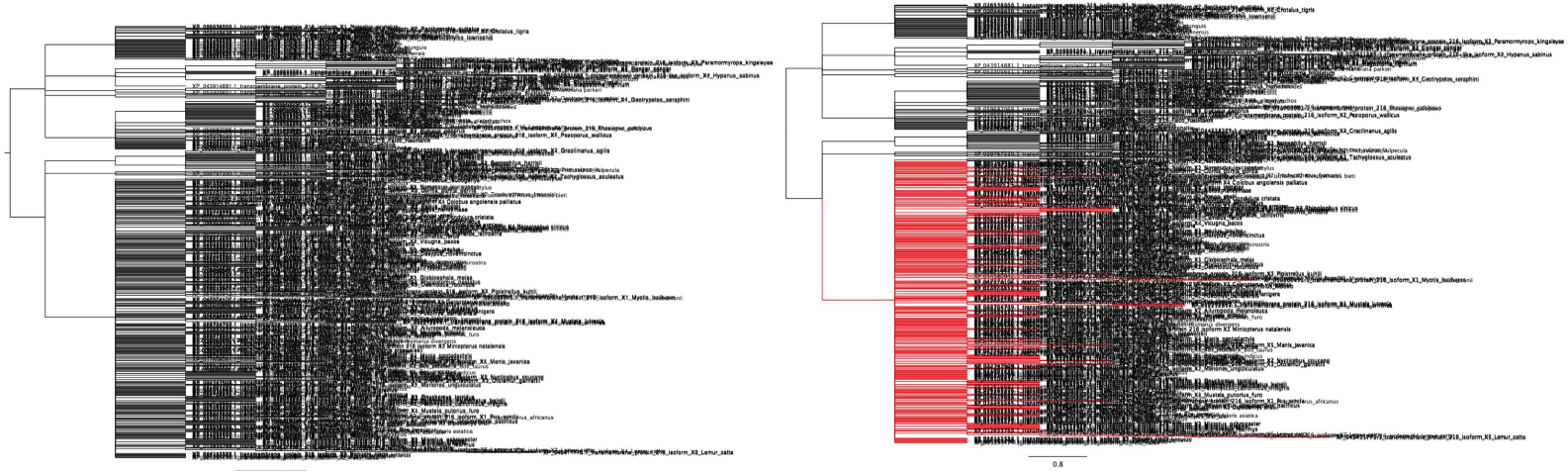
- MEGAX (Neighbor-Joining Method)





Rooting the Phylogenetic Trees

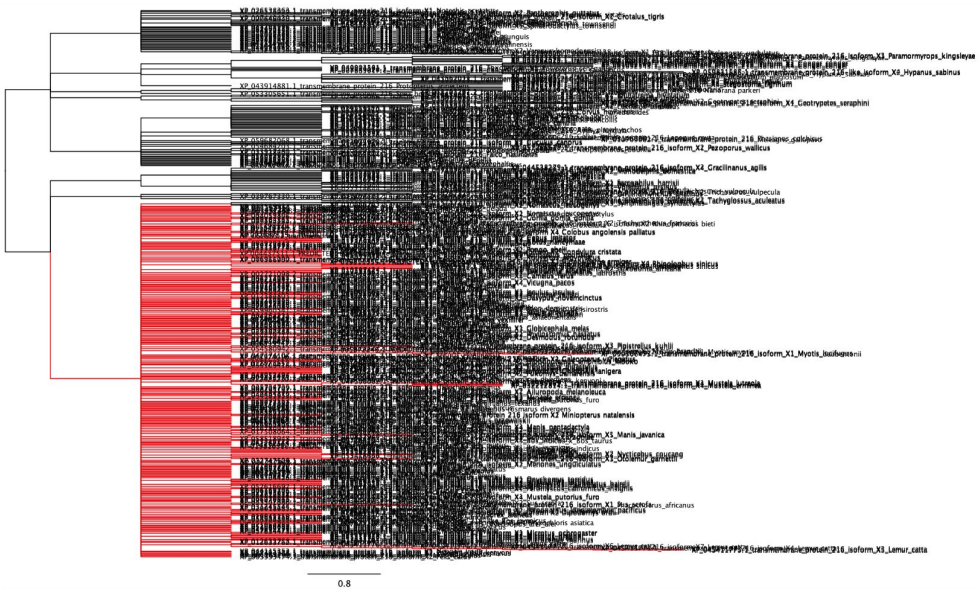
- FigTree (Midpoint Rooting)



