MUSCLE_Gblocks_FastTree_V2.2

July 31, 2019

1 Create species tree based on a meta-alignment of 1-to-1 orthologs

Given a directory with OrthoGropus containg the protein sequences, this script align them with MUCLE, trim the alignemnts with GBlocks, concatenates the alignments and infers a phylogenetic tree with FastTree.

```
In [ ]: # Load libs
        import os, subprocess, sys,glob
        from Bio.Seq import Seq
        from Bio import SeqIO,AlignIO, Phylo
        from collections import defaultdict
```

1.1 Individual alignments + Gblocks

The 1-to-1 orthologs will be aligned, and their alignments polished with GBlocks.

```
mkdir v2.2
mkdir v2.2/MUSCLE_out/

cp -r  ~/data_disk/Cricket_genome_annotation/Comparative_Genomics_V2/Orthofinder/Orthofinder_i

ls v2.2/Single_Copy_Orthologue_Sequences | wc -l
#  732 # Single copy orthologs
```

```
In [ ]: maindir=os.path.join(os.getcwd(),"v2.2")
        inputdir=os.path.join(maindir,"Single_Copy_Orthologue_Sequences")
        output_align_dir=os.path.join(maindir,"MUSCLE_out")

        ##Number of files
        number=0
        for file in os.listdir(inputdir):
            number=number+1
        print("Number of files=",number)
```

```
In [ ]: # for each file on the directory

        for file in os.listdir(inputdir):
```

```
    #print(os.path.join( inputdir,file))
    inputfas=os.path.join(inputdir,file)

    muscle_out=os.path.join(output_align_dir,os.path.splitext(file)[0])

    #run MUSCLE
    MUSCLEcommand="~/data_disk/Software/muscle3.8.31_i86linux64 -in %s -out %s"
    subprocess.run(MUSCLEcommand % (inputfas, muscle_out) , shell=True)

    # run GBlocks
    Gblockcommand="~/data_disk/Software/Gblocks_0.91b/Gblocks %s -t=p  -b4=5 -b5=a"
    subprocess.run(Gblockcommand % muscle_out , shell=True)
```

1.2 Join the alignments in a single file

```
In [ ]: # Concatenate Alignments
        ## https://yueyvettehao.github.io/2018/09/using-biopython-to-concatenate-aligned-seque

        Alignemntfileslist=[]

        for file in os.listdir(output_align_dir):
            if file.endswith('-gb'):
                print(file)
                Alignemntfileslist.append(os.path.join(output_align_dir,file))

        Alignemntfileslist

        # Join all fasta alignments  in single file
        All_alignments=os.path.join(output_align_dir,"All_alignments.fa")

        catcommand="cat %s > %s" % (' '.join(Alignemntfileslist),All_alignments )

        print(catcommand)

        subprocess.run(catcommand, shell=True)
```

1.3 Concatenate the alignments in a Meta-alignment

```
In [ ]: sequence_map = defaultdict(str)


        #All_alignments="/home/guillem/data_disk/Cricket_genome_annotation/Comparative_Genomic
        #output_handle = open("Example.fa", "w")
        output_handle = open(os.path.join(output_align_dir,"Meta_alignment.fa"), "w")

        for sequence in SeqIO.parse(All_alignments, "fasta"):
```

```
        sequence.name=sequence.name.split("_")[0]  #  only retain spp name
        sequence_map[sequence.name] += str(sequence.seq)


    tmp= open("tmp","w+")

    for key, seq in sequence_map.items() :
        #print (key)
        tmp.write("".join((">",key) ))
        tmp.write("\n")
        tmp.write(seq)
        tmp.write("\n")
        #print ("".join((">",key) ))
        #print (seq)
    tmp.close()

    ## Reformat to good fasta file
    alignments = AlignIO.parse("tmp", "fasta")
    AlignIO.write(alignments, output_handle, "fasta")
    output_handle.close()
    os.remove("tmp")
```

1.4 Run FastTree

I used the FastTreMP whichuses multiple threads

```
In [ ]: metalignment_file= os.path.join(output_align_dir,"Meta_alignment.fa")
        outtree=os.path.join(maindir, "Fasttree_out.tree")

        Fasttree_Command="/home/guillem/data_disk/Software/FastTreeMP -gamma < %s > %s" % (meta

        print(Fasttree_Command)

        Fasttree_result =subprocess.run(Fasttree_Command , shell=True,  stdout=subprocess.PIPE

In [19]: #print("Stdout:\n",str(Fasttree_result))
         #print("Stderr:\n",str(Fasttree_result))
         ## Rename tree with full species names:


         dictionary_spp_file= {"Api" : "Acyrthosiphon_pisum",
                 "Fca" : "Folsomia_candida",
                 "Lmi" : "Locusta_migratoria",
                 "Rma" : "Rhopalosiphum_maidis",
                 "Ago" : "Aphis_gossypii",
                 "Msa" : "Melanaphis_sacchari",
                 "Sfl" : "Sipha_flava",
                 "Bge" : "Blattella_germanica",
```

```
                "Gbi" : "Gryllus_bimaculatus",
                "Mpe" : "Myzus_persicae",
                "Zne" : "Zootermopsis_nevadensis",
                "Cle" : "Cimex_lectularius",
                "Hha" : "Halyomorpha_halys",
                "Nlu" : "Nilaparvata_lugens",
                "Cse" : "Cryptotermes_secundus",
                "Lst" : "Laodelphax_striatella",
                "Oci" : "Orchesella_cincta",
                "Dci" : "Diaphorina_citri",
                "Lko" : "Laupala_kohalensis",
                "Dme" : "Drosophila_melanogaster",
                "Tca" : "Tribolium_castaneum",
                "Bmo" : "Bombyx_mori",
                "Ame" : "Apis_mellifera",
                "Foc" : "Frankliniella_occidentalis",
                "Phu" : "Pediculus_humanus"
                            }

        ### Tree, from Abbrebiations to full Names

        treefile=open(outtree, "r")

        outtreerenamed=open(os.path.join(maindir, "Fasttree_out_fullnames.tree"), "w+")

        for line in treefile:
            for abb, spp in dictionary_spp_file.items():
                #print(abb, spp)
                line = line.replace(abb, spp)
            print(line.lower())
            #outtreerenamed.write(line.lower())
            outtreerenamed.write(line)

        outtreerenamed.close()
```

```
(((blattella_germanica:0.09437,(cryptotermes_secundus:0.04688,zootermopsis_nevadensis:0.04993)
```

