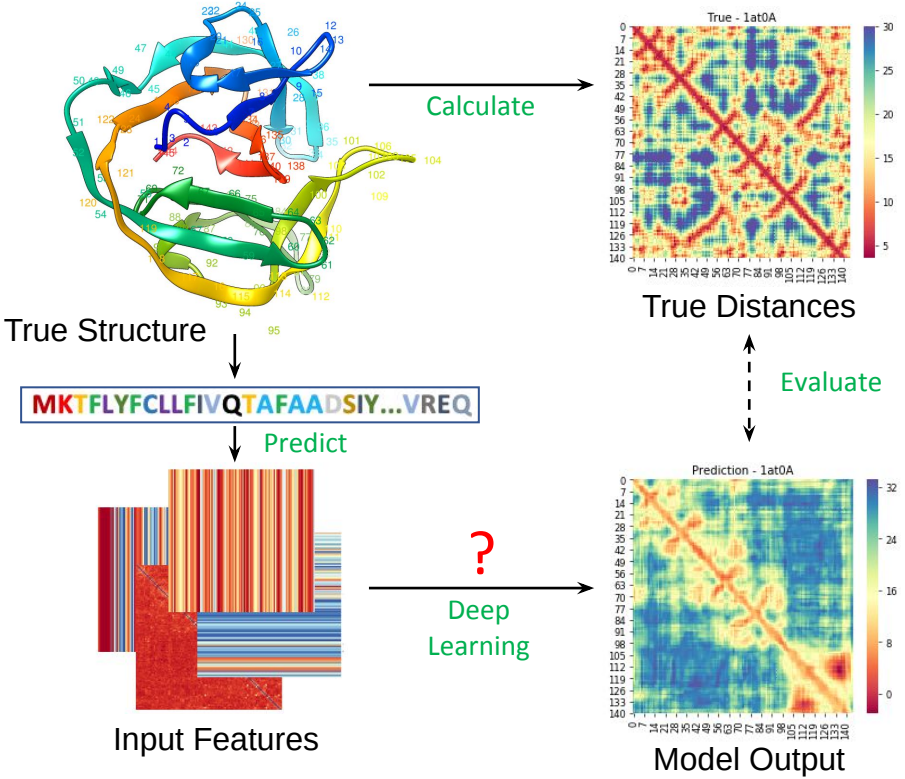
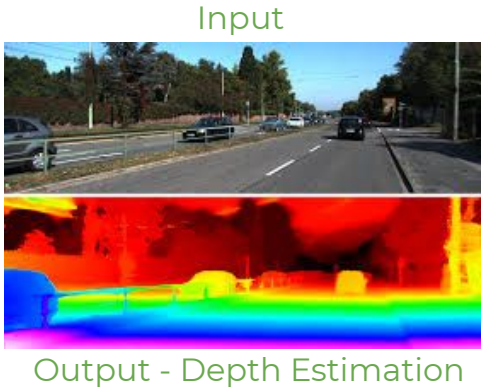
ProteinDistNet: A dataset for deep learning protein inter-residue distances and contacts

Badri Adhikari
Assistant Professor of CS
Department of Mathematics & Computer Science
University of Missouri-St. Louis

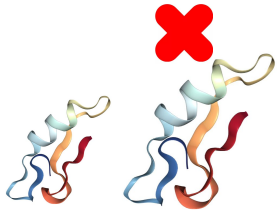
Protein inter-residue distance prediction (PIDP)

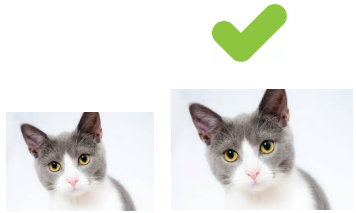


Unique features of PIDP problem (compared to other DL problems)



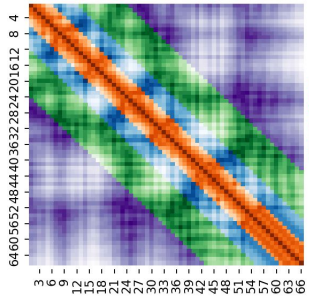
- Large number of input channels
  - Other examples: plant genotype prediction from hyperspectral images
- Input features are 0D, 1D, or 2D
- Visualization is less meaningful
  - Predicted distance/contact maps can be visualized and compared
  - But, the visualizations do NOT enable us to study and debug what the filters are learning
- Non-scalability of protein structures
  - An object in the real world (for example, a chair) may be tiny or large
  - Proteins can also be large or small but the size of structure patterns are always physically fixed
  - The size of an alpha helix is the same in proteins of any size





