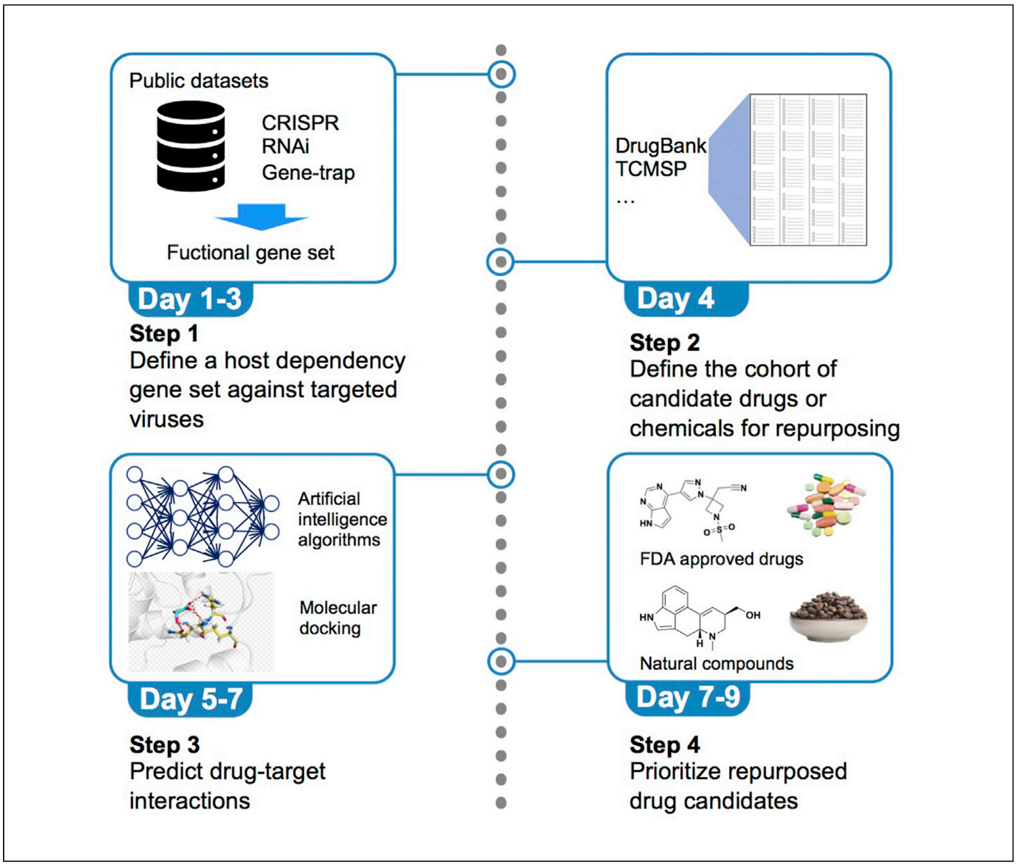
STAR Protocols

CellPress
OPEN ACCESS

Protocol

An *in silico* drug repositioning workflow for host-based antivirals


Public datasets
CRISPR
RNAi
Gene-trap
Fuctional gene set
Day 1-3
Step 1
Define a host dependency gene set against targeted viruses
DrugBank
TCMSP
...
Day 4
Step 2
Define the cohort of candidate drugs or chemicals for repurposing
Artificial intelligence algorithms
Molecular docking
Day 5-7
Step 3
Predict drug-target interactions
FDA approved drugs
Natural compounds
Day 7-9
Step 4
Prioritize repurposed drug candidates

Drug repositioning represents a cost- and time-efficient strategy for drug development. Artificial intelligence-based algorithms have been applied in drug repositioning by predicting drug-target interactions in an efficient and high throughput manner. Here, we present a workflow of *in silico* drug repositioning for host-based antivirals using specially defined targets, a refined list of drug candidates, and an easily implemented computational framework. The workflow described here can also apply to more general purposes, especially when given a user-defined druggable target gene set.

Zexu Li, Yingjia Yao, Xiaolong Cheng, Wei Li, Teng Fei

zeki2019@163.com (Z.L.)
feiteng@mail.neu.edu.cn (T.F.)

Highlights

A step-by-step protocol for host-based antiviral drug repositioning

Drug-target interaction predicted by artificial intelligence-based algorithms

Protocol applicable to other scenarios given a druggable target gene set

Li et al., STAR Protocols *2*, 100653
September 17, 2021 © 2021 The Author(s).
https://doi.org/10.1016/j.xpro.2021.100653

STAR Protocols

CellPress
OPEN ACCESS

Protocol

An *in silico* drug repositioning workflow for host-based antivirals

Zexu Li,[1,2,5,6,*] Yingjia Yao,[1,2,5] Xiaolong Cheng,[3,4] Wei Li,[3,4] and Teng Fei[1,2,7,*]

[1]College of Life and Health Sciences, Northeastern University, Shenyang 110819, People's Republic of China

[2]Key Laboratory of Data Analytics and Optimization for Smart Industry (Northeastern University), Ministry of Education, Shenyang 110819, People's Republic of China

[3]Center for Genetic Medicine Research, Children's National Hospital, 111 Michigan Ave NW, Washington, DC 20010, USA

[4]Department of Genomics and Precision Medicine, George Washington University, 111 Michigan Ave NW, Washington, DC 20010, USA

[5]These authors contributed equally

[6]Technical contact

[7]Lead contact

*Correspondence: zeki2019@163.com (Z.L.), feiteng@mail.neu.edu.cn (T.F.)

https://doi.org/10.1016/j.xpro.2021.100653

SUMMARY

Drug repositioning represents a cost- and time-efficient strategy for drug development. Artificial intelligence-based algorithms have been applied in drug repositioning by predicting drug-target interactions in an efficient and high throughput manner. Here, we present a workflow of *in silico* drug repositioning for host-based antivirals using specially defined targets, a refined list of drug candidates, and an easily implemented computational framework. The workflow described here can also apply to more general purposes, especially when given a user-defined druggable target gene set.

For complete details on the use and execution of this protocol, please refer to Li et al. (2021).

BEFORE YOU BEGIN

Overview

Artificial intelligence-based algorithms have been applied in drug repositioning as well as other relevant fields (Hao et al., 2016; Pushpakom et al., 2019; Tanoli et al., 2021; Wang et al., 2020; Yang et al., 2020; Zhou et al., 2020). This protocol below describes the specific steps of *in silico* drug repositioning for antivirals against *Coronaviridae* viral families including SARS-CoV-2 (severe acute respiratory syndrome coronavirus 2), SARS-CoV (severe acute respiratory syndrome coronavirus) and MERS-CoV (Middle East respiratory syndrome coronavirus) using *Coronaviridae*-specific host dependency gene set, refined drug candidate list covering 2457 marketed drugs and 1062 natural compounds, and DeepCPI algorithm for drug-target interaction (DTI) prediction. Moreover, this workflow can be extended for broader drug repositioning purposes, given a user-defined target gene set, a custom list of candidate drug chemicals and implementation of more DTI prediction algorithms.