Phylogenetic tree of 500 sequences file

Obtaining the Fasta File of the Subtree

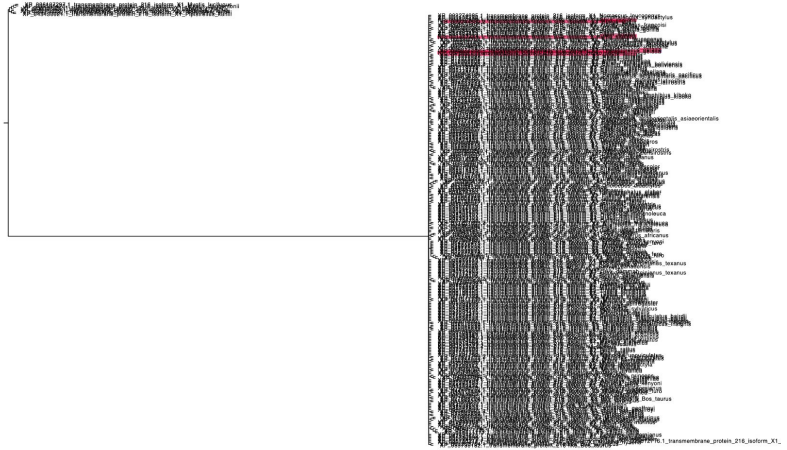
- Python




```python
import re

def extract_xp_words_from_file(filename):
    try:
        with open(filename, 'r') as file:
            newick_string = file.read()
            pattern = r'[XN]P_\d+\.\d+'
            xp_matches = re.findall(pattern, newick_string)
            return xp_matches
    except FileNotFoundError:
        print(f"Error: File '{filename}' not found.")
        return []
filename = "input.nwk"
xp_words = extract_xp_words_from_file(filename)
def extract_sequences_from_fasta(input_fasta, xp_identifiers, output_fasta):
    try:
        with open(input_fasta, 'r') as input_file, open(output_fasta, 'w') as output_file:
            in_xp_list = False
            for line in input_file:
                if line.startswith('>'):
                    identifier = line.strip()[1:]
                    index=identifier.find(" ")
                    aa=identifier[0:index]
                    in_xp_list = identifier[0:index] in xp_identifiers
                    if in_xp_list:
                        output_file.write(line)
                elif in_xp_list:
                    output_file.write(line)
    except FileNotFoundError:
        print(f"Error: File '{input_fasta}' not found.")
input_fasta = "input.fasta"
output_fasta = "output.fasta"
extract_sequences_from_fasta("500.fas", xp_words, "500_pruned.fas")
```

Rooting the Phylogenetic Trees and Second Subtree

- FigTree (Midpoint Rooting)
- Python

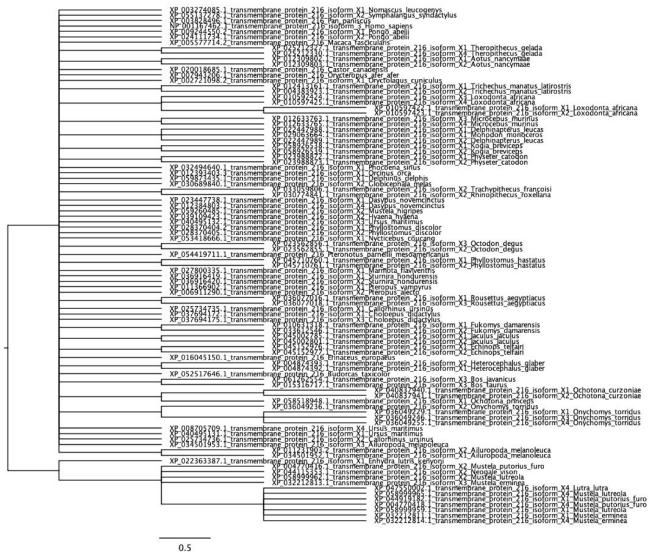


Phylogenetic tree of first subtree

```python
import re
def parse_nwk_file(file_path):
    with open(file_path, 'r') as file:
        content = file.read()
    return content

def process_nwk_file(file_path, min_bootstrap=0.7, pattern=r'[XN]P_\d+\.\d+'):
    regex = re.compile(r'\[&Bootstrap=(\d+\.\d+|\d+\.\d+|\d+)\]|([XN]P_\d+\.\d+)')
    with open(file_path, 'r') as file:
        input_nwk = file.read()
    matches = regex.finditer(input_nwk)
    current_bootstrap = None
    current_species_code = None
    output_nwk = input_nwk
    uneliminated_species = set()
    for match in matches:
        groups = match.groups()
        if groups[0] is not None:
            # Found a bootstrap value
            current_bootstrap = float(groups[0])
        elif groups[1] is not None:
            # Found a species code
            current_species_code = groups[1]
            uneliminated_species.add(current_species_code)
            if current_bootstrap is not None and current_bootstrap < min_bootstrap:
                # Remove clade if bootstrap is less than 0.7
                uneliminated_species.remove(current_species_code)
    return list(uneliminated_species)
uneliminated_species_list = process_nwk_file("subtree_pruned1.nwk")

def extract_sequences_from_fasta(input_fasta, xp_identifiers, output_fasta):
    try:
        with open(input_fasta, 'r') as input_file, open(output_fasta, 'w') as output_file:
            in_xp_list = False
            for line in input_file:
                if line.startswith('>'):
                    identifier = line.strip()[1:]
```

