JAMIA Open, 4(3), 2021, 1–5
doi: 10.1093/jamiaopen/ooab079
Application Notes

Application Notes

PhenClust, a standalone tool for identifying trends within sets of biological phenotypes using semantic similarity and the Unified Medical Language System metathesaurus

Jennifer L. Wilson [1], Mike Wong[2], Nicholas Stepanov[3], Dragutin Petkovic[2,3], and Russ Altman[4,5]

[1]Department of Chemical and Systems Biology, Stanford University, Stanford, California, USA, [2]CoSE Computing for Life Science, San Francisco State University, San Francisco, California, USA, [3]Department of Computer Science, San Francisco State University, San Francisco, California, USA, [4]Department of Bioengineering, Stanford University, Stanford, California, USA, and [5]Department of Genetics, Stanford University, Stanford, California, USA

Corresponding Author: Russ Altman, MD, PhD, Department of Bioengineering, Shriram Center, Stanford University, 443 Via Ortega, Room 209, Altman Lab, MC: 4245, Stanford, CA 94305-4145, USA; rbaltman@stanford.edu

Received 7 May 2021; Revised 12 August 2021; Editorial Decision 17 August 2021; Accepted 2 September 2021

ABSTRACT

Objectives: We sought to cluster biological phenotypes using semantic similarity and create an easy-to-install, stable, and reproducible tool.
Materials and Methods: We generated Phenotype Clustering (PhenClust)—a novel application of semantic similarity for interpreting biological phenotype associations—using the Unified Medical Language System (UMLS) metathesaurus, demonstrated the tool's application, and developed Docker containers with stable installations of two UMLS versions.
Results: PhenClust identified disease clusters for drug network-associated phenotypes and a meta-analysis of drug target candidates. The Dockerized containers eliminated the requirement that the user install the UMLS metathesaurus.
Discussion: Clustering phenotypes summarized all phenotypes associated with a drug network and two drug candidates. Docker containers can support dissemination and reproducibility of tools that are otherwise limited due to insufficient software support.
Conclusion: PhenClust can improve interpretation of high-throughput biological analyses where many phenotypes are associated with a query and the Dockerized PhenClust achieved our objective of decreasing installation complexity.

Key words: systems biology, phenotype analysis, high-throughput analysis, network analysis, computational tools, Docker containers

LAY SUMMARY

We created Phenotype Clustering (PhenClust), a computational tool that clusters biological phenotypes, such as diseases, based on their relatedness and a Docker container to stably release an easy-to-install version of our tool. PhenClust measures disease similarity using Unified Medical Language System (UMLS), however, installing this dependency is tricky and a failed installation prevents usage of PhenClust. Our Docker system is a stable release of PhenClust with applications to multiple biological analyses.

BACKGROUND

Tools that extract biological meaning from computational analyses advance biological research, yet a lack of stable and portable releases of these tools limit their utility and reproducibility. We discovered that network methods identified tens to hundreds of phenotypes significantly associated with a drug's protein interaction network. The protein network generated for Metformin (published in[1]) was associated with "Diabetes mellitus, type 1," "Diabetes mellitus, type 2," "Pleural neoplasms," and "Neoplasm of the rectum" among others. In this case, we could manually identify disease groups (eg, "oncology" or "metabolic disease"). However, some drug networks were associated with hundreds of phenotypes and we preferred a computational method for finding disease groups. We developed the Phenotype Clustering (PhenClust) tool to group diseases and used a novel application of semantic similarity to generate clusters of biological phenotypes. At the time, we were unaware of other tools for completing this task. We further encountered limitations in the tool's dissemination because of difficult-to-install dependencies.

We developed PhenClust around the Unified Medical Language System (UMLS) metathesaurus, which has many advantages. The UMLS metathesaurus contains, to the best of our knowledge, one of the largest biomedical ontologies, making it an attractive resource for understanding relationships between diseases. Additionally, we gathered our disease names from multiple databases and the UMLS meta-map[2] tools provided a robust method for mapping syntactically distinct disease names to a common identifier system (the UMLS system uses the Concept Unique Identifier, or CUI term). Lastly, the umls-interface and umls-similarity[3] tools provided useful methods for interfacing with the UMLS database and calculating similarity between concepts. Calculating similarity between concepts was essential for understanding relationships between diseases and eventually discovering disease clusters, yet successfully installing a robust version of UMLS was difficult and this dependency was sensitive to routine operating system updates.

