GIGANTIC

A SIMPLE COMPREHENSIVE SEMIAUTOMATED PHYLOGENOMIC PIPELINE TO EXPLORE GENE AND SPECIES EVOLUTION

JAN HSIAO, LOLA CHENXI DENG, SHREK CHALASANI, AND ERIC EDSINGER

18 JAN 2022

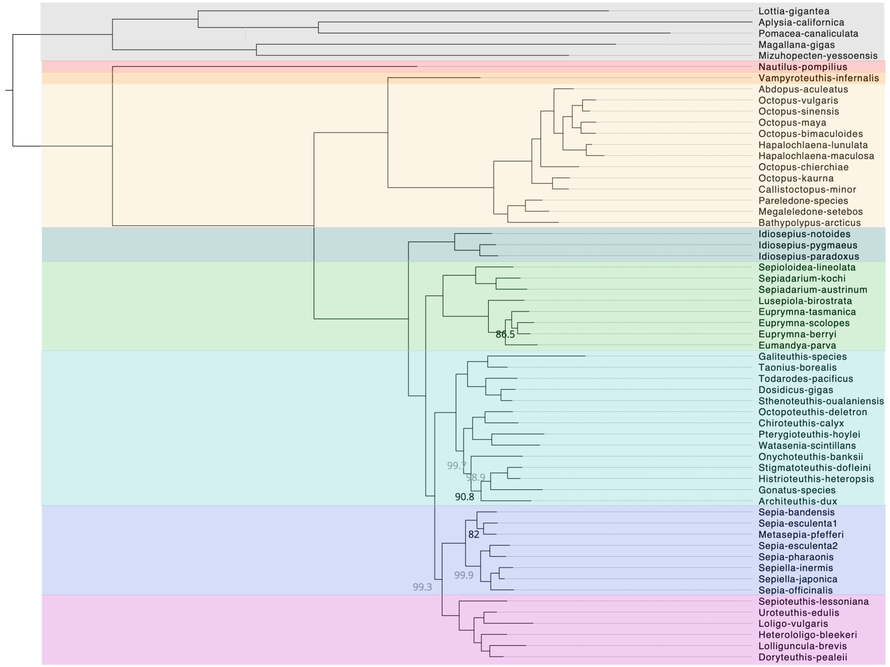
GIGANTIC takes as little as - a reference gene set fasta - and - a list of genome-sequenced species - and produces - rigorous gene and species phylogenetic analyses.

GIGANTIC functionality includes:

1) Integration of public and user-provided genomes
2) Accurate sensitive genome-scale homolog identification
3) Gene family sequence identity characterization
4) Genome gene cluster / broken gene model identification
5) ML BUSCO-based superalignment or supertree species tree
6) Representative ML gene family / superfamily tree
7) Phylonames-based Clade and user-defined Group ML gene trees
8) Automated user-guided tree label and tree color annotation
9) Browser-based viewer to explore numerous trees and data
10) Cloud-friendly semi-autonomous fully accessible pipeline

GIGANTIC in ACTION: Lola's 2021 SICB talk

Ceph50: 8 Major Cephalopod Clades



Gastropods & bivalves

Nautilus
Vampyroteuthis

Octopuses

Pygmy squids

Bobtail squids

Oegopsid squids

Cuttlefishes

Myopsid squids

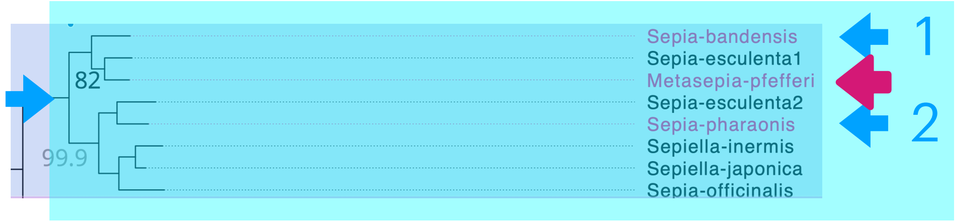
Quadrupedal Walking

Flambouyant Cuttlefish
Metasepia pfefferi

Bipedal Walking 1

Dwarf Cuttlefish
Sepia bandensis

Did quadrupedal walking evolve from arm walking in cuttlefish?



Bipedal Walking 2

Pharaoh Cuttlefish
Sepia pharaonis

GIGANTIC in ACTION: Jan's 2022 SICB poster

TRP Ion Channel Diversity And Evolution Highlights Molecular Sensors In Animal Diversification And Biology

Jan Hsiao[1,2], Lola Chengxi Deng[1,3], Sreekanth Chalasani[1], and Eric Edsinger[1,4]

[1] Salk Institute for Biological Studies, [2] University of California San Diego, [3] University of Edinburgh, [4] University of Chicago Marine Biological Laboratory

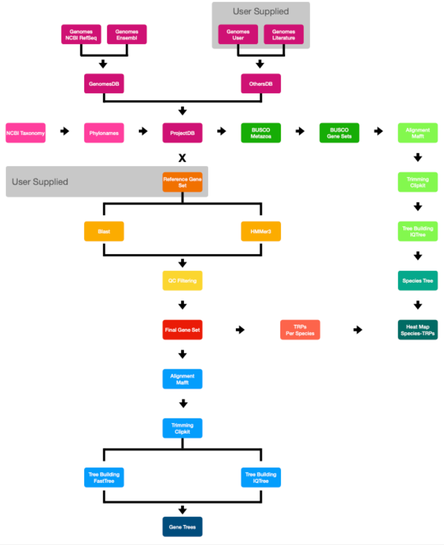
Abstract

Transient Potential Receptor (TRP) ion channels are a diverse superfamily of multimodal molecular sensors that respond to a wide variety of stimuli, including mechanical, chemical, and thermal. TRP channels are present in most eukaryotes but best understood in mammalian, worm, and fly genetic models, where they are expressed in diverse cell-types and commonly associated with the nervous system. Here, we characterized TRP superfamily gene and genome evolution to better understand origins and evolution of molecular sensors, brains, and behavior in animals. We developed a flexible push-button phylogenomics pipeline, Gigantic, that generated genome-based gene and species trees and enabled phylogenetic characterization of challenging remote homologs and distantly-related organisms deep in evolution. We identified complete sets of TRP superfamily ion channels, with over 3200 genes in 26 species representing 26 animal phyla and having publicly-available sequenced genomes, including 3 unicellular outgroups. We then identified clusters of TRP family members in genomes, evaluated gene models per cluster, and identified and repaired split gene models. In addition, we produced whole-organism PacBio transcriptomes for 3 species to independently validate our gene model assessment and repair models. We find that many TRP families exhibited numerous and often extensive expansions in different phyla. Some expansions represent local clusters on respective genome, a trend that is likely undercounted due to varied quality in genome assemblies and annotations of non-model organisms. Overall, our work expands known TRP diversity across animals, including addition of previously uncharacterized phyla and identification of unrecognized TRP family homologs in previously characterized species, with gene family expansions as possible fodder for innovation, diversification, and tuning of biological systems in animals.

Challenge

TRP and other mechanosensitive ion channels are fundamental to diverse biologies and central to emerging genetic technologies, including ultrasound-based sonogenetics, BUT gene family diversity and evolution are poorly understood.

1. We built a phylogenomic pipeline (GIGANTIC) to identify distant homologs and characterize gene & species evolution.



2. We characterized genome-scale evolution of 6 mechanosensitive ion channel gene families or superfamilies in 26 and 307 animal species.

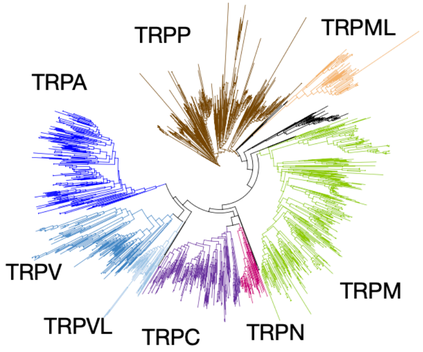
Metazoa26 Species Tree

Mechanosenitive Ion Channel Gene Family Counts Per Species

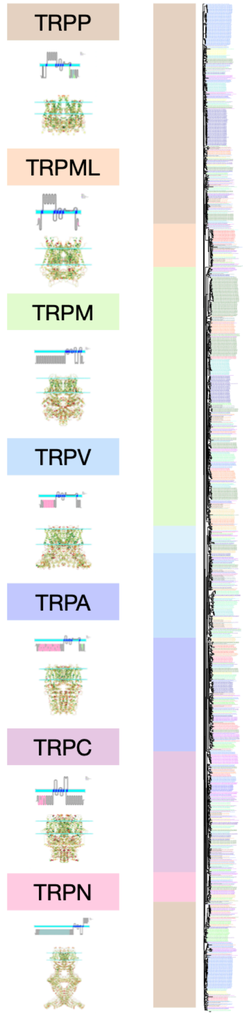
	TRP	PIEZO	ANO	TMC	DEG	KCNK
Sphaeroforma arctica	0	1	1	0	0	3
Capsaspora owczarzaki	4	1	1	1	0	0
Monosiga brevicollis	8	2	3	3	0	3
Ephydatia muelleri	34	10	7	16	0	1
Hormiphora californensis	49	1	11	6	24	21
Trichoplax adhaerens	26	1	5	5	12	1
Nematostella vectensis	72	0	12	5	30	35
Hofstenia miamia	92	7	9	0	77	15
Asterias rubens	43	2	5	2	25	22
Saccoglossus kowalevskii	91	1	7	4	36	30
Branchiostoma floridae	222	1	8	6	45	24
Styela clava	41	1	5	2	15	7
Homo sapiens	33	2	10	8	9	15
Priapulus caudatus	24	1	6	6	13	12
Ramazzottius varieornatus	35	4	9	4	0	8
Drosophila melanogaster	17	2	5	1	34	11
Caenorhabditis elegans	23	1	2	2	43	47
Adineta ricciae	181	3	13	3	42	31
Macrostomum lignano	239	9	35	8	173	23
Dicyema japonicum	9	1	1	3	0	1
Notospermus geniculatus	94	3	14	10	134	15
Phoronis australis	59	2	8	6	34	14
Lingula anatina	86	3	7	10	58	12
Bugula neritina	46	2	9	6	42	15
Crassostrea gigas	113	1	8	7	24	11
Owenia fusiformis	58	1	8	4	117	9

3. We find TRP superfamily tree structure is unstable but support for each gene family is strong.

Ultrafast Bootstrap Supported TRP Families

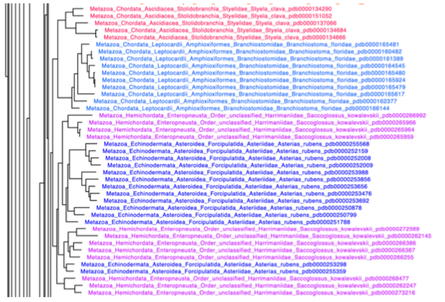


Metazoa26 TRP Superfamily

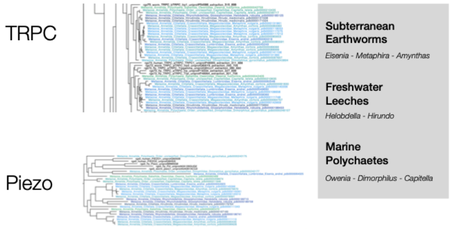


4. We also find numerous lineage-specific expansions in TRP and other mechanosensitive ion channel families.

Close-up: TRPC Lineage-specifc Expansions In Invertebrate Deuterostomes.



5. Evolution of subterranean touch-centric biology in earthworms correlates with their expansion of mechanosensory ion channels within annelids.



Future Directions

- Engineer various identified mechnosensitive ion channels to advance ultrasound-based sonogenetic technology.
- Expand ion channel evolution charcterization to other Kingdoms.
- Enable push-button automation (Juypter Notebook and SnakeMake) and stand-alone functionality (Singularity) for GIGANTIC.

Acknowledgements

We thank the National Health Institute, the Salk Institute, and the University of California San Diego for their funding and support.

GIGANTIC Features

Current databases - Latest software - User-friendly codeware

NCBI RefSeq
Ensembl

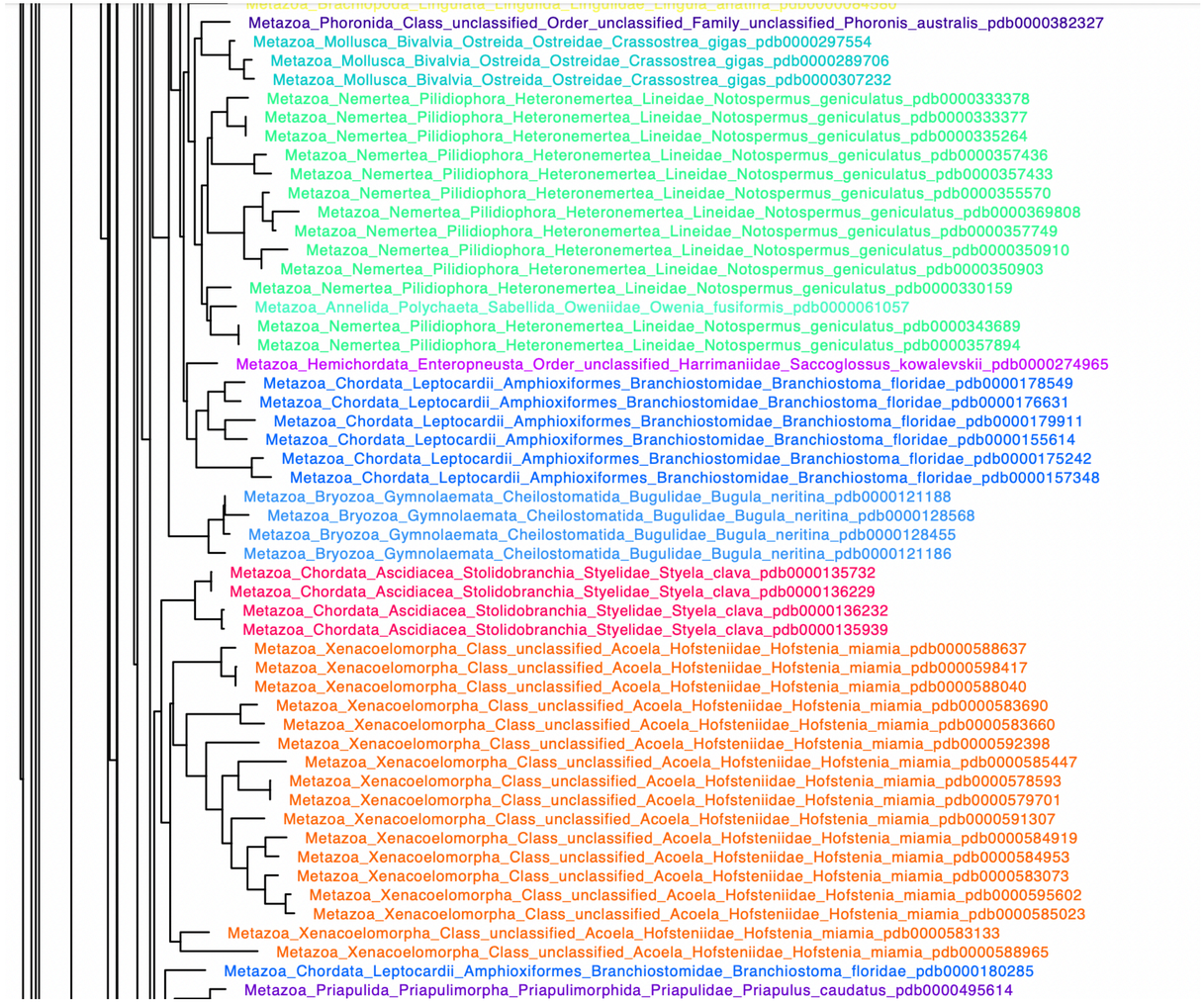
Mafft
Clipkit
FastTree / IQTree / PhyloBayes / Astral
FigTree / iTOL

