Prediction of protein subcellular localization

Machine Learning project

Raffaelli Claudia, Chaudhry Abdullah

claudia.raffaelli@stud.unifi.it, abdullah.chaudhry@stud.unifi.it

Department of Information Engineering
University of Florence
Via di Santa Marta 3, Florence, Italy

Overview

Subcellular localization

Subcellular localization is a multi-class classification problem where a protein sequence is assigned to one of different cell compartments.

We will see:

- Subcellular localization prediction **background**.
- **State-of-the-art**.
- **Networks and methodologies** according to Almagro et al. [1].
- Baseline [2] and achieved **results comparison**.
- Overview on **ELMo** [3] representations.
- Evaluation of new findings.

Subcellular localization background

Some facts about **proteins** [4]:

- Consist of single or multiple chains, called **polypeptides**, each of thousands of amino acids.
  - The amino acids alphabet is composed by 20 symbols.
- Are produced in the ribosomes, located in a specific zone of the cell,
  - in a process known as **translation**.
- Are transported to their destination by checking for molecular **tags**,
  - i.e. amino acid subsequences that act as shipping signals to specific subcellular sites.
  - Usually located at the start (N-terminus) or at the end (C-terminus) of the amino acid chain.
- The delivery, carried by a protein complex, occurs in about **10 sites**.

Applications

Applications of subcellular localization involve:

- Target identification during drug discovery process.
- Better understanding of overall cell function.
- Identification of proteins as vaccine candidates.

Common **means of locating** a protein are the detection of:

- particular shipping **signals** at the beginning or end of the chain,
- **motifs**, i.e. amino acid arrangements that add functionalities.
  - Some motifs are also highly correlated with the destination codes.

State-of-the-art approaches

Relevant literature [5] regarding the topic:

- BLAST [6] search is an homology based method. Chooses as location the best hit in a database of annotated examples.
  - **Sequence homology** defines the shared ancestry in the evolutionary history of that protein.
- Signal-3L [7] sends PSI-BLAST profiles as input to an SVM.
  - **PSI-BLAST** derives a profile from multiple sequence alignment (more later).
- Parras et al. in [8] uncover evolutionary relationship while doing motif classification.
- DeepLoc [9][1] uses a recurrent neural network (RNN) and an attention mechanism to detect important protein regions.

DeepLoc paper

Main topics covered:

- **Dataset**,
  - a few observation on common issues.
- Neural networks **models**,
  - benefits of using convolution neural networks (CNN),
  - attention mechanism,
  - hierarchical tree of sorting pathways.
- Experiments
  - for **model selection**, comparing relative performances on different architectures,
  - to test **generalization** ability of DeepLoc dataset.

Datasets

Datasets **specifications**:

- **DeepLoc** [10] dataset in FASTA format, with about 14000 proteins,
  - constructed using protein data from **UniProt** dataset,
  - each sequence labeled:
    - for subcellular localization (10 classes),
    - as either membrane, soluble or unknown.
  - with careful **homology reduction** in train/test to 30%.
- **Multiloc** [11] FASTA dataset, with almost 6000 proteins,
  - split into 11 different classes for subcellular localization,
  - homology reduced to 80% of identity.

Processed datasets

Processed datasets:

- Their DeepLoc:
  - Two versions: sequences long 400 and 1000.
  - Encoded with either BLOSUM64, PSI-BLAST profiles or HSDM.
  - Only labeled for subcellular localization.
- Our DeepLoc:
  - Two versions: sequences long 400 and 1000.
  - Two versions: encoded with PSI-BLAST and with one-hot encoding.
  - Both labels.
- Our Multiloc:
  - Sequences long 1000.
  - Encoded with PSI-BLAST.
  - Only subcellular localization label available.

PSI-BLAST encoding

Profile analysis

Detects distantly related proteins by sequence comparison. Express the information of a group of sequences aligned by structural or sequence similarity in a position-specific scoring table PSSM or **profile** [12].

PSI-BLAST is a tool that:

- Derives the PSSM from the multiple sequence alignment above a given threshold using protein–protein BLAST.
- PSSM is used to further search the database for new matches.
- Detects distant relationships between proteins.
- We computed the 4 iteration PSSM.