For specific drug repurposing against *Coronaviridae* family viruses, we should firstly define the proper gene set for candidate drugs to target. In addition to limited number of virus-specific genes, host dependency genes (HDGs) with functional implications whose loss-of-function renders host resistance to specific viral infection may serve as an ideal target gene pool for inhibitory drugs to exert antiviral effect. Public datasets derived from functional genetic screens using techniques such as gene-trap, RNA interference (RNAi) and clustered regularly interspaced palindromic repeats

(CRISPR) have provided a wealth of resource about virus-specific HDGs. We have collected *Coronaviridae*-specific HDGs in our previous study (Li et al., 2021) and use them as target gene set in this protocol. HDGs for a broader range of RNA viruses can also be found in Li et al., 2021. For the interrogated drug candidates, we build a chemical cohort by collecting 2457 Food and Drug Administration (FDA) approved drugs (Database: DrugBank, version 5.1.7, released 2020-07-02; https://www.drugbank.ca) and 1062 selected natural compounds embedded in herbs of traditional Chinese medicine with favorable druggability (Li et al., 2021). This refined drug candidate list does not include experimental and investigational chemicals. Since FDA approved drugs and herbs of traditional Chinese medicine have already been applied in humans, this refined cohort may represent the safest drug candidates to be readily tested for clinical trials. Precise and efficient DTI prediction stands in a central position for successful drug repositioning. Multiple artificial intelligence-based algorithms have been developed to predict DTI between multiple drugs and targets. In this protocol, we employ DeepCPI, a computational framework using feature-embedding and deep learning, for DTI prediction (Wan et al., 2019). Compared to other pipelines, DeepCPI is quite computationally efficient which can be run even by a personal computer while maintaining decent predicting power (For example, in the current protocol, DeepCPI can be run on the MacBook Pro with 8 GB of memory to predict 405,405 hypothetical DTI pairs in about 1 h). Each drug-target pair is scored by DeepCPI for their potential interaction, and repurposed drug candidates are then prioritized according to their targeting range (the number of predicted targets) and strength for interrogated targets (targeting potential reflected by DTI score). For the top ranked drug candidates, molecular docking analysis is performed to take a closer examination for the binding interface and free energy of potential drug-target interaction. The workflow generates a ranked list of potential repurposed drug candidates against *Coronaviridae* viruses that are ready for in-depth experimental and clinical evaluation.

Software setup and installation

Timing: ~1 day

A personal computer with Linux- or Unix-based operating system is required to execute this protocol. The prerequisite software (in key resources table) can be downloaded from the corresponding websites. The accompanying user manuals provide detailed information about their functions and uses.

1. Set up the operating environment for DeepCPI.
   a. Requirement: Python2.7, Keras=1.2.2, Gensim=0.10.2, Tensorflow=1.2.0, RDKit.
   b. The source code of DeepCPI can be downloaded from https://github.com/FangpingWan/DeepCPI.
   c. We also recommend the user to install conda (environment management system) (https://docs.conda.io/projects/conda/en/latest/user-guide/install/index.html).
2. Install DeepCPI using command line under Unix or Linux system.
   a. Open Terminal
   b. conda create -n DeepCPI python=2.7 (#create a Python 2.7 environment)
   c. source activate DeepCPI (#activate virtual environment)
   d. conda install RDKit
   e. conda install Keras=1.2.2
   f. conda install Gensim=0.10.2
   g. cd [The path of DeepCPI] (e.g., "cd /.../DeepCPI-master". #Change directory to the home directory of the DeepCPI folder named "DeepCPI-master")
   h. python DeepCPI.py (#Run test data)
   i. For advanced help, please see page on the GitHub (https://github.com/FangpingWan/DeepCPI).
3. Download and install software for molecular docking analysis.

a. Download and install AutoDock software (http://autodock.scripps.edu; version 4.2.6) (Morris et al., 2009).
b. Download and install MGLTools software (http://mgltools.scripps.edu/downloads; version 1.5.6).
c. Download and install PyMOL software (https://pymol.org/2/, version 2.3.2, open-source project).

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Target gene set of *Coronaviridae*-specific HDGs with amino acid sequence information	Supplemental File S1	Coronaviridae_HDGs.txt
Approved drug list with InChI information	Supplemental File S1	Drugbank_Approved.txt
Selected natural compound list with InChI information	Supplemental File S1	TCM_selected.txt
Drug candidate cohort information	Table S1	Drug_cohort_information.xlsx
Predicted DTI and ranked list of repositioned drugs against *Coronaviridae* viruses	Table S2	DTI_and_ranked_drug_list.xlsx
DrugTargtPairGenerator.py	This study	https://github.com/zexuneu/computational-framework-of-host-based-drug-repositioning
MatricesGenerator.py	This study	https://github.com/zexuneu/computational-framework-of-host-based-drug-repositioning
FilterOutNonSignificant.py	This study	https://github.com/zexuneu/computational-framework-of-host-based-drug-repositioning
ZscoreNormalization.Rmd	This study	https://github.com/zexuneu/computational-framework-of-host-based-drug-repositioning
Software and algorithms		
DeepCPI	(Wan et al., 2019)	https://github.com/FangpingWan/DeepCPI
AutoDock	(Morris et al., 2009)	http://autodock.scripps.edu
MGLTools	MGLTools Website	http://mgltools.scripps.edu/downloads
PyMOL	Schrödinger	https://pymol.org/2/