Rooting the Phylogenetic Trees

- FigTree (Midpoint Rooting)



Phylogenetic tree of second subtree

Conservation Score Calculation

- Python

```python
filename = "../conservation_score/500subtree_aligned_fasta.fas"
sequence_dict = fastareader(filename)
alignment_sequences = list(sequence_dict.values())
amino_acids = ["A", "R", "N", "D", "C", "Q", "E", "G", "H", "I", "L", "K", "M", "F", "P", "S", "T", "W", "Y", "V"]

consensus_sequence = ""
consensus_scores = {}
for position in range(len(alignment_sequences[0])):
    amino_acid_percentages = dict.fromkeys(amino_acids, 0)
    amino_acids_at_position = ""
    for seq in alignment_sequences:
        amino_acids_at_position += seq[position]
    for amino_acid in amino_acids:
        count_amino_acid = amino_acids_at_position.count(amino_acid)
        amino_acid_percentages[amino_acid] = count_amino_acid / len(alignment_sequences)
    amino_acid_percentages = dict(sorted(amino_acid_percentages.items(), key=lambda x: x[1], reverse=True))
    consensus_amino_acid = list(amino_acid_percentages.keys())[0]
    max_amino_acid_score = list(amino_acid_percentages.values())[0]
    consensus_sequence += consensus_amino_acid
    consensus_scores[position] = {consensus_amino_acid: max_amino_acid_score}

print("Consensus Sequence:", consensus_sequence)
print("\nAmino Acid Scores:")
for pos, data in consensus_scores.items():
    print(f"Position {pos + 1}: {data}")

conservation_scores = [sum(data.values()) for data in consensus_scores.values()]
import matplotlib.pyplot as plt

plt.hist(conservation_scores, bins=20, color='blue', edgecolor='black')
plt.xlabel('Conservation Score')
plt.ylabel('Number of positions')
plt.title('Conservation Score Histogram')
plt.show()
```

Mutation Analysis

Mutations were retrieved from three different sources;

- Clinical Papers
  - Absolute pathogenic mutations were retrieved
- ClinVar Database
  - Variation and phenotype relationship
- GnomAD Database
  - Population Frequencies

Mapping the Mutations and Calculating Threshold Values

- Python

```python
import pandas as pd

def fastareader(filename):
    seqDict = {}
    infile = open(filename, 'r')
    for line in infile:
        if line[0] == ">":
            header = line.strip()[1:]
            seqDict[header] = ""
        else:
            seqDict[header] += line.strip()
    return seqDict

def map_mutation_position(aligned_filename):
    filename = "tmem216_proteinSequenceIso3.fas"
    original_sequence = list(fastareader(filename).values())[0]
    sequence_dict = fastareader(aligned_filename)
    aligned_sequence = sequence_dict['NP 001167462.1 transmembrane protein 216
    originalPos_alignedPos_dict = {}
    modified_aligned = aligned_sequence
    for i in range(len(original_sequence)):
        aa = original_sequence[i]
        position = modified_aligned.index(aa)
        originalPos_alignedPos_dict[i] = position
        modified_aligned = modified_aligned[:position] + "-" + modified_aligne
    return originalPos_alignedPos_dict
```

