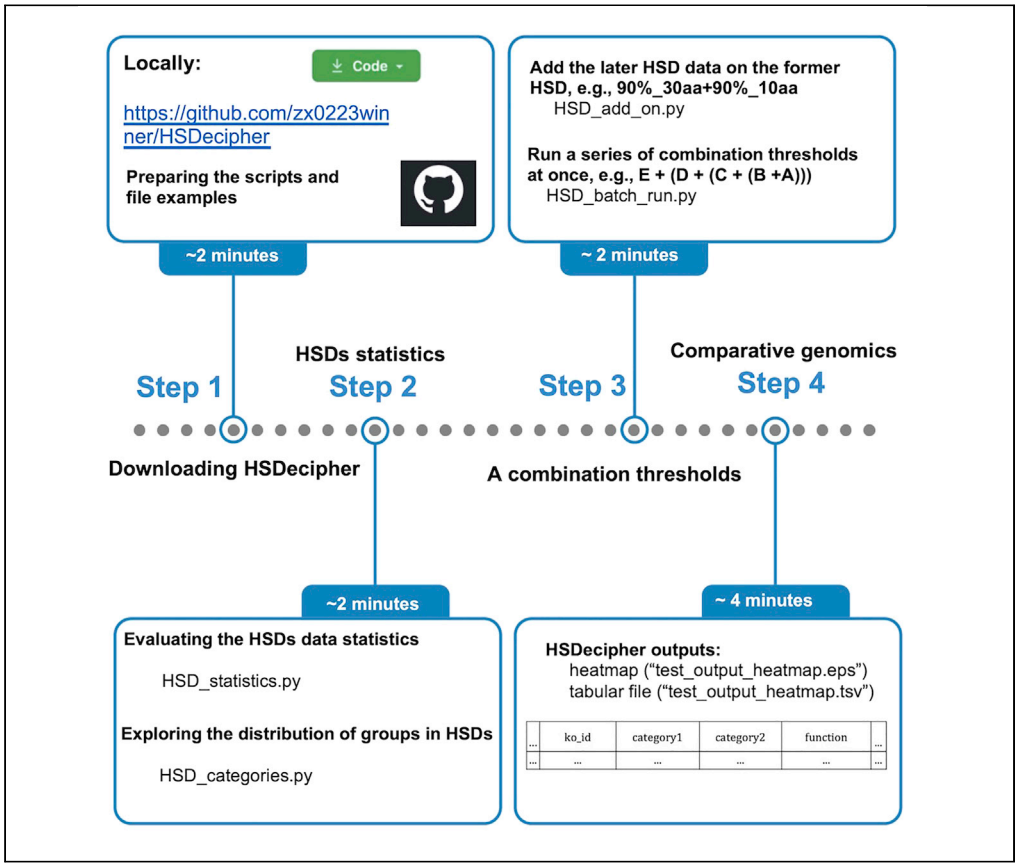
STAR Protocols

CellPress
OPEN ACCESS

Protocol

HSDecipher: A pipeline for comparative genomic analysis of highly similar duplicate genes in eukaryotic genomes



Many tools have been developed to measure the degree of similarity between gene duplicates within and between species. Here, we present HSDecipher, a bioinformatics pipeline to assist users in the analysis and visualization of highly similar duplicate genes (HSDs). We describe the steps for analysis of HSDs statistics, expanding HSD gene set, and visualizing the results of comparative genomic analyses. HSDecipher represents a useful tool for researchers exploring the evolution of duplicate genes in select eukaryotic species.

Publisher's note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.

Xi Zhang, Yining Hu, Zhenyu Cheng, John M. Archibald

xi.zhang@dal.ca (X.Z.)
john.archibald@dal.ca (J.M.A.)

Highlights

The HSDecipher pipeline analyzes highly similar duplicate genes (HSDs) in eukaryotes

HSDecipher pipeline statistics can be acquired using custom scripts

A larger dataset of HSDs is acquired by using a series of combination thresholds

Groups of HSDs can be visualized and compared within or between species

Zhang et al., STAR Protocols 4, 102014
March 17, 2023
https://doi.org/10.1016/j.xpro.2022.102014

STAR Protocols

CellPress
OPEN ACCESS

Protocol

HSDecipher: A pipeline for comparative genomic analysis of highly similar duplicate genes in eukaryotic genomes

Xi Zhang,[1,2,5,*] Yining Hu,[3] Zhenyu Cheng,[2,4] and John M. Archibald[1,2,6,*]

[1]Department of Biochemistry and Molecular Biology, Dalhousie University, Halifax, NS B3H 4R2, Canada

[2]Institute for Comparative Genomics, Dalhousie University, Halifax, NS B3H 4R2, Canada

[3]Department of Computer Science, Western University, London, ON N6A 5B7, Canada

[4]Department of Microbiology and Immunology, Dalhousie University, Halifax, NS, Canada

[5]Technical contact

[6]Lead contact

*Correspondence: xi.zhang@dal.ca (X.Z.), john.archibald@dal.ca (J.M.A.)
https://doi.org/10.1016/j.xpro.2022.102014

SUMMARY

Many tools have been developed to measure the degree of similarity between gene duplicates within and between species. Here, we present HSDecipher, a bioinformatics pipeline to assist users in the analysis and visualization of highly similar duplicate genes (HSDs). We describe the steps for analysis of HSDs statistics, expanding HSD gene sets, and visualizing the results of comparative genomic analyses. HSDecipher represents a useful tool for researchers exploring the evolution of duplicate genes in select eukaryotic species.

