Integration of a Multi-Head Attention Transformer Network with pCoMiC: Predicted Classification of Mutations in CFTR

Connor Clark-Baba - 20830237

University of Waterloo

BIOL 465

April 19, 2024

**1.0 Introduction**

The Cystic Fibrosis Transmembrane Conductance Regulator (CFTR) protein is located on the surface of the lungs and regulates transepithelial salt and water transport by controlling the permeability of chloride ions through its porous channel1. The CFTR protein is a member of the ATP-binding cassette (ABC) transporter superfamily and is the sole member which transports chloride ions alongside being the only ligand-gated ion channel which consumes its own ligand (ATP) during normal function2. In healthy individuals, it consists of 1,480 amino acids spanning 5 primary domains – two membrane spanning domains (MSDs), two nucleotide-binding domains (NDB1 and NBD2) and a regulatory (R) domain1,3.

Mutations in the CFTR protein that have been associated with cystic fibrosis (CF) have been broadly grouped into 5 classes based on the resulting levels of functional protein and differing phenotypes4. Class I depicts a protein production mutation, interfering with regular production of the CFTR protein and typically arising through nonsense mutations or mutations resulting in non-standard splicing isoforms4. Class II mutations, the most common, are protein processing and are characterized by improper folding giving rise to a non-functional 3-dimensional conformation4. Class III are deemed gating mutations where the mechanisms controlling the opening and closing of the protein’s ion channel are disrupted, particularly the binding and phosphorylation of ATP, preventing chloride transport altogether5. Class IV are conduction mutations which alter the properties of the interior of the CFTR channel such as the relative charge making the passive flow of chloride ions considerably more difficult and reducing efficiency4,5. Finally, class V mutations are called insufficient protein mutations whereby the CFTR protein will likely function as intended though is produced at significantly lower rates than normal or degraded at rapid rates4.

Depending on the type of mutation an individual possesses, different forms of treatment are required. Traditional mutation classification methods require the collection of blood, sweat or saliva samples and relatively extensive *in vitro* testing6,7. As such, the pipeline, predicted Classification of Mutations in CFTR (pCoMiC) was developed to limit the need for substantial testing and to quickly determine the nature of mutations in CF patients. Briefly, the first iteration of this pipeline relied on identifying the relative change in 3 important amino acid properties for determining mutant classifications based off changes in protein conformation. Changes in hydropathy, relative charge in the human body and hydrogen bonding capability were calculated around the mutated residue and run through custom scoring functions to determine the independent probability of classification for the 5 classes. A more extensive description of the methods used can be seen in the BIOL 469 paper, “The Development of pCoMiC: Predicated Classification of Mutations in CFTR” by Clark-Baba, Tewari and Adtani attached to this submission8.

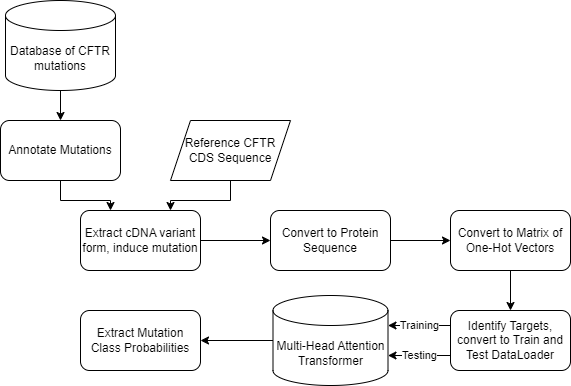
While the pipeline produced commendable results with 90.7% of 54 test samples being correctly characterized, it possessed several core limitations. The first and most important limitation is the restriction of the analysis to solely the region around the mutated residue. The pipeline failed to analyze how the mutated residue may affect interactions between domains in the CFTR protein. As a result, stringencies for classification needed to be reduced which led to many false positives with 50% of the benign mutations being classified as CF-causing. The second most glaring limitation was the inability to properly identify class V mutations by only analyzing the coding sequence (CDS) of the CFTR protein. Class V mutations typically arise due to alternative splicing events within non-coding regions splicing out important regulatory factors causing the protein to be translated at far lower rates or degraded entirely7. Of course, a pipeline needs to analyze these splicing domains on the nucleotide sequence in order to identify such mutations.

