Usher Syndrome and the evolution of microvillar sensory structures

Hayden Speck

March 9, 2017

1 Introduction

Usher Syndrome (USH), a genetic sensory disorder, is the most common cause of combined blindness-deafness in humans. The genes associated with Usher syndrome play key structural and functional roles in ciliated sensory cells of the vertebrate retina and inner ear. Usher genes form interciliary links and their anchoring complexes in photoreceptors and the mechanosensory hair cell ([**?**]). When a mutation occurs in one of these genes, mechanotransduction is abolished and the retina degenerates, resulting in blindness, deafness and impaired vestibular function.

Given the key role these genes play in vertebrate sensory structures, it is concievable that these genes may serve similar sensory functions in other Metazoan groups. Previously thought to be confined to vertebrates, USH homologs were identifed within the genome of the Echinoderm *Strongylocentrotus* ([**?**]). Recently, USH homologs have been shown to be upregulated in the choanocytes of the sponge *Ephydatia*, hinting that these genes may play a conserved role in the evolution of ciliated sensory structures of the Metazoa, and begging the question of how early these genes arose ([**?**]). By investigating the evolutionary history of the genes involved in Usher syndrome, this project can better determine how the suite of genes involved with Usher syndrome were assembled within the Metazoa and its close relatives, and what role these genes might have played in the sensory evolution of early animals.

2 Methods

To investigate the evolutionary history of Usher syndrome associated genes, we can build gene trees.

2.1 Overview

BLAST human sequences against select organism sequence databases on NCBI.

Parse the XML files recieved from NCBI to easily summarize the search results.

Gather Gene IDs from output, and download from NCBI

Download Sequences

Align sequences for single gene with MUSCLE

Build a tree with RAxML

Read format and label the tree in R

2.2 Code

Remote BLAST

```
def search_taxa_all_gene_delay(list_of_taxa):
    # blasts sequences in a file against a list of taxa
    # loop through the list and run blast for each one
    # will save each result to a separate xml file

    from Bio.Blast import NCBIWWW
# imports the NCBIWWW module to allow remote Searching

    import time
# delays imputs to the NCBI servers and get kicked off



    with open("USH_Search_seq.fasta", "r") as fasta_file:
        sequences = fasta_file.read()
        fasta_file.close()
        #reads in sequences we will be searching
```

```
    for i in list_of_taxa:

        result_handle = NCBIWWW.qblast("blastp",
# specifies the program for a protein-protein search
                                       "refseq_protein",
#  database of protein sequences
                                       sequences,
# our list of sequences we read in
                                       alignments = 100,
# asks for 100 best hits
                                       descriptions = 100,
                                       expect = 0.00001,
# specifies max E-value(likelihood of a random match for our query)
                                       entrez_query = str(i))
# specifies the taxa as we loop through it


        file_name=str("USH_Search_"+str(i)+".xml")
#this creates a name for the file

        save_file=open(file_name, "w")
#we are opening a file that does not yet exist to write to it

        save_file.write(result_handle.read())
#writing the result of our blast search to local file

        save_file.close()
#closing it to allow the file to actually write it

        result_handle.close()
#close the results handle



        print("created "+ file_name)
```

```
#this is just a nice way to track the progress of the program


        time.sleep(60)
# 1 minute delay writing the output and requests to the ncbi server

# NCBI is a shared resource, shouldn't monopolize computer time

Here is the list of taxa:



full_taxa_file_name=open("/home/eeb177-student/Desktop/eeb-177/
project/sandbox/Testrun_multi_genes_same_org/
full_list_taxa_NCBI.txt", "r")
```

Parsing the XML output

```
def parse_and_summarize(blast_output_xml):
# goes through the output of a BLAST xml file
# finds the relevant stats to summarize the search
    from Bio.Blast import NCBIXML
    from Bio.SeqRecord import SeqRecord
    #import the required libraries

    for file_name in blast_output_xml:
        result_handle = open(str(file_name), "r")
# sets the result handle

        blast_records = NCBIXML.parse(result_handle)
        #need to use parse if it has multiple records in it

        for blast_record in blast_records:
            org_desig=file_name.split("_")[2]\
.split("[")[0].replace(" ", "_")
 #properly formats the taxa id so to name things

            homo_sapiens = "[Homo sapiens]"
```

```
            blast_query=blast_record.query
            if homo_sapiens in blast_query:
                gene_name=blast_record.query.split("|")[4]\
.split("[")[0]\.replace(" ", "_").replace("_protein", "")\
.replace("_isoform_b3", "")
                formated_gene_name = gene_name[1:-1]
            else:
                formated_gene_name=blast_query.split("|")\
[4].split("_")[0]
#this conditional formats the gene name
#switches between two formats in spliting the name
```