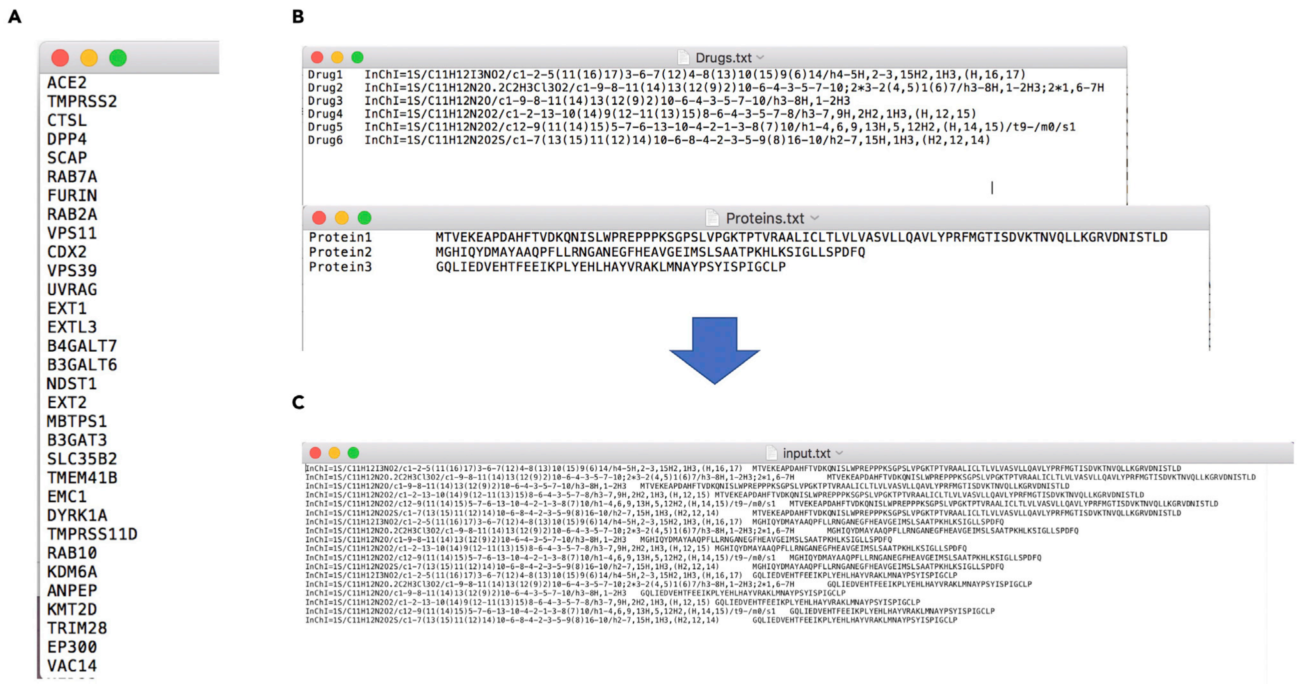
STEP-BY-STEP METHOD DETAILS

Define the druggable target gene set

Timing: ~3 days

Any user-defined target gene set can be used for this protocol towards more general applications. As a specific example, the definition of target gene set against *Coronaviridae* viruses is shown in the following steps.

1. Collect public datasets to define *Coronaviridae*-specific HDGs.
   a. Collect the references performing high throughput genetic perturbation screening for *Coronaviridae* virus resistance in human cells. In these studies, gene-trap, RNAi and CRISPR techniques are employed to perturb a gene's function. For example, use the search key word "SARS-CoV-2 AND screen" to collect SARS-CoV-2 virus-related screening references from PubMed (https://pubmed.ncbi.nlm.nih.gov/). References for other *Coronaviridae* viruses such as MERS-CoV and SARS-CoV can be collected similarly. Pinpoint the datasets reporting the viral resistance HDGs associated with these references.
   b. Collect scattered HDGs for *Coronaviridae* viruses from individual literatures in which specific genes are shown to be critical or essential for complete viral life cycle (non-screen study).



```
ACE2
TMPRSS2
CTSL
DPP4
SCAP
RAB7A
FURIN
RAB2A
VPS11
CDX2
VPS39
UVRAG
EXT1
EXTL3
B4GALT7
B3GALT6
NDST1
EXT2
MBTPS1
B3GAT3
SLC35B2
TMEM41B
EMC1
DYRK1A
TMPRSS11D
RAB10
KDM6A
ANPEP
KMT2D
TRIM28
EP300
VAC14
```

```
Drug1   InChI=1S/C11H12I3NO2/c1-2-5(11(16)17)3-6-7(12)4-8(13)10(15)9(6)14/h4-5H,2-3,15H2,1H3,(H,16,17)
Drug2   InChI=1S/C11H12N2O.2C2H3Cl3O2/c1-9-8-11(14)13(12(9)2)10-6-4-3-5-7-10;2*3-2(4,5)1(6)7/h3-8H,1-2H3;2*1,6-7H
Drug3   InChI=1S/C11H12N2O/c1-9-8-11(14)13(12(9)2)10-6-4-3-5-7-10/h3-8H,1-2H3
Drug4   InChI=1S/C11H12N2O2/c1-2-13-10(14)9(12-11(13)15)8-6-4-3-5-7-8/h3-7,9H,2H2,1H3,(H,12,15)
Drug5   InChI=1S/C11H12N2O2/c12-9(11(14)15)5-7-6-13-10-4-2-1-3-8(7)10/h1-4,6,9,13H,5,12H2,(H,14,15)/t9-/m0/s1
Drug6   InChI=1S/C11H12N2O2S/c1-7(13(15)11(12)14)10-6-8-4-2-3-5-9(8)16-10/h2-7,15H,1H3,(H2,12,14)
```

```
Protein1        MTVEKEAPDAHFTVDKQNISLWPREPPPKSGPSLVPGKTPTVRAALICLTLVLVASVLLQAVLYPRFMGTISDVKTNVQLLKGRVDNISTLD
Protein2        MGHIQYDMAYAAQPFLLRNGANEGFHEAVGEIMSLSAATPKHLKSIGLLSPDFQ
Protein3        GQLIEDVEHTFEEIKPLYEHLHAYVRAKLMNAYPSYISPIGCLP
```

```
InChI=1S/C11H12I3NO2/c1-2-5(11(16)17)3-6-7(12)4-8(13)10(15)9(6)14/h4-5H,2-3,15H2,1H3,(H,16,17)  MTVEKEAPDAHFTVDKQNISLWPREPPPKSGPSLVPGKTPTVRAALICLTLVLVASVLLQAVLYPRFMGTISDVKTNVQLLKGRVDNISTLD
InChI=1S/C11H12N2O.2C2H3Cl3O2/c1-9-8-11(14)13(12(9)2)10-6-4-3-5-7-10;2*3-2(4,5)1(6)7/h3-8H,1-2H3;2*1,6-7H        MTVEKEAPDAHFTVDKQNISLWPREPPPKSGPSLVPGKTPTVRAALICLTLVLVASVLLQAVLYPRFMGTISDVKTNVQLLKGRVDNISTLD
InChI=1S/C11H12N2O/c1-9-8-11(14)13(12(9)2)10-6-4-3-5-7-10/h3-8H,1-2H3   MTVEKEAPDAHFTVDKQNISLWPREPPPKSGPSLVPGKTPTVRAALICLTLVLVASVLLQAVLYPRFMGTISDVKTNVQLLKGRVDNISTLD
InChI=1S/C11H12N2O2/c1-2-13-10(14)9(12-11(13)15)8-6-4-3-5-7-8/h3-7,9H,2H2,1H3,(H,12,15) MTVEKEAPDAHFTVDKQNISLWPREPPPKSGPSLVPGKTPTVRAALICLTLVLVASVLLQAVLYPRFMGTISDVKTNVQLLKGRVDNISTLD
InChI=1S/C11H12N2O2/c12-9(11(14)15)5-7-6-13-10-4-2-1-3-8(7)10/h1-4,6,9,13H,5,12H2,(H,14,15)/t9-/m0/s1   MTVEKEAPDAHFTVDKQNISLWPREPPPKSGPSLVPGKTPTVRAALICLTLVLVASVLLQAVLYPRFMGTISDVKTNVQLLKGRVDNISTLD
InChI=1S/C11H12N2O2S/c1-7(13(15)11(12)14)10-6-8-4-2-3-5-9(8)16-10/h2-7,15H,1H3,(H2,12,14)        MTVEKEAPDAHFTVDKQNISLWPREPPPKSGPSLVPGKTPTVRAALICLTLVLVASVLLQAVLYPRFMGTISDVKTNVQLLKGRVDNISTLD
InChI=1S/C11H12I3NO2/c1-2-5(11(16)17)3-6-7(12)4-8(13)10(15)9(6)14/h4-5H,2-3,15H2,1H3,(H,16,17)  MGHIQYDMAYAAQPFLLRNGANEGFHEAVGEIMSLSAATPKHLKSIGLLSPDFQ
InChI=1S/C11H12N2O.2C2H3Cl3O2/c1-9-8-11(14)13(12(9)2)10-6-4-3-5-7-10;2*3-2(4,5)1(6)7/h3-8H,1-2H3;2*1,6-7H        MGHIQYDMAYAAQPFLLRNGANEGFHEAVGEIMSLSAATPKHLKSIGLLSPDFQ
InChI=1S/C11H12N2O/c1-9-8-11(14)13(12(9)2)10-6-4-3-5-7-10/h3-8H,1-2H3   MGHIQYDMAYAAQPFLLRNGANEGFHEAVGEIMSLSAATPKHLKSIGLLSPDFQ
InChI=1S/C11H12N2O2/c1-2-13-10(14)9(12-11(13)15)8-6-4-3-5-7-8/h3-7,9H,2H2,1H3,(H,12,15) MGHIQYDMAYAAQPFLLRNGANEGFHEAVGEIMSLSAATPKHLKSIGLLSPDFQ
InChI=1S/C11H12N2O2/c12-9(11(14)15)5-7-6-13-10-4-2-1-3-8(7)10/h1-4,6,9,13H,5,12H2,(H,14,15)/t9-/m0/s1   MGHIQYDMAYAAQPFLLRNGANEGFHEAVGEIMSLSAATPKHLKSIGLLSPDFQ
InChI=1S/C11H12N2O2S/c1-7(13(15)11(12)14)10-6-8-4-2-3-5-9(8)16-10/h2-7,15H,1H3,(H2,12,14)        MGHIQYDMAYAAQPFLLRNGANEGFHEAVGEIMSLSAATPKHLKSIGLLSPDFQ
InChI=1S/C11H12I3NO2/c1-2-5(11(16)17)3-6-7(12)4-8(13)10(15)9(6)14/h4-5H,2-3,15H2,1H3,(H,16,17)  GQLIEDVEHTFEEIKPLYEHLHAYVRAKLMNAYPSYISPIGCLP
InChI=1S/C11H12N2O.2C2H3Cl3O2/c1-9-8-11(14)13(12(9)2)10-6-4-3-5-7-10;2*3-2(4,5)1(6)7/h3-8H,1-2H3;2*1,6-7H        GQLIEDVEHTFEEIKPLYEHLHAYVRAKLMNAYPSYISPIGCLP
InChI=1S/C11H12N2O/c1-9-8-11(14)13(12(9)2)10-6-4-3-5-7-10/h3-8H,1-2H3   GQLIEDVEHTFEEIKPLYEHLHAYVRAKLMNAYPSYISPIGCLP
InChI=1S/C11H12N2O2/c1-2-13-10(14)9(12-11(13)15)8-6-4-3-5-7-8/h3-7,9H,2H2,1H3,(H,12,15) GQLIEDVEHTFEEIKPLYEHLHAYVRAKLMNAYPSYISPIGCLP
InChI=1S/C11H12N2O2/c12-9(11(14)15)5-7-6-13-10-4-2-1-3-8(7)10/h1-4,6,9,13H,5,12H2,(H,14,15)/t9-/m0/s1   GQLIEDVEHTFEEIKPLYEHLHAYVRAKLMNAYPSYISPIGCLP
InChI=1S/C11H12N2O2S/c1-7(13(15)11(12)14)10-6-8-4-2-3-5-9(8)16-10/h2-7,15H,1H3,(H2,12,14)        GQLIEDVEHTFEEIKPLYEHLHAYVRAKLMNAYPSYISPIGCLP
```