In the present work, the former limitation will be addressed. While ensuring class V mutations are properly identified is of high priority, it requires additional knowledge of alternative splicing mutations and their analysis, both outside the scope of this report. To address the lacking analysis of interactions between domains, I propose constructing and training a neural network to learn and extract complex features from mutated protein sequences. Key considerations for determining the optimal architecture include the type of neural network and its depth. Indels and nonsense mutations will inevitably alter the length of the protein sequence meaning input length to the neural network will not be consistent. As such, a recurrent neural network (RNN) and a multi-head attention transformer (MHAT) were considered.

Both RNNs and MHATs can accept varying input lengths by padding the data without losing any information or adding additional background noise for the network to filter9. For RNNs, this requires the use of a variable rate RNN, described by Gao and colleagues10. However, RNNs process sequential information in series (i.e. one position after another) leading to considerably higher computational costs and the inability to consider all positions in the sequence simultaneously, failing to capture the relationship between elements distant from one another10,11. Conversely, MHATs process information in parallel and consider the entire sequence at once, far more attractive features for the current classification problem11,12.

**2.0 Methods**

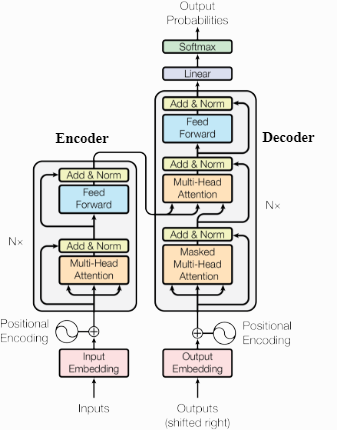
2.1 Pipeline Workflow



**Figure 1**: *Predicted Classification of Mutations in CFTR (pCoMiC) workflow with multi-head attention transformer network.*

The workflow for the pipeline is highlighted in Figure 1. Briefly, mutations from an extensive database obtained from Clinical and Functional Translation of CFTR (CFTR2) were annotated with their associated class13. Mutation annotations were derived from information in the CFTR2 database and various submissions in ClinVar Miner14. The cDNA variant form was processed allowing mutations to be induced in a reference CFTR CDS, obtained from NCBI GenBank, through a custom function15. Mutated CDS’ were converted to protein sequences then to matrices of one-hot vectors. Values in a one-hot vector take on the value of either a 1 or a 0. In this case, each vector is of length 20 and whichever position the 1 appears in indicates which residue the vector is representative of corresponding to the alphabetical order of their one-letter codes. The matrix was 1480x20 as the CFTR protein is 1480 amino acids long and 20 different amino acids exist. For mutations decreasing the length, the matrix was padded with rows of 0’s until a length of 1480 was reached. For mutations increasing the length, rows were removed until a length of 1480 assuming the mutated residue was not located in rows subject to removal. Targets were identified based off the previous manual annotations and represented as multi-hot vectors of length 5 where a 1 indicates the mutation is known to be of class index + 1. Training and testing subsets were derived from 80% and 20% of the original dataset, respectively. Singleton classes, that is, classes which are only represented once, were automatically filtered into the testing set. The MHAT was trained for 200 epochs at a learning rate of 0.001 using the NAdam optimizer from PyTorch and the binary cross-entropy loss function. The testing set was then passed through the MHAT and both training and testing accuracies were recorded. Finally, mutation class probabilities for both datasets were extracted.

2.2 Neural Network Architecture



**Figure 2:** *The multi-head attention transformer network used was adapted from the basic architecture shown by Vaswani and colleagues12.*

Figure 2 displays the basic architecture for a multi-head attention transformer network, as described in the paper, “Attention is All You Need”, by Vaswani and colleagues12. In the encoder (Fig. 2, left), the input protein sequence is initially represented as embeddings where each residue is mapped to a higher-dimensional space, capturing all features associated with said position. The position of each residue in the sequence is added to each embedding matrix through the positional encoding function. The multi-head attention layers learn specific types of attention patterns from the input sequence. The purpose of the attention layers can be briefly described as mapping query, key and value vectors to a specific output where the output represents the weighted sum of values. Such weights are computed by determining the compatibility between queries and keys, essentially, the query determines how much attention a given residue should pay towards other residues based off their keys. The association between residues is learned in this sense. For example, if the query from residue X computes a high compatibility with the key from residue Y, the value from residue Y will be weighted more heavily when computing the value for residue X. The raw input to the multi-head attention layer is then added to its output and normalized across the feature dimensions before being sent through a basic feed-forward network.