Model architectures

Recurrent Neural Network (RNN)

Used for ordinal or temporal problems when data is sequential. Info from prior inputs (memory) is used to influence current input and output.

Layers used:

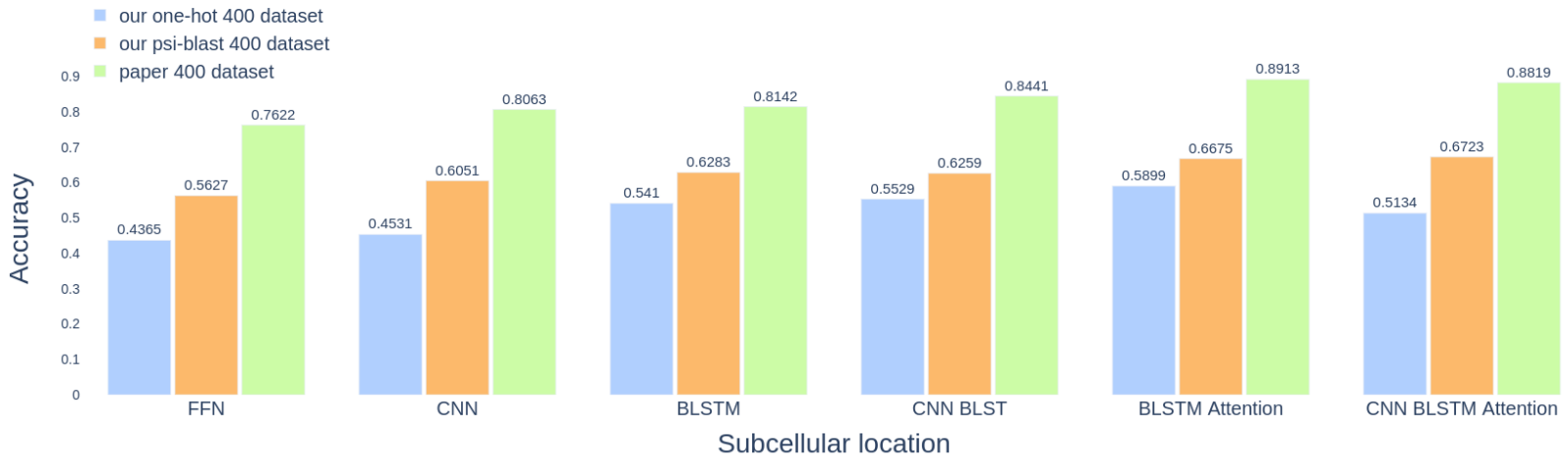
- **Convolutional** to extract motif information of given length.
- **Long short-term memory (LSTM)** stores long-term dependencies
  - and holds info from the start and the end of a sequence (BLSTM).
- **Bahdanau's Attention** [13] to focus on relevant sequence parts:
  1. The encoder produces hidden states for each input.
  2. From which are computed attention weights, later softmaxed.
  3. Is obtained the context vector = attention weights * hidden states.
  4. The decoder yields new output from the older and the context vector.