Python / Unix
Jupyter Notebook / Snakemake
Singularity

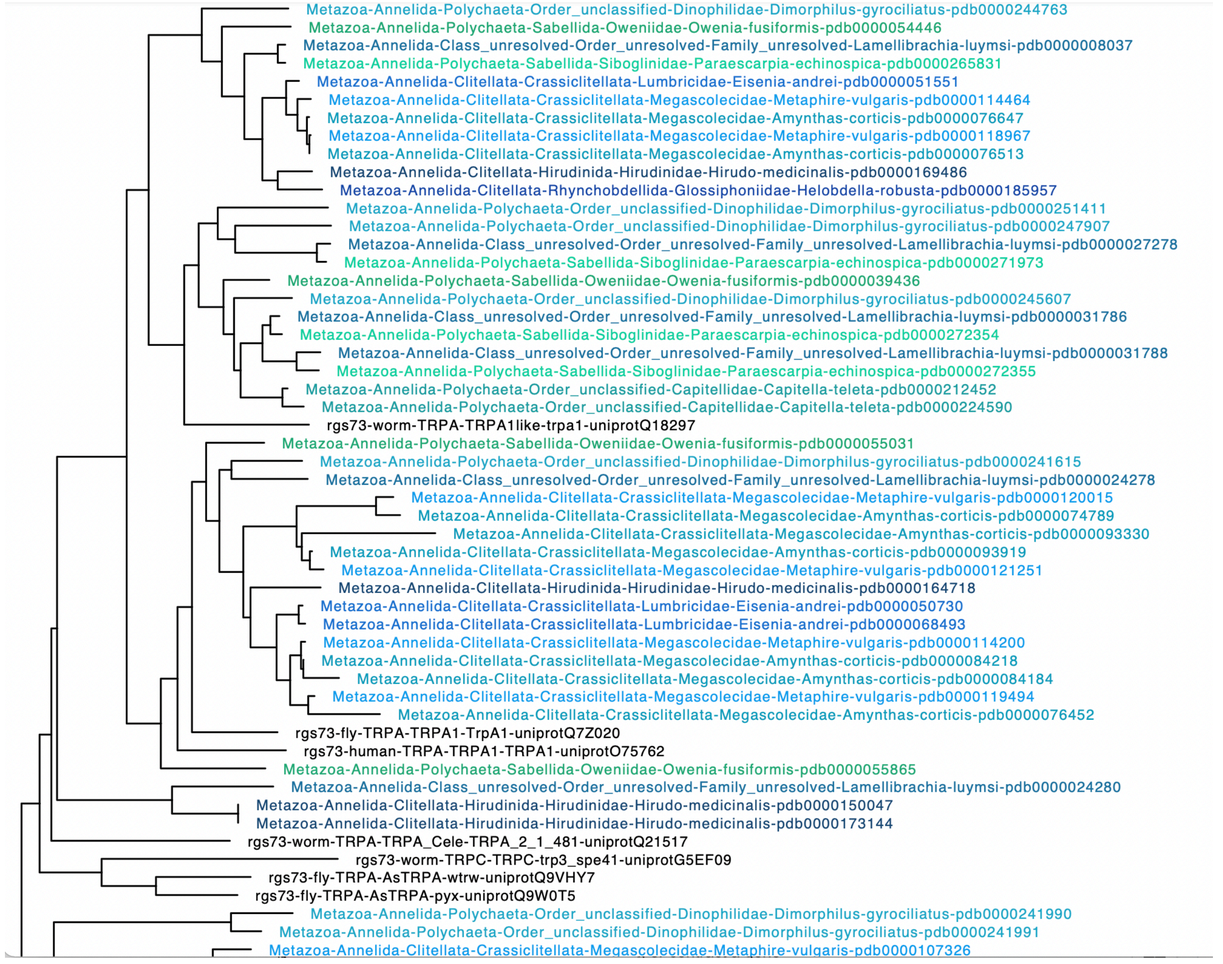
GIGANTIC Features

NCBI Taxonomy-based Phylonames enables annotation and non-expert interpretation of trees

Metazoa26 Representative TRP Superfamily Tree Closeup



Metazoa307 Clade Annelida TRP Superfamily Tree Closeup



GIGANTIC Features

Blast and a novel HMMer-based homolog detection strategy (giganticHMMer) identify diverse and distant sequences

giganticHMMer

Phase 1

Consolidate projectdb genome and reference genomes.
Collect reference sequence fastas.
Choose keyword for genome annotation parameter file.
Update homolog filtering parameter file.

Phase 2

Annotate reference genomes.
Concatenate all genomes.
Build search database.

Phase 3

Set desired HMMER jackhmmer protein substitution matrix.
Set maximum number of jackhmmer rounds per sequence query.
Set sequence hits reporting e-value threshold.

Phase 4

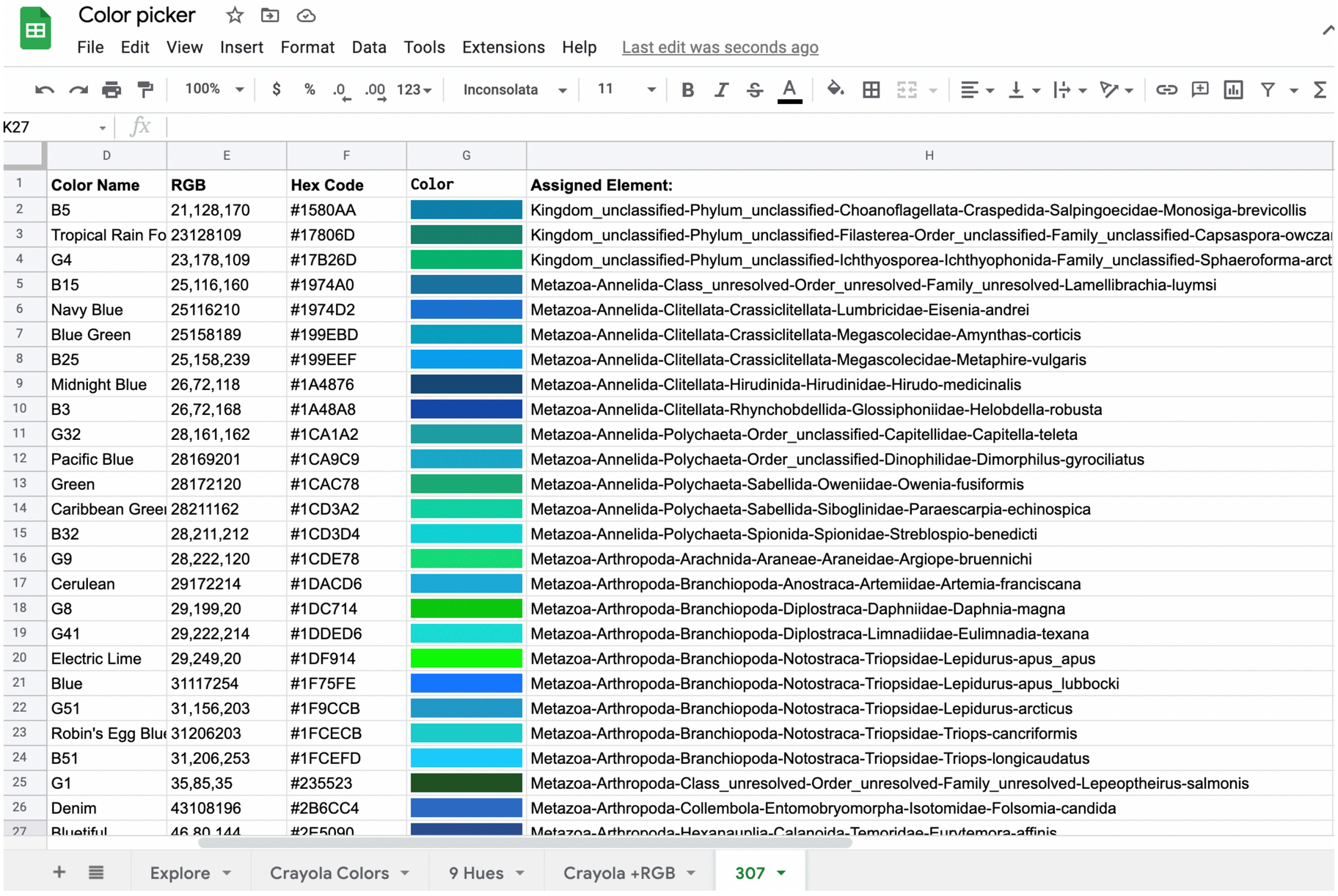
Iteratate new jackhmmer query sequences.
Update a database of filtered homologs per iteration based on hit report.
Continue until no new sequences are discovered.

Phase 5

Make homolog sequences non-redundant to create a final giganticHMMer CGS fasta.
Append reference sequences to create a final giganticHMMer AGS fasta.

GIGANTIC Features

AutoAno assigns user-provided and Crayola color palettes



Color picker

File Edit View Insert Format Data Tools Extensions Help Last edit was seconds ago

100% $ % .0 .00 123 Inconsolata 11

K27

	D	E	F	G	H
1	**Color Name**	**RGB**	**Hex Code**	**Color**	**Assigned Element:**
2	B5	21,128,170	#1580AA		Kingdom_unclassified-Phylum_unclassified-Choanoflagellata-Craspedida-Salpingoecidae-Monosiga-brevicollis
3	Tropical Rain Fo	23128109	#17806D		Kingdom_unclassified-Phylum_unclassified-Filasterea-Order_unclassified-Family_unclassified-Capsaspora-owczai
4	G4	23,178,109	#17B26D		Kingdom_unclassified-Phylum_unclassified-Ichthyosporea-Ichthyophonida-Family_unclassified-Sphaeroforma-arct
5	B15	25,116,160	#1974A0		Metazoa-Annelida-Class_unresolved-Order_unresolved-Family_unresolved-Lamellibrachia-luymsi
6	Navy Blue	25116210	#1974D2		Metazoa-Annelida-Clitellata-Crassiclitellata-Lumbricidae-Eisenia-andrei
7	Blue Green	25158189	#199EBD		Metazoa-Annelida-Clitellata-Crassiclitellata-Megascolecidae-Amynthas-corticis
8	B25	25,158,239	#199EEF		Metazoa-Annelida-Clitellata-Crassiclitellata-Megascolecidae-Metaphire-vulgaris
9	Midnight Blue	26,72,118	#1A4876		Metazoa-Annelida-Clitellata-Hirudinida-Hirudinidae-Hirudo-medicinalis
10	B3	26,72,168	#1A48A8		Metazoa-Annelida-Clitellata-Rhynchobdellida-Glossiphoniidae-Helobdella-robusta
11	G32	28,161,162	#1CA1A2		Metazoa-Annelida-Polychaeta-Order_unclassified-Capitellidae-Capitella-teleta
12	Pacific Blue	28169201	#1CA9C9		Metazoa-Annelida-Polychaeta-Order_unclassified-Dinophilidae-Dimorphilus-gyrociliatus
13	Green	28172120	#1CAC78		Metazoa-Annelida-Polychaeta-Sabellida-Oweniidae-Owenia-fusiformis
14	Caribbean Greei	28211162	#1CD3A2		Metazoa-Annelida-Polychaeta-Sabellida-Siboglinidae-Paraescarpia-echinospica
15	B32	28,211,212	#1CD3D4		Metazoa-Annelida-Polychaeta-Spionida-Spionidae-Streblospio-benedicti
16	G9	28,222,120	#1CDE78		Metazoa-Arthropoda-Arachnida-Araneae-Araneidae-Argiope-bruennichi
17	Cerulean	29172214	#1DACD6		Metazoa-Arthropoda-Branchiopoda-Anostraca-Artemiidae-Artemia-franciscana
18	G8	29,199,20	#1DC714		Metazoa-Arthropoda-Branchiopoda-Diplostraca-Daphniidae-Daphnia-magna
19	G41	29,222,214	#1DDED6		Metazoa-Arthropoda-Branchiopoda-Diplostraca-Limnadiidae-Eulimnadia-texana
20	Electric Lime	29,249,20	#1DF914		Metazoa-Arthropoda-Branchiopoda-Notostraca-Triopsidae-Lepidurus-apus_apus
21	Blue	31117254	#1F75FE		Metazoa-Arthropoda-Branchiopoda-Notostraca-Triopsidae-Lepidurus-apus_lubbocki
22	G51	31,156,203	#1F9CCB		Metazoa-Arthropoda-Branchiopoda-Notostraca-Triopsidae-Lepidurus-arcticus
23	Robin's Egg Blu	31206203	#1FCECB		Metazoa-Arthropoda-Branchiopoda-Notostraca-Triopsidae-Triops-cancriformis
24	B51	31,206,253	#1FCEFD		Metazoa-Arthropoda-Branchiopoda-Notostraca-Triopsidae-Triops-longicaudatus
25	G1	35,85,35	#235523		Metazoa-Arthropoda-Class_unresolved-Order_unresolved-Family_unresolved-Lepeoptheirus-salmonis
26	Denim	43108196	#2B6CC4		Metazoa-Arthropoda-Collembola-Entomobryomorpha-Isotomidae-Folsomia-candida
27	Bluetiful	46,80,144	#2E5090		Metazoa-Arthropoda-Hexanauplia-Calanoida-Temoridae-Eurytemora-affinis