Mapping the mutations

```python

def main():
    filename = "paper_mutations.txt"
    mutations = open(filename, 'r').read()
    mutations = mutations.split(",")
    position_list = []
    #
    filename = "tmem216_proteinSequenceIso3.fas"
    original_sequence = list(fastareader(filename).values())[0]
    #
    for mutation in mutations:
        substr = mutation[3:]
        if substr[1].isalpha(): #Then the mutation position is a 1-digit number
            position = substr[0]
        elif substr[2].isalpha(): #Then the mutation position is a 2-digit number
            position = substr[:2]
        else: #Otherwise
            position = substr[:3]
        if int(position) <= len(original_sequence):
            position_list.append(int(position))
    aligned_filename = "500_aligned.fas"
    position_dict = map_mutation_positions.map_mutation_position(aligned_filename)
    conservation_scores = pd.read_csv("output.tsv", sep = "\t")
    cons_score_list = mutation_pos_cons_score(position_list, position_dict, conservation_scores)
    #
    aligned_filename = "500subtree_aligned_fasta.fas"
    position_dict = map_mutation_positions.map_mutation_position(aligned_filename)
    conservation_scores = pd.read_csv("output2.tsv", sep = "\t")
    cons_score_list2 = mutation_pos_cons_score(position_list, position_dict, conservation_scores)
    #
    aligned_filename = "500_subtree_pruned.fas"
    position_dict = map_mutation_positions.map_mutation_position(aligned_filename)
    conservation_scores = pd.read_csv("output3.tsv", sep = "\t")
    cons_score_list3 = mutation_pos_cons_score(position_list, position_dict, conservation_scores)
```

Calculating Threshold Values

Classification/Variant Analysis of Variants in TMEM216 Gene

- Python

76	562	p.Thr78Ala	0.9719298245614036	True	4.793148263031543e-06
77	563	p.Thr78Arg	0.9719298245614036	True	6.570561257342603e-06
78	564	p.Gly80Glu	1.0	True	1.5933563413989032e-06
79	565	p.Asn81Lys	0.9929824561403509	True	1.5930060661671e-06
80	566	p.Cys83Tyr	1.0	True	3.185098833616807e-06
81	572	p.Lys86Gln	0.9894736842105264	True	3.6009996374993697e-06
82	576	p.Pro88Thr	0.9614035087719298	True	6.843043840644669e-07
83	579	p.Leu89Val	0.9929824561403509	True	3.4212456618605007e-06
84	580	p.Leu89Phe	0.9929824561403509	True	5.3293871080886465e-05
85	582	p.Ser90Gly	0.7789473684210526	False	3.6010082823190495e-06
86	583	p.Ser90Asn	0.7789473684210526	False	6.842266596601857e-07
87	584	p.Ser90Thr	0.7789473684210526	False	1.3684533193203714e-06
88	586	p.Ile91Met	0.968421052631579	True	2.7368317341113233e-06
89	587	p.Ser92Thr	0.9894736842105264	True	6.841938896012108e-07

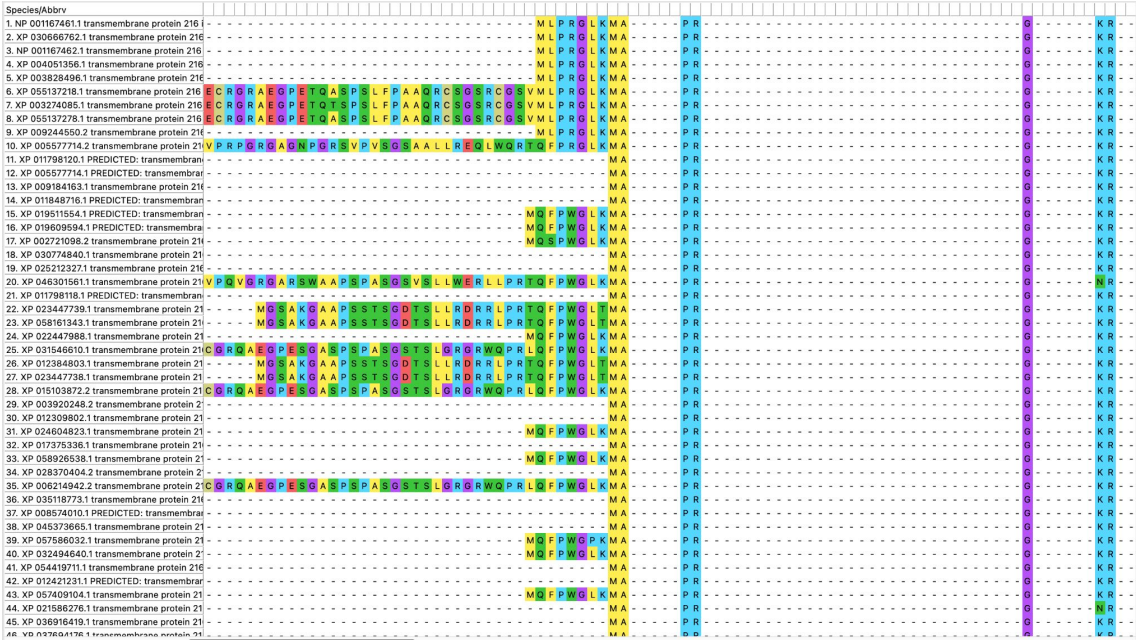
```
variants = pd.read_csv("gnomAD_all.csv")
cols_to_keep = ['HGVS Consequence', 'VEP Annotation', 'ClinVar Clinical Significance', 'Allele Count', 'Allele Number']
variants = variants[cols_to_keep]

missense_variants = variants.loc[variants['VEP Annotation'] == 'missense_variant']
missense_variants = missense_variants.loc[missense_variants['ClinVar Clinical Significance'].isnull()]

position_list = []
for i in list(missense_variants.index):
    row = missense_variants.loc[i]
    mutation = row[0]
    substr = mutation[5:]
    if substr[1].isalpha(): #Then the mutation position is a 1-digit number
        position = substr[0]
    elif substr[2].isalpha(): #Then the mutation position is a 2-digit number
        position = substr[:2]
    else: #Otherwise
        position = substr[:3]
    position_list.append(int(position))

conservation_thresholds = open("conservation_thresholds.txt", 'r').read().strip().split(" ")
conservation_thresholds = [float(score) for score in conservation_thresholds]

aligned_filename = "500_aligned.fas"
position_dict = map_mutation_positions.map_mutation_position(aligned_filename)
conservation_scores = pd.read_csv("output.tsv", sep = "\t")
cons_score_list = known_mutations.mutation_pos_cons_score(position_list, position_dict, conservation_scores)
threshold1 = conservation_thresholds[0]
isPathogenic = []
for score in cons_score_list:
    if score >= threshold1:
        isPathogenic.append(True)
    else:
        isPathogenic.append(False)

missense_variants1 = missense_variants
missense_variants1['cons_score'] = cons_score_list
missense_variants1['isPathogenic'] = isPathogenic
```