Unique features of PIDP problem (compared to other DL problems)

- Variable input feature volume
  - The length of a protein sequence can vary from a few residues to a few thousand residues
- The goal is to predict distances
  - The problem can be formulated as binary classification, multi-class classification, or regression
  - The ultimate goal is to develop methods that can predict raw physical distances (in Angstroms)
    - Similar to NMR experiments
- Long-range distances are important
  - A model that predicts top L/2 contacts accurately is not always the model with minimum loss
  - This imposes an additional challenge in model training and selection
- Symmetrical along diagonal



Distance prediction: One problem, many questions

"Many questions remain unanswered at this intersection of deep learning and distance prediction"

- Is the data that we have sufficient?
  - If so, why is it that in every CASP competition methods that use newer/larger sequence DBs win?
- Are the current deep learning methods "fit" for the distance prediction problem?
  - Are residual networks and minor variations the end?
- How to best engineer the features?
  - So many features and possible combinations - sequence features, coevolution-based algorithms, raw features such as pair frequencies matrix, covariance matrix, and precision matrix
- How similar/different is the structure prediction problem compared with other (well studied) deep learning problems?

Is coevolution information “the solution” to structure prediction?

- The ‘hope’ from coevolution information
  - MSAs and coevolution information appear to be the path for pushing structure prediction, mainly contact and distance prediction
- The irony:
  - Amino acid sequence in a cell, when folding, does NOT have access to any coevolution or conservation information
- Is coevolution information a ‘trap’ for us to push structure prediction?
  - How many years should we spend on developing coevolution based method?
  - How far can we push structure prediction using multiple sequence alignments?
  - We need to make that push as soon as possible
    - so we can either get back to the physics and chemistry of protein folding or find another paths that will lead us further

How can we speed up the progress(es) ?

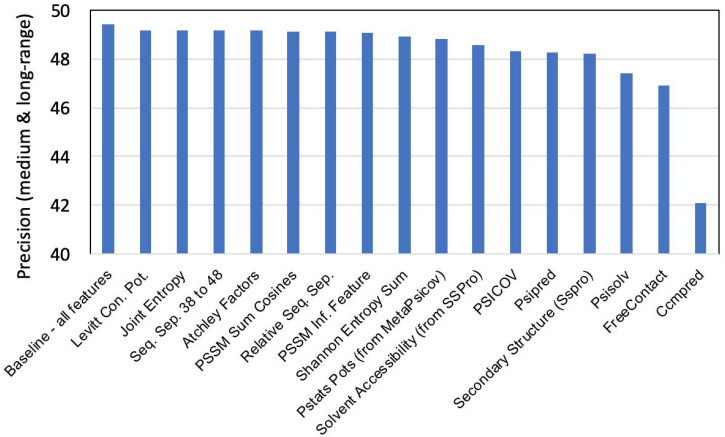
- We need a dataset that is small, representative, & packaged for instant development
- Why?
  - Such datasets allow rapid progress
  - For example, MNIST and ImageNet for computer vision problems
- Do we have such a dataset for protein distance/contact prediction?
  - ProteinNet (Mohammed AlQuraishi)
  - ProSPr (todo) (Wendy M Billings et al.)
- Our goal
  - create such a dataset

Preparation of the “ProteinDistNet” dataset

- Reference
  - The 3456 representative proteins and 150 test protein chains used to train, validate, and test the “DeepCov” method for protein contact prediction
- Removed structural gaps
  - Some chains had adjacent Cβ atoms are too far apart in the 3D space
  - For all the proteins that had such structural discontinuity we only kept the first structural domain
  - 3424 protein chains remained
- Input features and output distance maps
  - Trimmed the chains that are longer to 256 residues
- ProteinDistNet and ProteinDistNet128 datasets
  - Further trimmed the training and validation set to 128 residues (ProteinDistNet128)

Features generation and reduction

- Commonly used features
  - Predicted secondary structures, coevolution features, solvent accessibility, position-specific scoring matrix derived features, Atchley factors, many pre-computed statistical potentials, alignment statistics such as the number of effective sequences, Shannon entropy sum, mean contact potential, normalized mutual information, etc.
- Which of these contain complementary information and which are redundant?