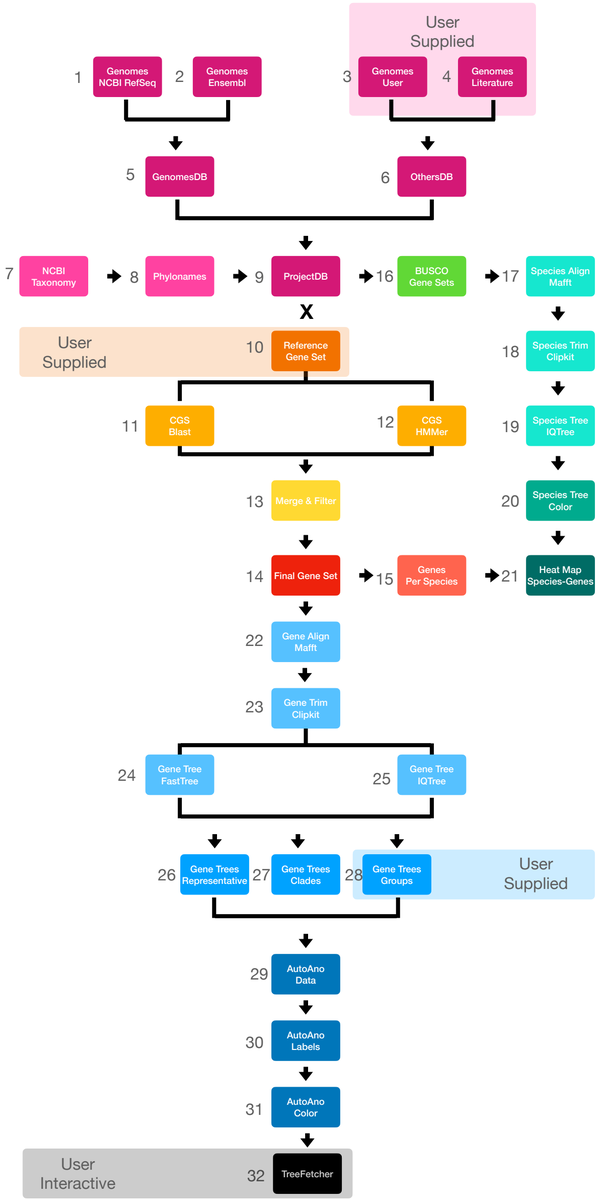
Explore Crayola Colors 9 Hues Crayola +RGB 307

GIGANTIC Features

AutoAno generates pdfs of numerous label and color annotations of all trees

GIGANTIC Structural Design

GIGANTIC Phylogenomic Pipeline



GIGANTIC is structured as:

GIGANTIC > BLOCK > STEP > SCRIPT > OUTPUT

SCRIPTs are Python, Unix, or Jupyter Notebook.

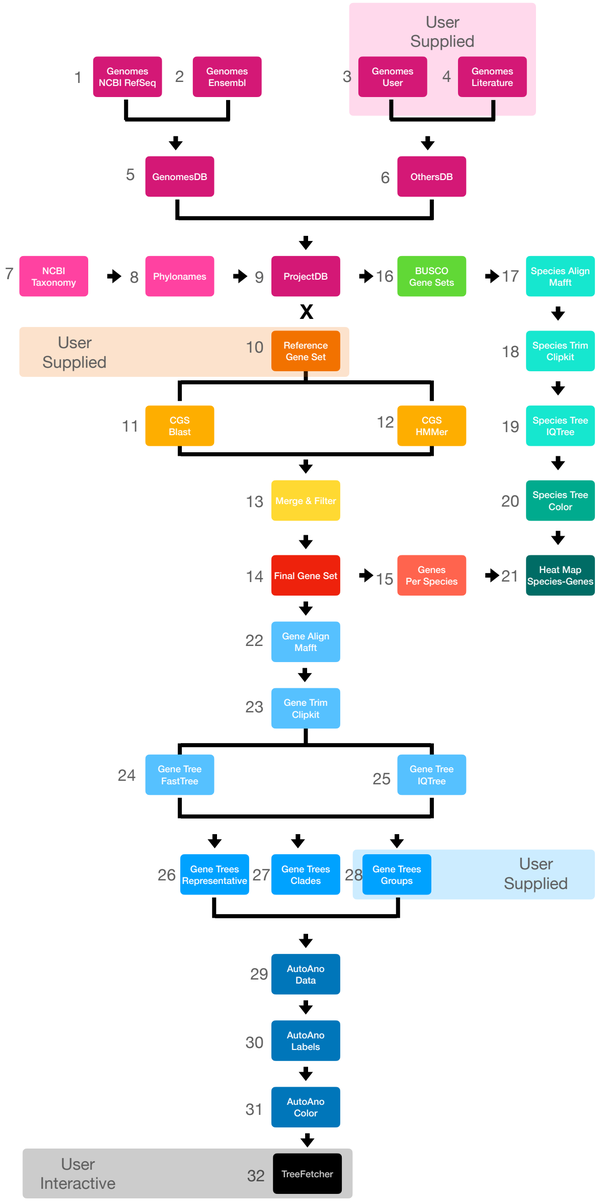
STEPs, **BLOCKs**, and/or **GIGANTIC** will be pushbutton in Jupyter Notebook and SnakeMake.

OUTPUT is structured text or pdf files.

INPUT is user-provided or previous **OUTPUT**.

GIGANTIC Structural Design

GIGANTIC Phylogenomic Pipeline



GIGANTIC > BLOCK

BLOCK 1: Databases

BLOCK 2: Homologs

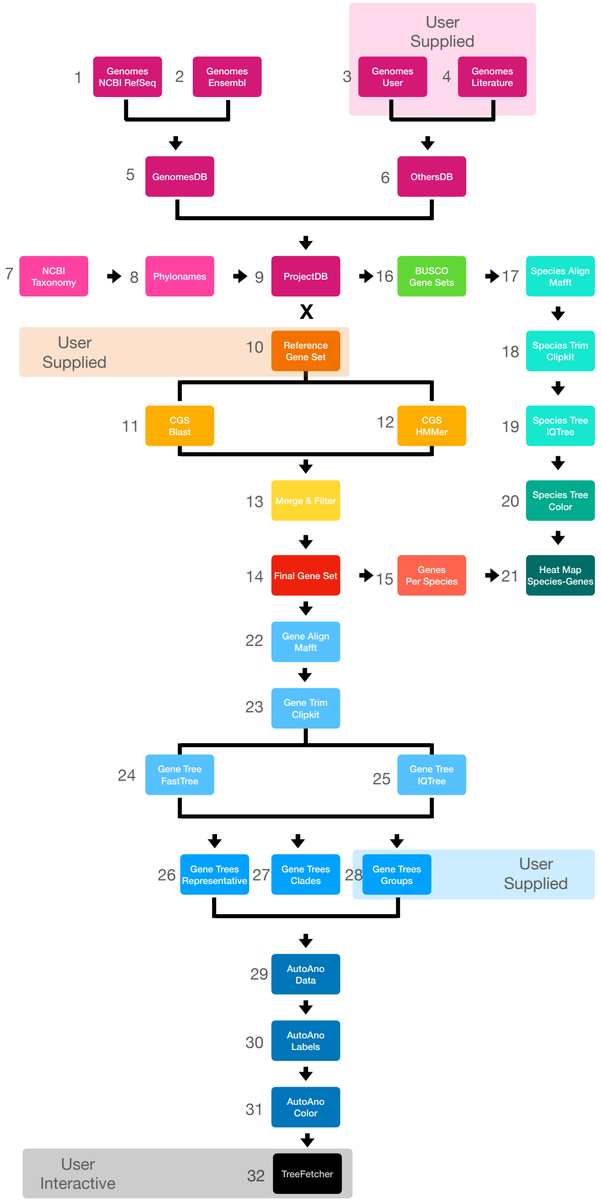
BLOCK 3: Species

BLOCK 4: Genes

BLOCK 5: Exploration

GIGANTIC > BLOCK > STEP

GIGANTIC Phylogenomic Pipeline



BLOCK 1: Databases

Step 1: Genomes NCBI RefSeq
Step 2: Genomes Ensembl
Step 3: Genomes User
Step 4: Genomes Outside
Step 5: GenomesDB
Step 6: OthersDB
Step 7: NCBI Taxonomy
Step 8: Phylonames
Step 9: ProjectDB

BLOCK 2: Homologs

Step 10: Reference Gene Set
Step 11: Candidate Gene Set Blast
Step 12: Candidate Gene Set HMMer
Step 13: Merge and Filter
Step 14: Final Gene Set
Step 15: Genes Per Species

BLOCK 3: Species

Step 16: BUSCO Gene Sets
Step 17: Species Align Mafft
Step 18: Species Trim Clipkit
Step 19: Species Tree IQTree
Step 20: Species Tree Color
Step 21: Heat Map Species-Genes

BLOCK 4: Genes

Step 22: Gene Align Mafft
Step 23: Gene Trim Clipkit
Step 24: Gene Tree FastTree
Step 25: Gene Tree IQTree
Step 26: Gene Trees Representative
Step 27: Gene Trees Clades
Step 28: Gene Trees Groups
Step 29: AutoAno Data
Step 30: AutoAno Labels
Step 31: AutoAno Colors

BLOCK 5: Exploration

Step 32: TreeFetcher

GIGANTIC Code Contributions

Lola

Species Tree and Gene Tree

Jan

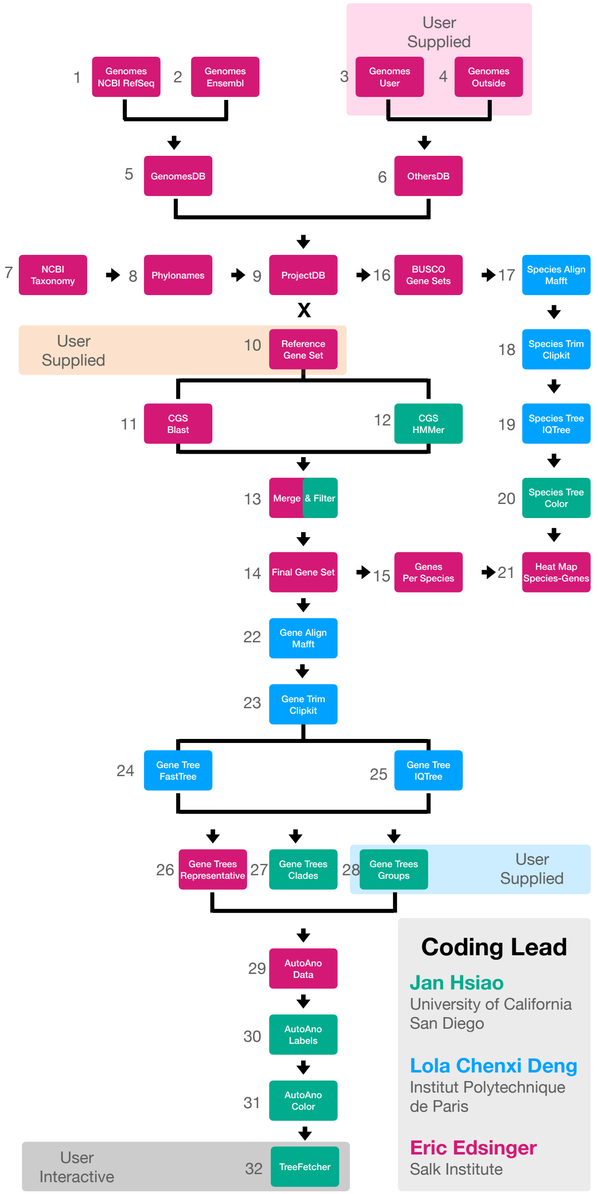
CGS HMMer

Gene Trees Clades and Gene Trees Groups

AutoAno and **TreeFetcher**

Jupyter Notebooks

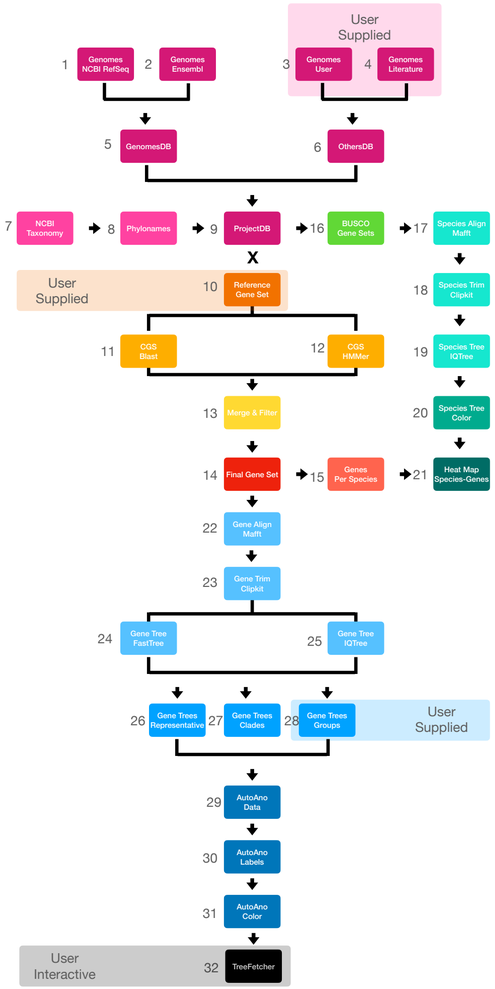
GIGANTIC Contributions



GIGANTIC: 300+ scripts - 10,000+ lines of code

Lola's Code

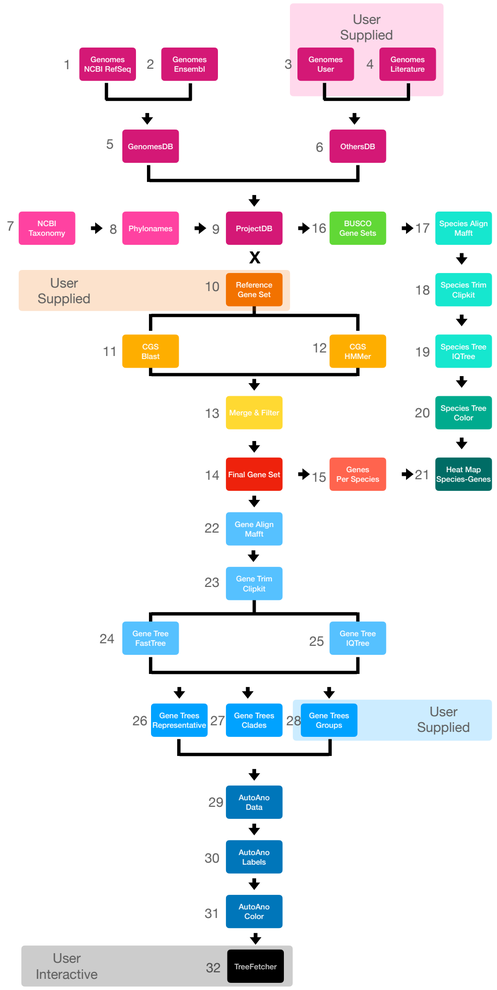
GIGANTIC




```
#!/usr/bin/python
import os
import re
from os import path

class Origin:
    def __init__(self, filename):
        self.id = filename.split('-')[-1]
        self.info = {}
        seq = []
        with open('./4-clipkit/' + filename,'r') as f:
            file=f.read()
            for species in file.split('>'):
                if species.strip():
                    i=species.strip().split('\n')
                    header=i[0]
                    j=header.strip().split('-')
                    name='-'.join(j[5:7])
                    seq=species.strip().split('\n')[1:]
                    self.info[name]=seq

        self.fill=[]
        block=[]
        for i in seq:
            self.fill.append('X' * len(i))
        for i in self.fill:
            block.append(len(i))
            self.len=sum(block)

    def write_data(self, all_species):
        for i in all_species:
            if i in self.info:

                # Create files for appending
                with open('./5-catsequences/'+i, 'a') as f:
                    for output in self.info[i]:
                        f.write(output)
                       # f.write('\n')

            # Fill space with 'X's if a gene is absent in certain species.
            else:
                with open('./5-catsequences/'+ i,'a') as f:
                    for output in self.fill:
                        f.write(output)
                        #f.write('\n')
```