Interpreting relationships among biological phenotypes has applications beyond protein interaction network analysis. For instance, selecting a gene target for further drug development may require analysis of whether that gene is associated with multiple biological processes or side effects. A scientist may prefer a gene target with a narrow set of phenotypic associations or may also consider a gene's association to both safety and efficacy phenotypes. In a recent search of the PheWAS catalogue[4]—a database that links genetic variants to clinical phenotypes—a search for the tumor necrosis factor alpha gene recovered associations to 279 phenotypes and a search for the proprotein convertase subtilisin/kexin type 9 gene yielded associations to 62 phenotypes. For the case of drug target prioritization, we anticipate that the number of gene–phenotype relationships will increase, and that having tools like PhenClust to summarize those relationships will be of broad utility.

Because PhenClust was viable as a general research tool and because PhenClust required difficult-to-install dependencies, our goal with this application note was to release a stable, easy-to-install version of PhenClust and demonstrate the tool's application to multiple biological analyses. Dockerized systems are increasingly popular for releasing software tools with required dependencies and environment variables through a container object.[5] Docker containers have the advantage of extensive software support, which is not always available for academic software tools. Indeed, installing the Docker software and PhenClust container is drastically less complex than installing PhenClust from GitHub and the UMLS metathesaurus. The GitHub release of PhenClust (via the PathFX repo[1]) required the user to install the UMLS metathesaurus and umls-interface.pl and umls-similarity.pl[3] dependencies. Furthermore, our original release of PhenClust used UMLS 2017AA, and we have updated and included a new version using UMLS 2020AA.

Here, we present an introduction to the PhenClust tool, a novel application of semantic similarity for clustering biological phenotypes, and two case studies to orient new users. This application achieves our objective of releasing a stable, easy-to-install, and reproducible implementation of the PhenClust tool.

SIGNIFICANCE

PhenClust is significant because the tool increased the biological understanding of high-throughput computational analyses. The Docker is valuable because it contains a stable and reproducible implementation of the PhenClust tool and it reduces the complexity of installation, thus making PhenClust broadly accessible for research applications.

METHODS

We developed two Dockerized systems that contain either the 2017AA or 2020AA UMLS metathesaurus releases, umls-interface.pl, umls-similarity.pl, PathFX code, PhenClust, and all required python dependencies. The Docker containers allow users to run PhenClust as part of PathFX analysis or use the code in a standalone fashion. Because access to the UMLS metathesaurus required registration, we leveraged the National Library of Medicine (NLM) UMLS Terminology Service (UTS) authorization API to restrict access to NLM-registered users using the UTS API.

We further engineered the Docker container for relatively easy use and installation. We created a copy of the mysql directory during build time which is then mounted in the correct folder at run time, which decreased the initial database loading time to 5 min (the conventional loading method is >2 h). Installation of the Docker takes about 30 min, but this load time is not incurred after the Docker container is installed. Users have the flexibility to use the Dockerized UMLS metathesaurus image separately or for further development. To ease communicating with Docker, we provided a shell script that includes a help command for available algorithm options and helps deploy and stop containers safely without running into memory leaks. The shell script can detect when the database is

actively loading and will wait for the mysql folder to be mounted at run time before executing commands. These scripts currently install the UMLS-2017AA or UMLS-2020AA versions.

To access the Dockerized container, users can create an account on PathFX-web[6] (https://www.pathfxweb.net/). After logging in, users navigate to the "Download" page. On this page, users are guided to install Docker Desktop. Users are also reminded of the requirement to have an account with the NLM UTS. After establishing an NLM account, users can login, and the container download becomes available (Figure 1).