Figure 1. Prepare DeepCPI input file

(A) Gene symbol list of target gene set exemplified by 165 high confidence HDGs for *Coronaviridae* viruses.

(B) The structure, layout and information of the text files for drugs and targets.

(C) The structure, layout and information of the merged text file generated as DeepCPI input.

2. Filter the collected data to pinpoint HDGs.

   If a host gene or its encoding protein is shown only to physically interact with viral proteins or regulated by viral genes but without functional implication on viral life cycle upon gene's loss-of-function, the gene is not classified as a HDG.

   A gene is defined as a HDG only when it meets any of the following criteria:

   a. Its loss-of-function impedes or reduces viral infection or activity by experimental evidence in non-screen studies.
   b. It has been clearly classified into HDG group in screen studies.
   c. When HDG group is not specified in screen studies, arbitrarily take the top ~5% of all the interrogated genes in the positive selection list as HDGs with a custom log-fold change cutoff in CRISPR knockout or RNAi screens. For example, in a typical result output generated by MAGeCK (Li et al., 2015; Li et al., 2014) analytic pipeline for CRISPR screens, genes can be ranked according to their negative or positive selection trend by jointly considering the log-fold change and statistical significance of their corresponding guide RNAs. HDGs can be arbitrarily defined as the top ~5% of all the genes with a log-fold change of 1.0 (loose cutoff) or 2.0 (stringent cutoff).

3. Define high confidence HDG gene set for *Coronaviridae* family viruses.

As there are several independent studies and datasets for HDG identification against *Coronaviridae* family viruses, we only take a subset of HDGs that occurs more than once among different datasets as high confidence HDGs for further analysis. A total of 165 high confidence HDGs are defined for *Coronaviridae* viruses (Figure 1A). After that, prepare a HDG file in the structure of "gene symbol + amino acid sequence" (Figure 1B, Supplemental File S1).

Note: In addition to PubMed, public integrated database such as "CRISP-view" (http://crispview.weililab.org/) can also be used to search high throughput genetic screen studies or datasets (Cui et al., 2021). In addition, virus-specific HDGs for 10 families and 29 species of RNA viruses can be downloaded from (Li et al., 2021).

Define the cohort of candidate drugs or chemicals for repurposing

Timing: ~1 day

4. Collect FDA approved drug information.
   a. Drug information is extracted from Database: DrugBank (version 5.1.7, released 2020-07-02; https://www.drugbank.ca) (Wishart et al., 2018). Open DrugBank website -> Download -> Structures -> Structure External Links -> Approved -> Download. (#Download FDA approved drug data with InChI (the IUPAC International Chemical Identifier) information from the DrugBank website)
   b. Extract the DrugBank ID and InChI, and save them as separate files in the structure of "DrugBank ID + InChI" (Supplemental File S1).
   c. A total of 2457 FDA approved drugs are collected with InChI information. Note that the InChI value is required for DeepCPI.
5. Collect natural compound information.
   a. Natural compound information is downloaded from Database: Traditional Chinese Medicine Systems Pharmacology (TCMSP) (version 2.3, released 2014-05-31; https://tcmspw.com/tcmsp.php) which is a unique systems pharmacology platform of Chinese herbal medicines (Ru et al., 2014).
   b. Filter the pool of 1455 natural compounds for better druggability by requiring each candidate passing the criteria of oral bioavailability (OB) ≥ 30.0%, drug-likeness (DL) ≥ 0.18 and blood-brain barrier (BBB) ≥ -0.30. Finally, 1062 selected natural compounds with InChI information are kept for the downstream DTI analysis.
   c. Extract the compound ID and InChI, and save them as separate files in the structure of "compound ID + InChI" (Supplemental File S1).

Note: The above drug cohort information used in this protocol can be found in Table S1.

Prepare DeepCPI input file

Timing: ~2 h (variable)

DeepCPI requires two layers of information for DTI prediction: "the InChI information of drugs" and "the amino acid sequence of target gene-encoding proteins".