Statistical Testing: t-test

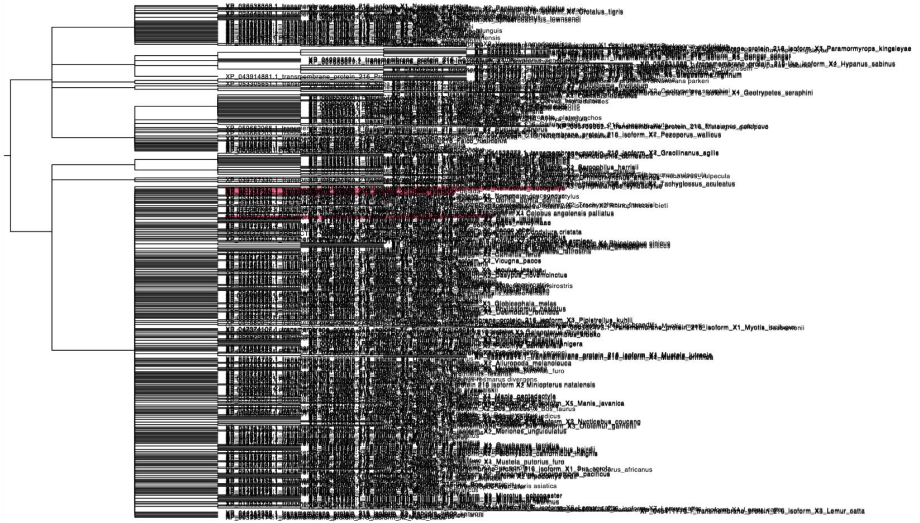
- Python

```python
from scipy import stats as st
import pandas as pd


outputList=["classification_output.csv","classification_output2.csv","classification_output3.csv"]
for i in outputList:
    df = pd.read_csv(i)

    a = df.loc[df['isPathogenic'] == True, 'Allele_frequency'].to_numpy()
    b = df.loc[df['isPathogenic'] == False, 'Allele_frequency'].to_numpy()

    pvalue = st.ttest_ind(a=a, b=b, equal_var = True).pvalue

    print(f"P-value for {i}: {pvalue}")
```

