A Novel Computational Model to Classify Subcellular Protein Localizations

Kevin Hu

Introduction

Proteins are the building blocks of all life. There are an estimated 20 000 different types of human proteins, which perform specific tasks within the cell's organelles and perform the necessary functions to build, repair, and give signals needed for the body to survive. However, scientists have barely scratched the surface of the human proteome - the majority of proteins have yet to be catalogued into a classification system. In 2018, consumer medications targeted a meagre 672 proteins - just over 3%. Additionally, these targeted proteins can occur in multiple areas of the body, causing unintended damage to non-targeted areas in the form of side effects. A more secure grasp of the human proteome would allow pharmaceutical drug firms to develop more efficient, protein-specific medications. Doctors could also make more effective diagnoses by isolating malfunctioning proteins and predicting the early onset of disease.

The Human Protein Atlas (HPA) is the largest initiative that seeks to catalogue every protein into a comprehensive protein map of the human body. Some information the atlas provides is the gene that expresses the protein, the tissue type it is expressed in, the protein function, and the antibodies that target it. A key piece of information to develop such an analysis is finding out the subcellular localization of the protein in question, namely where specifically in the cell it is found. This is because the protein's function and behaviour are primarily dependent on where it occurs; knowing the subcellular localization would allow the protein to be much more easily isolated for further study.

With advances in high-throughput confocal microscopy, the atlas also contains a rapidly growing collection of high-quality biomedical images of cells tagged with antibodies to highlight certain key features. Manually sorting through these images to determine subcellular localizations proves to be increasingly obsolete and inaccurate, however, if the process was automated, it would accelerate the understanding of human cells and disease.

Previous attempts at automating the process of subcellular localization include the analysis of DNA and amino acid sequences, and machine learning algorithms such as decision trees, and support vector machines; however, these approaches have only worked for a subset of all the possible subcellular locales and do not take into account proteins with multiple subcellular localizations. This only gives a fragmented glimpse of a small part of the interconnected cellular processes. To allow for a fuller view of these processes, attempts at an overall classifier have used machine learning combined with massive-scale citizen science, where many volunteers willingly helped to classify data within a video game. While transfer learning is a well-suited, modern approach to image classification, having non-experts classify this biomedical image data leads to greater risk of the mislabelling of data, creating disjunction and bias in image classification.

Provided sufficient data, convolutional neural networks (CNNs) have had unparalleled success with image and scene recognition. They are based on the principle of feature extraction - an image is split into many sections and passed through many layers of mathematical operations to isolate and identify certain features. CNNs have large architectures and millions of parameters, making runtime and optimization prevalent issues in developing an effective model. Examples of convolutional neural networks include VGGNet, DenseNet, GoogleInceptionNet, ResNet, and ResNeXt. These networks are often pre-trained on large standard image databases; transfer

learning can then be applied to re-train these networks on a new dataset. They are also capable of multiclass and multilabel classification - multiclass being a type of classification problem where there are more than two categories to sort an image into, and similarly multilabel where the image can sorted into multiple classes at once. This combination makes CNN's a powerful tool and one well-suited to this biomedical image analysis problem.

Open-source image data is available for download from both Human Protein Atlas and Kaggle, an open source community for data scientists. The information is highly standardized and contains over 120 000 samples: each image in the HPA Cell Atlas is split into four channels, acquired sequentially with a Leica SP5 confocal microscope (Figure 1). They depict twenty-seven morphologically different cell types that are immunofluorescently labelled for one protein of interest (green) and three subcellular location reference markers (blue for the nucleus, yellow for the endoplasmic reticulum, and red for the microtubules). The HPA defines a total of 28 possible subcellular locations, or classes (Figure 2), in which the protein can occur, making it a multiclass problem. Furthermore, some proteins can have multiple subcellular locations, or labels, making the problem multilabel as well.

