protein-phylogeny pipeline setup instructions

overview

This software pipeline is used for creating phylogenetic trees to identify orthologous proteins for set of proteins involved in a chemical pathway. Starting from a set of proteins involved in a specific signaling pathway in humans, it aims to identify orthologs for these proteins, and eventually pathway conservation across species. This document describes the outline and instructions for running this pipeline.

outline

> step1. select a specific pathway of interest from Reactome database
>
> step2. For this pathway, download protein set of this pathway from Reactome
>
> step3. create required files for pipeline and place them in specified location in folder hierarchy (done by script)
>
> step4. install all required software and dependencies. Setup required path variables and aliases
>
> step5. run pipeline for each protein in pathway [this is done in multiple steps]
>
> step6. check folder for orthologs that were identified by the pipeline
>
> step7. visualize phylogenetic tree

workflow of this pipeline

i. Reactome: Database of chemical reactions
ii. UniProt: Protein database
iii. BLASTp: BLAST for protein
iv. MAFFT: Multiple Sequence Alignment
v. HMMER: profile HMM
vi. Gblocks: clean sequence. size reduction. speedup.
vii. RAxML-NG: phylogenetic tree inference software based on Maximum Likelihood
viii. create species tree [this is based on NCBI Taxonomy information]
ix. NOTUNG: software for gene-tree species-tree reconcilation

software requirements

Below are software(versions) which have been tested.

i. Anoconda or Miniconda [1]
ii. Python version: 3.5.6 under conda environment [2]
iii. R version: 3.6.1 [3]
iv. HMMER version: 3.2.1 [4] , [5]
v. Gblocks version: 0.91b [6]
vi. NOTUNG version: 2.9.1.3 [7]
vii. RAxML-NG version [8],[9],[10]
viii. faSomeRecords *download link: faSomeRecords*
select your system (macOS/Linux etc..) and choose faSomeRecords.
ix. taxonkit [11]
x. ncbi-blast-2.10.0+ [12]
xi. UGENE [13]

protein-phylogeny pipeline

list of scripts used in this pipeline

1. reconcile1.sh
2. extract_protein_ids_from_tsv
3. extract_results_of_species_from_tsv
4. extract_seqs_from_hmmer_tsv_and_db
5. extract_seqs_from_tsv_and_db
6. extract_species_count_from_tsv
7. extract_headers_from_fasta.sh
8. 000_Merged_postRAxML_001to007.sh
9. 001_postRAxML.sh
10. 002_postRAxML.sh
11. 003_postRAxML.sh
12. 004_executeNotung.sh
13. 005_getOrthologs.sh
14. 006_getOrthoinfo.sh
15. 007_map_ID_**for**_new_orthologs.sh
16. blastp_search_and_reports_multiple_queries
17. get_sign_seqs
18. hmm_build
19. hmm_search
20. mafft_blast
21. mafft_hmm
22. raxml_it
23. remove_gaps
24. 001_**for**_raxml_ver9_specnr2specnames.py
25. 002_resolvePolytomy.R
26. 003_ete2notung.py
27. 004_RAxML2Notung.py
28. 005_get_orthologs.py
29. 006_get_orthoinfo.py
30. 007_map_ID_**for**_new_orthologs.py
31. pre_RAxML_resolve_duplicates.sh
32. 001_pre_RAxML_processing.sh
33. 002_pre_RAxML_processing.sh
34. 00_get_uniprot_data.py
35. 00_prepare_data.sh
36. (optional) batch_reconcile1.sh
37. phenotype_enrichment.py
38. phenotype_enrichment.sh

additional scripts only for referenceList of scripts that were included in ver.2018 of this pipeline.

*scripts below are not used in current workflow - myfunc.py - find_least_dupes.py - compare_orthologs.py - check_domains.py

list of databases used in this pipeline

i. protein.aliases.v11.0.txt 14, 15,16,17
ii. protein.sequences.v11.0.fa 14, 15 (this is used as BLASTDB)
iii. HomoSapiens_geneprot_database
iv. MusMusculus_geneprot_database
v. RattusNorvegicus_geneprot_database
vi. OryctolagusCuniculus_geneprot_database
vii. DanioRerio_geneprot_database
viii. CaenorhabditisElegans_geneprot_database
ix. DictyosteliumDiscoideum_geneprot_database

other data files

The above species databases only contain information for known orthologous proteins between humans and that species. To map protein ID of proteins newly identified as orthologs by this pipeline, we need mapping between ENSEMBL protein ID and ENSEMBL gene ID. This is required for phenotype enrichment.(next step after orthology mapping)

Following data files from BioMart (https://www.ensembl.org/biomart/martview/) contain this information. (*columns to include in these databases is upto the user as long as protein ID can be mapped to gene ID.)