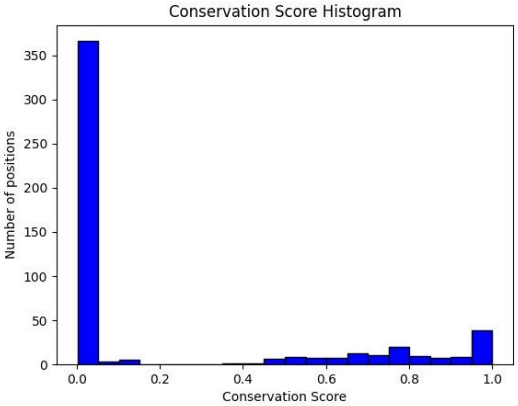
500 Aligned Sequence



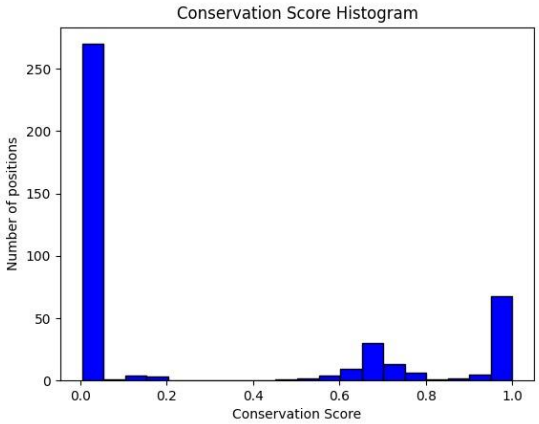
TMEM216 isoforms



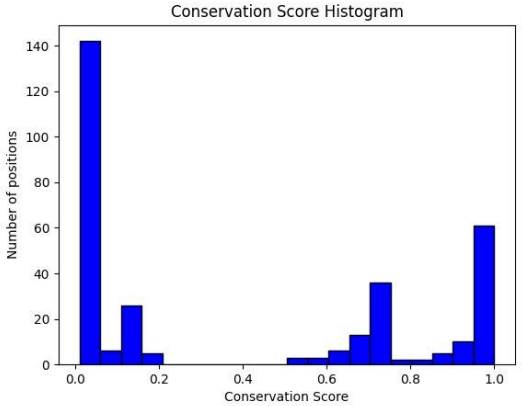
Conservation Score Distributions for Different Aligned Sequences



Distribution of 500 aligned sequences



Distribution of pruned aligned sequences



Distribution of second time pruned aligned sequences

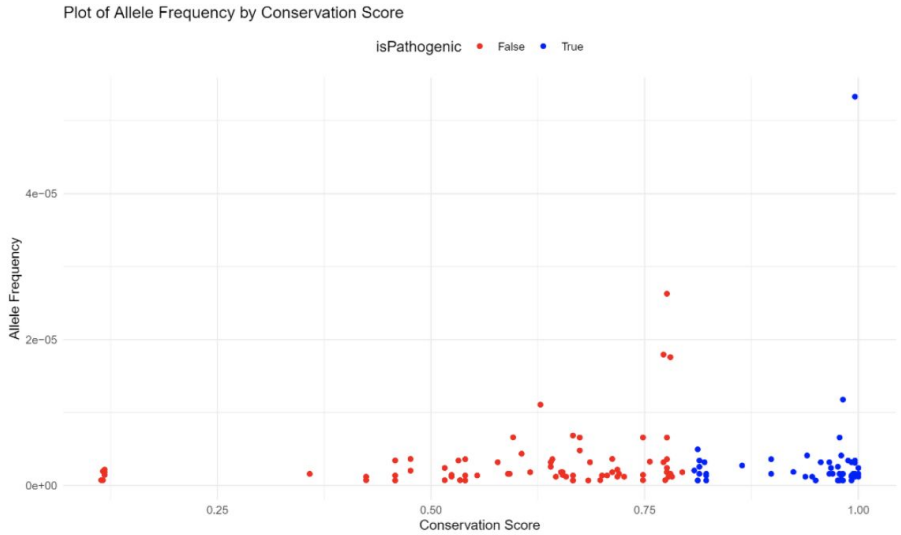


Fig.28 Conservation Scores and Allele Frequency scatter plot of mutations for 500 hits data

Classification Results

1.Output file

		HGVS Consequence	cons_score	isPathogenic	Allele_frequency
1	85	p.Leu2Arg	0.114	False	7.232666192203764e-07
2	90	p.Arg4Trp	0.116	False	7.226812050853631e-07
3	91	p.Arg4Gly	0.116	False	1.9532267305588833e-06
4	94	p.Gly5Arg	0.118	False	2.1679527444113793e-06
5	99	p.Leu6Met	0.118	False	1.445308096326894e-06
6	100	p.Leu6Pro	0.118	False	1.8227951469902007e-06
7	103	p.Met8Thr	0.652	False	1.8163192653351836e-06
8	104	p.Ala9Ser	0.654	False	1.8158707100054476e-06
9	106	p.Ala9Glu	0.654	False	1.4464892981489276e-06
10	109	p.Arg11Pro	0.616	False	1.8162664826183299e-06
11	208	p.Gly12Val	0.534	False	7.23194436609838e-07
12	209	p.Gly12Asp	0.534	False	7.23194436609838e-07