GIGANTIC: 300+ scripts - 10,000+ lines of code

GIGANTIC

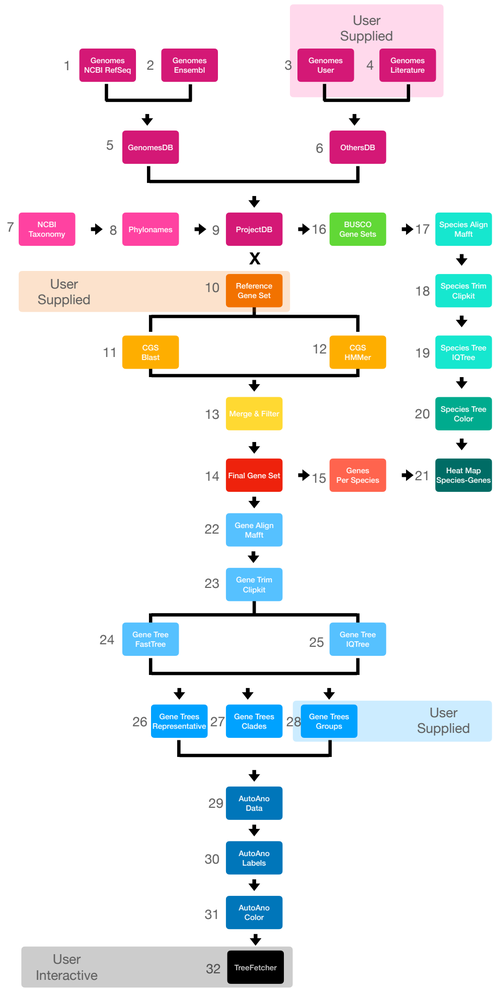


Jan's Code

```
#! python
import os
import sys

### User Input ###
j_blosum="BLOSUM45"
j_evalue="0.001"
j_rounds="2"

### User Input Stop###
j_cpu = "10"
if len(sys.argv) > 1:
    if sys.argv[1] == '--cpu':
        j_cpu=sys.argv[2]

### No Need to Modify ###

constraints = {}
constraintsfile = open('../userinput/constraints.txt','r')
for nextline in constraintsfile:
    info = nextline[:-1].split('\t')
    family = info[0]
    outgrouplist = info[1].split(',')
    constraints[family] = outgrouplist
constraintsfile.close()

active_iter = True
itercount = 0

while(active_iter):
    this_iteration = str(itercount)
    previous_iteration = str(itercount-1)

    os.system("mkdir jackhmmer-iteration-"+this_iteration)
    os.chdir("jackhmmer-iteration-"+this_iteration)

    newcount = 0
    passedcount = 0

    sequencedict = {}
    inputfasta = open("../../1-process-fastas/local-projectdb.aa",'r')
    for nextline in inputfasta:
        if nextline[0] == '>':
            header = nextline[1:-1]
            sequencedict[header] = ""
        else:
            sequencedict[header] += nextline[:-1]
```

GIGANTIC: 300+ scripts - 10,000+ lines of code

GIGANTIC



Eric's Code

```
#! python


##### USER INPUT

input_fastas = open( 'output/4-list-fastas', 'r' )
input_hits = open( 'output/15-all-blastp-all-reports', 'r' )
input_rgs_ids = open( 'output/8-map-source-to-reference-identifiers', 'r' )

output_fasta = open( 'output/16-CGS-final-sequences-by-blastp-RBF.fasta', 'w' )
output_filtered = open( 'output/16-dropped-queries-no-rgs-top-hit-in-rgs-genome', 'w' )

model_species = [ 'human','mouse','fly','worm','anemone' ]

##### BEGIN SCRIPT

rgs_ids = []

for next_line in input_rgs_ids:

        info = next_line[ :-1 ].split( '\t' )
        projectdb_id = info[ 0 ]
        rgs_id = info[ 1 ]
        rgs_ids.append( rgs_id )
        rgs_ids.append( projectdb_id )

keepers = []
queries = []

for next_hit in input_hits:

        info = next_hit.split( '\t' )
        query = info[ 0 ]
        queries.append( query )

        hit = info[ 1 ]
        hit_info = hit.split( '-' )
        name = hit_info[ 1 ]

        if name in model_species:

                # drop RGS genes
                if query in rgs_ids:

                        pass
```

GIGANTIC: Salk Machines

GIGANTIC

Chi

Linux Debian 4

CPU Count: 64
GenuineIntel Intel(R) Xeon(R) CPU E5-2697A v4 @ 2.60GHz
2 chips x 16 cores : 32 hyperthread cores
0.5 Tb RAM

Eccles

Linux Debian 4

CPU Count: 48
GenuineIntel Intel(R) Xeon(R) Gold 6146 CPU @ 3.20GHz
2 chips x 12 cores : 24 hyperthread cores
1 Tb RAM

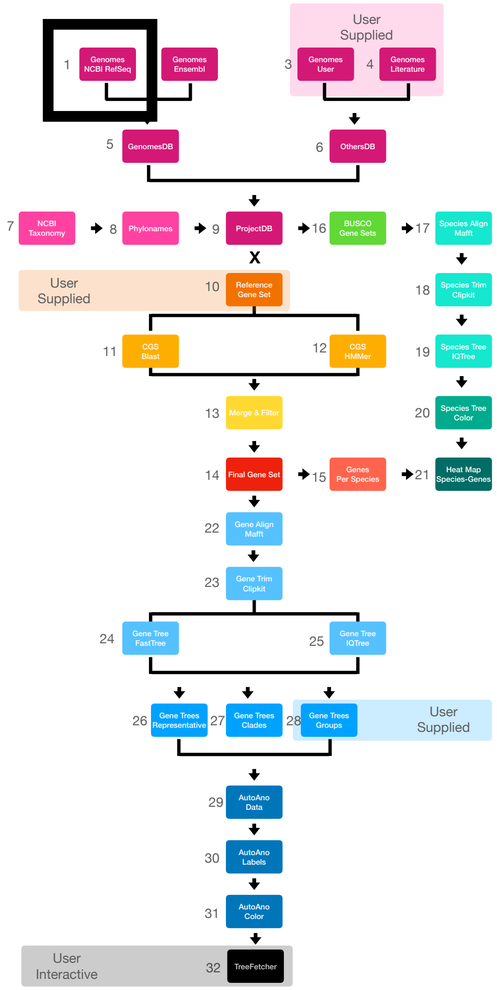
Ika

Linux CentOS 7

CPU Count: 80
GenuineIntel Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz
2 chips x 20 cores : 40 hyperthread cores
0.3 Tb RAM

Genomes NCBI RefSeq

GIGANTIC Phylogenomic Pipeline



STEP 1 **Genomes NCBI RefSeq** provides automated download of NCBI RefSeq genomes and processing to remove isoforms.

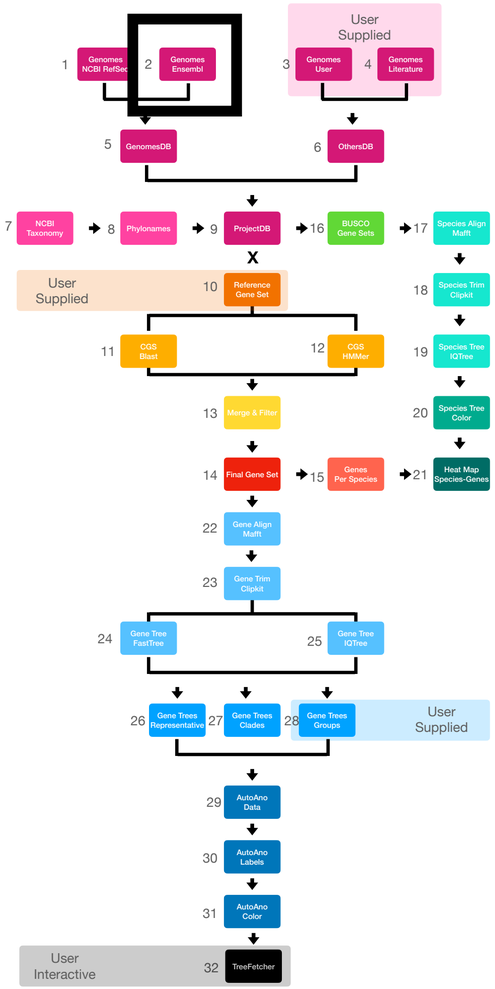
SCRIPTS

genomesdb 1-downloaded

```
001-mkdir-downloaded-directories
008-wget-ncbiRefSeq-mammalia-info
009-python-command-wget-ncbi-mammalia
010-wget-ncbi-refseq-mammalia-vertebrates
011-wget-ncbiRefSeq-vertebrates-other-info
012-python-command-wget-ncbi-vertebrate-other
013-wget-ncbi-refseq-vertebrates-other
014-wget-ncbiRefSeq-invertebrates-info
015-python-command-wget-ncbi-invertebrates
016-wget-ncbi-refseq-invertebrates
```

Genomes Ensembl

GIGANTIC Phylogenomic Pipeline



STEP 2 **Genomes Ensembl** provides automated download of Ensembl genomes and processing to remove isoforms.

SCRIPTS

genomesdb 1-downloaded

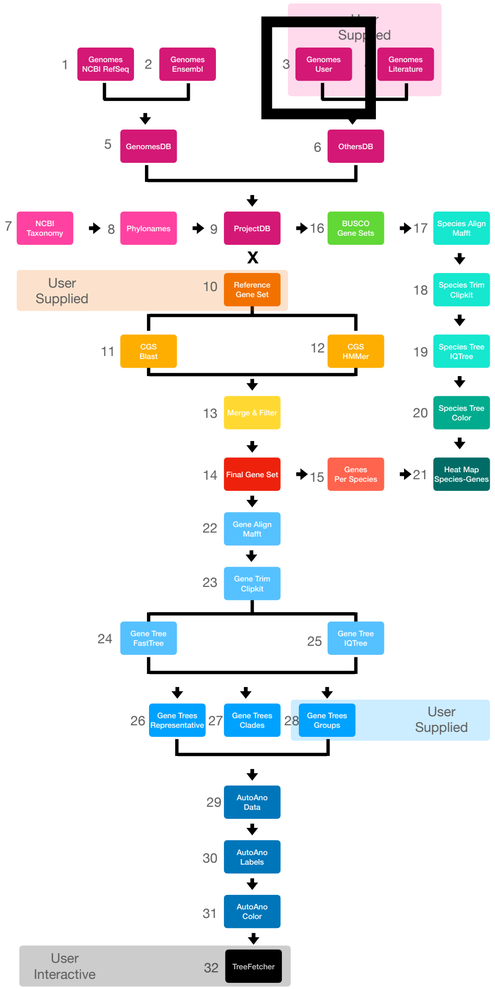
```
002-wget-ensembl-vertebrates-info
003-python-command-wget-ensembl-vertebrates
004-wget-ensembl-vertebrates-genomes
005-wget-ensembl-metazoa-info
006-python-command-wget-ensembl-metazoa
007-wget-ensembl-metazoa-genomes
017-gunzip
```

genomesdb 2-longest-transcript

```
001-ls-gffs-fastas
002-python-process-gffs
003-python-process-proteomes
008-ls-genome-T1-fastas
```

Genomes User

GIGANTIC Phylogenomic Pipeline



STEP 3 **Genomes User** enables inclusion of user-provided in-house genome data sets.

SCRIPTS

User supplied fastas and gffs

`001-scp-or-other-command-providing-file-sources`

Area for development with collaborators

Requirements

Standardized filename

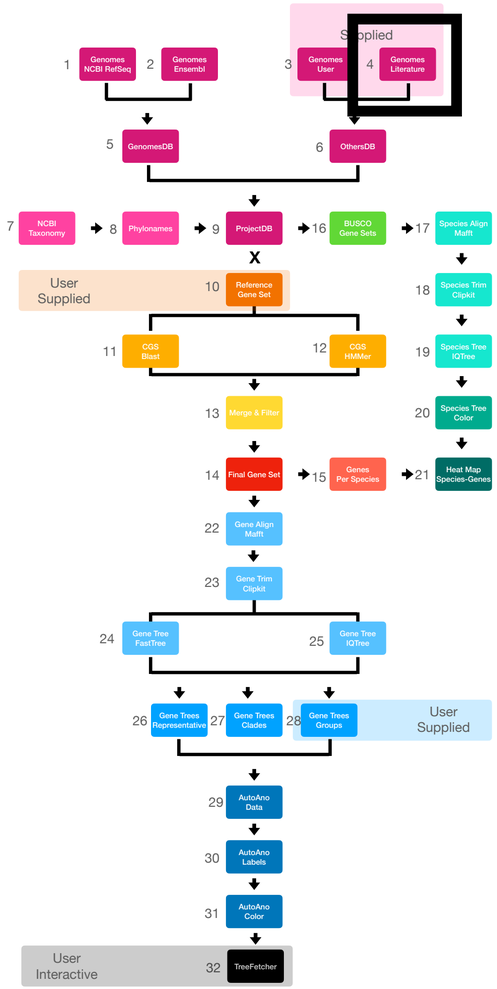
Phylum-Genus-Species_subspecies-source_genome_name.fasta

Standardized header

>Source_gene_model_id original_header

Genomes Outside

GIGANTIC Phylogenomic Pipeline



STEP 4 **Genomes Outside** enables inclusion of user-provided outside genome data sets not available in NCBI RefSeq or Ensembl.