For complete details on the use and execution of this protocol, please refer to Zhang et al. (2021)[1] and Zhang et al. (2022).[2]

BEFORE YOU BEGIN

Gene duplication has long been recognized as an important process in molecular evolution. Due to interest in identifying the possible role of duplicate genes in organismal adaptation,[3] bioinformatics tools have been developed to aid in their detection.[4] However, distinguishing orthologs (i.e., genes that differ due to speciation) from recently-evolved paralogs (genes that arose by gene duplication) can still be difficult. Hundreds of highly similar duplicate genes (HSDs) were recently identified in the genome of an Antarctic green alga *Chlamydomonas* sp. UWO241 (renamed *Chlamydomonas priscuii*).[5–7] The HSDs were found using HSDFinder[1] and characterized alongside those in other eukaryotic genomes in HSDatabase.[2] In a previously published protocol the use and application of HSDFinder was presented.[8] Here we describe the step-by-step use of custom scripts in HSDecipher, which allows researchers to carry out a downstream analysis of HSDs in eukaryotic species of interest.

Requirements for setting up the pipeline

HSDecipher contains a set of custom Python scripts to visualize and interpret the data generated by HSDFinder. Sample output files can be found at the following link: https://github.com/zx0223winner/HSDecipher. Here, five Python scripts are presented and applied sequentially in a pipeline. To run locally, pre-installed Python (preferably Python 3) and Linux (e.g., Ubuntu 20.04 LTS) environments are required. The other necessary scripts and data can be accessed via the links in the key resources table.

Note: To allow the comparative analysis data to be visualized in a heatmap, the minimum specification is a computer with 2 cores, 4 GB of RAM and 128 GB storage.

KEY RESOURCES TABLE

RESOURCE	SOURCE	IDENTIFIER*
Deposited data		
Chlamydomonas reinhardtii	GenBank: GCA_000002595.3[9]	https://www.ncbi.nlm.nih.gov/genome/?term=txid3055[orgn]
Arabidopsis thaliana	GenBank: GCA_000001735.2 [10,11]	https://www.ncbi.nlm.nih.gov/genome/?term=GCA_000001735.2
Homo sapiens	GenBank: GCF_000001405.39[12,13]	https://www.ncbi.nlm.nih.gov/genome/?term=txid63221[Organism:noexp]
Software and algorithms		
Python 3	The Python community	SCR_008394; https://www.python.org/downloads/
pandas v1.2.2	Python Data Analysis Library	https://pandas.pydata.org
HSD_statistics.py, *HSD_categories.py*, *HSD_add_on.py*, HSD_batch_run.py, and *HSD_heatmap.py*	This study	https://github.com/zx0223winner/HSDecipher

*Note: Identifier is used from the RRID portal (https://scicrunch.org/resources).

MATERIALS AND EQUIPMENT

The software implementation was written in Python 3 using the following custom scripts and platforms: *HSD_statistics.py*, *HSD_categories.py*, *HSD_add_on.py*, *HSD_batch_run.py*, and *HSD_heatmap.py*. For example, (1) *HSD_statistics.py* is a python script that calculates the statistics of HSDs using a variety of HSDFinder thresholds. The output file is written in a table with the following headers: File name, Candidate HSDs, Non-redundant gene copies, Gene copies, True HSDs, Space, Incomplete HSDs, Capturing value, and Performance score. 'Capturing value' and 'Performance score' are two parameters used to evaluate the HSD results.[1] (2) *HSD_categories.py* counts the number of HSDs with two, three, and more than four categories, which is helpful when evaluating the distribution of duplicate groups within HSDs. (3) Since the similarity of duplicate genes within and among genomes can vary significantly, by using the scripts *HSD_add_on.py* and *HSD_batch_run.py*, users can add newly curated HSDs using a combination of thresholds to assemble a larger dataset of HSD candidates. (4) *HSD_heatmap.py* visualizes the collected HSDs in a heatmap and compares HSDs sharing the same predicted biochemical pathway function. The KEGG database has been used to provide KO accession numbers for each gene model identifier.[14] In this step-by-step protocol, we use the results of an HSD analysis of the genomes of *Chlamydomonas reinhardtii*, *Arabidopsis thaliana* and *Homo sapiens* to illustrate how to perform downstream analysis with these custom Python scripts.

STEP-BY-STEP METHOD DETAILS

Downstream analysis of HSD statistics

Timing: ~2 min (Depending on file sizes and internet speed)

This step performs a preliminary evaluation of the HSD results obtained using the HSDFinder tool.[8]

Note: By using different thresholds for amino acid length and pairwise identities, users can filter the groups of HSDs from the all-against-all BLAST protein sequence similarity search (E-value cut-off ≤1e-10). We used a short form of the sequence similarity assessment metrics, such as 90%_10aa, which refers to amino acid pairwise identity ≥90%, and amino acid aligned length variance ≤10. When naming the files, users should adhere to this format (species_name.identity_length.txt; e.g., "Chlamydomonas_reinhardtii.90_10.txt"), thereby allowing recognition of the output by downstream scripts.

1. Users can first acquire the HSDecipher package from GitHub (https://github.com/zx0223winner/HSDecipher).

   Note: In the HSDs folder, we have prepared an HSDFinder analysis for three model species: *C. reinhardtii*, *A. thaliana*, and *H. sapiens*. To save processing time, the comparatively small genome of *C. reinhardtii* is used as the study case.

```
# Clone the package and move to the HSDecipher/directory

>git clone https://github.com/zx0223winner/HSDecipher

# install python3 and relevant libraries

pip3 install python

# Chlamydomonas_reinhardtii.90_10.txt

XP_001689821.1 XP_001689821.1; XP_001690281.2 241; 241 Pfam PF00011; PF00011 Hsp20/alpha crystallin family; Hsp20/alpha crystallin family 2.0E-10; 2.8E-10 IPR002068; IPR002068 Alpha crystallin/Hsp20 domain; Alpha crystallin/Hsp20 domain
```