1.Threshold Value:
0.8074285714285713

2.Output file

		HGVS Consequence	cons_score	isPathogenic	Allele_frequency
1	85	p.Leu2Arg	0.1298245614035087	False	7.232666192203764e-07
2	90	p.Arg4Trp	0.1333333333333333	False	7.226812050853631e-07
3	91	p.Arg4Gly	0.1333333333333333	False	1.9532267305588833e-06
4	94	p.Gly5Arg	0.1719298245614035	False	2.1679527444113793e-06
5	99	p.Leu6Met	0.1578947368421052	False	1.445308096326894e-06
6	100	p.Leu6Pro	0.1578947368421052	False	1.8227951469902007e-06
7	103	p.Met8Thr	0.6175438596491228	False	1.8163192653351836e-06
8	104	p.Ala9Ser	0.6140350877192983	False	1.8158707100054476e-06
9	106	p.Ala9Glu	0.6140350877192983	False	1.4464892981489276e-06
10	109	p.Arg11Pro	0.6035087719298246	False	1.8162664826183299e-06
11	208	p.Gly12Val	0.519298245614035	False	7.23194436609838e-07
12	209	p.Gly12Asp	0.519298245614035	False	7.23194436609838e-07
13	215	p.Lys13Glu	0.6175438596491228	False	7.230835754458172e-07
14	216	p.Lys13Arg	0.6175438596491228	False	1.2006300906715843e-06
15	217	p.Arg14Gly	0.6631578947368421	False	7.230532514258615e-07

2.Threshold Value:
0.8401002506265665

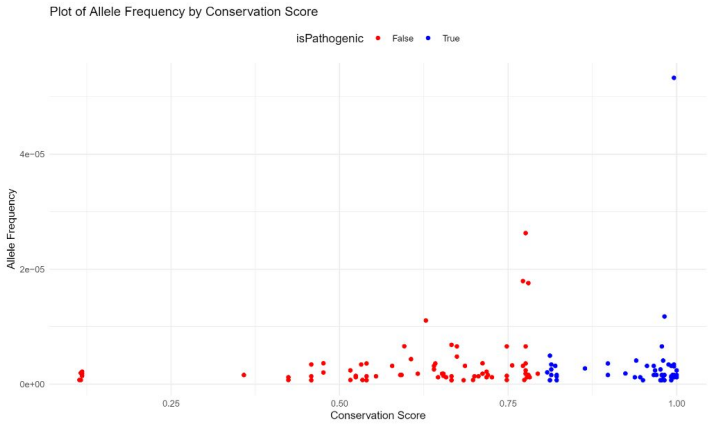
3.Output file

		HGVS Consequence	cons_score	isPathogenic	Allele_frequency
1	85	p.Leu2Arg	0.1041666666666666	False	7.232666192203764e-07
2	90	p.Arg4Trp	0.1041666666666666	False	7.226812050853631e-07
3	91	p.Arg4Gly	0.1041666666666666	False	1.9532267305588833e-06
4	94	p.Gly5Arg	0.1875	False	2.1679527444113793e-06
5	99	p.Leu6Met	0.1666666666666666	False	1.445308096326894e-06
6	100	p.Leu6Pro	0.1666666666666666	False	1.8227951469902007e-06
7	103	p.Met8Thr	0.6458333333333334	False	1.8163192653351836e-06
8	104	p.Ala9Ser	0.6666666666666666	False	1.8158707100054476e-06
9	106	p.Ala9Glu	0.6666666666666666	False	1.4464892981489276e-06
10	109	p.Arg11Pro	0.6458333333333334	False	1.8162664826183299e-06
11	208	p.Gly12Val	0.6145833333333334	False	7.23194436609838e-07
12	209	p.Gly12Asp	0.6145833333333334	False	7.23194436609838e-07
13	215	p.Lys13Glu	0.6875	False	7.230835754458172e-07

3.Threshold Value:
0.8705357142857143

Conservation Score and Allele Frequency Plots

- R Studio




```r
library(readr)
library(ggplot2)
library(dplyr)

df <- read_csv('classification_output.csv', show_col_types = FALSE)
df <- df %>% select(-...1)

df$isPathogenic <- factor(df$isPathogenic, labels = c("False", "True"))

p <- ggplot(df, aes(x = cons_score, y = Allele_frequency, color = isPathogenic)) +
  geom_point() +
  scale_color_manual(values = c("False" = "red", "True" = "blue")) +
  labs(
    x = 'Conservation Score',
    y = 'Allele Frequency',
    color = 'isPathogenic',
    title = 'Plot of Allele Frequency by Conservation Score'
  ) +
  theme_minimal() +
  theme(
    legend.title = element_text(size = 12),
    legend.position = "top"
  )

ggsave('allele_freq_corr_s=500_r.pdf', p, width = 10, height = 6, units = 'in')

print(p)
```