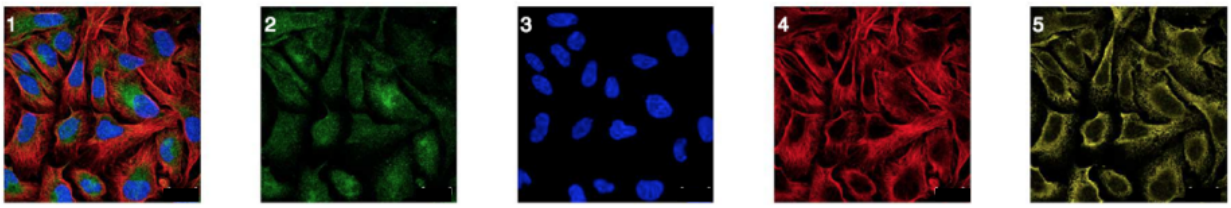

Figure 1: An RGBY image and its four component channels

Objective

The objective of this project is to develop a robust, generalized algorithm capable of multilabel, multiclass classification of all the possible subcellular localizations of unknown proteins from fluorescent microscope image data.

The model must address several key challenges: the first is four-channel input (the dataset images are in RGBY format, which is an uncommon input format), extreme class imbalance (some classes being far more prevalent than others), and complexity arising from a multiclass-multilabel problem. The successful model could then be used to determine the subcellular locations of an antibody-tagged fluorescent protein. This model would be key to develop an information profile on the protein and to enable it to be studied in further detail.

Procedures:

All instances of the algorithm were implemented and tested on a Google Cloud platform. During the building of the algorithm, numerous parameters were shifted and tuned to improve model accuracy, namely learning rates, training time, dropout, RGB vs RGBY input, image size/resolution, data augmentation, different loss functions, and various forms of cross-validation strategy. Stacking the best results created a machine learning framework optimized to identify subcellular protein localizations.

Stage 1: Collecting and preprocessing image data:

Data from the HPA and Kaggle were loaded into a scientific Python 3 development environment and processed using standard libraries including OpenCV for image recognition, Pytorch for deep learning, and Scikit-learn for data processing. The original 2048x2048 images

were saved in a TIFF format and were all resized to 800x800 pixels to reduce computational time and noise in the data samples. The less-represented classes in the dataset were oversampled (copies of the images were created) to mitigate the data imbalance problem. Additionally, the dataset was augmented (translated/rotated/flipped) to create new training examples, which yielded an increase in the number of training images and relevant data. This, in turn, increased model performance.

Stage 2: Developing the computational model

The base classification model made use of a pre-trained CNN for transfer learning. The network used was a 34-layers-deep residual neural network (ResNet34). ResNet is a recent type of neural network that is capable of both multiclass and multilabel problems, making it well suited to the problem. This network architecture was observed to achieve higher accuracy and shorter runtime than other CNN architectures. During training, the input for the model was the dataset of RGBY images as well as a CSV file containing labels mapping to each image in the dataset.

To address the unique RGBY image input, the ResNet's input layer parameters were adjusted to initialize a new channel with new weights in order to accept four channels instead of three (RGB). Additionally, to address the problem of data imbalance, the focal loss function was used to deal with more rarely-occurring classes (e.g. endosomes, lysosomes). The framework's optimal learning rate of 0.1 was calculated using the Adam optimizer and gradually changed over a training time of 24-epochs (the dataset was iterated through 24 times). A dropout rate of 50% (randomly ignoring a subset of connections in the network) was used in the network to reduce redundant neural connections and drastically speed up runtime of the model.

Stage 3: Evaluating model performance

The performance of the machine learning model was evaluated using multiple test sets and validated using a combination of metrics: F1 score, and accuracy. To avoid overfitting the model and improve the generalization of predictions, 10-fold stratified cross-validation was used to split the training dataset into 10 smaller, equally-sized validation sets containing an equal representation of each class. Additionally, test-time augmentation (TTA) was used to augment the validation data by applying geometric transformations to the images, allowing for errors in identification to be smoothed out for more accurate classification of images. F1 score then took a weighted average of a confusion matrix (containing true positive, false positive, false negative, and true negative predictions) to generate a score representative of the model's accuracy. Finally, the model was also tested on two additional datasets comprising a total of 120 000 extra images which remained completely disjoint from the training set.

Results and Discussion:

The resulting framework after training takes input through four channels - one for each of the RGBY image filters - and produces a CSV file with predicted labels for the aforementioned images (Figure 3). The model succeeded in reaching an average F1 score of 79% on the validation sets and an average of 52% on two completely unknown test sets of over 120 000 additional images. This performance is 10% better than the current state-of-the-art classifier, which scored 72% on validation sets and had no additional testing. The model additionally

eliminates the need for potentially flawed citizen science classification and sets a new precedent for the models attempting to classify the entirety of the human proteome.