For the decoder (Fig. 2, right), the same basic methods are applied though it is instead initialized to the output from the encoder rather than the input sequence. The two networks are concatenated together before being flattened and sent through a linear (dense, fully connected) layer and finally sent through the Softmax activation function to provide probability outputs. The description provided of the basic MHAT is significantly reduced and the original paper by Vaswani and colleagues should be referenced for a more extensive description12.

In the present model, an embedding size of 128, 2 transformer layers for both decoder and encoder with 8 attention heads each and a feedforward layer of size 256 are used. Three final linear layers are included, and the sigmoid activation function is used rather than Softmax given that the probabilities should be independent of one another.

2.3 Testing pCoMiC

To test pCoMiC, the same 54 known mutations used in the previous iteration will again be tested. The class representations are listed in table 1. Mutations were predicted to be in a class if pCoMiC output a probability of >60%. Initially, classifications were only deemed correct if the output vector exactly matched the target vector. That is, for a target vector of [0 1 1 0 0], indicating a mutation of classes II and III (protein processing and gating), correct classification required an output vector of [<60 >60 >60 <60 <60]. This was later revised where a predicted class which overlapped with the expected classes in any fashion was indicative of a correct classification. For the same target vector, if the predicted probability vector was any combination including class II or III, classification was deemed correct due to the overlapping nature of mutations (Ex. [0 1 0 0 0] and [0 0 1 1 0] would be deemed correct)7.

**Table 1.** *Distribution of mutation classes in the dataset, 54 known mutations are represented. Note that the total mutations number exceeds 54 as several mutations fall under more than one class.*

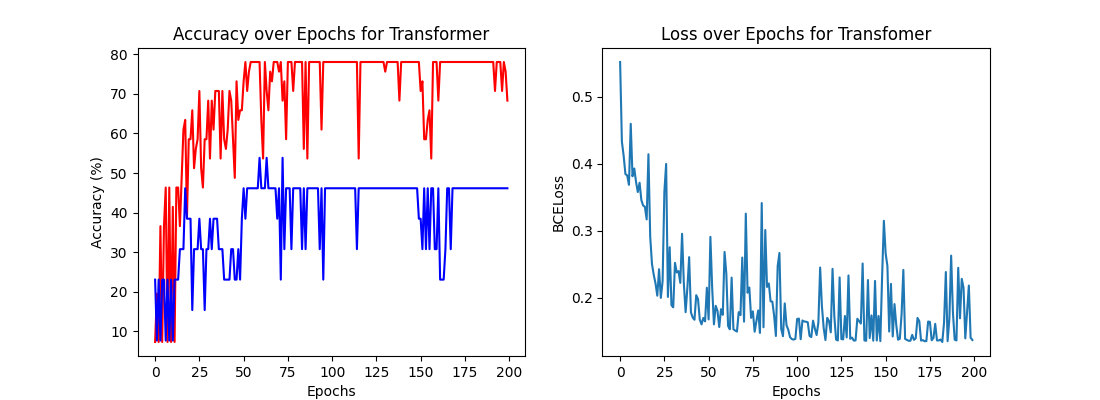
|  |  |
| --- | --- |
| Mutation Class | Number in Dataset |
| Class I | 22 |
| Class II | 20 |
| Class III | 14 |
| Class IV | 8 |
| Class V | 0 |
| Benign | 4 |

2.4 Data and Code Availability

The entire pipeline was coded in Python (v3.11.5) and all data and code used are available upon request. A draft version of the pipeline may be available at <https://github.com/cclarkba>, depending on if the files have been cleaned sufficiently.

**3.0 Results**

3.1 Model Accuracy and Semi-Supervised Learning Rules

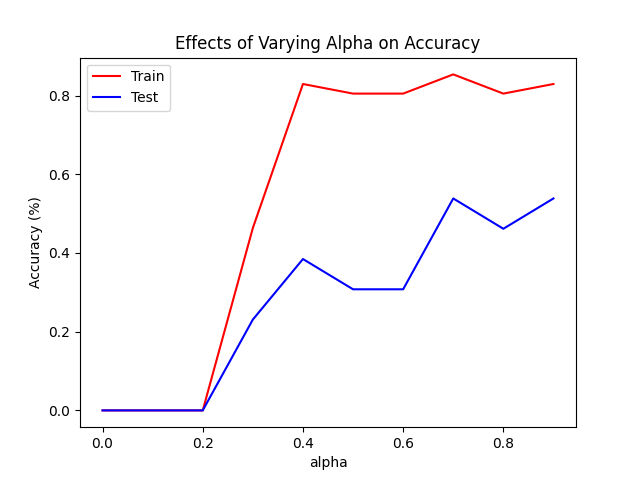


**Figure 3.** *Accuracy (left) and learning (right) curves for the MHAT network employed. Red depicts the training accuracy and blue the testing accuracy. The model was trained for 200 epochs with BCELoss.*