SCRIPTS

User supplied fastas and gffs

`001-wget-or-other-command-providing-file-sources`

Area for development with collaborators

Requirements

Standardized filename

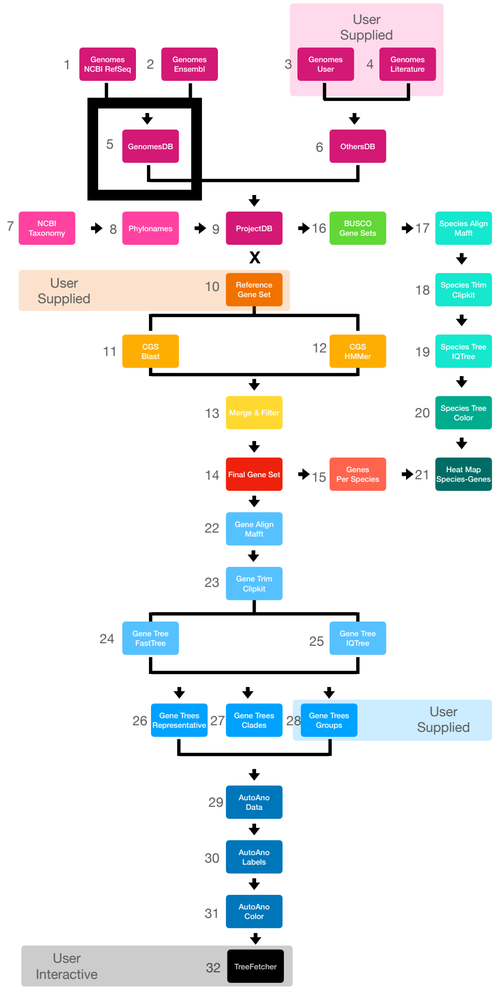
Phylum-Genus-Species_subspecies-source_genome_name.fasta

Standardized header

>Source_gene_model_id original_header

GenomesDB

GIGANTIC Phylogenomic Pipeline



STEP 5 **GenomesDB** selects top genomes and removes redundancy between NCBI RefSeq and Ensembl based on BUSCO scores.

SCRIPTS

genomesdb 3-busco

```
001-ls-phylogenome-paths
002-python-busco-commandlines
003-busco-CONDA-ACTIVATE-BUSCO4
003-busco-metazoa.ini
```

genomesdb 4-one-genome-per-species

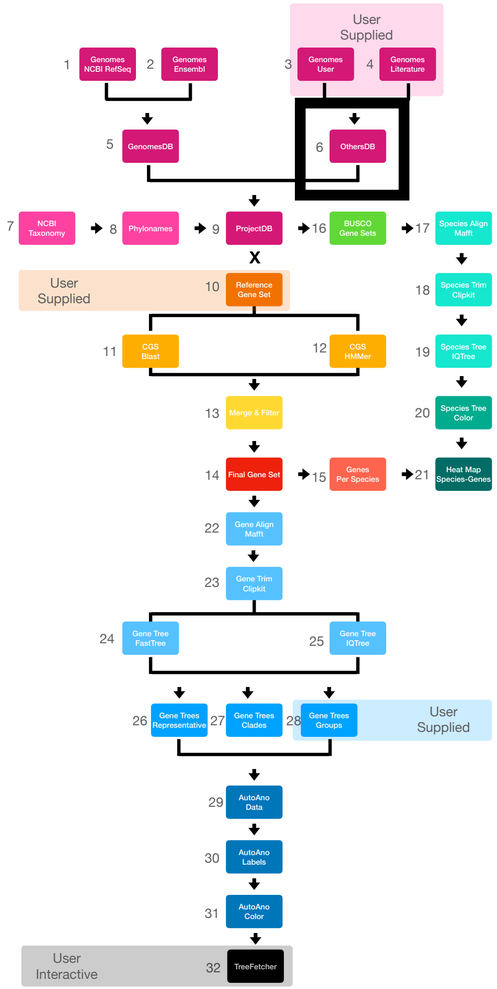
```
001-cat-busco-short-summaries
002-python-parse-busco-screen-capture
003-python-select-top-busco-per-species
004-removed-from-top-genome-per-spp-by-hand
```

genomesdb 5-genomesdb

```
001-mkdir
002-python-commandline-cp-top-phylogenome-fastas-to-genomesdb
```

OthersDB

GIGANTIC Phylogenomic Pipeline



STEP 6 **OthersDB** selects top genomes in User and Outside genome data sets.

SCRIPTS

othersdb

```
001-ls-source-fastas
009-python-command-busco
010-busco-metazoa.ini
010-busco-metazoa-otherdb-CONDA
```

genomesdb 4-one-genome-per-species

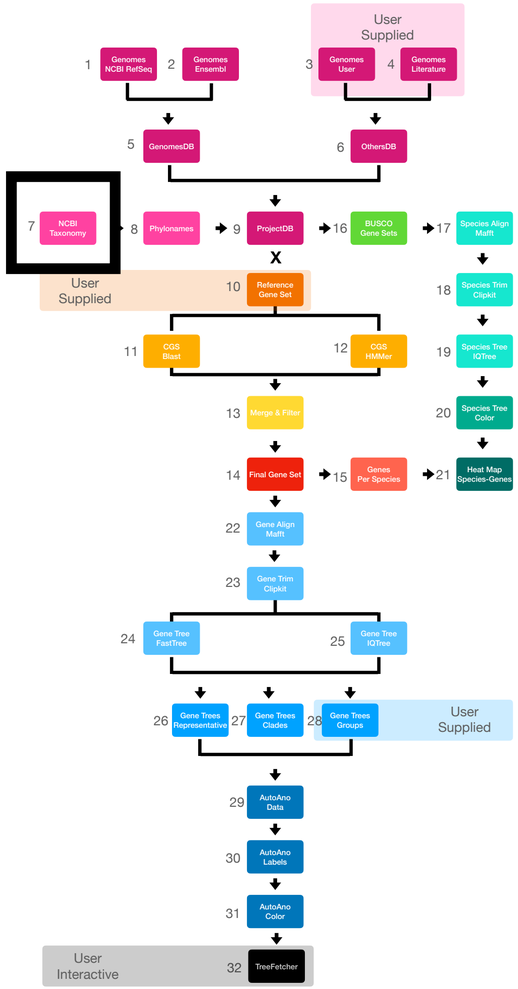
```
001-cat-busco-short-summaries
002-python-parse-busco-screen-capture
003-python-select-top-busco-per-species
004-removed-from-top-genome-per-spp-by-hand
```

genomesdb 5-genomesdb

```
001-mkdir
002-python-commandline-cp-top-phylogenome-fastas-to-genomesdb
```

NCBI Taxonomy

GIGANTIC Phylogenomic Pipeline



STEP 7 **NCBI Taxonomy** provides automated download of current NCBI Taxonomy database.

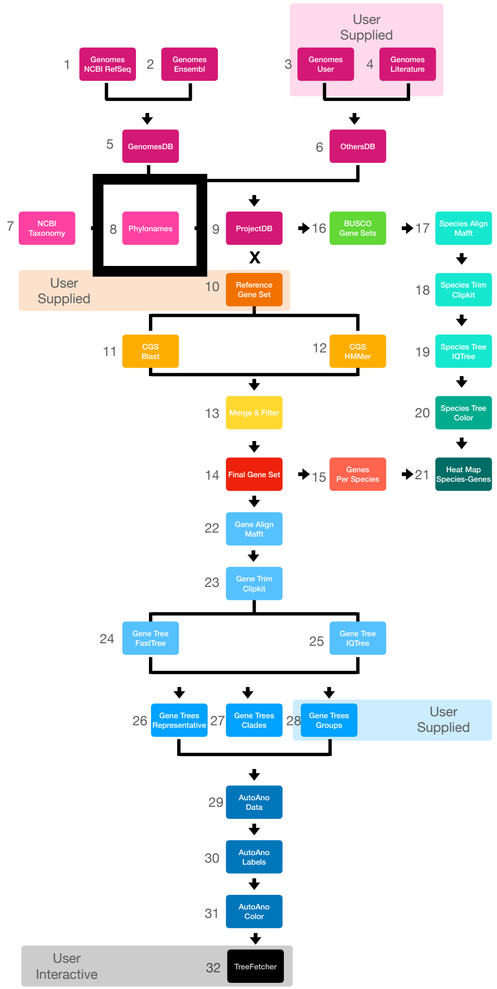
SCRIPTS:

genomesdb 2-longest-transcript

004-wget-ncbi-taxa-dump
005-gunzip

Phylonames

GIGANTIC Phylogenomic Pipeline



STEP 8 **Phylonames** produces Phyloname identifiers for all species in NCBI based on parsing of NCBI Taxonomy database.

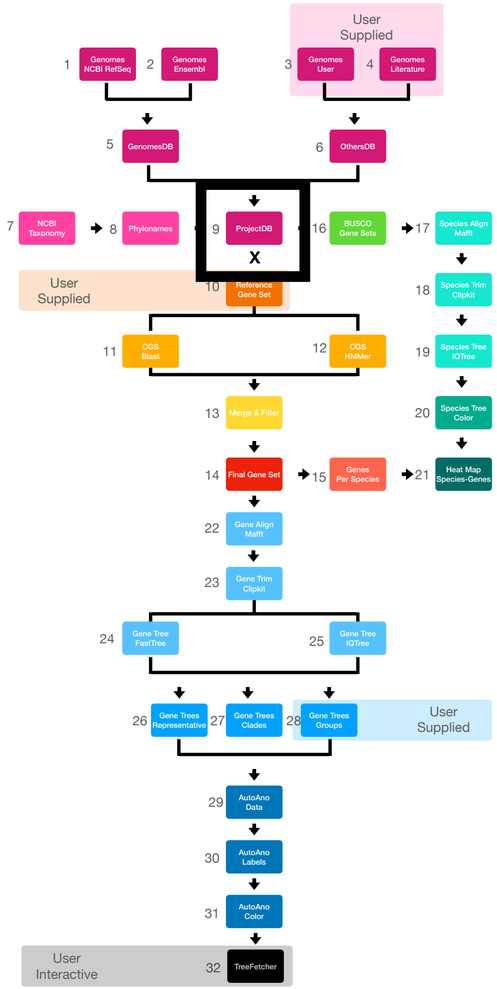
SCRIPTS

genomesdb 2-longest-transcript

006-mkdir
007-python-phyloNames-ncbi-taxonomy

ProjectDB

GIGANTIC Phylogenomic Pipeline



STEP 9 **ProjectDB** produces a single top genome per user-specified species - Phyloname headers and single longest transcript per gene - based on BUSCO and user-request.

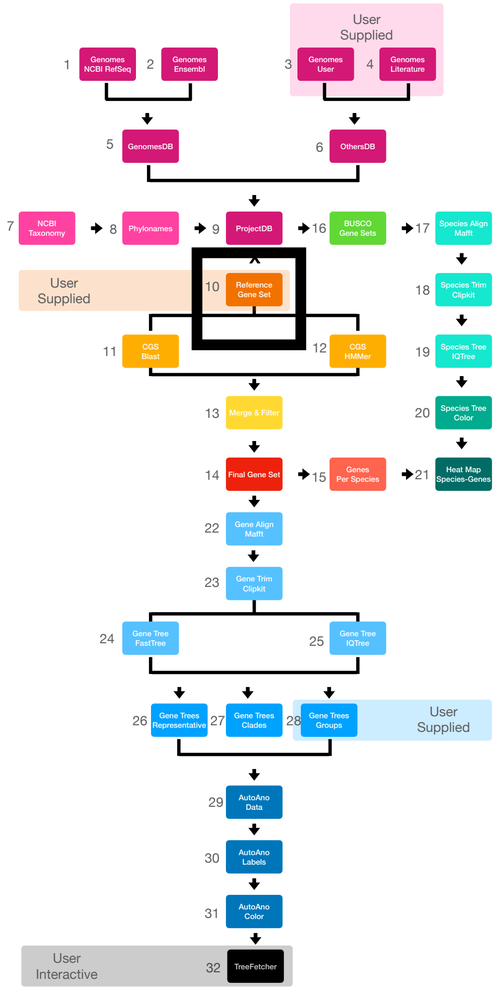
SCRIPTS

pipeline-blocks 1-databases 3-projectDB

```
000-user-input-specific-species
001-cp-fastas-genomedb-otherdb
002-ls-fastas
003-python-projectdb-seqids
004-ls-fastas
005-python-cleanup-fasta
006-sed-asterik
007-python-command-busco
008-busco-metazoa.ini
008-busco-metazoa-otherdb-CONDA
009-cat-busco-short-summaries
010-python-makeblastdb
011-makeblastdb
```

Genomes User

GIGANTIC Phylogenomic Pipeline



STEP 10 **Reference Gene Set** enables user input of a structured Reference Gene Set fasta per gene family and with all gene family members known per reference species.

SCRIPTS

User supplied fastas

```
*000-wget-or-other-command-providing-file-sources
000-source-fastas
001-mkdir-ls
002-python-cleanup-fasta
```

Area for development with collaborators

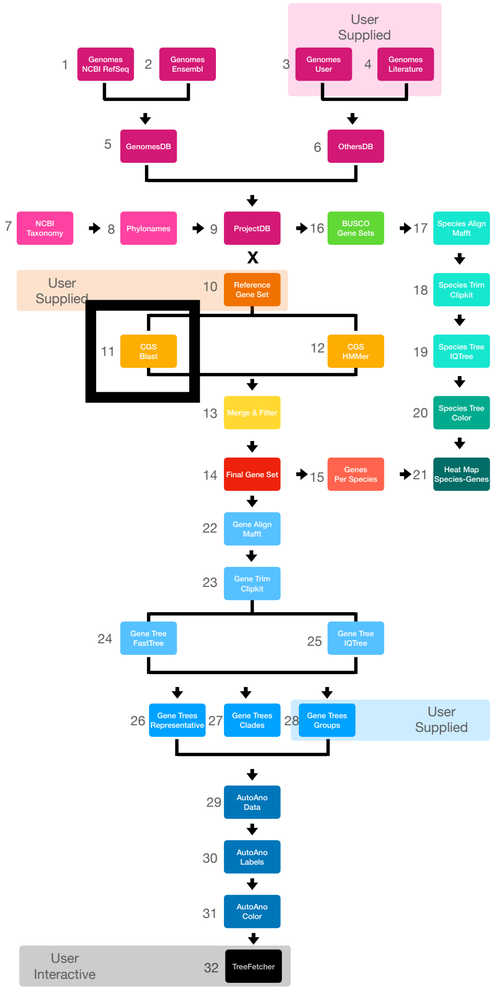
Requirements

Standardized filename + map name_TO_Phylum-genus-species
gene_family_id-reference_species_name_1-reference_species_name_2-DDmonYYYY.fasta

Standardized header
>Source_gene_id original_header

CGS Blast

GIGANTIC Phylogenomic Pipeline



STEP 11 **Candidate Gene Set Blast** identifies gene homologs in ProjectDB genomes using Blast-based reciprocal best hit to family or superfamily in reference genomes.