The PhenClust code requires minimal inputs. If used as part of PathFX analysis, the user can specify PhenClust as an option to the algorithm. With this option, PathFX will pass network results to PhenClust. To use PhenClust as a standalone tool, the user only needs to specify a text file with a CUI term on a new line. To demonstrate use of the tool, we created two examples (contained in *run_phenclust_ex1.py* and *run_stand_alone_phen_clust_ex2.py*, both examples explained in Supplementary Material S1, and also included in the Docker container).

PhenClust consists of two scripts: first, there is a python wrapper (calc_lin_matrix_umls_SO.py) to umls-similarity.pl that takes an input file, calls the UMLS tools, and generates a matrix file of similarity values. The second script (plot_and_cluster_phenotypes_SO.py) uses the results from UMLS to cluster diseases and generates text and image outputs. For a full description of the scripts and the parameters see Supplementary Material S1. All code and example analyses are included in the Docker container.

RESULTS

The Dockerized versions of PhenClust are stable and portable. Furthermore, the containers are distinct from PathFXweb where PhenClust is not accessible as a standalone tool and from the PathFX GitHub repository that requires the user installs the UMLS metathesaurus (Supplementary Table S1). We measured run times for these implementations and observed 10 min—44 h depending on the number of input CUI terms and the computing environment (Supplementary Table S2). However, changing certain parameters can greatly reduce run times to under 1 h (Supplementary Table S2).

We endeavored to use PhenClust to identify phenotype clusters for diseases associated with a network analysis from PathFX and to understand diseases associated with two candidate gene targets for developing a novel therapy for systematic lupus erythematosus (SLE) using a gene signature published by Arasappan et al.[7] Running PhenClust with each example dataset created two images and a summary table of results (Figure 2, Supplementary Figures S1–S4, and Supplementary Material S1 and S3). The labeled dendrogram repre-