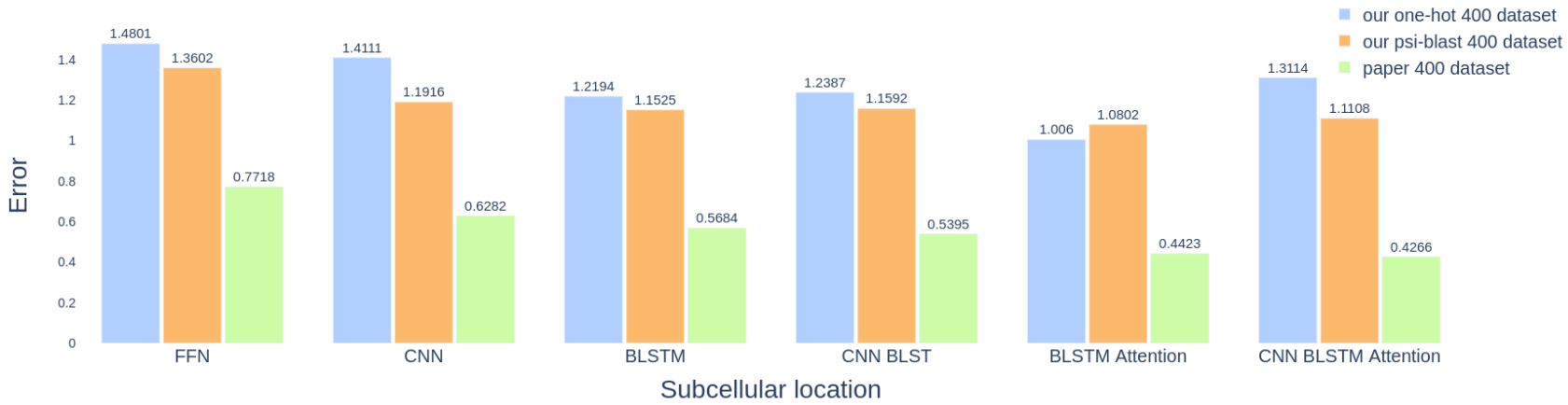
Hyperparameters optimization

Two **optimization** methods:

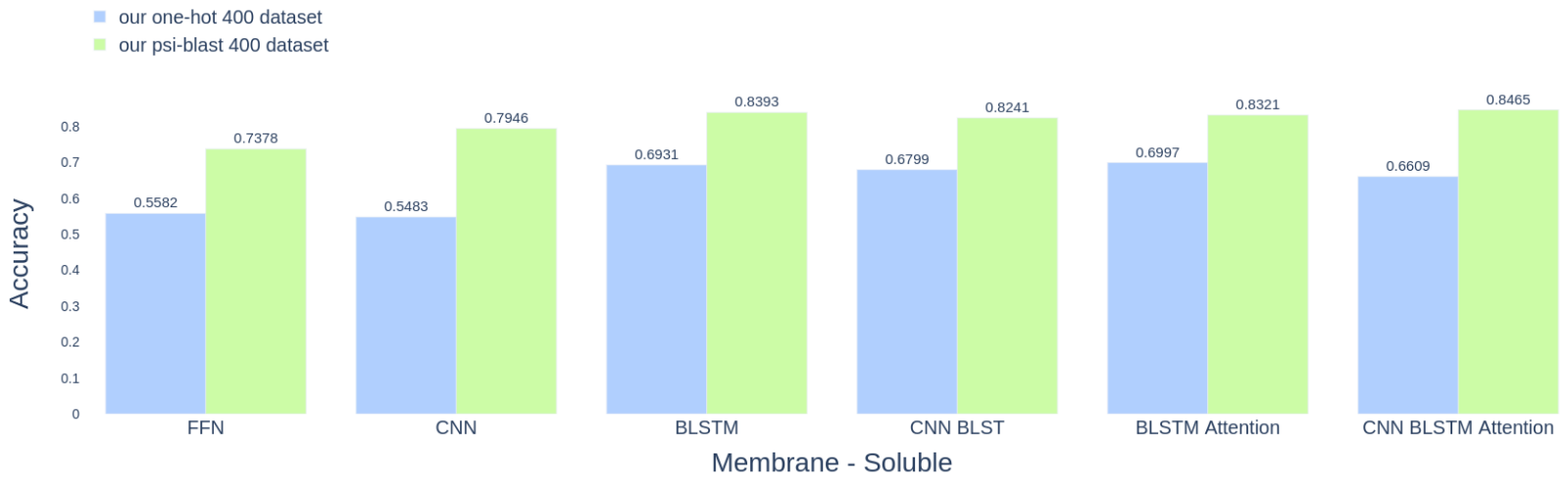
- Talos **Random search** defines a search space as a bounded domain of hyperparameter values.
  - Randomly sample points in that domain.
  - Chooses the best configuration so far.
- Keras Tuner **Bayesian search**, finds best values in a search space.
  - Builds a surrogate probability model of the objective function and finds the hyperparameters that perform best on the surrogate.
  - Applies the hyperparameters found to the true objective function.
  - Aims to become less wrong with more data inputs by updating the surrogate model after each objective function's evaluation epoch.