i. ENSEMBL_Mouse_Gene_Prot_ID_mapping.txt
ii. ENSEMBL_Rat_Gene_Prot_ID_mapping.txt
iii. ENSEMBL_Zebrafish_Gene_Prot_ID_mapping.txt
iv. ENSEMBL_Celegans_Gene_Prot_ID_mapping.txt
v. ENSEMBL_Rabbit_Gene_Prot_ID_mapping.txt
vi. ENSEMBL_SlimeMould_Gene_Prot_ID_mapping.txt

other public databases used in this workflow

- Reactome 18, 19
- UniProt 20
- STRING 16, 17

Folder structure

This section describes folder structure for this pipeline. Scripts used in this pipeline assume files and folders are assigned names according to a specific format.This means folder names will have EXACTLY the same names as mentioned below.(including capital letters etc..) Additionally, these files/folders must be placed according to a specific folder hierarchy. Failure to do so will lead to errors.

i. create a folder called "DARTpaths" in your home directory.
ii. Below this, create the following four folders.

```
(1) software
(2) Proteins
(3) List_of_species
(4) databases
```

Following files should be placed within this folder. (directly under DARTpaths folder)

```
(1) protein.aliases.v11.0.txt
(2) extract_species_count_from_tsv
(3) extract_protein_ids_from_tsv
(4) extract_seqs_from_tsv_and_db
(5) extract_seqs_from_hmmer_tsv_and_db
(6) extract_results_of_species_from_tsv
```

i. Create following folders under "software" folder. (DARTpaths > software >)

```
(1) reconcile_scripts
(2) pre_RAxML
(3) blastdb
```

Following file/folders should be placed within this folder.(DARTpaths > software >)

```
[Folders]
 (1) Notung-2.9.1.3*
 (2) ncbi-blast-2.10.0+ (can be placed in another location as long as PATH is set correctly)
 (3) hmmer-3.3
 (4) Gblocks_0.91b
```

*if multiple versions are installed in your system, this can be placed under "Notung". (software > Notung > Notung-2.9.1.3). Modify script accordingly.

```
(1) ugeneInstaller_64bit
(2) taxonkit
(3) faSomeRecords
(4) reconcile1.sh
(5) pre_RAxML_resolve_duplicates.sh
(6) extract_headers_from_fasta.sh
(7) 000_Merged_postRAxML_001to007.sh
(8) 001_postRAxML.sh
(9) 002_postRAxML.sh
(10) 003_postRAxML.sh
(11) 004_executeNotung.sh
(12) 005_getOrthologs.sh
(13) 006_getOrthoinfo.sh
```

i. Following files should be placed within "reconcile_scripts" folder.(DARTpaths > software > reconcile_scripts >)

```
(1) 001_for_raxml_ver9_specnr2specnames.py
(2) 002_resolvePolytomy.py
(3) 003_ete2notung.py
(4) 004_RAxML2Notung.py
(5) 005_get_orthologs.py
(6) 006_get_orthoinfo.py
(7) blastp_search_and_reports_multiple_queries
(8) mafft_blast
(9) mafft_hmm
(10) hmm_search
(11) hmm_build
(12) remove_gaps
(13) raxml_it
(14) get_sign_seqs
```

ii. Following files should be placed under "pre_RAxML" folder. (DARTpaths > software > pre_RAxML >)

```
(1) 001_pre_RAxML_processing.sh
(2) 002_pre_RAxML_processing.sh
```

iii. Following files should be placed under "List_of_species" folder.(DARTpaths > List_of_species >)

```
(1) Subset_species_DHHC_filtered_speciesname.txt
(2) Subset_species_DHHC_filtered_ID.txt
```

iv. "blastdb" folder

This folder contains database files from STRING.(protein.sequences.v11.0.fa) This is created using ncbi-blast-2.10.0+ . (note: database files from BLASTp are not used. BLAST is only used for BLAST search, not as source of data. No database from BLAST is downloaded or used.) --> See **Setup BLAST** section.

Setup BLAST

i. download blast software from: *https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/*

(e.g.) ncbi-blast-2.10.0+-x64-macosx.tar.gz.md5 ncbi-blast-2.10.0+-x64-macosx.tar.gz

2.first check if it is downloaded properly. follow the steps below.