The MHAT network output mediocre results with a final training accuracy of 68.29% and testing accuracy of 46.15%, though the final average training accuracy appears to reach closer to 80% (Fig. 3). From the learning curve in figure 3, the model was rather volatile though consistently displayed a downwards trend. Additionally, the accuracy curve from figure 3 displays rather significant overfitting of the model to the training dataset as shown through the clear divergence in the two curves. Since the MHAT network is rather large while the training dataset consisted of only 41 samples, overfitting was expected.

To combat the effects of overfitting, a semi-supervised learning rule was employed. Here, 80 additional unlabeled samples were included during training and the weighted effect from the losses of either network on the combined loss were varied through the parameter, *alpha* (1).

[1]



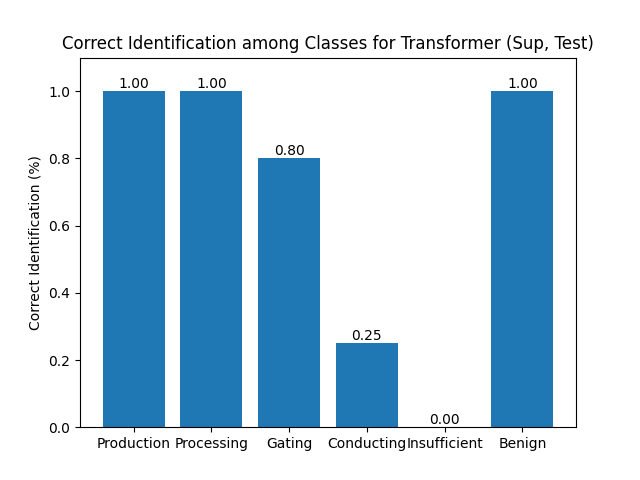
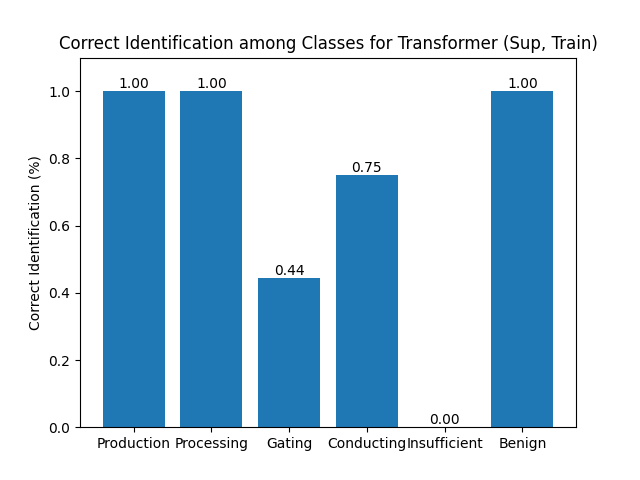
**Figure 4.** *The effects of varying the influence of the unlabeled and labeled training sets during semi-supervised learning of the MHAT model through the parameter, alpha.*

Optimal values of *alpha* were determined to range from 0.7-0.9 indicating moderate influence from the unlabeled dataset. It is possible that including more than 80 unlabeled samples may improve this further though extremely long training times (upwards of 4 hours) and limited labeled data were the core issues. In figure 5, the model is noticeably less volatile as shown through increased smoothness and infrequent spiking in the learning curve (Fig. 5, right). There are also far fewer negative deflections in the accuracy curves, further demonstrating the beneficial effects of the semi-supervised model (Fig. 5, left). With the semi-supervised learning rule, the MHAT model attained a final training accuracy of 78.05% and testing accuracy of 46.15%, comparable to the solely supervised learning rule. Overall, the MHAT model with both learning rules performed less than satisfactorily.