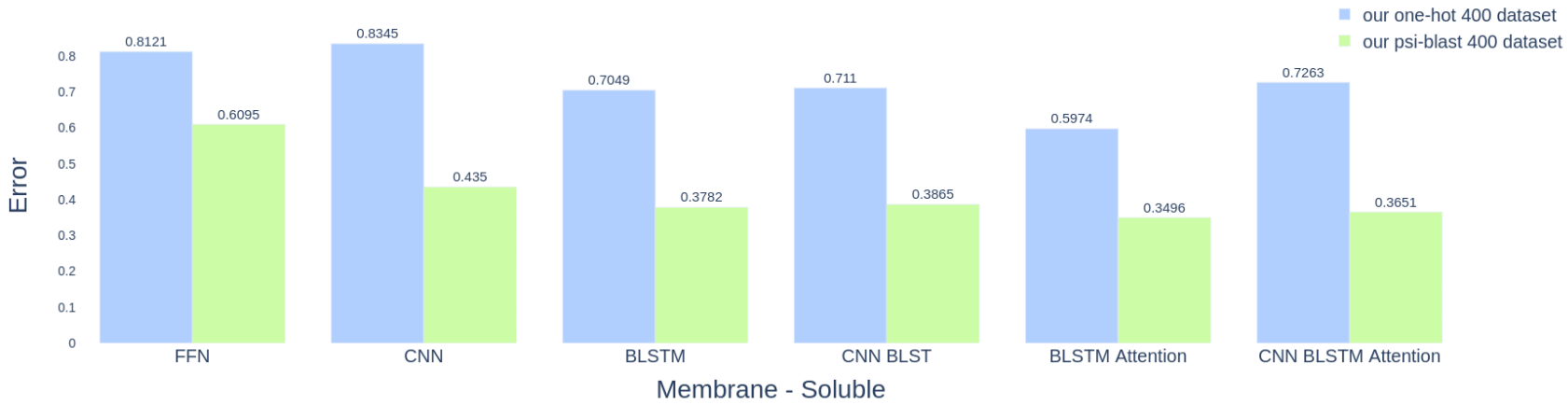
Experiment: models comparison



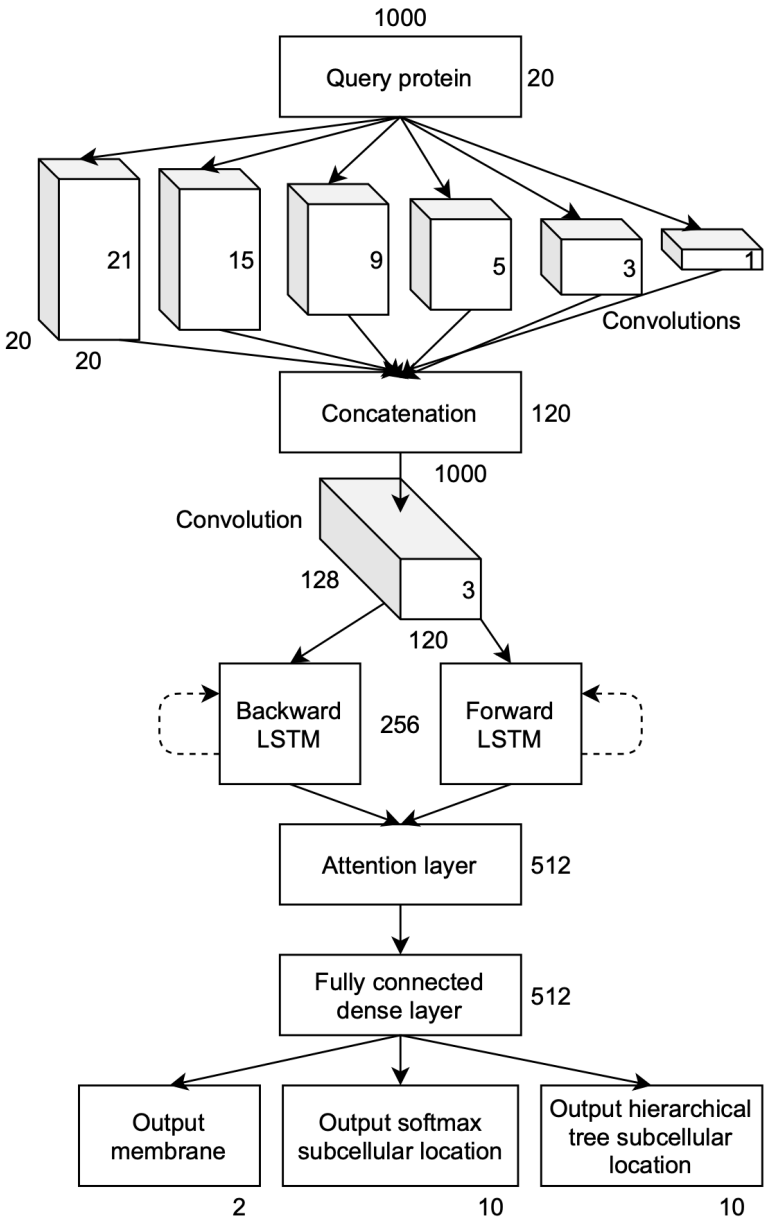


Experiment: models comparison (cont.)

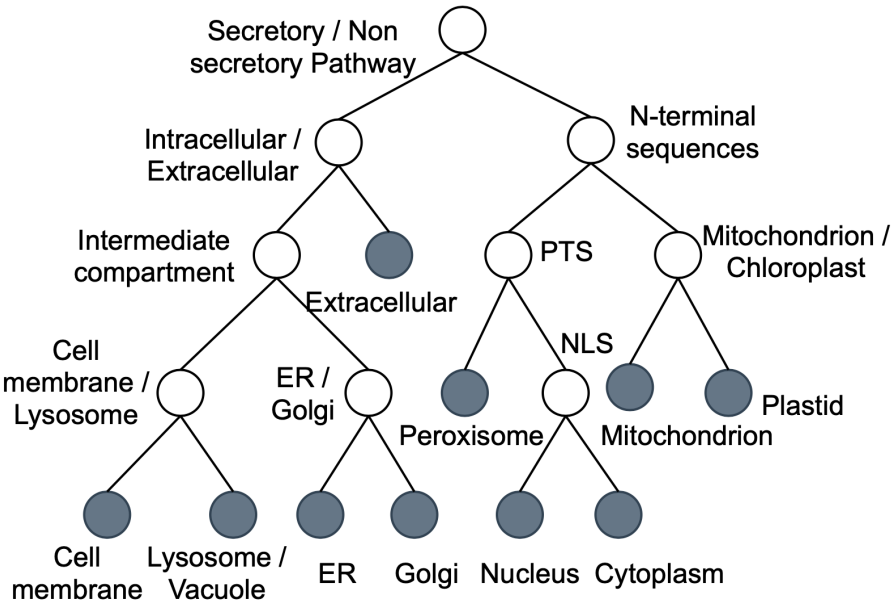




Experiment: Complete network



- Training:
  - 4-fold validation for softmax,
  - 4-fold validation for hierarchical tree.
  - Hyperparameters of each fold chosen with Random + Bayesian search.
- **Hierarchical tree**:

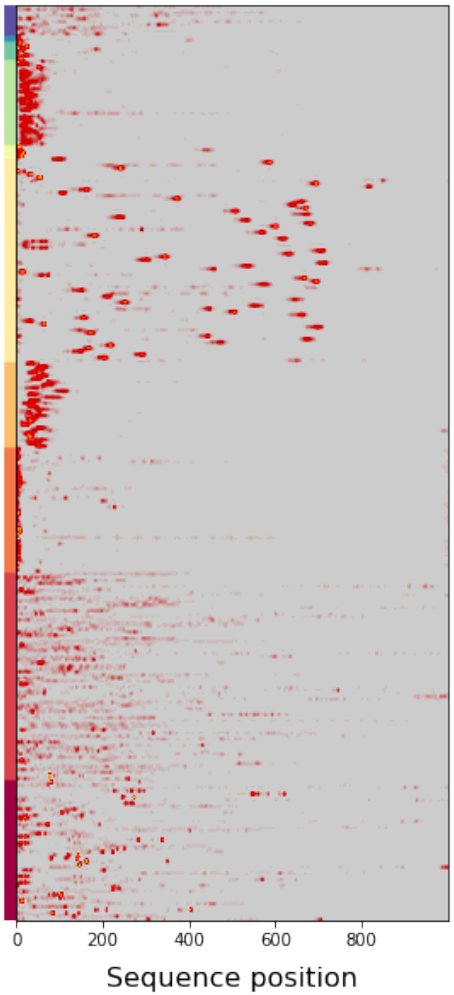


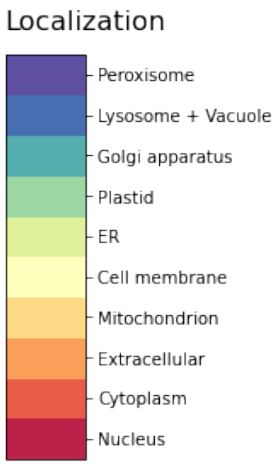
Experiment: Complete network (cont.)

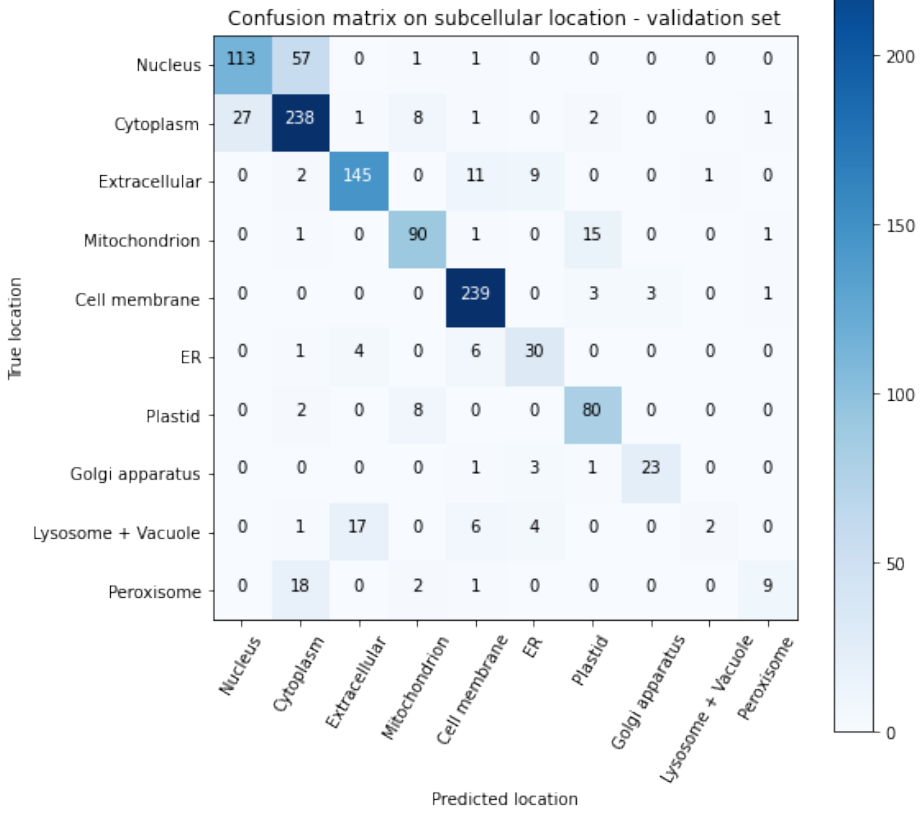
		Softmax						Hierarchical tree					
		Subcellular			Membrane			Subcellular			Membrane		
		Accuracy	Loss	Gorodkin	Accuracy	Loss	MCC	Accuracy	Loss	Gorodkin	Accuracy	Loss	MCC
Paper	Random	0.708	0.799	0.658	N/A	N/A	N/A	0.681	0.876	0.621	N/A	N/A	N/A
	Random + Bayesian	**0.757**	0.690	0.724	N/A	N/A	N/A	0.750	0709	0.713	N/A	N/A	N/A
PSI-Blast	Random	0.505	1.552	0.315	0.789	0.446	0.468	0.502	1.551	0.317	0.801	0.454	0.469
	Random + Bayesian	**0.546**	1.439	0.299	0.809	0.411	0.422	0.550	1.447	0.313	**0.811**	0.406	0.459