Discussion

- As the number of known pathogenic mutations increases, the classification of unknown mutations becomes more accurate.
- As the genetic relevance of the sequences increases in a set of sequences, conservation scores and the threshold value also increase.
- Number of mutations which identified as pathogenic and the classification thresholds are very close for the second and third data.
- With the observation of p-values, there is no statistically significant difference in allele frequencies between pathogenic and nonpathogenic variants.
- According to allele frequency scatter plot of mutations there is no negative correlation between conservation scores and allele frequencies.

Issues

- Why sequence file of 5000 was not used?
- Why sequence files of 100 and 250 were not used?
- Why sequence file of 1000 was not used?

References

Agnihotry, S., Pathak, R. K., Singh, D. B., Tiwari, A., & Hussain, I. (2022). Protein structure prediction. *Bioinformatics*, 177–188. https://doi.org/10.1016/b978-0-323-89775-4.00023-7

ClinVar. (n.d.). *TMEM216[gene] - ClinVar - NCBI*. https://www.ncbi.nlm.nih.gov/clinvar/?term=TMEM216%5Bgene%5D&redir=gene

Edvardson, S., Shaag, A., Zenvirt, S., Erlich, Y., Hannon, G. J., Shanske, A., Gomori, J. M., Ekstein, J., & Elpeleg, O. (2010c). Joubert Syndrome 2 (JBTS2) in Ashkenazi Jews Is Associated with a TMEM216 Mutation. *The American Journal of Human Genetics*, *86*(2), 294. https://doi.org/10.1016/j.ajhg.2010.01.022

FigTree. (n.d.). http://tree.bio.ed.ac.uk/software/figtree/

GnomAD. (n.d.). *gnomAD Browser*. https://gnomad.broadinstitute.org/gene/ENSG00000187049?dataset=gnomad_r4

Hillis, D. M., & Bull, J. J. (1993). An empirical test of bootstrapping as a method for assessing confidence in phylogenetic analysis. *Systematic Biology*, *42*(2), 182–192. https://doi.org/10.1093/sysbio/42.2.182

Koichiro Tamura, Glen Stecher, and Sudhir Kumar (2020) MEGAX: Molecular Evolutionary Genetics Analysis version X. *Molecular Biology and Evolution* 38:3022-3027. https://www.megasoftware.net/

OMIM. *Transmembrane protein 216; TMEM216*. (n.d.). https://www.omim.org/entry/613277

Parisi, M. (2017, June 29). *Joubert Syndrome*. GeneReviews - NCBI Bookshelf. https://www.ncbi.nlm.nih.gov/books/NBK1325/

Protein BLAST: search protein databases using a protein query. (n.d.). https://blast.ncbi.nlm.nih.gov/Blast.cgi?PROGRAM=blastp&PAGE_TYPE=BlastSearch&LINK_LOC=blasthome

UniProt. (n.d.). https://www.uniprot.org/

U.S. National Library of Medicine. (2017, July 1). *Joubert syndrome*. MedlinePlus. https://medlineplus.gov/genetics/condition/joubert-syndrome/#:~:text=Joubert%20syndrome%20typically%20has%20an,and%20symptoms%20of%20the%20condition

U.S. National Library of Medicine. (n.d.). *Joubert syndrome 2* . National Center for Biotechnology Information. https://www.ncbi.nlm.nih.gov/medgen/334114

U.S. National Library of Medicine. (n.d.). *TMEM216[gene] - ClinVar - NCBI*. https://www.ncbi.nlm.nih.gov/clinvar/?term=TMEM216%5Bgene%5D&redir=gene

U.S. National Library of Medicine. (n.d.). *TMEM216 transmembrane protein 216 [homo sapiens (human)]* . National Center for Biotechnology Information. https://www.ncbi.nlm.nih.gov/gene/51259

U.S. National Library of Medicine. (2021, March 26). *What are proteins and what do They do?*. MedlinePlus. https://medlineplus.gov/genetics/understanding/howgeneswork/protein/

Valente, E. M. et.al.(2010). Mutations in TMEM216 perturb ciliogenesis and cause Joubert, Meckel and related syndromes. *Nature Genetics*, *42*(7), 619–625. https://doi.org/10.1038/ng.594