2. Users can run the Python scripts HSD_statistics.py and HSD_categories.py, which can be found in the GitHub main directory.

```
# HSD_statistics.py

>python3 HSD_statistics.py <path to HSD species folder> <format of HSD file. e.g., 'txt' or 'tsv'> <output file name. e.g., species_stat.tsv>

# In our case of HSD data in C. reinhardtii genome

>Python3 HSD_statistics.py /HSDs_folder/Chlamydomonas_reinhardtii txt Chlamy_stat.tsv
```

```
# HSD_categories.py

>python3 HSD_categories.py <path to HSD species folder> <format of HSD file. e.g., 'txt' or 'tsv'> <output file name. e.g., species_groups.tsv>

# In our case of HSD data in C. reinhardtii genome

>Python3 HSD_categories.py /HSDs_folder/Chlamydomonas_reinhardtii txt Chlamy_groups.tsv
```

⚠ CRITICAL: In Table 1, 'Candidate HSDs' indicates the number of highly similar gene duplicate candidates; True HSDs are duplicate groups satisfying the respective thresholds and gene copies containing the same domain(s); Non-redundant gene copies are the number of unique gene copies in each group of HSDs; Gene copies are the total number of gene copies in each group of HSDs; The number of spaces indicates the number of gene copies encoding the putative function without any conserved domain(s) with hits to the Pfam database (e.g., hypothetical proteins); Capturing value indicates the levels of predicted HSDs; Performance score is a value that allows users to assess the performance of the HSD retreival process. Troubleshooting 1.

⚠ CRITICAL: In Table 2, '2-group HSDs' refers to the number of HSD categories containing only two gene copies. Troubleshooting 2.

Using a series of combination thresholds to expand an HSD gene dataset

Timing: ~2 min (Depending on file sizes, computing power, and internet speed) (for step 3)

Users will require the Python scripts *HSD_add_on.py* and *HSD_batch_run.py* to run the following analysis.

3. *HSD_add_on.py* can add newly acquired HSD data to original HSD output, thereby enlarging the HSD candidate dataset.

```
# HSD_add_on.py

#HSD_add_on.py python3 HSD_add_on.py -i <inputfile> -a <adding_file> -o <output file>

# In our case of HSD data in C. reinhardtii genome

>Python3 HSD_add_on.py -i /HSDs_folder/Chlamydomonas_reinhardtii/ Chlamydomonas_rein-
hardtii.90_10.txt -a Chlamydomonas_reinhardtii.90_30.txt -o
Chlamydomonas_reinhardtii.90_10_90_30.txt
```

Note: For example, HSDs identified at a threshold of 90%_30aa were added to those identified at a threshold of 90%_10aa (denoted as "90%_30aa+90%_10aa").

⚠ CRITICAL: Any redundant candidate HSDs acquired at each combination threshold are removed if the more relaxed threshold (e.g., 90%_30aa) retrieves the identical genes from the stricter cut-off (e.g., 90%_10aa).

Troubleshooting 3.

```
# HSD_batch_run.py

>python3 batch_run.py -i <inputfolder>

# In our case of HSDs data

>Python3 HSD_categories.py /HSDs_folder/
```

⚠ CRITICAL: *HSD_batch_run.py* can execute a series of combination threshold analyses at once. Users should back up the original HSDs folder before running the *HSD_batch_run.py* script. To minimize redundancy and to acquire a larger dataset of HSD candidates, we processed each selected species with the following combination of thresholds:

Troubleshooting 4.

```
# Chlamydomonas_reinhardtii.90_10.txt

XP_001689450.1 XP_001689450.1; XP_001700901. 1 280; 276 Pfam PF01459; PF01459 Eukaryotic
porin; Eukaryotic porin 1.3E-34; 3.5E-39 IPR027246; IPR027246 Eukaryotic porin/Tom40; Eu-
karyotic porin/Tom40

XP_001689455.1 XP_001689455.1; XP_001698498.1 194; 161 Pfam PF08534; PF08534 Redoxin; Re-
doxin 4.2E-35; 1.1E-36 IPR013740; IPR013740 Redoxin; Redoxin
```

Note: The resulting output file of HSDs based on a combination of thresholds will appear in HSDs_folder, e.g., "Chlamydomonas_reinhardtii.90_10.txt", "Arabidopsis_thaliana.90_10.txt" and "Homo_sapiens.90_10.txt".

Downstream comparative genomic analysis of HSDs in eukaryotic genomes

Timing: ~4 min (Depending on the size of the data, computing power, and internet speed) (for step 4)

In this step, users can apply the *HSD_heatmap.py* script on the previous generated HSD results to perform a comparative analysis.

4. Users can compare different thresholds of HSDs in one genome or HSDs retrieved from different genomes in a heatmap (Figures 1 and 2).

Note: the data can be derived from multiple genomes with a species or the genomes of different species. The generated tabular file (Table 3) collects gene duplicates predicted to be involved in the same biological process or biochemical pathway, which can be used for natural selection analysis.

```
# HSD_heatmap.py

# For intra-species

>python3 HSD_heatmap.py -f <HSD file folder> -k <KO file folder> -r <width of output heatmap, e.g., 30 pixels> -c <height of output heatmap, e.g., 20 pixels>

>python3 HSD_heatmap.py -f /HSDs_folder/Chlamydomonas_reinhardtii/ -k /ko/ -r 30 -c 20
```

Note: The generated examples can be found in the heatmap folder under the HSDecipher main directory, such as the high resolution heatmap file "Chlamydomonas_reinhardtii_output_heatmap.eps" and the tabular file "Chlamydomonas_reinhardtii_output_heatmap.tsv".

```
# HSD_heatmap.py, for inter-species analysis

>python3 HSD_heatmap.py -f <HSD file folder> -k <KO file folder> -r <width of output heatmap, e.g., 30 pixels > -c <height of output heatmap, e.g., 20 pixels >

>python3 HSD_heatmap.py -f /HSDs_folder/ -k /ko/ -r 30 -c 20
```

Note: The inter-species analysis example can be found in the heatmap folder under the HSDecipher main directory with the name "test_output_heatmap.eps" and "test_output_heatmap.tsv".