- **Matthews correlation coefficient (MCC)** [14] measures the quality of a binary classification. Correlation observed / predicted:
  - +1 perfect prediction,
  - 0 no better than random prediction,
  - -1 total disagreement between prediction and observation.
- **Gorodkin** [15] is a generalization of MCC that applies to k categories. More informative on class imbalance.

Experiment: Complete network (cont.)







Experiment: DeepLoc vs Multiloc

Generalization experiment:

- Both datasets encoded using PSI-BLAST.
- Four different trainings using hyperparameters from random search.
- The paper showed:
  - Multiloc achieving good performances only on Multiloc itself.
  - DeepLoc performing good even on Multiloc test set.

Training set	Test set	Accuracy	Gorodkin
DeepLoc	DeepLoc	0.475	0.355
DeepLoc	Multiloc	0.450	0.344
Multiloc	DeepLoc	0.446	0.327
Multiloc	Multiloc	0.710	0.658

Experiment: ELMo

Natural Language Processing approach (**NLP**):

- Consider proteins as sentences split into n-grams with a given stride.
- Swiss-prot and UniRef50 [16] to pre-train ELMo.
- Obtain embedding of DeepLoc.
- Subcellular localization using contextualized word vectors.

Embedding from Language Model (ELMo)

ELMo [17] is a way to represent words in vectors or embeddings. The bidirectional language model (**biLM**) which uses two BLSTM in its architecture, allows to obtain a contextualized representation of each word.

Experiment: ELMo (cont.)

Steps:

- Installation of the Tensorflow version of **biLM** [18] for computing ELMo representations.
- The network is initialized using pre-trained weights,
  - obtained with unsupervised learning on Swiss-prot database, using different combinations of stride and n,
  - or using ready-made weights learned from UniRef50 [16].
- Are now computed the ELMo representations of DeepLoc.
- Used to train CNN-LSTM-Attention network.

Experiment: ELMo (cont.)

Perplexity

Perplexity metric in NLP is a way to capture the degree of *uncertainty* a model has in predicting (assigning probabilities to) some text.

Results obtained with different pre-train data:

					Measures			
Model	**n**	**Stride**	**Perplexity**	**Pre-train time**	**Subcellular accuracy**	**Membrane accuracy**	**Gorodkin**	**MCC**
UniRef50	1	1	10.5	3w	-	-	-	-
UniRef50+Swiss+DeepLoc	1	1	15.1	3w+24h	**0.771**	**0.904**	**0.286**	**0.789**
Swiss + DeepLoc	1	1	17.6	24h	0.566	0.815	0.249	0.559
Swiss + DeepLoc	2	1	18.2	24h	0.599	0.825	0.270	0.622
Swiss + DeepLoc	2	2	290.4	24h	0.644	0.857	0.263	0.696
Swiss + DeepLoc	3	1	23.7	24h	0.484	0.761	0.149	0.384
Swiss + DeepLoc	3	2	342.7	24h	0.633	0.846	0.248	0.595
Swiss + DeepLoc	3	3	5044.2	24h	0.608	0.844	0.268	0.660

Conclusions and future works

- Improve proteins encoding using different combinations of various techniques as BLOSUM64, PSI-BLAST, HSDM. This could lead to,
  - Better results in accuracy and loss.
  - Better generalization of our network trained on DeepLoc.
- A much longer training on ELMo with a bigger dataset as UniRef50 or TrEMBL.
  - This hopefully would lead to better performances.
- Try other combinations of n-grams and stride for the embeddings.

Thanks! Any questions?

