deepDegron Documentation

Release 1.0.0

Collin Tokheim

Jun 16, 2021

CONTENTS

Author Collin Tokheim, Shirley Liu

Contact ctokheim # ds DOT dfci DOT harvard DOT edu

Lab Liu Lab

Source code GitHub

Q&A Biostars (tag: deepDegron)

The Ubiquitin-Proteasome System (UPS) is the primary means for selective protein degradation in cells. While the UPS may contribute upwards of 19% of mutated driver genes in cancer, a systems-level understanding of the UPS is lacking. The regulatory specificty of the UPS is thought to be governed by E3 ligases recognizing short amino acid sequence motifs, known as degrons, on substrate proteins. However, only a handful of E3 ligases has known degron motifs, hampering our capability to understand UPS regulation in normal physiology and disease.

deepDegron is a machine learning method to systematically predict the potential for a protein sequence to contain a degron. Leveraging this capability, deepDegron also allows the user to predict whether a mutation likely disrupts a degron, which may lead to increased protein stability. Furthermore, it also includes a statistical test to examine for enrichment of mutations leading to degron loss in a gene. Currently, deepDegron supports predictions for degrons at the c-terminus and n-terminus of proteins. Future updates may expand to the full proteome.

Contents:

CHAPTER

ONE

DOWNLOAD

1.1 deepDegron releases

- deepDegron v1.1.0 - 3/4/2021 - Command line interface now uses sub-parsers
- deepDegron v1.0.0 - 2/17/2021 - Initial release

1.2 Trained deepDegron models

C-terminal deepDegron:

- position specific model
- bag of amino acids model

N-terminal deepDegron:

- position specific model
- bag of amino acids model

CHAPTER

TWO

INSTALLATION

deepDegron has only been tested on linux operating systems. We recommend that you use **python 3.7** to run deepDegron.

2.1 Installation by PIP

The easiest way to install deepDegron is to use PIP.

```
$ pip install deepDegron
$ pyensembl install --release 75 --species human  # download human hg19 reference data
$ pyensembl install --release 95 --species human  # download human hg38 reference data
```

2.2 Installing from source

2.2.1 Releases

First download the deepDegron source code on on github.

Once downloaded, please change to the top-level directory in the deepDegron source code.

2.2.2 Installing dependencies

We recommend using conda to install the deepDegron dependencies.

```
$ conda env create -f environment.yml  # create environment for deepDegron
$ source activate deepDegron # activate environment for deepDegron
$ pyensembl install --release 75 --species human  # download human reference data
$ python setup.py install  # install deepDegron
```

Make sure the deepDegron environment is activated when you want to run deepDegron.

An alternative way to install the python dependencies is to use pip.

```
$ python -m pip install --upgrade pip
$ pip install -r requirements.txt  # install required packages
$ pyensembl install --release 75 --species human  # download human reference data
$ python setup.py install  # install deepDegron
```

CHAPTER

THREE

TUTORIAL

In this tutorial we will be investigating somatic mutations found in GATA3 in breast cancer samples from The Cancer Genome Atlas (TCGA). Note, this analysis could equally apply to other types of variants, such as germline or de novo variants, as well.

3.1 Scoring mutations for impacting degrons

deepDegron computes a degron potential score to represent the liklihood a protein sequence contains a degron. Mutations may lead to a change in degron potential. The difference between the degron potential of the mutant compared to wildtype sequence is what we call "delta degron potential". The more negative this score is, the more deepDegron predicts a degron has likely been disrupted by a mutation.

The first step to score mutations is to download the trained c-terminal degron models, as well as the necessary data file of mutations. Here, we are using mutations found in the GATA3 gene in breast cancer.

```
$ wget https://github.com/ctokheim/deepDegron/raw/master/models/cterm/neural_network_pos_
↪specific.pickle
$ wget https://github.com/ctokheim/deepDegron/raw/master/models/cterm/neural_network_bag_
↪of_amino_acids.pickle
$ wget https://raw.githubusercontent.com/ctokheim/deepDegron/master/tests/data/gata3_
↪mutations.txt
```

To obtain delta degron potential scores for mutations, you can use the deepDegron_score command.

```
$ deepdegron score -i gata3_mutations.txt -c models/cterm/neural_network_pos_specific.
↪pickle,models/cterm/neural_network_bag_of_amino_acids.pickle -o GATA3_delta_degron_
↪potential.txt
```

You will notice the output file will contain all mutations that impact the c-terminal protein sequence of GATA3, as well as the delta degron potential scores for each mutation. Notice for GATA3 that many are frameshift indels that have a very negative delta degron potential, indicating likely degron loss. You results should match the results seen here.