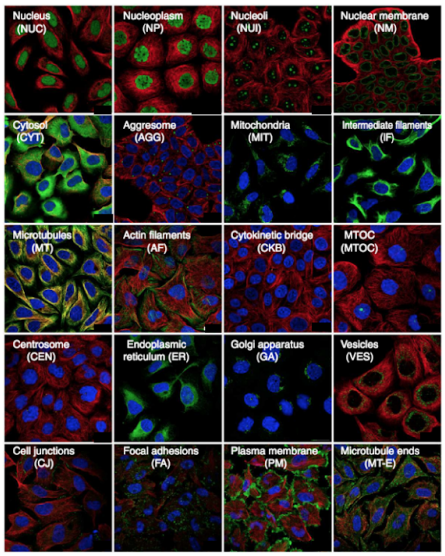

Figure 2: a subset of the 28 possible subcellular localizations

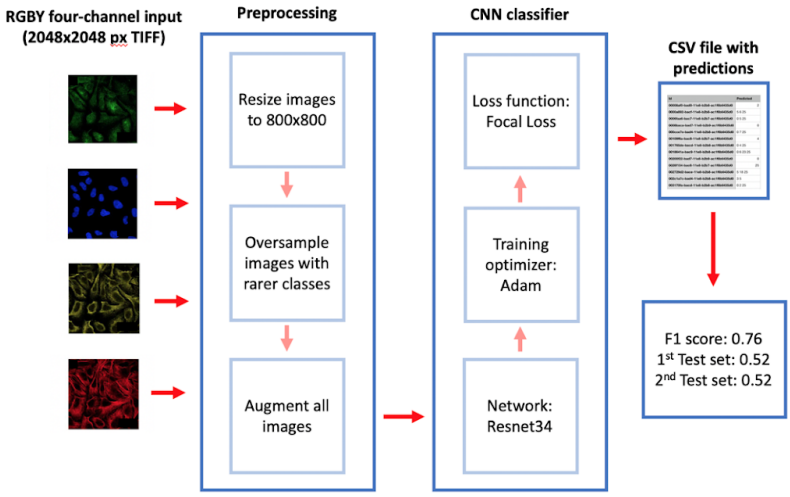

Figure 3: a simplified version of the machine learning framework

The 800x800 TIFF images were seen to be an optimal size when measuring results (Figure 4), with enough reduction in noise yet still retaining detail. Other image sizes tested were 256x256 PNG, 512x512 PNG, and 512x512 TIFF. The PNG images had noticeably lower resolution and had lower scores.

The dropout rate of 50% was consistently more effective than both lower and higher dropout rates across different resolution levels, thus it is likely that close to half of the network's connections/weights carry redundant information (Figure 5). The learning rate of 0.1 as calculated by the Adam optimizer proved to be more effective than others as well (Figure 6).

Additionally, using the Focal Loss method compared to more standard loss functions for transfer learning helped greatly. Unlike standard loss functions such as binary cross-entropy loss, the focal loss function worked very well with the unbalanced cell images dataset - this is likely due to rarer subcellular classes appearing once per every thousand instances of a more common class. However, these rarer classes still remained a problem, and are the main reason why the model's F1 and test accuracy scores remain lower - when constructing an overall classifier that took multi-localization into account, certain rarer classes could have been overlooked due to a lack of training examples and the algorithm being more focused on classifying the prominent features of more common classes such as nucleoplasm. Particularly rare classes were endosomes, lysosomes, rods & rings, and liquid droplets. Without these classes, the model's F1 score reached the mid-80s range.

The model's results ascertain that universal image classification of all subcellular protein localizations is an intricate and difficult task, with model improvement being very incremental especially when compared to classification of only a subset of localizations.