Bibliography

[1] Jose Armenteros et al. "DeepLoc: prediction of protein subcellular localization using deep learning". In: *Bioinformatics* 33 (Sept. 2017). DOI: 10.1093/bioinformatics/btx548.

[2] Abdullah Chaudhry Claudia Raffaelli. *Protein subcellular localization*. URL: https://github.com/ClaudiaRaffaelli/Protein-subcellular-localization.

[3] Matthew E. Peters et al. "Deep contextualized word representations". In: *Proc. of NAACL*. 2018.

[4] Khan Academy. *DNA to RNA to protein*. URL: https://www.khanacademy.org/science/biology/gene-expression-central-dogma/translation-polypeptides/v/rna-transcription-and-translation.

[5] Henrik Nielsen et al. "A Brief History of Protein Sorting Prediction". In: *The Protein Journal* 38 (June 2019). DOI: 10.1007/s10930-019-09838-3.

Bibliography (cont.)

[6] Stephen Altschul et al. "Gapped blast and psi-blast:A new generation of protein database search programs". In: *Nucl. Acids. Res.* 25 (Nov. 1996), pp. 3389–3402.

[7] Yi-Ze Zhang and Hong-Bin Shen. "Signal-3L 2.0: A Hierarchical Mixture Model for Enhancing Protein Signal Peptide Prediction by Incorporating Residue-Domain Cross Level Features". In: *Journal of chemical information and modeling* 57 (Mar. 2017). DOI: 10.1021/acs.jcim.6b00484.

[8] Marcos Parras et al. "Classification of protein motifs based on subcellular localization uncovers evolutionary relationships at both sequence and functional levels". In: *BMC bioinformatics* 14 (July 2013), p. 229. DOI: 10.1186/1471-2105-14-229.

[9] Søren Kaae Sønderby et al. "Convolutional LSTM Networks for Subcellular Localization of Proteins". In: *Lecture Notes in Computer Science* (2015), pp. 68–80. ISSN: 1611-3349. DOI: 10.1007/978-3-319-21233-3_6. URL: http://dx.doi.org/10.1007/978-3-319-21233-3_6.

Bibliography (cont.)

[10] Jose Armenteros et al. *DeepLoc dataset*. URL: http://www.cbs.dtu.dk/services/DeepLoc/data.php.

[11] Annette Höglund et al. "MultiLoc: prediction of protein subcellular localization using N-terminal targeting sequences, sequence motifs and amino acid composition". In: *Bioinformatics (Oxford, England)* 22 (June 2006), pp. 1158–65. DOI: 10.1093/bioinformatics/btl002.

[12] Michael Gribskov, A.D. McLachlan, and D. Eisenberg. "Profile analysis: detection of distantly related proteins. Proc NatlAcadSci USA84(13), 4355-8". In: *Proceedings of the National Academy of Sciences of the United States of America* 84 (Aug. 1987), pp. 4355–8. DOI: 10.1073/pnas.84.13.4355.

[13] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. 2016. arXiv: 1409.0473 [cs.CL].

Bibliography (cont.)

[14] B.W. Matthews. "Comparison of the predicted and observed secondary structure of T4 phage lysozyme". In: *Biochimica et Biophysica Acta (BBA) - Protein Structure* 405.2 (1975), pp. 442–451. ISSN: 0005-2795. DOI: https://doi.org/10.1016/0005-2795(75)90109-9. URL: https://www.sciencedirect.com/science/article/pii/0005279575901099.

[15] J. Gorodkin. "Comparing two K-category assignment by a K-category correlation coefficient". In: *Computational biology and chemistry* 28 (Dec. 2004), pp. 367–74. DOI: 10.1016/j.compbiolchem.2004.09.006.

[16] Michael Heinzinger et al. "Modeling aspects of the language of life through transfer-learning protein sequences". In: *BMC bioinformatics* 20.1 (2019), p. 723.

[17] Matthew E. Peters et al. *Deep contextualized word representations*. 2018. arXiv: 1802.05365 [cs.CL].

[18] AllenAI. *Bidirectional language model*. URL: https://github.com/allenai/bilm-tf.