For macOS: 1). Download the file you want to check and open the download folder in Finder. 2). Open the Terminal, from the Applications / Utilities folder. 3). Type md5 followed by a space. Do not press Enter yet. 4). Drag the downloaded file from the Finder window into the Terminal window. 5). Press Enter and wait a few moments. 6). The MD5 hash of the file is displayed in the Terminal. 7). Open the checksum file provided on the Web page where you downloaded your file from. 8). The file usually has a .cksum extension. NOTE: The file should contain the MD5 sum of the download file. For example: md5sum: 25d422cc23b44c3bbd7a66c76d52af46 9). Compare the MD5 hash in the checksum file to the one displayed in the Terminal.If they are exactly the same, your file was downloaded successfully. Otherwise, download your file again.

from command line:

> md5 ncbi-blast-2.10.0+-x64-macosx.tar.gz (press enter) cat ncbi-blast-2.10.0+-x64-macosx.tar.gz.md5 (press enter)

if the checksum is identicle, OK!

Extract the file in the folder.

> tar zxvpf $HOME/DARTpaths/software/ncbi-blast-2.10.0+-x64-macosx.tar.gz

The following command will show information on how to setup this file (protein.sequences.v11.0.fa) as BLASTDB. Follow the instructions.

> makeblastdb -help

Example of setting up protein.sequences.v11.0.fa as **BLASTDB** in your environment: Assumes protein.sequences.v11.0.fa is placed under the following location.

DARTpaths/blastdb/protein.sequences.v11.0.fa

> makeblastdb -in protein.sequences.v11.0.fa -out string_proteins -dbtype prot -title string_proteins -parse_seqids

Using STRING files as database

After setting up BLAST, set the following as blastdb.

1). protein.sequences.v11.0.fa

2). protein.aliases.v11.0.txt

Alias and path settings

Add the following in your**.bash_profile** file(or equivalent).

> export PATH=$PATH:$HOME/DARTpaths/software/ncbi-blast-2.10.0+/bin/ export

> BLASTDB=$BLASTDB:$HOME/DARTpaths/blastdb/

following might be required depending on your system's configuration.

> export LD_LIBRARY_PATH=$HOME/lib:$LD_LIBRARY_PATH export PKG_CONFIG_PATH=$HOME/lib/pkgconfig:$PKG_CONFIG_PATH

Create species databases

The details for this are included in

Run the pipeline

step1. Data preparation

First you will need a file of proteins in the pathway downloaded from Reactome. (see sample file:) This file contains protein identifiers/names of pathway. Keep this file under (DARTpaths > Pathway_files >). Make sure you have created a folder with the name of pathway under (DARTpaths > Pathways >).

For example if your pathway is called "ABC", a folder called "ABC" shoud be kept (DARTpaths > Pathways > ABC).

script ***00_prepare_data.sh*** will create folders and generate necessary files under this folder(this pathway's folder). This script will create a folder called **PATHWAYNAME_PROTEINNAME_searches_literature** under this pathway's folder. Below this folder, it will keep three files which contain FASTA sequence of proteins downloaded from UniProt and necessary information.

> 00_prepare_data.sh [*PathwayName*] [*PathwayFile*]

Example.

> 00_prepare_data.sh ABC ABC_pathway_proteins_R-HSA-1234567.tsv

step2. reconcile1.sh

This script takes the FASTA file from the previous step and runs BLAST search(BLASTp), MAFFT, HMMER & GBlocks and creates a multiple sequence alignment. change to directory of protein in the pathway. This script is run for each protein in the pathway separately.

> reconcile1.sh [Pathwayname_ProteinName]

step 3. pre-RAxML processing

Before passing this MSA to RAxML-NG for creating phylogeny, we will filter data based on list of species. This is to save time as large datasets can take very long to compute trees even on cluster. The resulting file only contains data for this list of species.

> 001_pre-RAxML.sh 002_pre-RAxML.sh

step4. create phylogeny (reconcile2)

With the MSA from the previous step, this creates a phylogeny. This can be done in two ways depending on the size of MSA.(on your computer OR on cluster) Options specified for creating this phylogeny is given in the sample file (.pbs file)

on your computer Run the following script. See instructions inside the script. (also see comment in reconcile1.sh)

reconcile2.sh

Run on cluster 1. prepare PBS file (see example file) 2. prepare MSA for RAxML-NG. see9.

> raxml-ng --check raxml-ng --parse

The current version(used in this workflow) of RAxML-NG will remove identical sequences with different names randomly. This means if (for example) your query protein(the one you started with(human pathway protein)) has an identical sequence, it might be removed. Therefore, it is advised always to make sure your query sequence is included in the file you give to RAxML-NG.

Tips for RAxML-NG on cluster

some sets can take hours to just load on the cluster.---> use binary format file(.rba file) to save time. This can be generated by, --parse command on your local machine.

step 5. post RAxML-NG processing (identify orthologs and retrieve its information)

Once the phylogeny (gene tree) is created, download this(tree files generated by RAxML-NG) from the cluster (or keep in folder if done locally) .