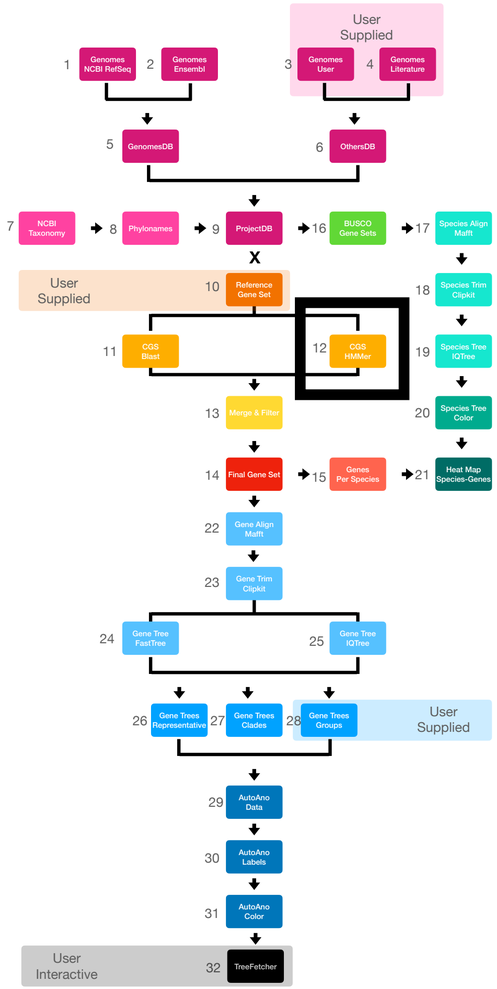
SCRIPTS

pipeline-blocks 2-homologs 1-blastp-pipeline

```
001-ls-blastdbs
002-python-command-blastp
003-blastp_X_projectDB
004-ls-reports-and-fastas
005-python-gene-set-fasta
006-blastp-rgs_X_rgs-genomes
007-ls-rgs-reports-fastas
008-python-update-reference-genomes
009-ls-reference-genome-fastas
010-python-command-makeblastdb
011-blastp-makedb
012-ls-blastp-rgs-genomes
013-python-command-blastp
014-blastp-hits_X_RGS-genomes
015-cat-all-blastp-reports
016-python-RBF-CGS-each-RGS-genome-fasta
017-cat-RGS-CGS-fasta
```

CGS HMMer

GIGANTIC Phylogenomic Pipeline



STEP 12 Candidate Gene Set HMMer

identifies gene homologs in ProjectDB genomes using novel HMMer-based identification based on convergent iterations of JackHMMer rounds.

SCRIPTS

Jan 1-process-fastas

001-process-refs
002-python-update-full
003-annotate-refs
004-python-localdb-post-process

Jan 2-jackhmmer-iterations

000-python-autoiterate
001-python-autoiterate-stats

Jan 3-generate-final-CGS

001-merge-family-genesets
002-make-AGS

Jan projectdb

000-prep
26-projectdb-dmnds
26-projectdb-fastas

Jan scripts

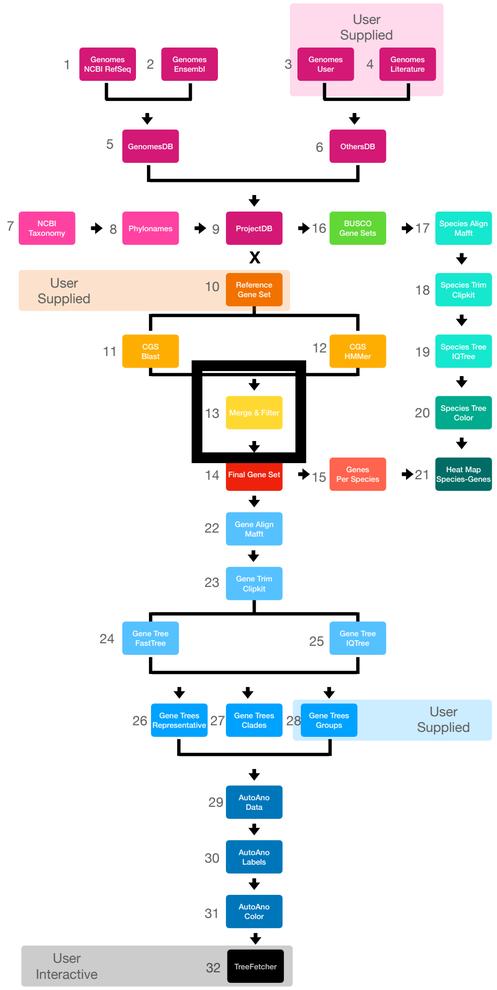
update-reference-genomes.py
find-shortest-seqlen.py
keep-sequences.py

Jan userinput

annotate-references.txt
constraints.txt

Genomes User

GIGANTIC Phylogenomic Pipeline



STEP 13 **Merge & Filter** merges Blast and HMM CGS and filters for bacterial contamination.

SCRIPTS

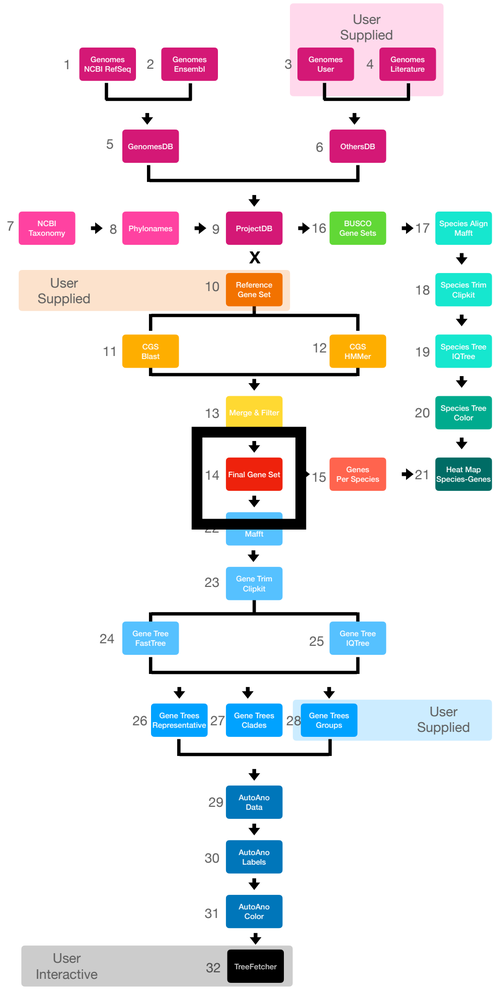
pipeline-blocks 2-homologs 3-merge-quality-control

```
001-setup
002-ls-redundant-ags-fastas
003-python-remove-RGS-sequences-make-nonredundant
004-diff-unique-sequences-per-run
005-python-make-diff-clean-nonredundant-sequences
006-blastp-CGS_X_refseq_select-full-length
007-python-CGS-filter-bacterial-contamination-full-length
008-python-potential-bacteria-sequences
009-interproscan-RGS-AGS-full-length
010-grep-pfam-interproscan-output
011-python-pfam-filter-AGS-sequences
012-cat-potential-pipeline-bacteria-pfam-false-positives
013-python-make-fasta-nonredundant
014-INPUT-ncbi-verified-pfam-false-positives
015-python-remove-false-positives-generate-KEEPERS
```

Broken gene models Jan

Final Gene Set

GIGANTIC Phylogenomic Pipeline



STEP 14 **Final Gene Set** generates a gene set per family that includes user-provided reference and GIGANTIC-identified candidate genes for gene family phylogenetic analysis.

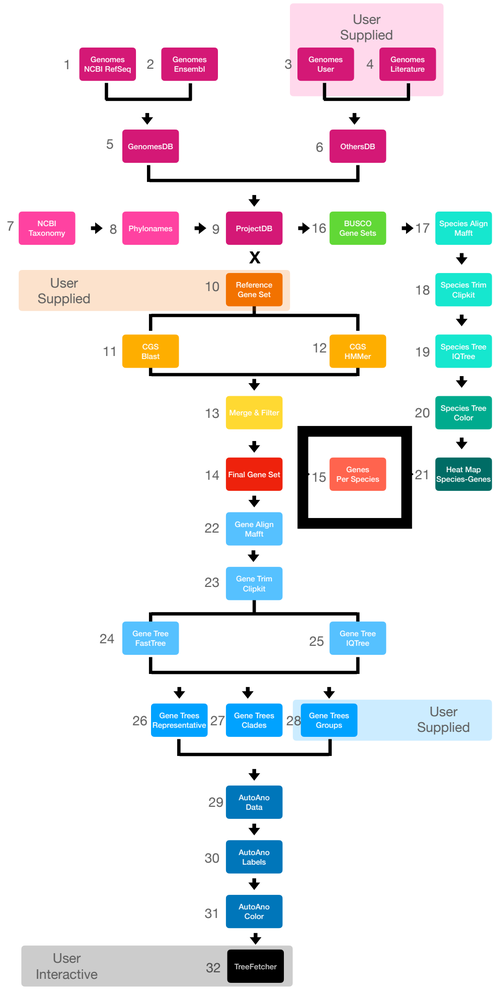
SCRIPTS

pipeline-blocks 2-homologs 3-merge-quality-control

`016-python-AGS-all-drop100-drop150`

Genes Per Species

GIGANTIC Phylogenomic Pipeline



STEP 15 **Genes Per Species** generates counts of genes per family per species.

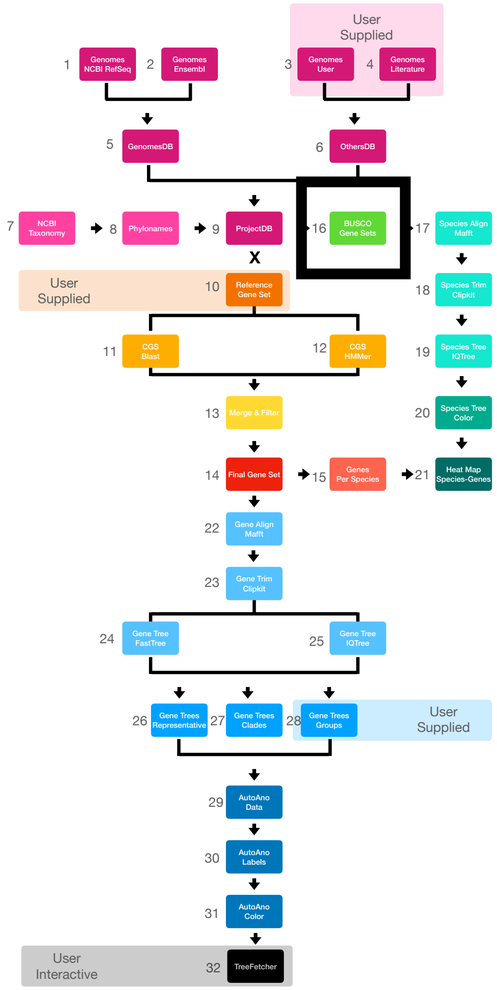
SCRIPTS

pipeline-blocks 2-homologs 3-merge-quality-control

```
001-input-final-ags-fastas
001-python-parse-fastas-to-count-genes
```

BUSCO Gene Sets

GIGANTIC Phylogenomic Pipeline



STEP 16 **BUSCO Gene Sets** generates BUSCO evaluation of ProjectDB genomes and BUSCO Gene Set fastas for ProjectDB species phylogenetic analysis.

SCRIPTS

othersdb

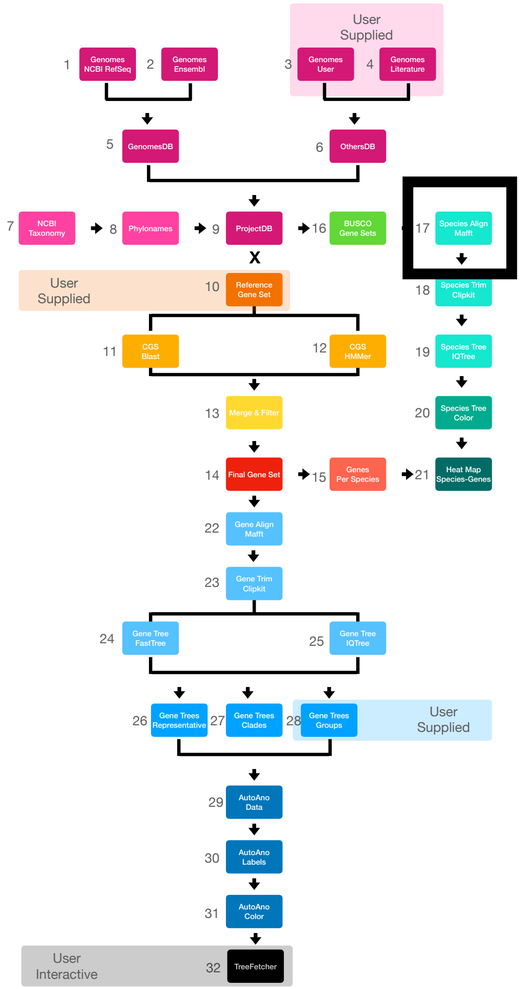
```
008-ls-othersdb-T1-fastas
009-python-command-busco
010-busco-metazoa.ini
010-busco-metazoa-otherdb-CONDA
```

tool-development busco-scripts

```
001-mkdir
002-find-busco-faa
003-python-busco_vs_species-busco-genesets
```

Species Align Mafft

GIGANTIC Phylogenomic Pipeline



STEP 17 **Species Align Mafft** performs catenation and generates a Mafft alignment / superalignment of BUSCO Gene Set sequences for ProjectDB species phylogenetic analysis.