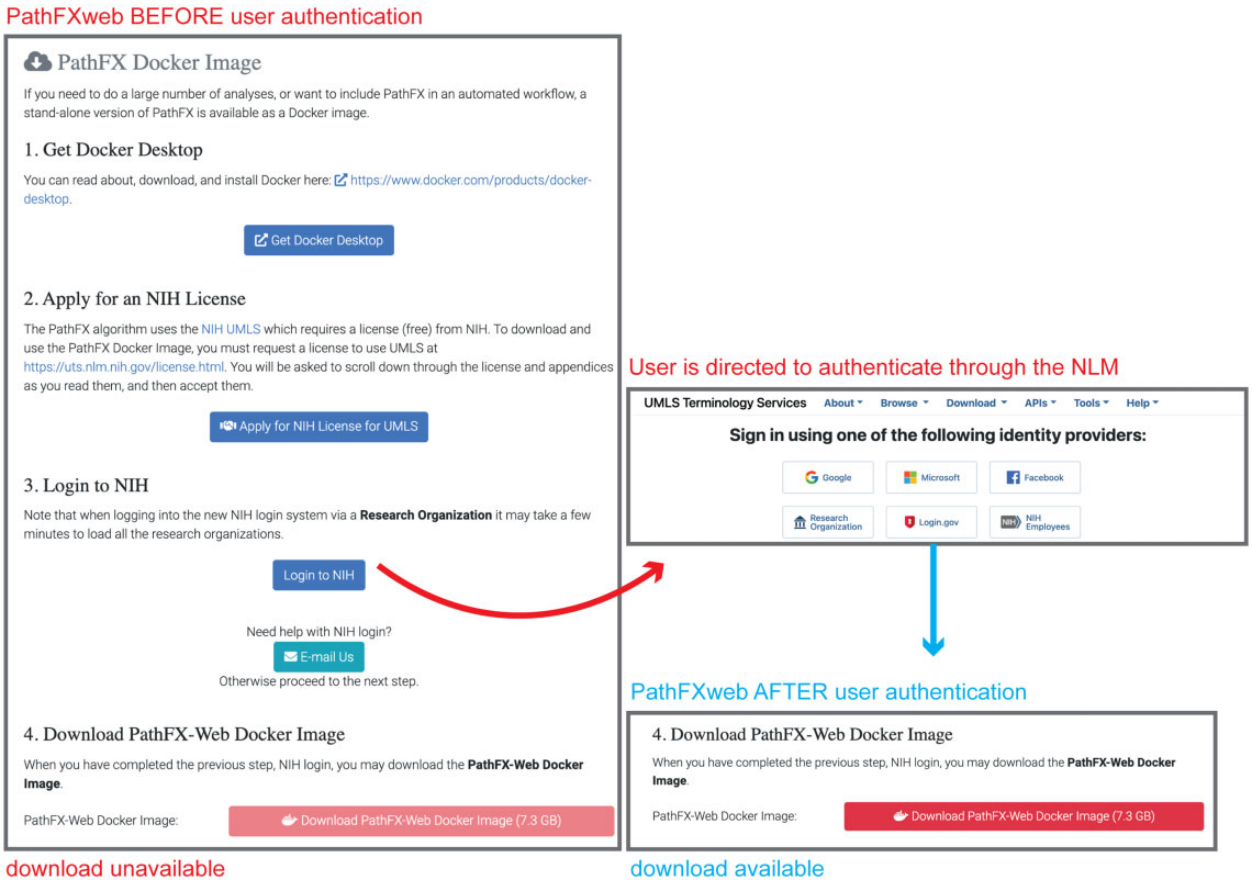

Figure 1. How to download Phenotype Clustering (PhenClust) and PathFX Docker container via PathFXweb. Users are guided to install Docker desktop and reminded of the requirement to have a National Library of Medicine (NLM) account. The Docker container is available for download after user authentication via the Unified Medical Language System terminology service. The user navigates to the download page, selects "Login to NIH," is redirected to the NLM page. On the NLM page the user selects the identity provider of their choice. After the license is verified, the user is redirected to PathFXweb and the Docker container is available for download.