⚠ CRITICAL: It is important to name the KEGG pathway KO file and HSD result file correctly so that they can be recognized by the HSDecipher scripts. For example, the KO information file for each species should be formatted as follows: "species_name.ko.txt" (e.g., Chlamydomonas_reinhardtii.ko.txt); the HSDs results file should be named "species_name.thresholds_thresholds.txt" (e.g., Chlamydomonas_reinhardtii.90_10.txt).

Troubleshooting 5.

Table 1. Example of HSDecipher statistics file based on the output file from HSDFinder

File_name	Candidate_ HSDs#	Non-redundant gene copies#	Gene copies#	True HSDs#	Space#	Incomplete HSDs#	Capturing value	Performance score
Chlamydomonas_reinhardtii.50_10	577	1625	1662	508	301	69	88.04	11.2
Chlamydomonas_reinhardtii.50_100	1089	3746	4954	915	487	174	84.02	8.67
Chlamydomonas_reinhardtii.50_30	822	2518	2676	710	380	112	86.37	10.19
Chlamydomonas_reinhardtii.50_50	942	3037	3462	814	422	128	86.41	10.34
Chlamydomonas_reinhardtii.50_70	1029	3389	4168	873	452	156	84.84	9.24
Chlamydomonas_reinhardtii.60_10	475	1380	1388	423	267	52	89.05	11.91
Chlamydomonas_reinhardtii.60_100	864	2910	3109	753	437	111	87.15	10.54
Chlamydomonas_reinhardtii.60_30	649	2034	2036	575	339	74	88.6	11.8
Chlamydomonas_reinhardtii.60_50	741	2409	2431	661	374	80	89.2	12.69
Chlamydomonas_reinhardtii.60_70	809	2657	2791	711	402	98	87.89	11.29
Chlamydomonas_reinhardtii.70_10	405	1204	1210	365	242	40	90.12	12.88
Chlamydomonas_reinhardtii.70_100	694	2355	2566	623	394	71	89.77	12.82
Chlamydomonas_reinhardtii.70_30	538	1704	1704	486	307	52	90.33	13.53
Chlamydomonas_reinhardtii.70_50	599	1981	1999	549	333	50	91.65	15.98
Chlamydomonas_reinhardtii.70_70	649	2161	2181	590	361	59	90.91	14.63
Chlamydomonas_reinhardtii.80_10	335	1030	1082	307	213	28	91.64	14.79
Chlamydomonas_reinhardtii.80_100	533	1910	1922	483	329	50	90.62	13.47
Chlamydomonas_reinhardtii.80_30	444	1450	1456	402	270	42	90.54	13.4
Chlamydomonas_reinhardtii.80_50	474	1648	1750	435	287	39	91.77	15.55
Chlamydomonas_reinhardtii.80_70	499	1772	1826	457	306	42	91.58	15.12
Chlamydomonas_reinhardtii.90_10	293	858	867	275	196	18	93.86	19.58
Chlamydomonas_reinhardtii.90_100	409	1486	2103	364	269	45	89	10.96
Chlamydomonas_reinhardtii.90_30	347	1128	1519	316	229	31	91.07	13.56
Chlamydomonas_reinhardtii.90_50	372	1281	1774	337	240	35	90.59	13.03
Chlamydomonas_reinhardtii.90_70	391	1380	1968	352	252	39	90.03	12.28

EXPECTED OUTCOMES

HSDecipher is a set of custom scripts for users who are interested in performing downstream analysis of highly similar gene duplicates obtained using HSDFinder. In the first two steps, analysis of HSD statistics and categories can help users evaluate the distribution and quality of HSD data. The third step allows users to expand their dataset of HSDs based on consideration of multiple sequence similarity assessment metrics; in other words, HSD datasets can be enlarged by adding more data using relaxed thresholds following removal of duplicates retrieved using different thresholds. For example, HSDs identified at a threshold of 80%_50aa can be added to those identified at a threshold of 80%_30aa (denoted as "80%_50aa+80%_30aa"); If the more relaxed threshold (i.e., 80%_50aa) contains identical genes acquired using the stricter cut-off (i.e., 80%_30aa), the combined HSD candidates can be filtered to remove the redundancy. In the last step, users can carry out a comparative genomics analysis of intra-/inter-genomic analysis of HSD data using a heatmap, which shows the functional distribution of HSDs or the levels of HSD sequence similarity shared between different species. Users can easily visualize and compare those significant enriched HSDs. Users are also provided with a tabular file to compare HSDs with the same KEGG pathway function, thereby allowing them to conveniently choose HSDs for downstream comparative genomic analysis (e.g., identification of signatures of natural selection).