6. Prepare a txt file (e.g., "Drugbank_Approved.txt" or "TCM_selected.txt") containing the InChI information for each drug (Figure 1B, Supplemental File S1).
7. Prepare a txt file (e.g., "Coronaviridae_HDGs.txt") containing the amino acid sequence for each target protein (Figure 1B, Supplemental File S1). The amino acid sequences are extracted from UniProt database (https://www.uniprot.org/).
8. Save the two files ("Coronaviridae_HDGs.txt" and "Drugbank_Approved.txt") under the same directory.
9. Open Terminal.
10. Change directory to where the files ("Coronaviridae_HDGs.txt" and "Drugbank_Approved.txt") are located by typing "cd /your/working/path".
11. Run python script "DrugTargtPairGenerator.py" by typing "python DrugTargtPairGenerator.py –f1 Coronaviridae_HDGs.txt –f2 Drugbank_Approved.txt" to generate a merged txt file (e.g., "Drug_Target_Pair.txt") with each possible drug-target pair (Figure 1C).

DTI prediction by DeepCPI

Timing: ~2h

12. Run the DeepCPI pipeline and calculate the DeepCPI score for drug-target pair.
    a. Paste the merged input file (e.g., "Drug_Target_Pair.txt") into the DeepCPI folder and rename it as "example.tsv". (#DeepCPI uses "example.tsv" as default input file)
    b. Open Terminal.
    c. Activate conda environment by typing "source activate DeepCPI".
    d. Change directory to the home directory of the DeepCPI folder named "DeepCPI-master" by typing "cd [The path of DeepCPI]". (e.g., "cd /.../DeepCPI-master")
    e. Run the DeepCPI pipeline under the DeepCPI folder by typing the command "python DeepCPI.py".
    f. A file named "Prediction_results.tsv" is generated at the end of the run. Each drug-target pair is assigned a DeepCPI score (range 0–1) representing their interaction potential. The higher score indicates higher interaction potential.
    g. Change directory to where the files ("Prediction_results.tsv", "Coronaviridae_HDGs.txt", and "Drugbank_Approved.txt" stored under the same directory) are located by typing "cd /your/working/path".
    h. Run python script "MatricesGenerator.py" by typing "python MatricesGenerator.py –f1 Prediction_results.tsv –f2 Coronaviridae_HDGs.txt –f3 Drugbank_Approved.txt" to create a score matrices T_{CPI} named "Prediction_results.matrix.txt" with DeepCPI score for each drug-target pair (x_{CPI}), where l refers to the length of drug list and k refers to the length of target list:

$$T_{CPI} \in \mathbb{R}^{l \times k}, x_{CPI} \in T_{CPI}$$

    i. Run python script "FilterOutNonSignificant.py" by typing "python FilterOutNonSignificant.py -f Prediction_results.matrix.txt -c 0.892" to filter out the non-significant DTI scores and only keep the confident scores. The output file is "Prediction_results.matrix.filtered.txt". The optimal standardized DeepCPI score threshold (0.892, sensitivity: 37.2%, specificity: 86.8%) is determined by receiver operating characteristics (ROC) analysis with benchmark datasets (Li et al., 2021). This pre-defined threshold may change when different benchmark datasets are used to evaluate DeepCPI performance. Once defined, such threshold is applicable to any DTI analysis using DeepCPI for different target gene sets and drug sets.

$$T_{CPI\_sig} = \begin{cases} x, & if\ x \geq 0.892 \\ 0, & if\ x < 0.892 \end{cases} \quad x \in T_{CPI}$$

Optional: When more DTI prediction algorithms are applied to alleviate the bias of each algorithm and improve the prediction precision, each method generates a prediction score for the same drug-target pair. However, the score distribution pattern is usually different between different methods. To make these DTI scores comparable, a z-score based normalization is recommended as exemplified in the following steps to standardize DeepCPI score. DTI scores derived from other prediction algorithms can be normalized in the similar manner.

    j. Open and run R script "ZscoreNormalization.Rmd" to generate z-score matrices Z_{CPI} named "z_Prediction_results.txt", where, μ is mean value of the original scores and σ is standard deviation of the original scores:

$$z_{CPI} = \frac{x_{CPI} - \mu_{CPI}}{\sigma_{CPI}}, \quad x_{CPI} \in T_{CPI}$$

    k. Open Terminal.

l. Change directory to where the files (“z_Prediction_results.txt”, “Coronaviridae_HDGs.txt”, and “Drugbank_Approved.txt” stored under the same directory) are located by typing “cd /your/working/path”.

m. Run python script “MatricesGenerator.py” by typing “python MatricesGenerator.py –f1 z_Prediction_results.txt –f2 Coronaviridae_HDGs.txt –f3 Drugbank_Approved.txt”. This command will create a z-score matrices Z_{CPI} named “z_Prediction_results.matrix.txt” with standardized DeepCPI score for each drug-target pair (z_{CPI}), where l refers to the length of drug list and k refers to the length of target list:

$$Z_{CPI} \in \mathbb{R}^{l \times k}, \; z_{CPI} \in Z_{CPI}$$

n. Run python script “FilterOutNonSignificant.py” by typing “python FilterOutNonSignificant.py -f z_Prediction_results.matrix.txt -c 0.641”. This command will filter out the non-significant DTI scores and only keep the confident scores. The output file is “z_Prediction_results.matrix.filtered.txt”. The optimal standardized DeepCPI score threshold (0.641, sensitivity: 73%, specificity: 51.9%) is determined by receiver operating characteristics (ROC) analysis with benchmark datasets (Li et al., 2021). This pre-defined threshold may change when different benchmark datasets are used. Once defined, such threshold for standardized DeepCPI score is applicable for different target gene sets and drug sets.

$$Z_{CPI\_sig} = \begin{cases} z, \; if \; z \geq 0.641 \\ 0, \; if \; z < 0.641 \end{cases} \; z \in Z_{CPI}$$

Prioritize repurposed drug candidates

Timing: ∼10 min

Repurposed drug candidates are ranked primarily according to their targeting range (the number of target) and targeting strength (the interaction potential of target).

13. Prioritize the drug candidates using *P_score* that only considers the HDG target-associated DTIs. *P_score* is calculated for each drug candidate by the following formula, where x_{CPI_sig} represents filtered DeepCPI score for each drug-target pair and *k* refers to the length of target list.

$$P\_score_{CPI} = \sum_{i=1}^{k} x_i^{CPI\_sig} \Big/ k, \; x_{CPI\_sig} \in T_{CPI\_sig}$$

a. Open the file “Prediction_results.matrix.filtered.txt” using Excel sheet. Drugs are listed in rows and targets are listed in columns.
b. For each drug, calculate *P_score* using the above formula (AVERAGE function). The higher of *P_score*, the better the corresponding drug is prioritized.
c. The drug candidates can be ranked according to their *P_score.*

Optional: If using normalized z-score, calculate *P_score* for each drug candidate corresponding to each DTI prediction method by the following formula exemplified by DeepCPI, where z_{CPI_sig} represents filtered DeepCPI score for each drug-target pair and k refers to the length of target list. Drug candidates can be ranked by integrative consideration of multiple *P_score* derived from each DTI prediction methods.