3.2 Statistical test

DeepDegron also can test whether there is a significant enrichment for mutations that likely lead to degron loss. This helps to avoid non-significant cases where a degron loss mutation may have happened by chance, and may not play a role in a given phenotype/condition.

To run the deepDegron statistical test on GATA3, use the deepDegron_test command.

```
$ deepdegron test -i gata3_mutations.txt -ns 100 -c models/cterm/neural_network_pos_
↪specific.pickle,models/cterm/neural_network_bag_of_amino_acids.pickle -o GATA3_result.
↪txt
```

Because in this example we are analyzing c-terminal degrons, we supplied the trained deepDegron models using the -c flag. However, for analyzing n-terminal degrons, the -n flag should be used. Additionally, for this toy example, we used only 100 simulations (-ns parameter), but in practical applications this should be much larger (e.g. 10,000). Note, increasing the number of simulations increases precision but has a longer run time.

Your result should show a delta degron potential of -23 and a p-value of 0.0 (beyond resolution of the 100 simulations) for the GATA3 data. It should match the results available here.

CHAPTER

FOUR

FILE FORMATS

4.1 Mutations

Mutations are provided in a Mutation Annotation Format (MAF) file (specification here). Columns can be in any order, and only a few columns in the MAF file are needed. The following is a list of the required columns.

- Hugo_Symbol
- Chromosome
- Start_Position
- End_Position
- Reference_Allele
- Tumor_Seq_Allele2
- Tumor_Sample_Barcode
- Variant_Classification

The remaining columns in the MAF specification can be left empty or not included.

Only coding variants found in the Variant_Classification column will be used, which includes the following: 'Missense_Mutation', 'Silent', 'Nonsense_Mutation', 'Splice_Site', 'Nonstop_Mutation', 'Translation_Start_Site', 'Frame_Shift_Ins', 'Frame_Shift_Del', 'In_Frame_Ins', or 'In_Frame_Del'.

CHAPTER

FIVE

FAQ

Who should I contact if I encounter a problem?

If you believe your problem may be encountered by other users, please post the question on biostars. Check to make sure your question has not been already answered by looking at posts with the tag deepDegron. Otherwise, create a new post with the deepDegron tag. We will be checking biostars for questions. You may also contact me directly at ctokheim AT ds DOT dfci DOT harvard DOT edu.

Can I run deepDegron using mutations annotated on hg19 and/or hg38?

Yes, deepDegron supports both hg19 and hg38. You will, however, need to download the relevant reference data for pyensembl first. We recommend ensembl release 75 for hg19:

```
$ pyensembl install --release 75 --species human  # download hg19 human reference data
```

For hg38, we recommend ensembl release 95:

```
$ pyensembl install --release 95 --species human  # download hg38 human reference data
```

To correctly specify which reference genome you are using, please supply the relevant ensembl release number to –ensembl-release flag in deepDegron.

```
$ deepdegron test [options] --ensembl-release 75 # for hg19
$ deepdegron test [options] --ensembl-release 95 # for hg38
```

Where can I obtain the training data for deepDegron?

You can obtain the set of mutations used for training from github for c-terminal degrons and n-terminal degrons.

What file format should I use for mutations?

deepDegron currently only reads MAF files. Please also see the *File formats* page.

CHAPTER

SIX

CITATION

Tokheim, C., Wang, X., Timms, R.T., Zhang, B., Mena, E.L., Wang, B., Chen, C., Ge, J., Chu, J., Zhang, W., et al. (2021). Systematic characterization of mutations altering protein degradation in human cancers. Mol Cell. link