The file that will be used in the following steps is the one which has the suffix**support** . This file contains bootstrap support values. Keep these files under **PathwayName_proteinName _trees** folder.

(DARTpaths > Pathways > PathwayName > PathwayName_proteinName > PathwayName_proteinName _trees)

Once the files are kept in the correct location, run**000_Merged_postRAxML_001to007.sh** . This will create necessary folders and keep results under them. Orthologs will be kept under **PathwayName_ProteinName_orthologs** folder. This script is run for each protein in the pathway separately.

> 000_Merged_postRAxML_001to007.sh [PathwayName_ProteinName] [STRING ID of this human protein]

Example.

> 000_Merged_postRAxML_001to007.sh AHR_ARNT "9606.ENSP00000351407"

step 6. Preparing data for phenotype enrichment

After running the above for all the proteins in the pathway, each protein folder will contain information on orthologs that were identified. Script 001 to 006 are used to identify orthologs. Script 007 is used to map gene IDs for these orthologs proteins and organize them per species and keep a copy of this just below the pathway's folder in "Orthologs_PATHWAYNAME_pathway" folder. For example, "Orthologs_Hedgehog_pathway" folder will be located in the following path:

DARTpaths/Pathways/Hedgehog/Orthologs_Hedgehog_pathway

Below this there are folders for each species where these ortholog files of each protein in this pathway for that species will be kept after running the script *000_Merged_postRAxML_001to007.sh*.

Once 000_Merged_postRAxML_001to007.sh has been completed for all the proteins in the pathway, generate a merged file per species. This is used in the next step for phenotype enrichment.

NOTE: This step retrieves ENSEMBL Gene IDs for orthologous proteins identified by this pipeline. In some cases, Gene IDs are not found. This could be due to "Retired" gene IDs etc. A file with "ID_NOT_FOUND" for each species stores information for these orthologs. Search them manually.

-Example- Go to the species' ortholog folder of this pathway.
DARTpaths/Pathways/Hedgehog/Orthologs_Hedgehog_pathway/Mouse_orthologs

Following command will create a merged file of all the orthologs for this species' orthologs in this pathway.

> cat * > Merged_Hedgehog_Mouse_orthologs.txt

In the next step, in phenotype enrichment, this will be combined with the ones extracted from species database (ENSEMBL compara).

Phenotype Enrichment of orthologs

This is second step of DARTpaths pipeline. The aim of this is to get an idea on phenotypes that might be observed for orthologous genes between humans and other species. Using proteins/genes in the human signaling pathway, it takes orthologous genes in species and collects information from phenotype databases for these genes.

Databases

Below are the databases used in this step. These can be downloaded from their website. Databases for orthologs(ENSEMBL compara) which are used in orthology mapping pipeline ("species database") are also used in this step. *** Download recent version of database files for your analysis. Below is just example.

(1) C.elegans phenotype data

WormBase (https://wormbase.org)

ftp://ftp.wormbase.org/pub/wormbase/releases/WS264/ONTOLOGY/phenotype_association.WS264.wb

ontology data:

file: phenotype_association.WS264.obo.txt

(2) Mouse phenotype data

IMPC (https://www.mousephenotype.org)

ftp://ftp.ebi.ac.uk/pub/databases/impc/all-data-releases/release-11.0/csv/ALL_genotype_phenotype.csv.gz

ontology data:

file: Mpheno_OBO.ontology.txt

(3) Zebrafish phenotype data

ZFIN (https://zfin.org)

https://zfin.org/downloads/phenoGeneCleanData_fish.txt

This is the file "Phenotype of Zebrafish Genes" under Phenotype Data.

ontology data :

https://zfin.org/downloads , under Category "Anatomical Ontologies", download file "Zebrafish Anatomy Term".
file: https://zfin.org/downloads/anatomy_item_YYYY.MM.DD.txt

file(for GO term): go_annotation_obo

(4) Dictyostellium phenotype data

http://dictybase.org/db/cgi-bin/dictyBase/download/download.pl?area=mutant_phenotypes&ID=all-mutants-

ddb_g.txt

ontology data :

file: all-mutants-ddb_g.txt

(5) Mapping between ENSEMBL and Reactome

Reactome (https://reactome.org/download-data)

https://reactome.org/download/current/Ensembl2Reactome_All_Levels.txt

Script for phenotype enrichment

- phenotype_enrichment.py
- phenotype_enrichment.sh

phenotype_enrichment.py is called from phenotype_enrichemnt.sh. This (.py) can be run independently as long as necessary arguments are specified. Running this will create a summary files with results.

usage:

> ***python phenotype_enrichment.py*** [Reactome_ID of_pathway] [PATHWAY_NAME] [PATH to DARTpaths folder]

Friday, 30. October 2020 04:03PM