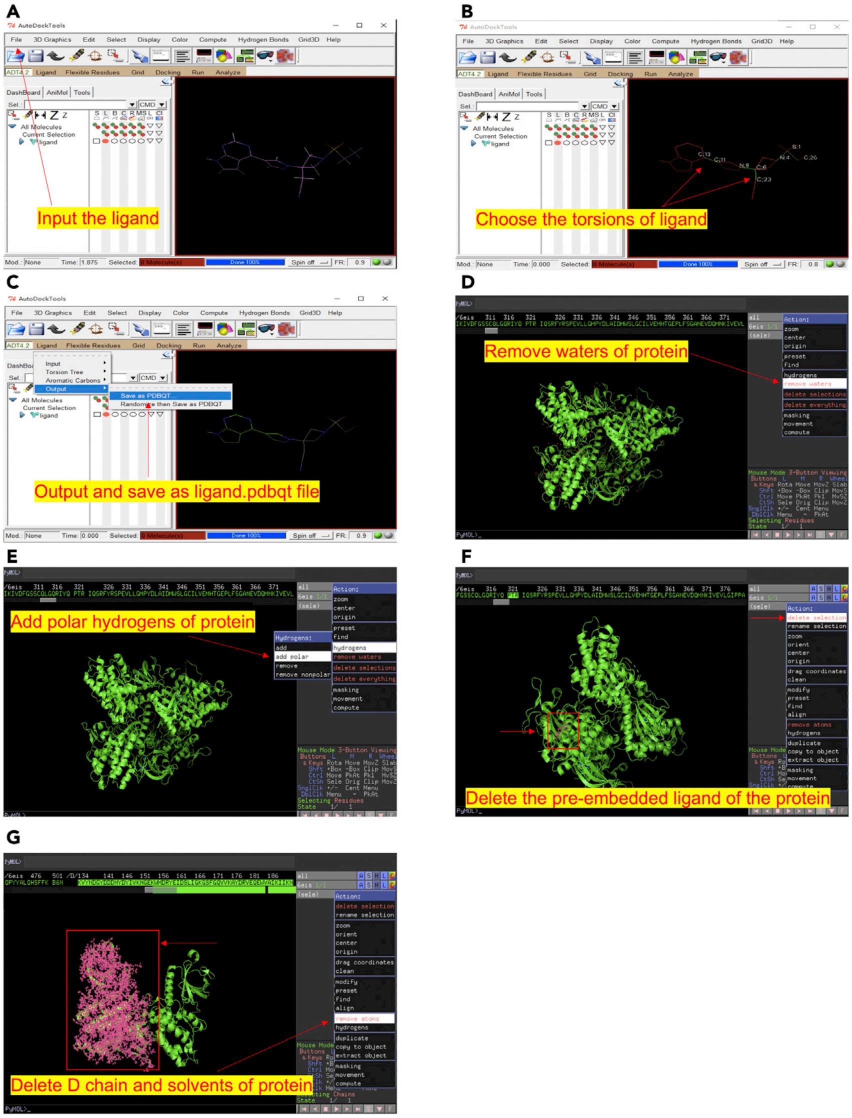
$$P\_score_{CPI} = \frac{\sum_{i=1}^{k} z_i^{CPI_{sig}}}{k}, \quad z_{CPI\_sig} \in Z_{CPI\_sig}$$

Molecular docking analysis of top ranked drugs

Timing: ~4 h

To further examine the potential binding interface and free energy between top ranked drugs and their predicted target proteins, molecular docking analysis can be performed. Using Baricitinib (one of the top ranked repurposed drugs against *Coronaviridae* viruses) and its predicted target DYRK1A as an example, molecular docking analysis is performed as in the following steps. The docking parameters may vary depending on the interrogated drug/target pair.

14. Prepare the ligand.
    a. Download the chemical structure file for Baricitinib (PubChem CID: 44205240) from PubChem website (https://pubchem.ncbi.nlm.nih.gov/) in SDF format (named as "Baricitinib.SDF").
    b. Open a PyMOL software browser and input the ligand file "Baricitinib.SDF".
    c. Export and save as "ligand.PDB" formatted file.
    d. Open the AutoDock software and input the "ligand.PDB" file (Figure 2A).
    e. Click "Ligand->Torsion Tree" and select "Choose Torsions" module (Figure 2B). The red chemical bond means un-rotatable, the green chemical bond means rotatable.
    f. Output and save as "ligand.pdbqt" formatted file (Figure 2C).
15. Prepare the protein receptor.
    a. The protein structure of DYRK1A (PDB: 6EIS) is downloaded from RCSB PDB website (http://www1.rcsb.org) in PDB format.
    b. Open a PyMOL software browser to input the file "6SIE.pdb".
    c. Remove waters (Figure 2D) and add polar hydrogens (Figure 2E).
    d. Choose the primary ligand of DYRK1A at the 321st amino acid position of A chain, and remove the pre-embedded ligand (Figure 2F).
    e. Delete the other chains (B, C, and D chains of DYRK1A in 6SIE.pdb) and solvents of the protein (Figure 2G).
    f. Save as "protein.pdb" formatted file.
    g. Open the AutoDock software and input the "protein.pdb" file.
    h. Set the atoms using "Assign AD4 type" module (Figure 3A).
    i. Compute the Gasteiger charges for protein molecules (Figure 3B).
    j. Export and save as "protein.pdbqt" formatted file (Figure 3C).
16. Set the grid box.
    a. Open the "Grid" module and input the "protein.pdbqt" file.
    b. Set map types and input the "ligand.pdbqt" file.
    c. Open "Grid Box" module to set the position of grid box.
    d. Set the center of grid box size: X center: -0.424, Y center: -16.948, Z center: -8.144. Then, set the number of points in X (60), Y (60) and Z (60) dimension of grid box to cover the active pocket (Figure 3D).
    e. Save as "dock.gpf" formatted file.
17. Analyze the grid docking.
    a. Choose the "Docking" module, and input the protein and ligand files ("protein.pdbqt" and "ligand.pdbqt").
    b. Click "Docking->Search Parameters" and choose "Genetic Algorithm" module.
    c. Click "Docking->Docking Parameters" and use the default settings.
    d. Output the Lamarckian GA result and save as "dock.dpf" formatted file (Figure 3E).



Input the ligand

Choose the torsions of ligand

Output and save as ligand.pdbqt file

Remove waters of protein

Add polar hydrogens of protein

Delete the pre-embedded ligand of the protein

Delete D chain and solvents of protein

Figure 2. Pre-processing procedures of molecular docking analysis

(A) Illustration of "Input the ligand" step by AutoDock software.

(B) Illustration of "Choose the torsions of the ligand" step in AutoDock.

(C) Illustration of "Output ligand.pdbqt file" step in AutoDock.