Figure 4: comparing image resolutions

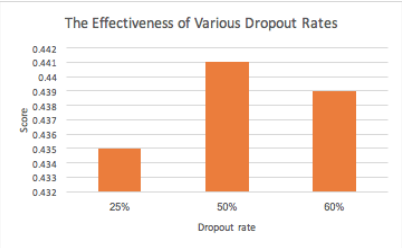

Figure 5: comparing dropout rates

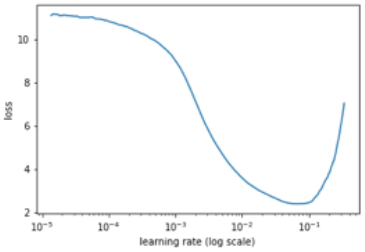

Figure 6: graphing the optimal learning rate

Conclusion

In conclusion, a novel machine learning algorithm was developed to automate the process of classifying subcellular localization of proteins. By tweaking many training parameters, the algorithm achieved a 10% improvement over previous state-of-the-art results using a CNN framework combined with other custom features such as focal loss, oversampling, and augmentation. The model maintains these results without any need for additional manual processing/citizen science, making it less prone to human error than currently existing models. Additionally, it can easily accommodate new features for greater performance, such as more advanced image preprocessing methods and further training optimizations.

By using the algorithm the 20 000 existing human proteins could be thoroughly mapped and researched at an ever accelerating rate, opening up new avenues of research that were previously inhibited by a lack of knowledge of the human proteome. Such applications range from developing more target-specific medications, nullifying harmful side effects caused by mislocalization of proteins, and earlier diagnoses of disease based on knowledge of protein localization behaviour. As proteins are a fundamental aspect of life and its vast set of processes, the algorithm's ability to decode key information about them gives it an incredibly wide range of application across all protein-related issues.

Future directions

A potential method to greatly boost the accuracy of these rarer classes could be to use metric learning as a postprocessing method. Metric learning could be used to adjust predictions of the Resnet34 network based on the ground-truth labels of similar images from the training set, allowing for rarer classes to be better classified.

Other strategies to improving model accuracy could be in preprocessing: More advanced preprocessing methods (e.g. AutoAugment) have the potential to extract more useful information out of raw image inputs.

Additionally, further training optimization such as averaging the predictions of several models, increasing training time or further optimizing training parameters such as learning/dropout rates could help model accuracy to rise several percent, which would be a substantial increase for a problem as incremental as subcellular protein localization.

Acknowledgements

I would like to thank Professor Forbes Burkowski of the University of Waterloo for his guidance and valuable discussions throughout the project. Furthermore, some computations were performed on a High-Performance Computing GPU cluster owned by SharcNet at the University of Waterloo.

References

Sullivan, D. P., Winsnes, C. F., Åkesson, L., Hjelmare, M., Wiking, M., Schutten, R., ... & Smith, K. (2018). Deep learning is combined with massive-scale citizen science to improve large-scale image classification. *Nature biotechnology*, *36*(9), 820. https://www.kaggle.com/c/human-protein-atlas-image-classification, www.proteinatlas.org

Coelho, L. P., Peng, T., & Murphy, R. F. (2010). Quantifying the distribution of probes between subcellular locations using unsupervised pattern unmixing. *Bioinformatics*, *26*(12), i7-i12.

Boland, M. V., & Murphy, R. F. (2001). A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of HeLa cells. *Bioinformatics*, *17*(12), 1213-1223.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Huang, G., Liu, S., Van der Maaten, L., & Weinberger, K. Q. (2018). Condensenet: An efficient densenet using learned group convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2752-2761).

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980-2988).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research*, *12*(Oct), 2825-2830.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Bellet, A., Habrard, A., & Sebban, M. (2013). A survey on metric learning for feature vectors and structured data. *arXiv preprint arXiv:1306.6709*.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., & Le, Q. V. (2018). Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*.

Bibliography

Baltruschat, I. M., Nickisch, H., Grass, M., Knopp, T., & Saalbach, A. (2018). Comparison of deep learning approaches for multi-label chest X-ray classification. *arXiv preprint arXiv:1803.02315*.

Pärnamaa, T., & Parts, L. (2017). Accurate classification of protein subcellular localization from high-throughput microscopy images using deep learning. *G3: Genes, Genomes, Genetics*, *7*(5), 1385-1392.

Hamilton, N. A., Pantelic, R. S., Hanson, K., & Teasdale, R. D. (2007). Fast automated cell phenotype image classification. *BMC bioinformatics*, *8*(1), 110.

Howard, J., etc. (2018). Fastai: https://github.com/fastai/fastai

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., ... & Lerer, A. (2017). Automatic differentiation in pytorch.