LIMITATIONS

There is a steep learning curve for researchers with limited knowledge of bioinformatics, especially those who are not familiar with the basic command lines and dash shell in a Linux/Unix environment. At the present time, a "one-click solution" does not exist because of the desire to retain flexibility in the usage of our scripts for different purposes. That said, HSDecipher is comparatively easier to use than some of the other options currently available, such as PhylomeDB[15] and OrthoFinder.[16,17] At present there are very few tools that can execute downstream comparative genomics analysis of

Table 2. Example of HSDecipher categories result file based on the output file from HSDFinder

File_name	2-group_HSDs#	3-group_HSDs#	>=4-group_HSDs#
Chlamydomonas_reinhardtii.50_10	420	87	70
Chlamydomonas_reinhardtii.50_100	689	185	215
Chlamydomonas_reinhardtii.50_30	554	137	131
Chlamydomonas_reinhardtii.50_50	622	157	163
Chlamydomonas_reinhardtii.50_70	662	168	199
Chlamydomonas_reinhardtii.60_10	344	69	62
Chlamydomonas_reinhardtii.60_100	589	131	144
Chlamydomonas_reinhardtii.60_30	437	107	105
Chlamydomonas_reinhardtii.60_50	509	112	120
Chlamydomonas_reinhardtii.60_70	553	125	131
Chlamydomonas_reinhardtii.70_10	291	63	51
Chlamydomonas_reinhardtii.70_100	482	94	118
Chlamydomonas_reinhardtii.70_30	369	85	84
Chlamydomonas_reinhardtii.70_50	413	90	96
Chlamydomonas_reinhardtii.70_70	449	92	108
Chlamydomonas_reinhardtii.80_10	232	57	46
Chlamydomonas_reinhardtii.80_100	357	80	96
Chlamydomonas_reinhardtii.80_30	297	76	71
Chlamydomonas_reinhardtii.80_50	316	78	80
Chlamydomonas_reinhardtii.80_70	331	82	86
Chlamydomonas_reinhardtii.90_10	210	47	36
Chlamydomonas_reinhardtii.90_100	267	57	85
Chlamydomonas_reinhardtii.90_30	220	56	71
Chlamydomonas_reinhardtii.90_50	246	51	75
Chlamydomonas_reinhardtii.90_70	259	53	79

highly similar duplicate gene data. HSDecipher thus fills a need for the bioinformatics and genomics community.

Since there is no golden rule to distinguish partial duplicates from more complete ones, a combination of thresholds is used to acquire a larger dataset of HSD candidates. But due to the limitation of this strategy, it should be noted that there are some large groups of HSD candidates in the database that likely diverged in function from one another. Users should thus proceed with caution when working with these types of datasets.

TROUBLESHOOTING

Problem 1

Why are non-redundant gene copies (i.e., gene duplicates) listed as a column in Table 1? (step 2).

Potential solution

Since the HSDs are filtered based on the all-against-all BLAST protein similarity search and the BLAST algorithms can limit the maximum target hits by default, it is possible that not all HSD gene copies group together based on a simple transitive link between the remaining genes, especially for genomes with many gene duplicates. Users can manually increase the setting of maximum target hits in their BLAST searches to solve this problem.

Problem 2

Do the HSD datasets include protein sequences from alternative splicing? (step 2).

Potential solution

To count the genuine gene copies in each group of HSDs, we suggest users remove isoforms derived from alternative splicing and keep the one with longest transcript length as the primary protein

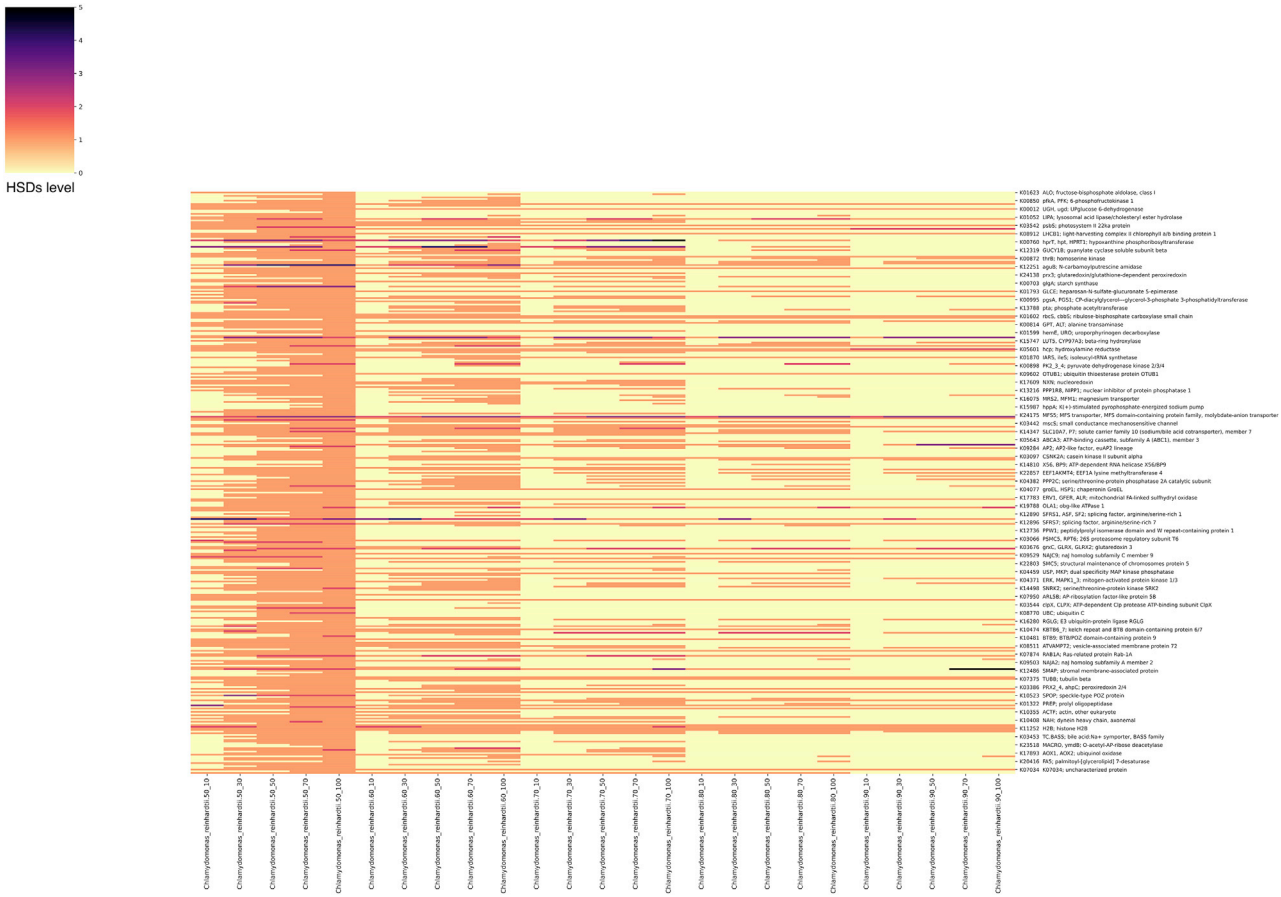


Figure 1. Heatmap illustrating results obtained using different thresholds for the detection of highly similar duplicates (HSDs) in the genome of *C. reinhardtii* using the HSDecipher pipeline