SCRIPTS

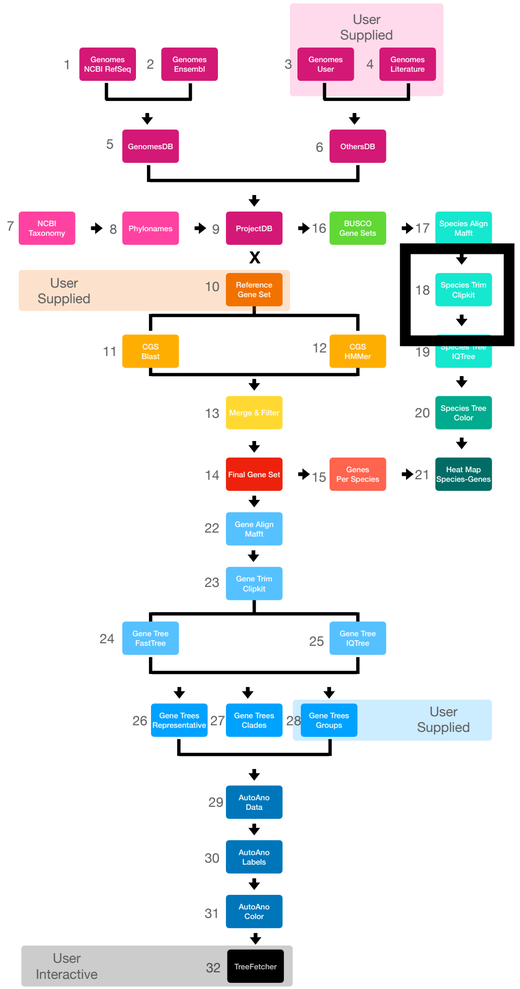
ldeng projects metazoa30

`C60-supermatrix1.sh`
`C60-supermatrix2.py`

Lola

Species Trim Clipkit

GIGANTIC Phylogenomic Pipeline



STEP 18 **Species Trim Clipkit** generates Clipkit trimmed Mafft-alignments for ProjectDB species phylogenetic analysis.

SCRIPTS

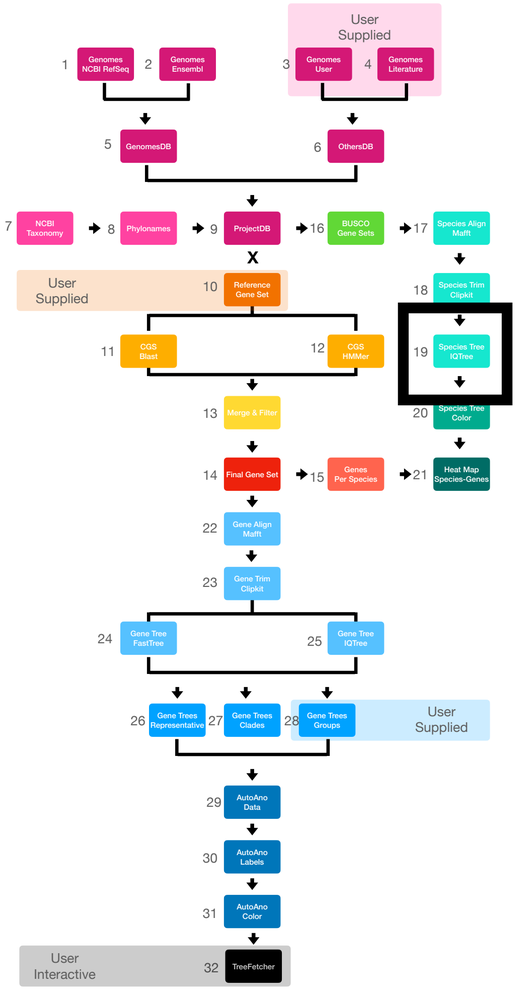
ldeng projects metazoa30

```
C60-supermatrix1.sh
C60-supermatrix2.py
```

Lola

Species Tree IQTree

GIGANTIC Phylogenomic Pipeline



STEP 19 **Species Tree IQTree** generates a BUSCO-based FastTree / IQTree maximum-likelihood species trees for ProjectDB species.

SCRIPTS

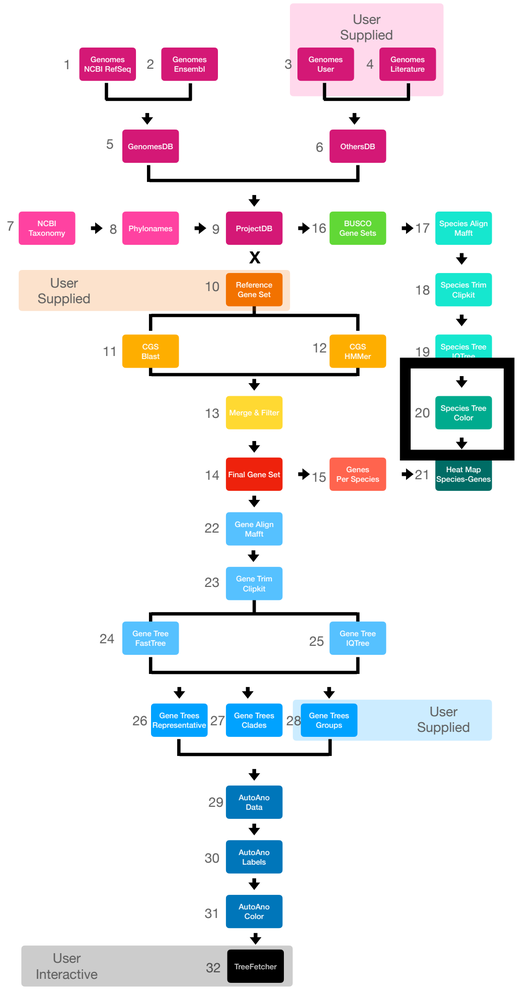
ldeng projects metazoa30

`C60-supermatrix3.sh`

Lola

Species Tree Color

GIGANTIC Phylogenomic Pipeline



STEP 20 **Species Tree Color** generates user-guided color annotation of the ProjectDB species tree.

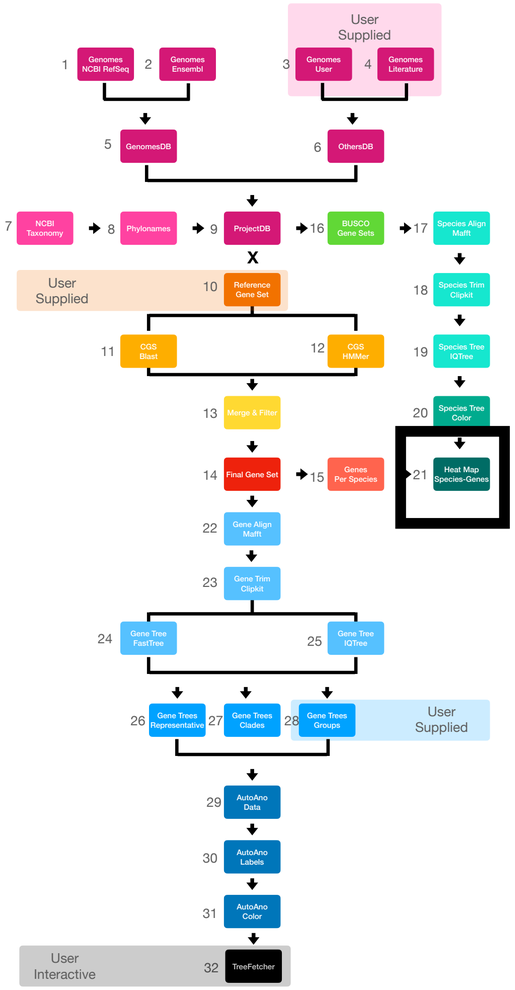
SCRIPTS
JAN

JAN

Jan

Heat Map Species-Genes

GIGANTIC Phylogenomic Pipeline



STEP 21 **Heat Map Species-Genes**

generates a matrix and associated basic heat map of species and number of genes per family.

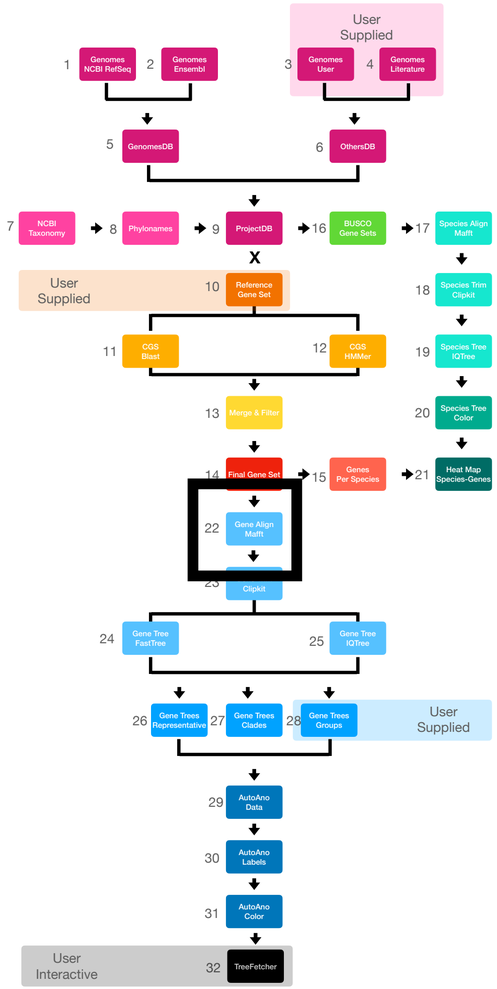
SCRIPTS

None

`Currently done by hand in Google Sheets -> Move to Python or R`

Gene Align Mafft

GIGANTIC Phylogenomic Pipeline



STEP 22 **Gene Align Mafft** generates Mafft or Mafft Dash alignment of Final Gene Set sequences for ProjectDB gene tree generation.

SCRIPTS

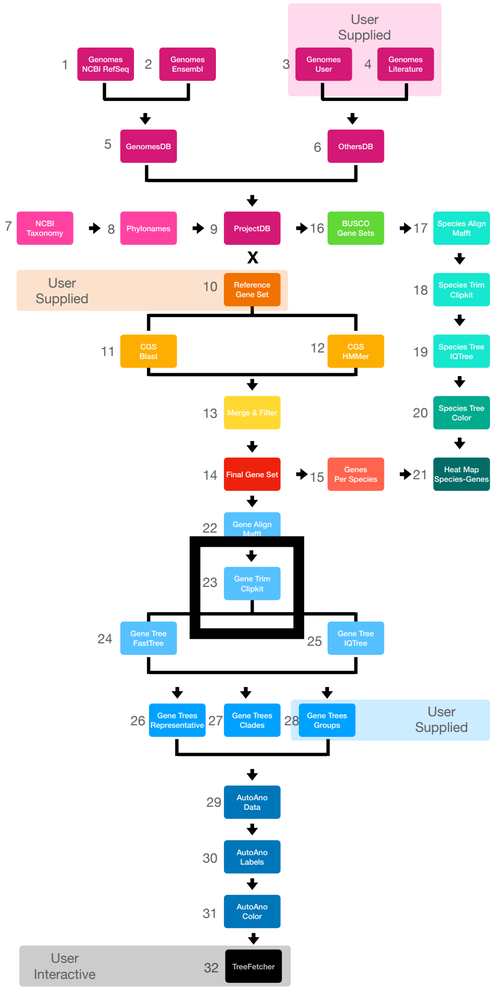
pipeline-blocks 4-trees 1-representative-trees

```
000-setup
001-sed-dashes-for-mafft
002-sed-U
003-mafft
```

Lola

Gene Trim Clipkit

GIGANTIC Phylogenomic Pipeline



STEP 23 **Gene Trim Clipkit** generates Clipkit trimmed Mafft-alignments for ProjectDB gene tree generation.

SCRIPTS

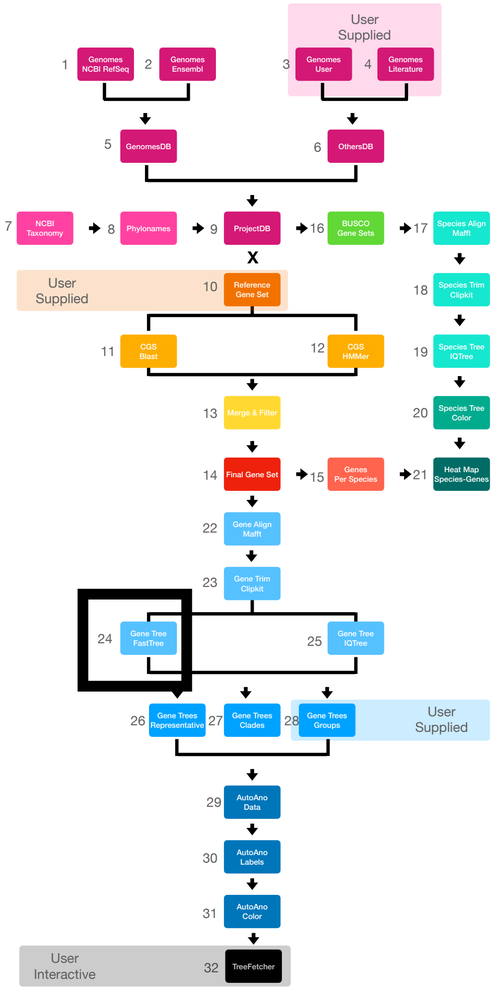
pipeline-blocks 4-trees 1-representative-trees

`004-clipkit-smartgap`

Lola

Gene Tree FastTree

GIGANTIC Phylogenomic Pipeline



STEP 24 **Gene Tree FastTree** generates a quick-to-finish (minutes to hours) medium quality maximum-likelihood-like ProjectDB gene tree.

SCRIPTS

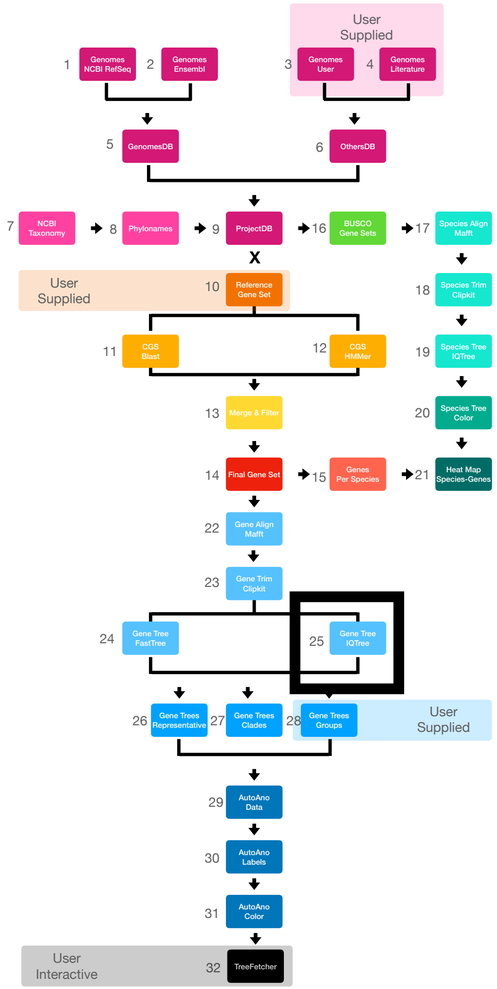
pipeline-blocks 4-trees 1-representative-trees

`006-fasttree`

Lola

Gene Tree IQTree

GIGANTIC Phylogenomic Pipeline



STEP 25 **Gene Tree IQTree** generates a slow-to-finish (hours to days) high quality maximum-likelihood ProjectDB gene tree.