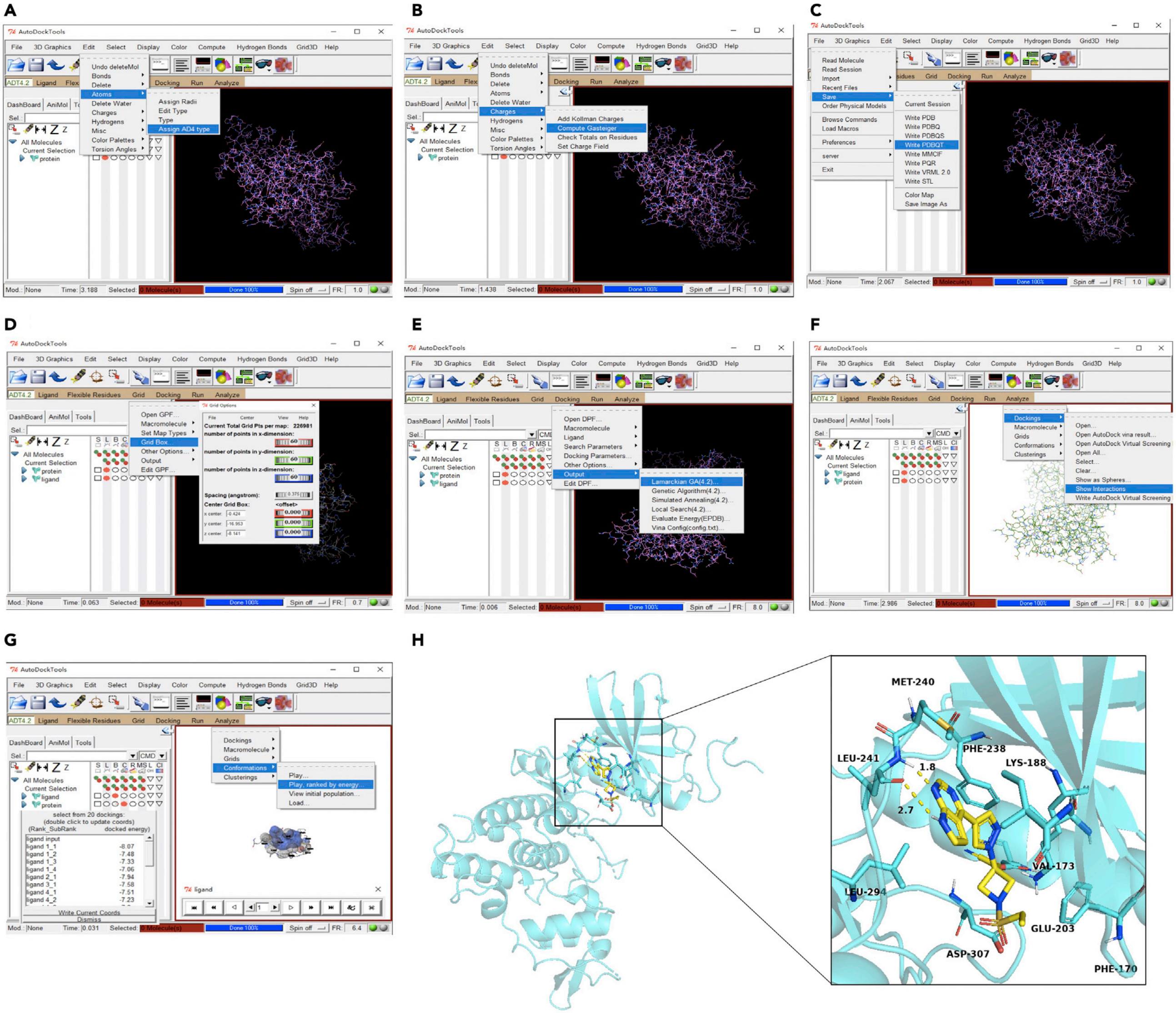
(D) Illustration of "Remove waters of protein" step in AutoDock.

(E) Illustration of "Add polar hydrogens of protein" step in AutoDock.

(F) Illustration of "Delete pre-embedded ligand" step in AutoDock.

(G) Example of "Delete the other chains and solvents of the protein" step (D chain of DYRK1A in 6SIE.pdb) in AutoDock.

e. Run the "AutoGrid" and "AutoDock" module with "dock.gpf" and "dock.dpf" file, respectively. A "dock.dlg" file is then generated.
f. Open the "dock.dlg" file and protein file ("protein.pdbqt").
g. Show the interactions between ligand and protein (Figure 3F).
h. Analyze the conformations of ligand and click this button ( ) (Figure 3G). The DashBoard shows the binding energy under different ligand conformations with the lowest binding energy of -8.07 kcal/mol for potential interaction between Baricitinib and DYRK1A A chain.
i. Output the complex interactions, and save as "result.pdbqt" formatted file.



```
select from 20 dockings:
(double click to update coords)
(Rank_SubRank           docked energy)
ligand input
ligand 1_1                  -8.07
ligand 1_2                  -7.48
ligand 1_3                  -7.33
ligand 1_4                  -7.06
ligand 2_1                  -7.94
ligand 3_1                  -7.58
ligand 4_1                  -7.51
ligand 4_2                  -7.23
```

Figure 3. Continued procedures of molecular docking analysis

(A) Illustration of setting the atoms using "Assign AD4 type" module in AutoDock software.
(B) Illustration of computing the Gasteiger charges for protein molecules in AutoDock.
(C) Illustration of exporting and saving as "protein.pdbqt" formatted file in AutoDock.
(D) Example of setting the center of grid box size to cover the active pocket in AutoDock.
(E) Illustration of outputting the Lamarckian GA result.
(F) Illustration of showing the interactions between ligand and protein.
(G) Illustration of analyzing different conformations of the ligand.
(H) Example of docking result showing the interaction between Baricitinib and DYRK1A.

18. Visualize the results of docking.
    a. Open the PyMOL browser and input the "result.pdbqt" file.
    b. Set the shape and color of the protein or the ligand.
    c. Display the background as "white".
    d. Output and save the picture of docking result as "docking.png" file (Figure 3H).

Note: Other molecular docking software can also be utilized. The binding interface and free energy may differ when using different molecular docking platforms.

Note: If there is no structure of interrogated target protein available in PDB website, protein structure prediction by homology modeling may be performed. If there is only apo-structure available where the target protein is not in complex with drugs or small molecules, binding pocket prediction or blind docking can be performed with molecular docking software.

Optional: If a deeper computational investigation on the binding-function relationship is needed, molecular dynamics (MD) simulation can be performed as elaborated in other literatures (Maximova et al., 2016; Mei et al., 2021; Yang et al., 2020).

EXPECTED OUTCOMES

In this protocol, we describe an *in silico* drug repositioning workflow to identify potential antiviral drugs against *Coronaviridae* viruses using HDGs as drug targets. A complete table listing the predicted DTI for each drug-target pair is generated, and a ranked list of the repurposed drug candidates is provided (Table S2). If there are positive control drugs with definite DTIs in other scenarios, they are expected to be present among the top positions of the ranked list. The binding details between top predicted drugs and targets are illustrated by molecular docking analysis. These results may expedite the drug development for infectious diseases caused by *Coronaviridae* viruses such as COVID-19. This strategy should be helpful to repurpose "old drug" for novel antiviral uses by facilitating the selection of lead compound for in-depth experimental and clinical evaluation.