Input features

1. Secondary structure predictions (PSIPRED)
2. Solvent accessibility predictions (PSIPRED)
3. Coevolutionary signals predicted using CCMpred
4. Coevolutionary signals predicted using FreeContact
5. Contact potentials calculated from multiple sequence alignments
6. Shannon entropy of the alignment column
7. Sequence profiles from the multiple sequence alignments

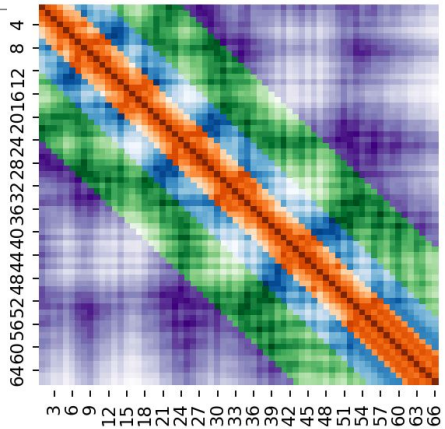
The ProteinDistNet dataset

- Number of chains
  - 3424 chains for training and validation with sequence lengths ranging from 50 to 256 residues
  - 150 test protein chains with lengths ranging from 50 to 266 residues
- Disk space
  - ProteinDistNet is only 736 MB & ProteinDistNet128 is only 360 MB (zipped)
- Scripts are all released
  - All the scripts used to curate the dataset, generate the input features and distance maps
  - Scripts with example deep learning models for training, validation and testing
- Development
  - Scripts for generating input features and distance maps are written in Python3 (Tensorflow)
  - The output files are standard *text files* containing lists of protein chain IDs, *pickle files* containing a dictionary of features, and *numpy files* containing numpy array of distance maps

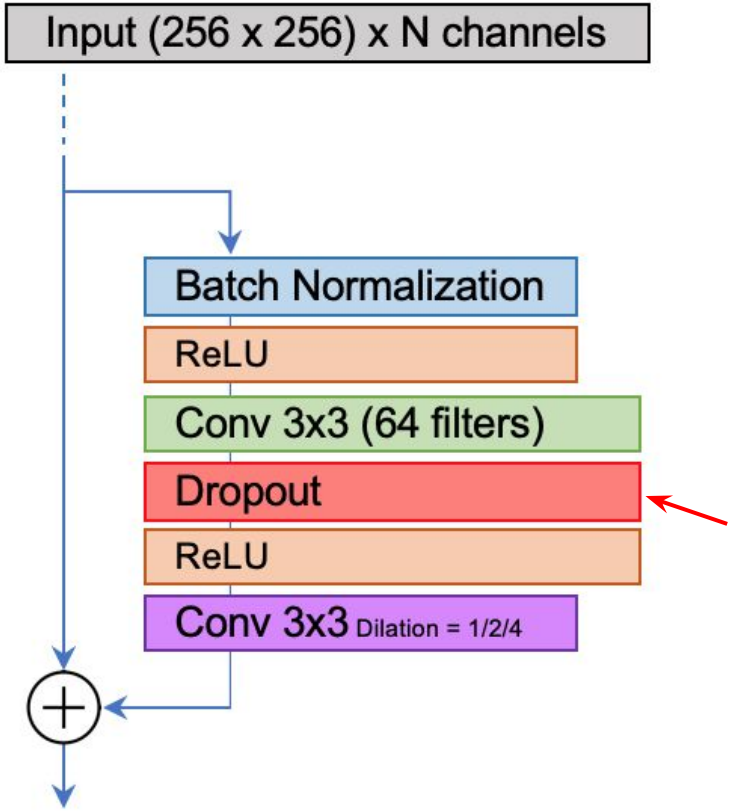
Evaluation of predicted distances

Prediction	Metric	Description
Distances	$M_{LR-L/5}$	Mean absolute error of smallest L/5 long-range distances (in Å)
	M_{LR-L}	Mean absolute error of smallest L long-range distances (in Å)
	$M_{MLR-L/5}$	Mean absolute error of smallest L/5 medium- or long-range distances (in Å)
	M_{MLR-L}	Mean absolute error of smallest L medium- or long-range distances (in Å)
Contacts	$P_{LR-L/5}$	Precision of top L/5 long-range contacts
	P_{LR-L}	Precision of top L long-range contacts
	$P_{MLR-L/5}$	Precision of top L/5 medium- and long-range contacts
	P_{MLR-L}	Precision of top L medium- and long-range contacts

Not in literature!