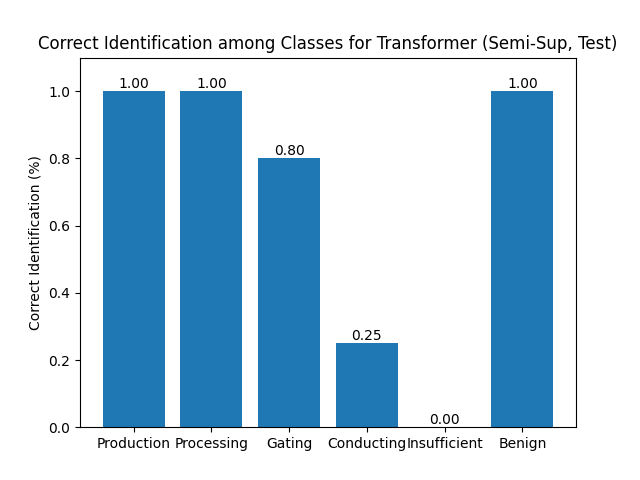
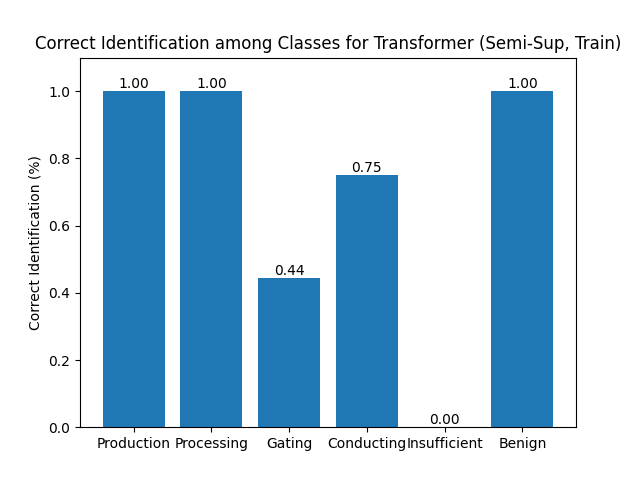
A graph of a number of different types of data

Description automatically generated with medium confidence**Figure 5*.*** *Accuracy (left) and learning (right) curves for the MHAT network employed using a semi-supervised learning rule with alpha=0.9. Red depicts training accuracy and blue the testing accuracy. The model was trained for 200 epochs with BCELoss*.

As mentioned, the accuracies presented are only indicative of a strict match between output and target vectors. Loosening the classification conditions to including any form of overlap between predicted classifications and target classes indeed improved the accuracy for both models. For both the supervised and semi-supervised learning rules, the training accuracy improved to 87.76% and testing accuracy to 78.95% with the highest rates arising from class I, II and benign mutations (Figs. 6 & 7). Interestingly, the distribution among classes were identical indicating the model likely reached maximum training capacity using both learning rules.



**Figure 6.***Correct identification among classes for the MHAT model using the supervised learning rule. Training set on the left and testing set on the right.*



**Figure 7.** *Correct identification among classes for the MHAT model using the semi-supervised learning rule. Training set on the left and testing set on the right.*

The identical distributions are likely also correlated to the rather small dataset used, where the “easier” to identify mutations were readily classified. Additionally, due to class I and II mutations being overrepresented, the model is likely better trained to classify these mutations resulting in lower rates for classes III and IV. Benign mutations appeared rather infrequently and were relatively easy to classify; if all 5 output scores were less than the threshold, the mutation was classified as Benign, explaining its exceptional classification rate. No class V mutations were analyzed for reasons previously discussed.

3.2 Alternative Neural Network Architecture

As an additional test to see if a neural network in general is a reasonable approach for classifying mutations, a simple, fully connected network was also constructed accepting a different form of input. The network consisted of 4 dense layers with 50 hidden neurons each, interspersed by the rectified linear unit (ReLU) activation function while using the same sigmoid activation as the final output. The input was adapted from the previous rendition of pCoMiC where three scores, hydropathy, relative charge and hydrogen bonding capability, were calculated in reference to the change in such characteristics between regions surrounding the mutated residue in the healthy protein sequence versus the mutated protein sequence8. Additionally, sequence length information was included as a ratio between the mutated sequence and the protein sequence for a total of 4 input nodes. As mentioned, this does eliminate the analysis of domain interaction since the full protein sequence is no longer being analyzed. Nevertheless, the network was constructed and tested to compare to the results of the MHAT model.

A graph of a network

Description automatically generated with medium confidence

**Figure 8.** *Accuracy (left) and learning (right) curves for the dense network employed. Red depicts training accuracy and blue the testing accuracy. The model was trained for 200 epochs with BCELoss*.