1.5 Visualize FastTree Tree

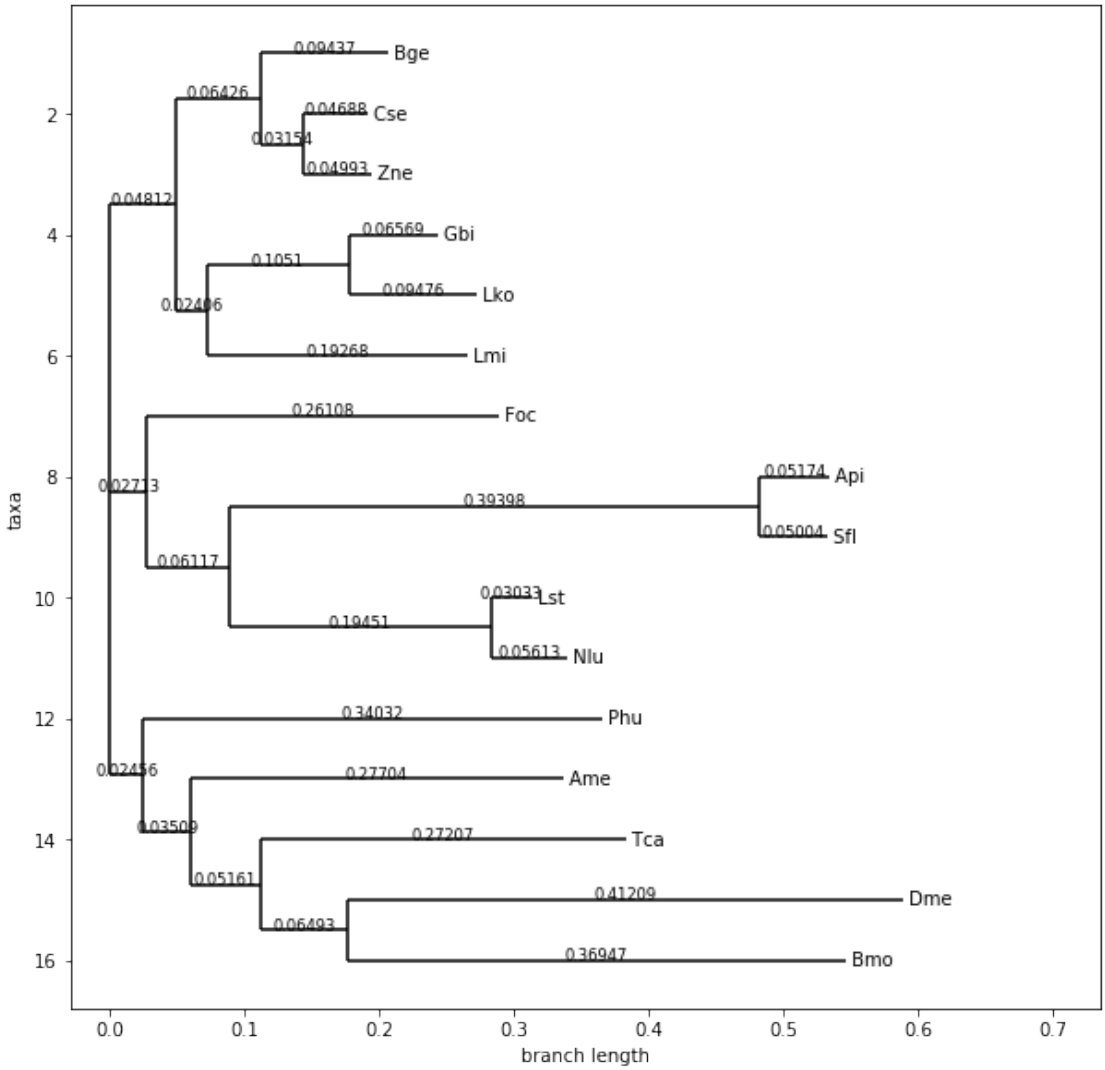
```
In [20]: tree = Phylo.read(outtree, 'newick')
         #print(tree)

         import matplotlib.pyplot as plt

         plt.rcParams['figure.figsize'] = [10, 10]
```

```
Phylo.draw(tree, branch_labels=lambda c: c.branch_length)

#Phylo.draw_ascii(tree)
```



##Preparing CAFE
RaxML/Fasttree r8s -> for cafe
https://groups.google.com/forum/#!searchin/hahnlabcafe/raxml|sort:date/hahnlabcafe/kEfPXEx1CN8/