LIMITATIONS

There are several limitations for this protocol. Firstly, the target gene set of *Coronaviridae*-specific HDGs may not be complete and the strength variation of perturbation impact between different HDGs is ignored. Secondly, only one DTI prediction algorithm (DeepCPI) is illustrated here and it is highly recommended to incorporate more independent algorithms to increase the precision and reduce the bias for DTI prediction. Thirdly, this protocol does not include the validation steps. The top ranked repurposed drug candidate should be readily selected and experimentally validated by performing in vitro assays for their cytotoxicity, antiviral activity and physical drug-target interaction before proceeding to more advanced evaluations.

TROUBLESHOOTING

Problem 1

The software and algorithms used in this protocol do not run through properly (Before you begin-Software setup and installation).

Potential solution

Double check the computer settings, make sure the downloaded versions of the software or algorithms are correct, and install them according to their manuals. Use the test data or files provided in this study to evaluate whether the software and algorithms are working properly.

Problem 2

Only a limited number of HDGs can be collected for specific type of virus (steps 1–3).

Potential solution

Insufficient number of target genes may decrease the probability and precision of drug repositioning due to low coverage of true HDGs. We recommend to expand the HDGs by additionally considering the HDG data from closely related viruses, for example, within the same viral family rather than only restricted to certain species of viruses.

Problem 3

DeepCPI is successfully installed and go through using the test data embedded in DeepCPI folder, however, it fails to generate results using user-provided data (steps 6–12).

Potential solution

Make sure to execute the program under the home directory of DeepCPI folder, double check the format of the input file, and remove any delimiter in the InChI value that may change the data structure.

Problem 4

It is difficult to determine the position of the grid box for the protein during molecular docking (step 16).

Potential solution

We recommend to try the following steps: firstly, refer to the literatures to identify potential active pocket of the protein; secondly, use "blind docking" or "binding pocket prediction" approach by AutoDock software.

Problem 5

The positive control drugs (if there are) are not in the top positions among the prioritized rank list of repurposed drugs (Expected Outcomes).

Potential solution

Carefully select the target gene set, make sure the positive control drugs are within the interrogated drug cohort, and/or apply multiple DTI prediction algorithms for drug repositioning.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Teng Fei (feiteng@mail.neu.edu.cn).

Materials availability

This study did not generate new unique reagents.

Data and code availability

This published article includes all datasets generated or analyzed during this study. The Python and R scripts can be found at the GitHub repository for this protocol (https://github.com/zexuneu/computational-framework-of-host-based-drug-repositioning).

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.xpro.2021.100653.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (31871344, 32071441), the Fundamental Research Funds for the Central Universities (N2020001, N182005005), the 111 Project (B16009), LiaoNing Revitalization Talents Program (XLYC1807212), and China Postdoctoral Science Foundation (2020T130012ZX).

AUTHOR CONTRIBUTIONS

Z.L. and Y.Y. performed the research. All the authors analyzed the data. Z.L., Y.Y., and T.F. wrote the manuscript with the input of all the other authors. T.F. supervised the study.

DECLARATION OF INTERESTS

W.L. reports serving as a consultant of Tavros Therapeutics. Other authors declare no conflict of interest.

REFERENCES

Cui, Y., Cheng, X., Chen, Q., Song, B., Chiu, A., Gao, Y., Dawson, T., Chao, L., Zhang, W., Li, D., et al. (2021). CRISP-view: a database of functional genetic screens spanning multiple phenotypes. Nucleic Acids Res. *49*, D848–D854.

Hao, G.F., Jiang, W., Ye, Y.N., Wu, F.X., Zhu, X.L., Guo, F.B., and Yang, G.F. (2016). ACFIS: a web server for fragment-based drug discovery. Nucleic Acids Res. *44*, W550–556.

Li, W., Koster, J., Xu, H., Chen, C.H., Xiao, T., Liu, J.S., Brown, M., and Liu, X.S. (2015). Quality control, modeling, and visualization of CRISPR screens with MAGeCK-VISPR. Genome Biol. *16*, 281.

Li, W., Xu, H., Xiao, T., Cong, L., Love, M.I., Zhang, F., Irizarry, R.A., Liu, J.S., Brown, M., and Liu, X.S. (2014). MAGeCK enables robust identification of essential genes from genome-scale CRISPR/Cas9 knockout screens. Genome Biol. *15*, 554.

Li, Z., Yao, Y., Cheng, X., Chen, Q., Zhao, W., Ma, S., Li, Z., Zhou, H., Li, W., and Fei, T. (2021). A computational framework of host-based drug repositioning for broad-spectrum antivirals against RNA viruses. iScience *24*, 102148.

Maximova, T., Moffatt, R., Ma, B., Nussinov, R., and Shehu, A. (2016). Principles and overview of sampling methods for modeling macromolecular structure and dynamics. PLoS Comput. Biol. *12*, e1004619.

Mei, L., Wu, F., Hao, G., and Yang, G. (2021). Protocol for hit-to-lead optimization of compounds by auto in silico ligand directing evolution (AILDE) approach. STAR Protoc. *2*, 100312.

Morris, G.M., Huey, R., Lindstrom, W., Sanner, M.F., Belew, R.K., Goodsell, D.S., and Olson, A.J. (2009). AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. J. Comput. Chem. *30*, 2785–2791.

Pushpakom, S., Iorio, F., Eyers, P.A., Escott, K.J., Hopper, S., Wells, A., Doig, A., Guilliams, T., Latimer, J., McNamee, C., et al. (2019). Drug repurposing: progress, challenges and recommendations. Nat. Rev. Drug Discov. *18*, 41–58.

Ru, J., Li, P., Wang, J., Zhou, W., Li, B., Huang, C., Li, P., Guo, Z., Tao, W., Yang, Y., et al. (2014). TCMSP: a database of systems pharmacology for drug discovery from herbal medicines. J. Cheminform. *6*, 13.

Tanoli, Z., Vaha-Koskela, M., and Aittokallio, T. (2021). Artificial intelligence, machine learning, and drug repurposing in cancer. Expert Opin. Drug Discov. 1–13.

Wan, F., Zhu, Y., Hu, H., Dai, A., Cai, X., Chen, L., Gong, H., Xia, T., Yang, D., Wang, M.W., et al. (2019). DeepCPI: A deep learning-based framework for large-scale in silico drug screening. Genomics Proteomics Bioinformatics *17*, 478–495.

Wang, F., Yang, J.-F., Wang, M.-Y., Jia, C.-Y., Shi, X.-X., Hao, G.-F., and Yang, G.-F. (2020). Graph attention convolutional neural network model for chemical poisoning of honey bees' prediction. Sci. Bull. *65*, 1184–1191.

Wishart, D.S., Feunang, Y.D., Guo, A.C., Lo, E.J., Marcu, A., Grant, J.R., Sajed, T., Johnson, D., Li, C., Sayeeda, Z., et al. (2018). DrugBank 5.0: a major update to the DrugBank database for 2018. Nucleic Acids Res. *46*, D1074–D1082.

Yang, J.F., Wang, F., Chen, Y.Z., Hao, G.F., and Yang, G.F. (2020). LARMD: integration of bioinformatic resources to profile ligand-driven protein dynamics with a case on the activation of estrogen receptor. Brief. Bioinform. *21*, 2206–2218.

Zhou, Y., Wang, F., Tang, J., Nussinov, R., and Cheng, F. (2020). Artificial intelligence in COVID-19 drug repurposing. The Lancet Digital health *2*, e667–e676.