The matrix in the heatmap refers the number of HSDs retrieved using different thresholds (e.g., 90_10, which refers to amino acid pairwise identity ≥90%, and amino acid aligned length variance ≤10) classified based on their KEGG functional categories.

sequence. This is because conserved sequences derived from alternative splicing can have similar functional domains, resulting in the misprediction of gene duplicates. We have developed a custom script called isoform2one (https://github.com/zx0223winner/isoform2one) to carry out this type of filtering before running the BLAST all-against-all search.

Problem 3

Will the original HSD file be modified after running the HSD_batch_run.py script? (step 3).

Potential solution

Yes, since the *HSD_batch_run.py* script can automatically run the *HSD_add_on.py* script multiple times based on a series of combination thresholds, the original files in HSDs folder will be modified. Users should back up a copy of the HSD files before running the *HSD_batch_run.py* script.

Problem 4

What criteria were used to collect the HSDs in *HSD_batch_run.py* script? (step 3).

Potential solution

Although there is no easy rule for distinguishing partial duplicates from complete duplicates, candidate HSDs generally have less than 50% amino acid length difference and similar predicted functions

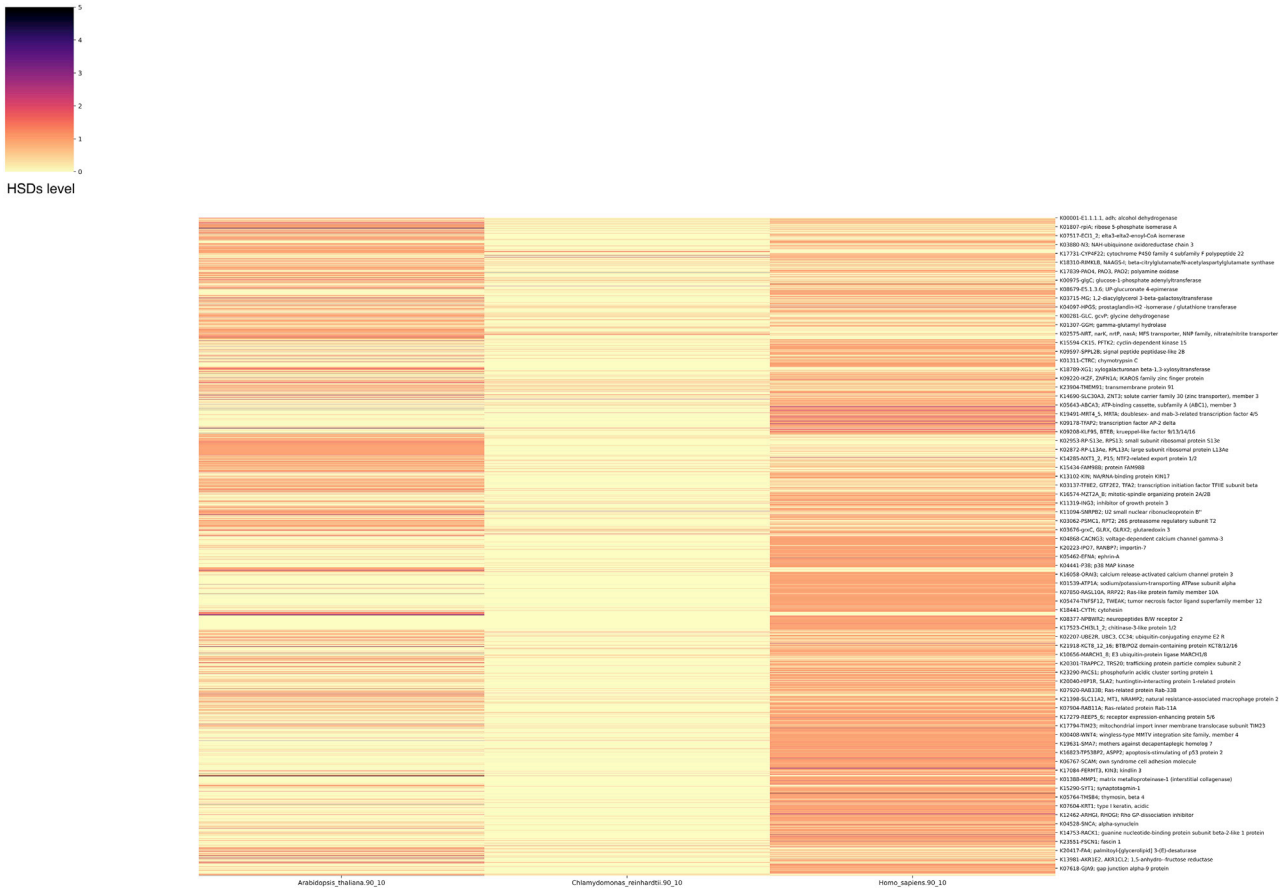


Figure 2. Heatmap showing results of running the HSDecipher pipeline on the predicted proteomes of *C. reinhardtii*, *A. thaliana* and *H. sapiens*

The matrix in the heatmap refers the number of HSDs across three eukaryotic species classified by their KEGG functional categories.

of conserved domains. To balance HSD detection sensitivity and accuracy, we suggest using a series of thresholds from 90%_10aa to 90%_100aa and from 50%_10aa to 50%_100aa. The combination threshold is selected using a series of thresholds: E + (D + (C + (B + A))).