A graph of a network

Description automatically generated

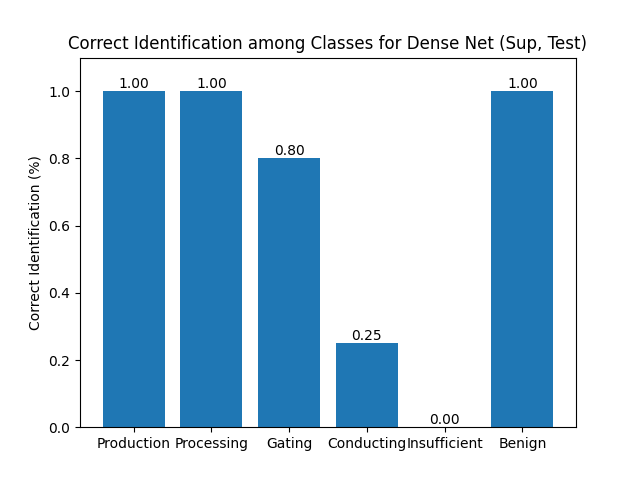
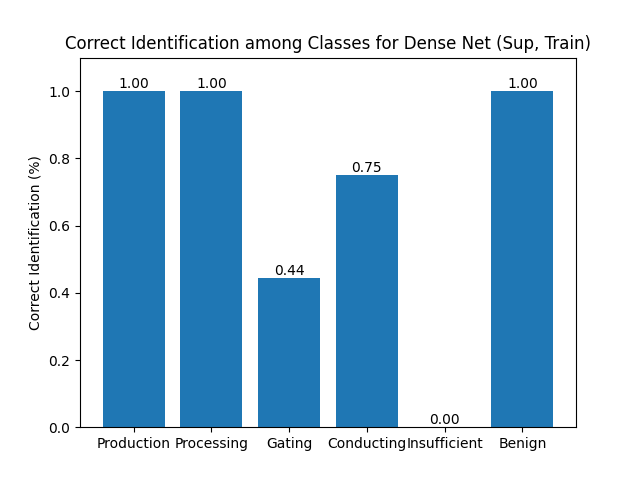
**Figure 9.** *Accuracy (left) and learning (right) curves for the dense network employed using a semi-supervised learning rule with alpha=0.9. Red depicts training accuracy and blue the testing accuracy. The model was trained for 200 epochs with BCELoss*.

Interestingly, the dense network displayed a higher training accuracy for both the supervised and semi-supervised learning rule compared to the MHAT model though the testing accuracies were the same between all 4, indicating no real improvement (Figs. 8 & 9, Table 2).

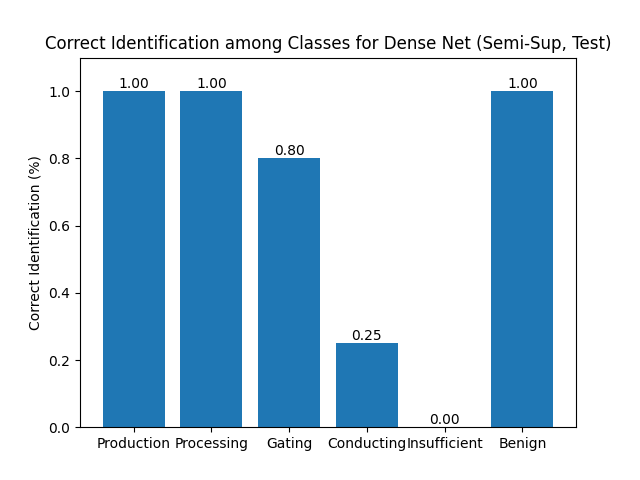
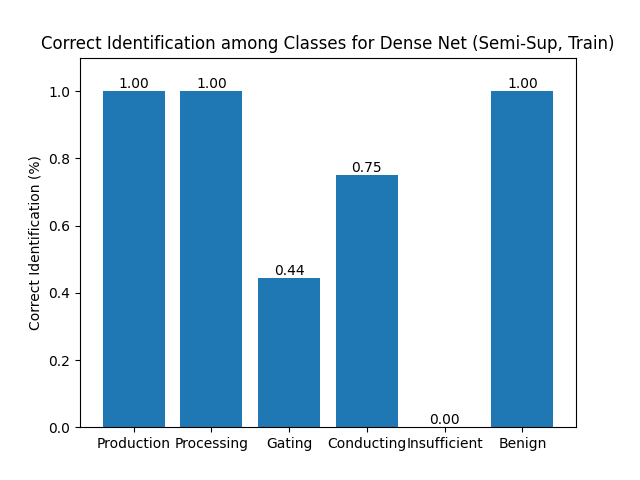
**Table 2**. *Training and testing accuracies for each model and learning rule used with strict classification conditions.*

|  |  |  |
| --- | --- | --- |
| Model and Learning Rule | Training Accuracy | Testing Accuracy |
| MHAT, Supervised | 78.05% | 46.15% |
| MHAT, Semi-Supervised | 68.29% | 46.15% |
| Dense, Supervised | 80.49% | 46.15% |
| Dense, Semi-Supervised | 82.93% | 46.15% |