Deep learning architecture used for obtaining benchmark results



Architecture:

Residual network with dropout and dilation (DEEPCON)
Similar to AlphaFold's architecture + Dropout layer

Number of residual blocks:

128

Number of parameters:

9.5 million

Epochs:

32

Last layer activation:

'sigmoid' or 'relu'

Cross-validation or ensembling:

None

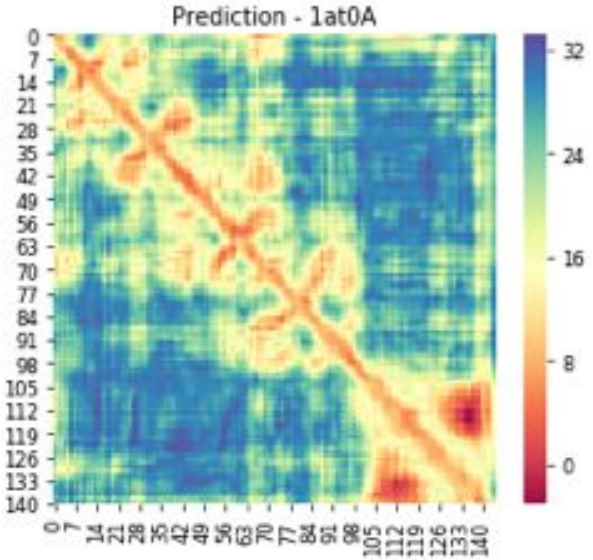
Contact prediction benchmark

Training set	Evaluation sets	$P_{LR-L/5}$
ProteinDistNet128	Validation	68.86
	Test	93.18
ProteinDistNet	Validation	76.16
	Test	93.46

The gain from using full 256 dataset (instead of 128) is much higher for validation set.

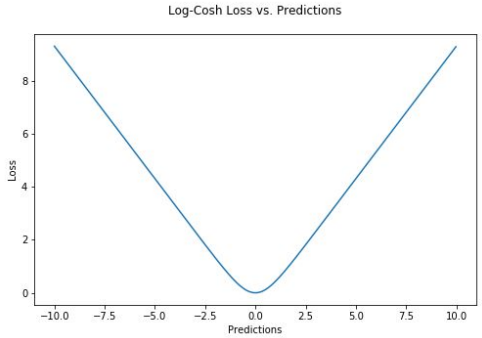
Distance prediction problem as a “regression problem”

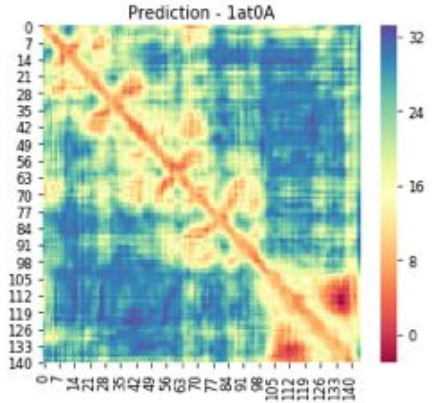
- We set the last layer’s activation as ‘ReLU’ instead of ‘sigmoid’
- It is more meaningful to predict inter-residue interactions than non-interactions
  - i.e. it is more important to predict smaller distances more accurately than larger distances
  - Makes sense from the perspective of structure prediction and binding-site prediction



Mae, mse, and logcosh losses do not work

- Will the loss functions such as mean squared error or mean absolute error work?
  - They will focus on optimizing the large distance values before the smaller ones
- ‘Logcosh’ loss (logarithm of hyperbolic of cosine) is found to be highly effective for many problems
  - It is behaves similar to the squared loss for smaller loss values and similar to absolute loss otherwise, i.e. the loss is not so strongly affected by the occasional incorrect predictions
  - This still does not focus on optimizing the smaller distances