A = 90%_100aa+(90%_70aa+(90%_50aa+(90%_30aa+90%_10aa))).

B = 80%_100aa+(80%_70aa+(80%_50aa+(80%_30aa+80%_10aa))).

C = 70%_100aa+(70%_70aa+(70%_50aa+(70%_30aa+70%_10aa))).

D = 60%_100aa+(60%_70aa+(60%_50aa+(60%_30aa+60%_10aa))).

E = 50%_100aa+(50%_70aa+(50%_50aa+(50%_30aa+50%_10aa))).

Problem 5

How can I acquire the KEGG pathway KO information for each genome? (step 4).

Potential solution

The detailed KO accession with each gene model identifier can be retrieved from the KEGG database. Our previous protocol provides a step-by-step guide.[8]

Table 3. Example of HSDecipher heatmap tabular file based on the output file from HSDFinder

ko_id	category1	category2	Function	Chlamydomonas_reinhardtii.90_10_hsds_id	Chlamydomonas_reinhardtii.90_10_hsds_genes	Chlamydomonas_reinhardtii.90_10_hsds_num	Arabidopsis_thaliana.90_10_hsds_id	Arabidopsis_thaliana.90_10_hsds_genes	Arabidopsis_thaliana.90_10_hsds_num	Homo_sapiens.90_10_hsds_id	Homo_sapiens.90_10_hsds_genes	Homo_sapiens.90_10_hsds_num
K00850	09101 Carbohydrate metabolism	00010 Glycolysis / Gluconeogenesis [PATH:ko00010]	pfkA, PFK; 6-phospho fructokinase 1	XP_001694148.1	XP_001694148.1; XP_001696305.1	1	NP_194651.1	NP_194651.1; NP_567742.1; NP_568842.1; NP_195010.1; NP_200966.2; NP_199592.1; NP_850025.1	1	NP_001341664.1	NP_001341664.1; XP_005252522.1; XP_016883857.1	1
K01623	09101 Carbohydrate metabolism	00010 Glycolysis / Gluconeogenesis [PATH:ko00010]	ALO; fructose-bisphosphate aldolase, class I	XP_001700318.1	XP_001700318.1; XP_001700659.1; XP_001701797.1	1	NP_178224.1	NP_178224.1; NP_568049.1; NP_565508.1; NP_001328708.1; NP_181187.1; NP_190861.1; NP_568127.1; NP_001329763.1	1	NP_000026.2	NP_000026.2; NP_005156.1; NP_001230106.1	1
K01006	09101 Carbohydrate metabolism	00010 Glycolysis / Gluconeogenesis [PATH:ko00010]	ppdK; pyruvate, orthophosphate dikinase	XP_042914963.1	XP_042914963.1; XP_042919927.1	1	NA	NA	NA	NA	NA	NA
K00627	09101 Carbohydrate metabolism	00010 Glycolysis / Gluconeogenesis [PATH:ko00010]	LAT, aceF, pdhC; pyruvate dehydrogenase E2 component (dihydrolipoamide acetyltransferase)	XP_001696403.1	XP_001696403.1; XP_042920693.1	1	NP_564654.1	NP_564654.1; NP_566470.1; NP_189215.1; NP_174703.1	1	NA	NA	NA
K00121	09101 Carbohydrate metabolism	00010 Glycolysis / Gluconeogenesis [PATH:ko00010]	frmA, AH5, adhC; S-(hydroxymethyl) glutathione dehydrogenase / alcohol dehydrogenase	XP_042919155.1	XP_042919155.1; XP_042919157.1	1	NP_173660.1	NP_173660.1; NP_001031079.1; NP_567645.1; NP_199040.1; NP_568453.1; NP_177837.1; NP_176652.3; NP_564409.1; NP_001190468.1	1	NP_000658.1	NP_000658.1; NP_000659.2; NP_000660.1; NP_001159976.1; NP_001095940.1; NP_000662.3; NP_001293100.1	1
K12957	09101 Carbohydrate metabolism	00010 Glycolysis / Gluconeogenesis [PATH:ko00010]	ahr; alcohol/ geraniol dehydrogenase (NAP+)	XP_001692728.2	XP_001692728.2; XP_042921179.1	1	NA	NA	NA	NA	NA	NA
K01895	09101 Carbohydrate metabolism	00010 Glycolysis / Gluconeogenesis [PATH:ko00010]	ACSS1_2, acs; acetyl-CoA synthetase	XP_001700210.2	XP_001700210.2; XP_001700230.1; XP_001702039.1	1	NA	NA	NA	NA	NA	NA
K00026	09101 Carbohydrate metabolism	00020 itrate cycle (TA cycle) [PATH:ko00020]	MH2; malate dehydrogenase	XP_001693118.1	XP_001693118.1; XP_001703167.2; XP_001702586.1	1	NP_179863.1	NP_179863.1; NP_001119199.1; NP_188120.1; NP_564625.1; NP_190336.1	1	NA	NA	NA
K00012	09101 Carbohydrate metabolism	00040 Pentose and glucuronate interconversions [PATH:ko00040]	UGH, ugd; UPglucose 6-dehydrogenase	XP_001698004.1	XP_001698004.1; XP_001703656.1	1	NP_173979.1	NP_173979.1; NP_189582.1; NP_197053.1; NP_198748.1	1	NA	NA	NA

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to John M. Archibald (john.archibald@dal.ca) and Technical Contact Xi Zhang (xi.zhang@dal.ca).