Similarly to the MHAT model, semi-supervised training produced far more stable learning and accuracy curves (Figs. 8 & 9). The increased training accuracy for the dense model is rather odd though likely attributed to the linear model overfitting training data much quicker due to the lack of complexity in the input data.



**Figure 10.** *Correct identification among classes for the dense model using the supervised learning rule. Training set on the left and testing set on the right.*



**Figure 11.** *Correct identification among classes for the dense model using the semi-supervised learning rule. Training set on the left and testing set on the right.*

Both learning rules for the dense model also produced the exact same distribution in identifcation rates among classes as the MHAT model, once again indicating that the model attained its maximun training capability and further modifications to the network architecture or extension of the labeled dataset will be required to attain any noticeable improvements (Figs. 10 & 11).

**4.0 Discussion**

The MHAT model proposed produced less than satisfactory results using true classification conditions indicating it did not efficiently extract features of the protein sequence to confidently classify mutations in the CFTR protein. While the best training accuracies exceeded 80% indicating commendable classification, testing accuracies above 50% were never reported leading to inconclusive results. Using the less strict classification conditions mentioned, training accuracy remained comparable though still improved at 87.76% while testing accuracy saw a drastic improvement up to 78.95%. While, of course, higher accuracies are expected with looser conditions, an increase of upwards of 30% was rather surprising and indicates promising results. The loosened classification conditions are justified by the overlapping nature of CFTR mutations. Often, mutations which affect the 3-dimensional conformation of a protein will impact the ability of the protein to perform its proper gating function which, in turn, prevents normal flow of chloride ions3,5,7. As such, it is not uncommon for the effects of class II, III and IV mutations to appear together and, indeed, 10 of the mutations in the database used represented of combination of the three classes. Importantly, only overlap between these 3 classes was considered as class I protein production mutations will of course effect the 3-dimensional structure, gating and conducting abilities of the protein given that a non-functional protein is produced3,4.

There is still considerable room for improvement in the MHAT extension of pCoMiC, specifically regarding the interpretation and availability of input data. Essentially all mutations in the current dataset had to be manually annotated, limiting the labeled dataset to only 54 known mutations, 41 of which were used to train the model. The CFTR2 database includes over 800 mutations albeit many of which are alternative splicing mutations, not currently analyzed by pCoMiC13. As such, extending the annotations of this database will be crucial for training the network to improve its accuracy. With a considerably larger training set, the efficacy of the model architecture can be assessed much better. At present, the model continuously overfits the training data, failing to capture the true patterns and hindering its ability to extract features. It is difficult to say if the current architecture is sufficient for such a daunting classification task and no further suggestions can be provided at this time.

The current input data for the MHAT model is simply the entire protein sequence converted to a representative 1480x20 matrix of one-hot vectors. While relatively easy to code, it does not offer much information for the network to extract. Many of the mutations involve a single point mutation or indel meaning the majority of input across all mutations is vastly the same. As an example, say the amino acid at position 5 is converted from a C to an A, resulting in a class II mutation (not the actual residue or mutation class). In another mutation, the amino acid at position 1205 is converted from a Q to an F producing a class III mutation. All other 1478 residues are still the same between both mutations and the only two differences in the input are 1200 positions apart from one another. While the latter issue is meant to be addressed through the positional encoding of the MHAT model, the former issue still stands, and it is very difficult for the network to extract the necessary features to differentiate between the two.

Thus, a better approach may be a combination of the two inputs presented in this report. The first approach to test would be dividing the protein sequence into subsets of around 5-15 residues and calculating their representative characteristics for each subsequence to have a total of 3 inputs. This would reduce the input to a (99-296)x3 matrix making relative changes in the input data considerably easier to deduce. A second approach would be constructing a multi-stream input neural network. Instead of accepting solely the protein sequence or the representative characteristics, both can be passed through separate networks, processed, then pooled together in some fashion to be passed through an additional fully connected network. This allows for the entire sequence to still be processed, addressing the first limitation proposed in this report, while also providing the network with more relevant information to use in classification.