“Inverse” logcosh loss

- As a solution, we propose a novel loss function that precisely focuses on optimizing the smaller distances first
  - When training the model, ‘reciprocate’ the true distances and then apply the standard logcosh loss

$$LohcoshLoss = mean(log(cosh(P - T))$$

$$InverseLogcoshLoss = K * mean(log(cosh(\frac{1.0}{P+e} - \frac{1.0}{T+e})$$

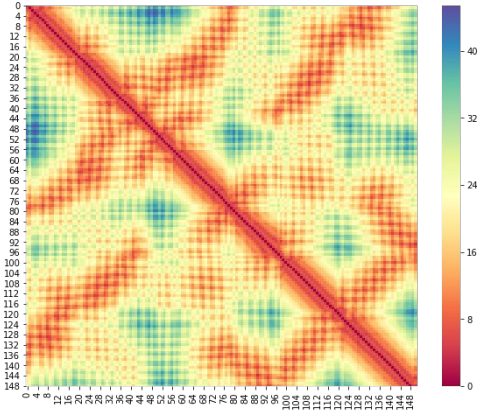
  - P is predicted distance
  - T is true distance
  - e is a small positive number (epsilon)
  - K is a scalar that simply scales the losses so the values do not underflow
    - We empirically set K to 100

Implementing the “inverse” logcosh loss

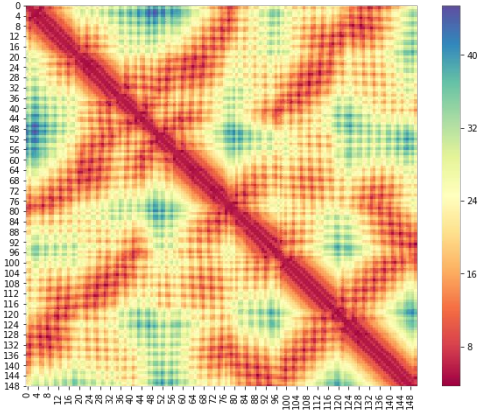
```
def _logcosh(x):
    return x + nn.softplus(-2. * x) - math_ops.log(2.)

def inv_log_cosh(y_true, y_pred):
    return K.mean(_logcosh(100.0 / (y_pred + epsilon) - 100.0 / (y_true + epsilon) ), axis=-1)
```

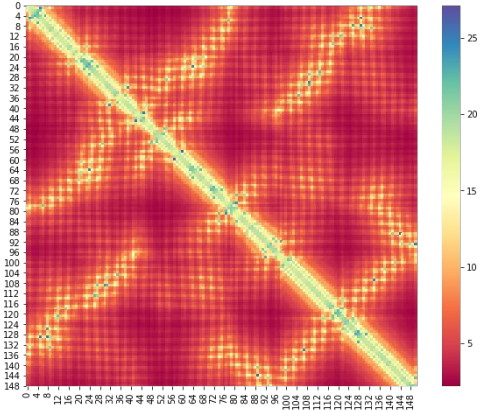
Trick: Invert the input instead of the loss function (reciprocate)