Materials availability

This study did not generate new unique reagents.

Data and code availability

The HSDecipher source code has been deposited at https://github.com/zx0223winner/HSDecipher. The archived version at Zenodo is https://doi.org/10.5281/zenodo.7437886.

ACKNOWLEDGMENTS

This work was supported by a Gordon and Betty Moore Foundation grant (GBMF5782) and a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada (RGPIN-2019-05058) awarded to J.M.A. This work was also supported by a Discovery Grant (RGPIN 04912) from the Natural Sciences and Engineering Research Council of Canada to Z.C. We thank David R. Smith for useful discussion of the manuscript.

AUTHOR CONTRIBUTIONS

The study was conceptualized by X.Z. The data and manuscript were analyzed and written by X.Z. Y.N.H. assisted with bioinformatics analysis and debugged the HSDecipher pipeline. Z.C. and J.M.A. edited the manuscript. All authors read, revised, and approved the final manuscript for peer review.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

1. Zhang, X., Hu, Y., and Smith, D.R. (2021). HSDFinder: a BLAST-based strategy for identifying highly similar duplicated genes in eukaryotic genomes. Front. Bioinform. *1*, 803176. https://doi.org/10.3389/fbinf.2021.803176.

2. Zhang, X., Hu, Y., and Smith, D.R. (2022). HSDatabase - a database of highly similar duplicate genes from plants, animals, and algae. Database *2022*, baac086. https://doi.org/10.1093/database/baac1086. 10.1101/2022.08.01.502183.

3. Kondrashov, F.A. (2012). Gene duplication as a mechanism of genomic adaptation to a changing environment. Proc. Biol. Sci. *279*, 5048–5057.

4. Zhang, X., and Smith, D.R. (2022). An overview of online resources for intra-species detection of gene duplications. Front. Genet. *13*, 1012788. https://doi.org/10.3389/fgene.2022.1012788.

5. Zhang, X., Cvetkovska, M., Morgan-Kiss, R., Hüner, N.P., and Smith, D.R. (2021). Draft genome sequence of the Antarctic green alga Chlamydomonas sp. UWO241. iScience *24*, 102084.

6. Cvetkovska, M., Szyszka-Mroz, B., Possmayer, M., Pittock, P., Lajoie, G., Smith, D.R., and Hüner, N.P.A. (2018). Characterization of photosynthetic ferredoxin from the Antarctic alga *Chlamydomonas* sp. UWO241 reveals novel features of cold adaptation. New Phytol. *219*, 588–604.

7. Stahl-Rommel, S., Kalra, I., D'Silva, S., Hahn, M.M., Popson, D., Cvetkovska, M., and Morgan-Kiss, R.M. (2022). Cyclic electron flow (CEF) and ascorbate pathway activity provide constitutive photoprotection for the photopsychrophile, Chlamydomonas sp. UWO 241 (renamed Chlamydomonas priscuii). Photosynth. Res. *151*, 235–250.

8. Zhang, X., Hu, Y., and Smith, D.R. (2021). Protocol for HSDFinder: identifying, annotating, categorizing, and visualizing duplicated genes in eukaryotic genomes. STAR Protoc. *2*, 100619.

9. Merchant, S.S., Prochnik, S.E., Vallon, O., Harris, E.H., Karpowicz, S.J., Witman, G.B., Terry, A., Salamov, A., Fritz-Laylin, L.K., Maréchal-Drouard, L., et al. (2007). The *Chlamydomonas* genome reveals the evolution of key animal and plant functions. Science *318*, 245–250.

10. Lamesch, P., Berardini, T.Z., Li, D., Swarbreck, D., Wilks, C., Sasidharan, R., Muller, R., Dreher, K., Alexander, D.L., Garcia-Hernandez, M., et al. (2012). The Arabidopsis Information Resource (TAIR): improved gene annotation and new tools. Nucleic Acids Res. *40*, D1202–D1210.

11. Rhee, S.Y., Beavis, W., Berardini, T.Z., Chen, G., Dixon, D., Doyle, A., Garcia-Hernandez, M., Huala, E., Lander, G., Montoya, M., et al. (2003). The Arabidopsis Information Resource (TAIR): a model organism database providing a centralized, curated gateway to Arabidopsis biology, research materials and community. Nucleic Acids Res. *31*, 224–228.

12. Lander, E.S., Linton, L.M., Birren, B., Nusbaum, C., Zody, M.C., Baldwin, J., Devon, K., Dewar, K., Doyle, M., FitzHugh, W., et al. (2001). Initial sequencing and analysis of the human genome. Nature *409*, 860–921.

13. Venter, J.C., Adams, M.D., Myers, E.W., Li, P.W., Mural, R.J., Sutton, G.G., Smith, H.O., Yandell, M., Evans, C.A., Holt, R.A., et al. (2001). The sequence of the human genome. Science *291*, 1304–1351.

14. Kanehisa, M., and Goto, S. (2000). KEGG: kyoto encyclopedia of genes and genomes. Nucleic Acids Res. *28*, 27–30.

15. Huerta-Cepas, J., Capella-Gutiérrez, S., Pryszcz, L.P., Marcet-Houben, M., and Gabaldón, T. (2014). PhylomeDB v4: zooming into the plurality of evolutionary histories of a genome. Nucleic Acids Res. *42*, D897–D902.

16. Emms, D.M., and Kelly, S. (2015). OrthoFinder: solving fundamental biases in whole genome comparisons dramatically improves orthogroup inference accuracy. Genome Biol. *16*, 157–214.

17. Emms, D.M., and Kelly, S. (2019). OrthoFinder: phylogenetic orthology inference for comparative genomics. Genome Biol. *20*, 238–314.