Finally, the current model is relatively “black-box-like” in the sense that the features the network is learning are not obvious. To address this, visualization of the attention layers in the transformers should be performed. This will display which portions of the input are paying the most attention to one another, potentially highlighting residues or domains in the input with high compatibility indicating they likely interact with one another frequently. Additionally, a dimensionality reduction algorithm such as UMAP, t-SNE or PCA may be performed on the data to identify any potential clustering along features. However, this may be difficult to draw any correlation to the features the neural network is learning unless they are integrated in some manner. Overall, integrating the MHAT model with pCoMiC produced mediocre results, though, nonetheless, alludes to a relatively promising future for the use of neural networks in the analysis of biological data, specifically the classification of mutations.

**5.0 References**

[1] Linsdell, P. (2005). Mechanism of chloride permeation in the cystic fibrosis transmembrane conductance regulator chloride channel. *Experimental Physiology*, *91*(1), 123–129. https://doi.org/10.1113/expphysiol.2005.031757

[2] Corradi, V., Vergani, P., & Tieleman, D. P. (2015). Cystic Fibrosis Transmembrane Conductance Regulator (CFTR). *Journal of Biological Chemistry*, *290*(38), 22891–22906. https://doi.org/10.1074/jbc.m115.665125

[3] Hwang, T. C., & Kirk, K. L. (2013). The CFTR ion channel: gating, regulation, and anion permeation. *Cold Spring Harbor perspectives in medicine, 3*(1), a009498. https://doi.org/10.1101/cshperspect.a009498

[4] Cystic Fibrosis Foundation. (2022). *Types of CFTR Mutations | Cystic Fibrosis Foundation*. www.cff.org; Cystic Fibrosis Foundation. https://www.cff.org/research-clinical-trials/types-cftr-mutations

[5] Lewis, H., et al. (2004). Structure of nucleotide-binding domain 1 of the cystic fibrosis transmembrane conductance regulator. *The EMBO Journal, 23*(2), 282-293. 10.1038/sj.emboj.7600040

[6] National Heart, Lung, and Blood Institute. (2022, March 24). *Cystic Fibrosis - Causes*. www.nhlbi.nih.gov. https://www.nhlbi.nih.gov/health/cystic-fibrosis/causes#:~:text=The%20CFTR%20protein%20has%20also

[7] Veit, G., et al. (2016). From CFTR biology toward combinatorial pharmacotherapy: expanded classification of cystic fibrosis mutations. *Molecular Biology of the Cell, 27*(3), 424-433. 10.1091/mbc.E14-04-0935

[8] Clark-Baba, C., Tewari, A. & Adtani, F. (2023). The development of pCoMiC: Predicted classification of mutations in CFTR. *University of Waterloo, BIOL 469*

[9] Garg, M., Ghosh, D. & Pradhan, P. (2023). Multiscaled multi-head attention-based video transformer network for hand gesture recognition. *IEEE Signal Processing Letters, 30*(), 80-84. 10.1109/LSP.2023.3241857

[10] Gao, Y. et al. (2022). Variable rate independently recurrent neural network (IndRNN) for action recognition. *Applied Sciences, 12*(7), 3281. https://doi.org/10.3390/app12073281

[11] Reza, S., Ferreira, M., Machado, J. & Tavares, J. (2022). A multi-head attention-based transformer model for traffic flow forecasting with a comparative analysis to recurrent neural networks. *Expert Systems with Applications, 202*(15). https://doi.org/10.1016/j.eswa.2022.117275

[12] Vaswani Ashish, Shazeer Noam, Parmar Niki, Uszkoreit Jakob, Jones Llion, Gomez Aidan N., Kaiser Łukasz, and Polosukhin Illia. 2017. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems. 6000–6010.

[13] The Clinical and Functional Translation of CFTR. (2023, April 7). *CFTR2 Variant List History*. Cftr2.org. https://cftr2.org/mutations\_history

[14] University of Utah Health. (2023). *ClinVar Miner*. https://clinvarminer.genetics.utah.edu/

[15] National Institutes of Health (NIH). (2024, April 20). CFTR CF transmembrane conductance regulator [Homo sapiens (human)] [Gene]. National Center for Biotechnology Information. Retrieved from https://www.ncbi.nlm.nih.gov/gene/1080