SCRIPTS

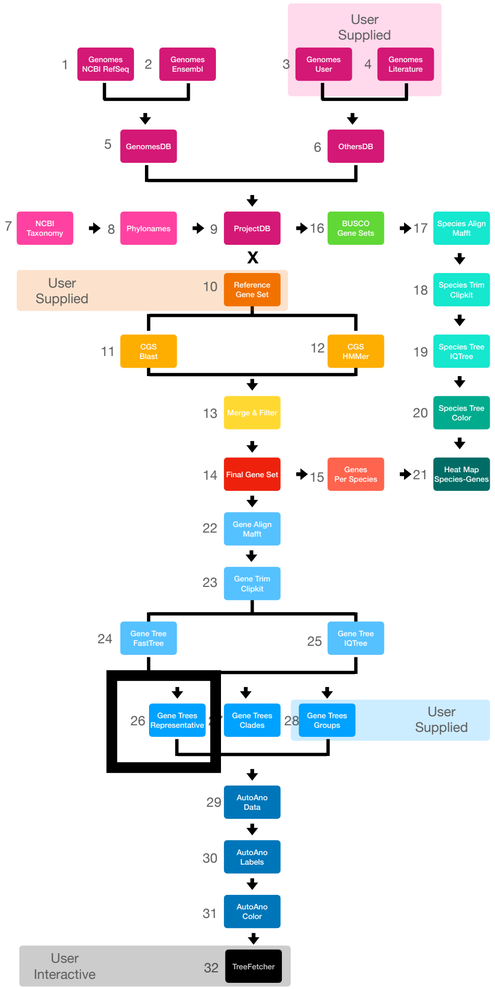
pipeline-blocks 4-trees 1-representative-trees

`005-iqtree2`

Lola

Gene Tree Represenative

GIGANTIC Phylogenomic Pipeline



STEP 26 Gene Tree Representative

generates a representative tree for the GIGANTIC run with species matching the species tree.

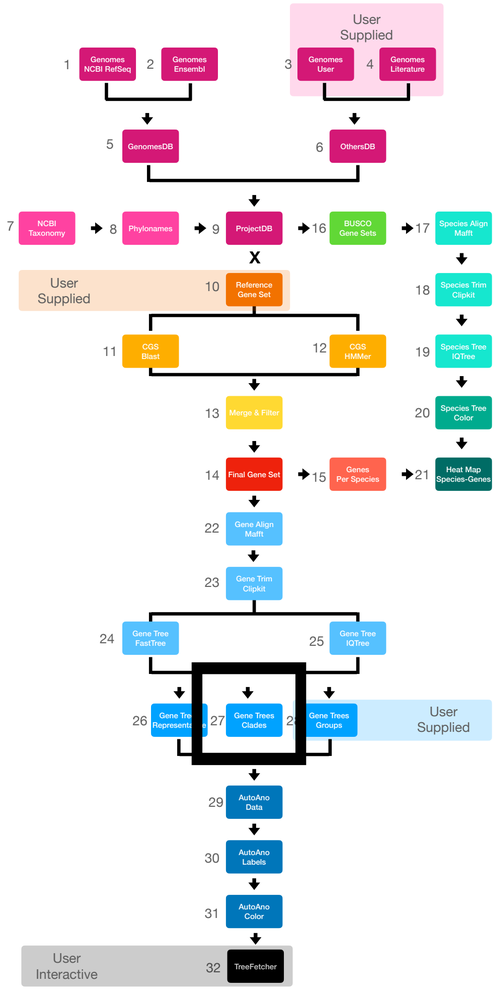
SCRIPTS

pipeline-blocks 4-trees 1-representative-trees

```
*000-user-provided-species-and-genes
```

Gene Tree Clades

GIGANTIC Phylogenomic Pipeline



STEP 27 **Gene Tree Clades** generates ML gene trees for all represented Phyloname Clades of species - and can include many additional species, as trees are phylum-level or less.

SCRIPTS

pipeline-blocks 4-trees 2-clade-trees

```
001-setup
002-python-clades
```

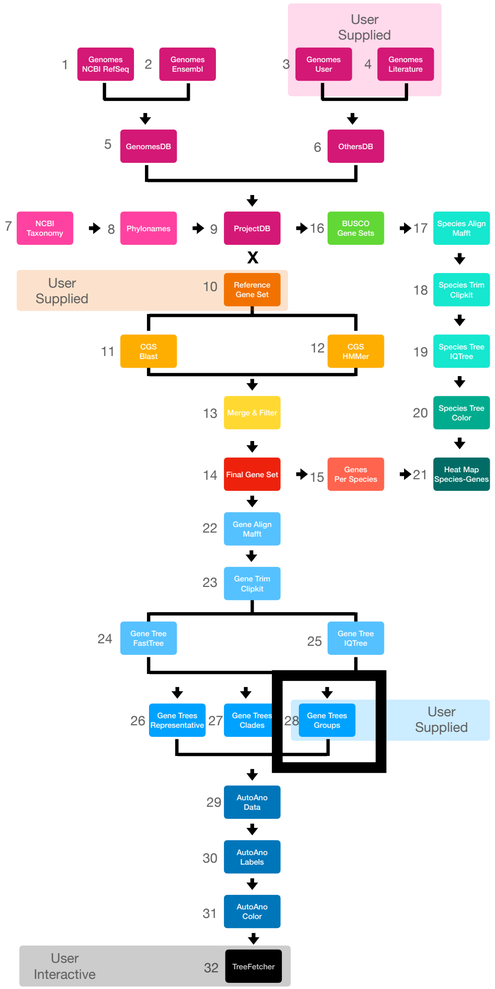
JAN

```
JAN
```

Jan

Gene Tree Groups

GIGANTIC Phylogenomic Pipeline



STEP 28 **Gene Tree Groups** generates ML gene trees for user-provided Groups of species - and can include many additional species, as Groups are typically small in number.

SCRIPTS

pipeline-blocks 4-trees 3-group-trees

```
000-python-group-sequences
000-rgs-counts-per-family
```

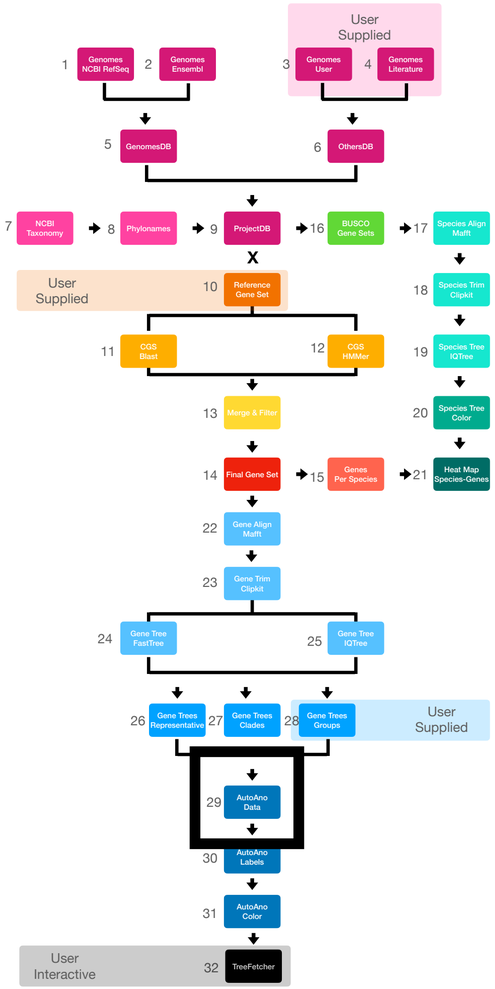
JAN

JAN

Jan

AutoAno Data

GIGANTIC Phylogenomic Pipeline



STEP 29 **AutoAno Data** generates Interproscan, genome coordinate, genome cluster, source identifier, and user-supplied annotation data sets.

SCRIPTS

pipeline-blocks 4-trees 4-pfam-trees interpro-scripts

```
000-python-ipr
001-ls-fastas
002-python-sed
003-sed-remove-asterisks
004-ls-fastas-no-asterisks
005-python-ipr-projectdb-genomes
006-interproscan-projectdb-genomes
007-interproscan-remaining-genomes
010-ls-tsv-files
```

pipeline-blocks 4-trees 4-pfam-trees interpro-scripts

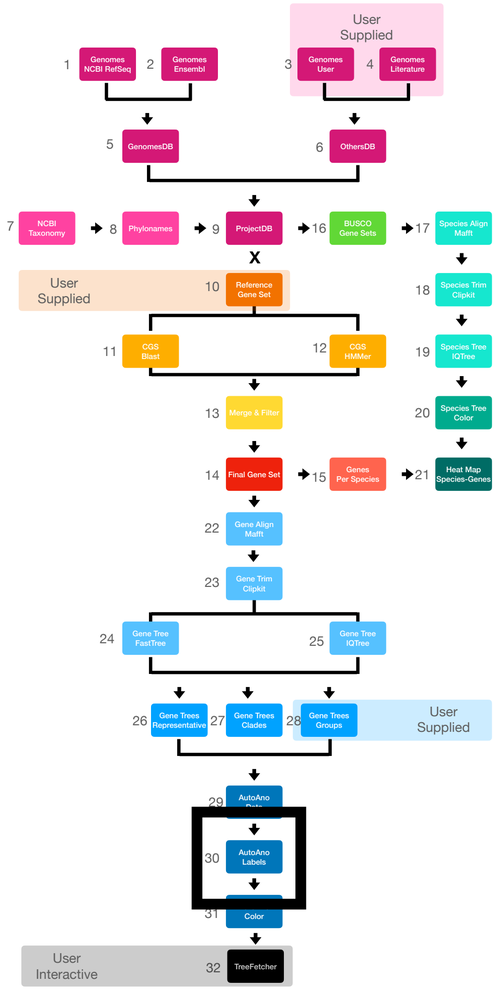
```
014-python-parse-interproscan-to-Pfam-tag-tree
```

```
001-cp-jan-analysis-files
002-ls-fasttree-files
003-python-update-treefiles-with-neighbor-status
004-python-intergenic-sizes
005-python-update-treefiles-with-neighbor-status-intergenic-size
MORE...
```

Jan

AutoAno Labels

GIGANTIC Phylogenomic Pipeline



STEP 30 **AutoAno Labels** generates trees re-annotated using annotation data from AnoAuto Data.

SCRIPTS

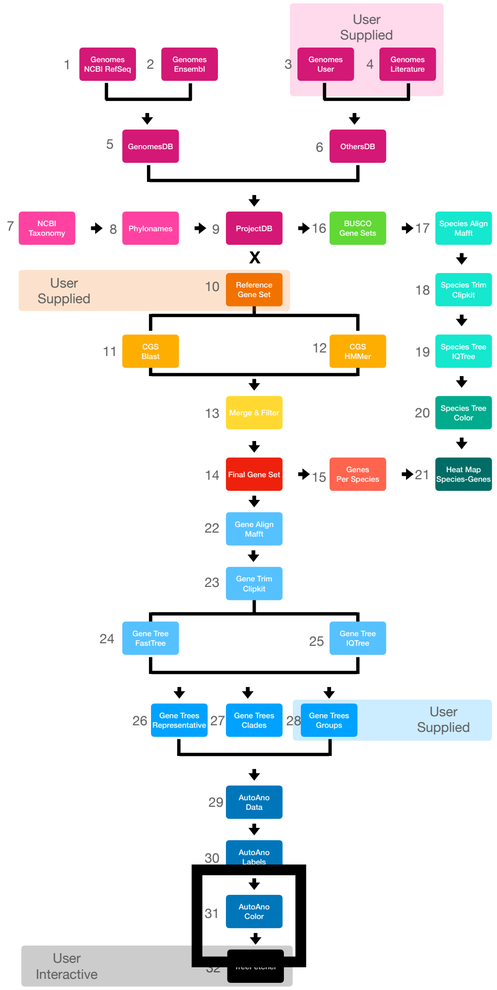
JAN Auto

run-ATP-pipeline.ipynb

Jan

AutoAno Color

GIGANTIC Phylogenomic Pipeline



STEP 31 **AutoAno Color** generates user-guided color annotated versions of all trees.

SCRIPTS

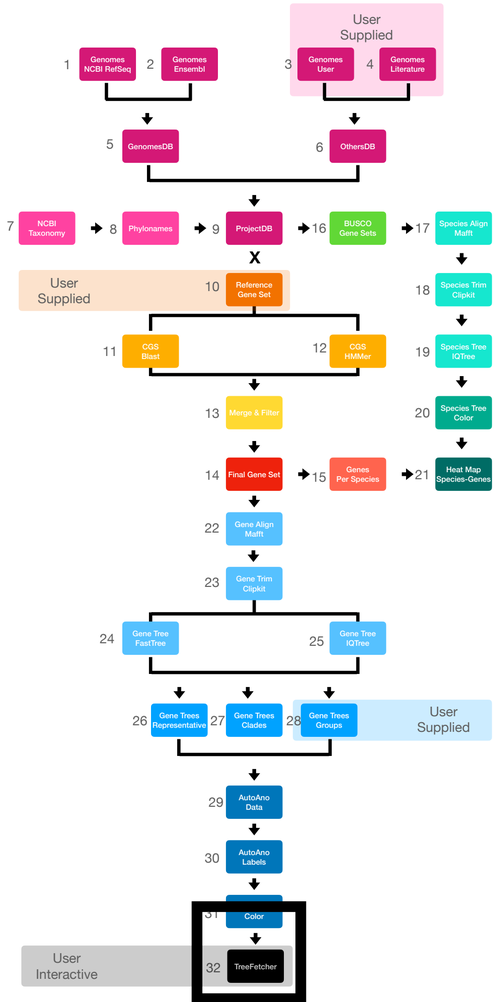
JAN Auto

run-ATP-pipeline.ipynb

Jan

TreeFetcher

GIGANTIC Phylogenomic Pipeline



STEP 32 **TreeFetcher** provides a lightweight HTML-based tool with a dropdown menu to view or download trees and other files in a web browser.

SCRIPTS
JAN

In progress

Jan

Thank you!