$$d_{ii} \neq 0$$

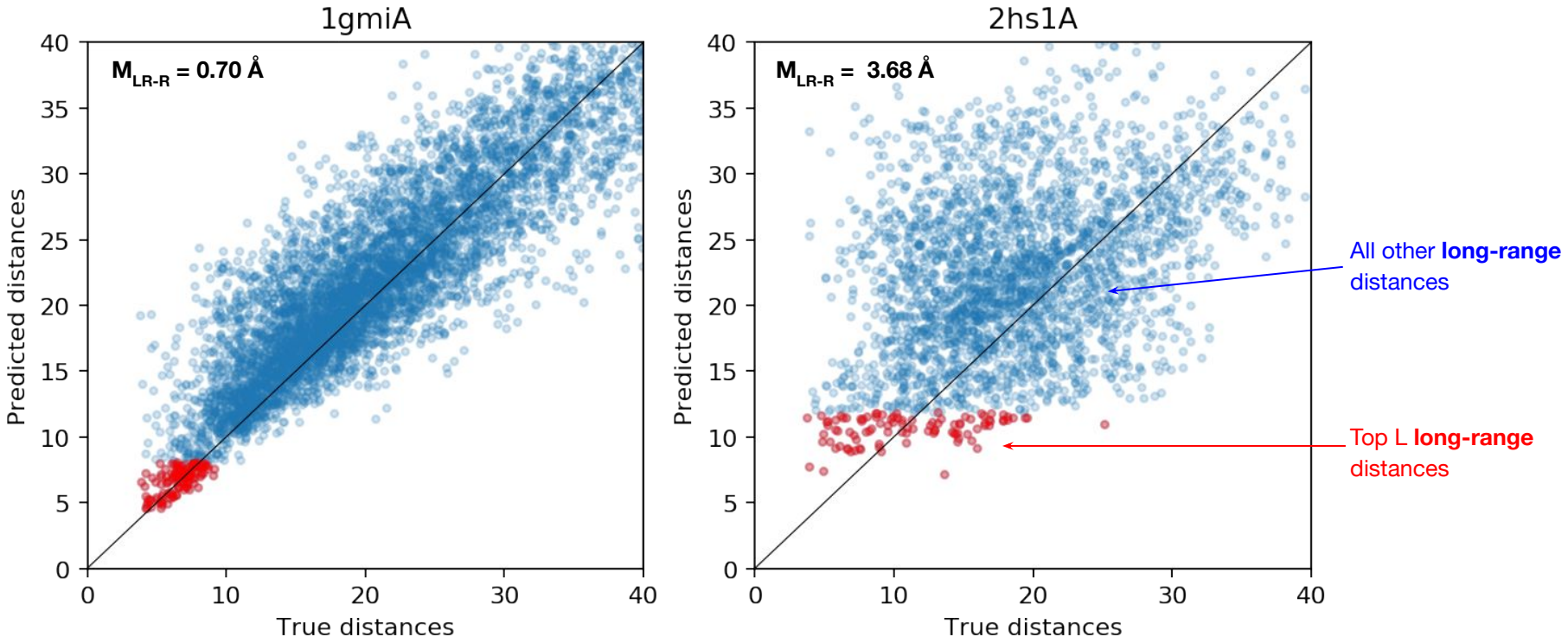


$$d_{ij} = \frac{100.0}{d_{ij}}$$



Inverting input works!

Comparison of true long-range distances and the distances predicted by the model



The model effectively focuses on correctly predicting the smaller long-range distances over larger long-range distances.

Distance prediction benchmark

Training set	Evaluation set	$M_{LR-L/5}$	M_{LR-L}	$P_{LR-L/5}$	P_{LR-L}
ProteinDistNet128	Validation	2.21	2.70	64.83	38.63
	Test	0.94	1.35	90.69	63.85
ProteinDistNet	Validation	1.99	2.56	72.52	45.92
	Test	0.92	1.32	91.68	66.09

Predict contacts (binary classification)

Training set	Evaluation sets	$P_{LR-L/5}$	P_{LR-L}
ProteinDistNet128	Validation	68.86	42.74
	Test	93.18	68.13
ProteinDistNet	Validation	76.16	49.98
	Test	93.46	69.31

The use of ProteinDistNet

- What deep learning works and does not work for distance prediction?
  - Loss function, architecture comparison, feature engineering, feature importance study, etc.
- Helpful for new students or postdocs
- How accurately can we predict ‘raw’ distances?
- The models built can be evaluated on CASP12 and CASP13
  - Because the dataset is curated before CASP12
- Machine learning experts unfamiliar with distance prediction can quickly jump in and contribute

Availability

- Github
  - https://github.com/ba-lab/ProteinDistNet
- All scripts are also available
  - All scripts used to generate the features
  - Can be used to train with a much larger dataset
- Manuscript in progress

Training and testing in Google colaboratory

Test.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

Editing

```
'''
Author: Badri Adhikari, University of Missouri-St. Louis,  11-13-2019
File: Contains the code for training, validating, and testing distance maps
'''

import tensorflow as tf
from tensorflow.keras.callbacks import ModelCheckpoint
import os
import sys
import numpy as np
import datetime

from dataio import *
from metrics import *
from generator import *
from models import *
from losses import *
```

Contributions/Conclusions

- A data set that is small, representative, and packaged for instant development
- A ‘trick’ to attack distance prediction as a regression problem

Acknowledgements

Research Contributions


Mrinal Rawool


Amarilda Dyrmishi


Anthony A.-Nyanzu


David Richards


Patrick Kong

IT Support

Philip Reiss
Kenneth Voss
Michael Remier
MU - Research Support (RCSS)

Computing Resources

Google

NVIDIA

University of Missouri System
COLUMBIA | KANSAS CITY | ROLLA | ST.LOUIS

Thank you all.

Questions?