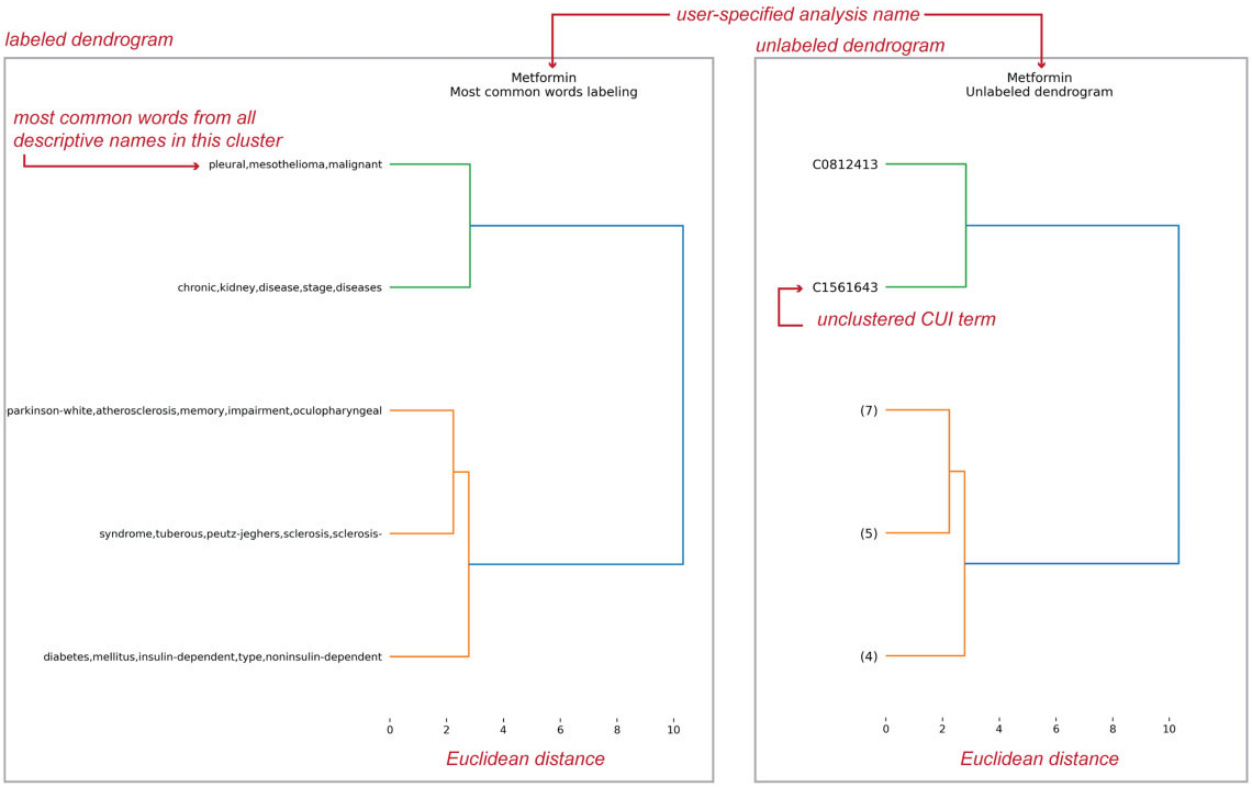

Figure 2. Clustering dendrograms from analysis of metformin-associated Concept Unique Identifier (CUI) terms with and without descriptive labels. Phenotype Clustering selected up to 5 of the most frequent words from disease names in the cluster as a label for the dendrogram (left) or provided the number of terms grouped into a cluster or the CUI identifier for un-clustered CUI terms (right). The *x*-axis represented the between-cluster Euclidean distance as calculated by the linkage function from the fastcluster Python module. The first line of the figure title is an analysis name parameter specified by the user. The branch colors are assigned by default from the scipy.cluster.hierarchy.dendrogram function. The color threshold is set with 0.7*max(linkage_distance) to approximate clusters in the data.

sented a summary of the clustered results and showed similar clusters proximal to each other (Figure 2 and Supplementary Figures S1–S4). Cluster labels represented the top words mentioned in all descriptor names for phenotypes associated with the cluster. PhenClust also generated a tabular output of all phenotypic descriptors in a file that has the extension *cluster_membership_*.txt* (Supplementary Table S3 and Supplementary Material S3). For a full description of the results files and example use cases see Supplementary Material S1. We used PhenClust to compare the genes, interleukin type 1 receptor 2 (IL1R2), and Fc fragment of IgG Receptor IIb (FCGR2B), as candidate druggable targets for SLE, an auto-immune disease. FCGR2B had associations to more auto-immune diseases and fewer potential side effects than IL1R2. PhenClust summary analysis could inform how gene targets are scrutinized before further development.

DISCUSSION

Interpretability (eg, visual and textual interpretation) is a major challenge for system-scale biological analyses and interpretability tools can drastically improve these efforts. In our own efforts, we discovered that protein networks around a drug's targets were enriched for several, and sometime hundreds, of disease phenotypes. We were motivated to consolidate these lists into broad disease groupings. Motivated by this finding, we generated the PhenClust tool that produced visual and textual interpretation of the input data—a list of phenotypes—and afforded the user a summarized version of this data. We further applied PhenClust to understand the uniqueness of disease-associated genes and discovered that PhenClust provided novel insights for considering these genes as druggable targets. In the latter example, we considered gene targets ranked for their association to SLE through a meta-analysis of microarray studies.[7] Meta-analyses are one technique used in prioritizing potential drug targets. However, a gene's association to other distinct or related diseases, or side effects may preclude a target from further development. We envisioned that PhenClust could be valuable to understanding a gene's association to multiple phenotypes aggregated from multiple databases.

Docker containers, a well-developed and documented commercial software platform, presented a valuable opportunity for enhancing dissemination of software tools and increasing reproducibility of computational studies. Even the best intended software releases can contain assumptions or hidden dependencies that limit successful installation.[8,9] Furthermore, although research publications make data and code available, it may not be sufficient for full replication of the analysis.[5] And despite the idea that computational code could be more portable, it remains difficult to completely replicate computational analyses.[5,9] A recent investigation of academic software tools discovered that many were "difficult" to install and this analysis further supported the idea that ease of use is essential for tool dissemination.[9] The dependency of PhenClust on a working installation of UMLS motivated us to pursue an alternative solution. The Docker platform was ideal because it allowed us to create a portable computational environment

that improved reproducibility of our analyses and enabled easier dissemination of the tool.

The UTS is one of the largest biomedical ontologies and the UMLS platform is amendable to the development and deployment of tools. The UMLS metathesaurus contains extensive resources, such as meta-map,[10] for mapping plain text terms from several other biomedical ontologies and sources to UMLS CUI identifiers. Furthermore, tools such as umls-interface.pl and umls-similarity.pl have enhanced access to the UMLS ontology.[3] Our project directly benefited from the UTS resources and allowed us to actively engage with this valuable resource. The Docker system removed hurdles to installing the metathesaurus and thus better enabled scientists to focus on the interpretation and use of our tool, as well as the UMLS metathesaurus. Furthermore, the platform was amenable to releasing a stable version of the original PhenClust tool (using UMLS version 2017AA) to support reproducibility of research and was amenable to data updates (using UMLS version 2020AA) to support evolving data sources.

CONCLUSIONS

We produced a standalone version of PhenClust, a tool that is valuable for grouping biological phenotypes based on semantic similarity. The tool is also broadly applicable to experiments where phenotypes can be mapped to UMLS identifiers. We extended the applications of the UMLS metathesaurus and anticipate that the tool will bring further users to the UMLS community. Lastly, packaging PhenClust with the UMLS metathesaurus in a Docker contain increased access to our tool by making it simpler to disseminate, and install and manage UMLS dependencies.

FUNDING

This work was supported by US Food and Drug Administration grant number U01FD004979, SPARK at Stanford, and by a Sanofi iDEA Award.

AUTHOR CONTRIBUTIONS

JLW conceived of the idea, conducted analysis, wrote and revised the manuscript. MW conducted analysis, wrote and revised the manuscript. NS conducted analysis. DP contributed resources, wrote and revised the manuscript. RA contributed resources and revised the manuscript.

SUPPLEMENTARY MATERIAL

Supplementary material is available at *JAMIA Open* online.

CONFLICT OF INTEREST STATEMENT

None declared.

DATA AVAILABILITY STATEMENT

The Docker container with data and code from this manuscript is available for download via http://pathfxweb.net/ for authenticated NLM users. Data and supplemental materials relevant to the manuscript are available in https://github.com/jenwilson521/PhenClust_Supplemental.

REFERENCES

1. Wilson JL, Racz R, Liu T, *et al.* PathFX provides mechanistic insights into drug efficacy and safety for regulatory review and therapeutic development. *PLoS Comput Biol* 2018; 14 (12): e1006614.
2. Aronson AR, Lang F-M. An overview of MetaMap: historical perspective and recent advances. *J Am Med Inform Assoc* 2010; 17 (3): 229–36.
3. McInnes BT, Pedersen T, Pakhomov SVS. UMLS-Interface and UMLS-Similarity: open source software for measuring paths and semantic similarity. *AMIA Annu Symp Proc* 2009; 2009: 431–5.
4. Denny JC, Ritchie MD, Basford MA, *et al.* PheWAS: demonstrating the feasibility of a phenome-wide scan to discover gene-disease associations. *Bioinformatics* 2010; 26 (9): 1205–10.
5. Boettiger C. An introduction to Docker for reproducible research. *Sigops Oper Syst Rev* 2015; 49 (1): 71–9.
6. Wilson JL, Wong M, Chalke A, *et al.* PathFXweb: a web application for identifying drug safety and efficacy phenotypes. *Bioinformatics* 2019; 35 (21): 4504–6.
7. Arasappan D, Tong W, Mummaneni P, *et al.* Meta-analysis of microarray data using a pathway-based approach identifies a 37-gene expression signature for systemic lupus erythematosus in human peripheral blood mononuclear cells. *BMC Med* 2011; 9: 65.
8. Cito J, Ferme V, Gall HC. Using Docker containers to improve reproducibility in software and web engineering research. In: International Conference on Web Engineering; 2016: 609–12.
9. Mangul S, Martin LS, Eskin E, Blekhman R. Improving the usability and archival stability of bioinformatics software. *Genome Biol* 2019; 20 (1): 47.
10. Demner-Fushman D, Rogers WJ, Aronson AR. MetaMap Lite: an evaluation of a new Java implementation of MetaMap. *J Am Med Inform Assoc* 2017; 24 (4): 841–4.