**Deep Learning Networks Synergize Pathology and Genomics**

Currently, the integration of heterogeneous data types is actively used in completely different fields and is a promising area of research, especially relevant for medical science, opening up new opportunities for both diagnosis and a deeper understanding of diseases.

This blog explores the approach of combining various data modalities: histopathological images and gene expressions – to study associations between cellular morphology and gene expression levels.

Histological examination of tissue specimens, obtained through biopsy, remains the gold standard for diagnosis and accurate evaluation of different diseases, and most diagnostic labs all over the world still use pathology images as the first step in diagnosis. To further confirm the diagnosis, the image inspection-based features can be augmented with sequencing or molecular data. Automated image analysis and computer vision algorithms help mitigate the subjectivity between the pathologists and provide an objective method for predicting patient prognosis directly from the image data. To use images (unstructured data) and to combine image information with gene expression (structured data in the form of a matrix), we need to transform images into feature vectors (embeddings). One way to do this is to train a deep-learning network to classify images and use it for getting embeddings.

In this paper, we use this popular deep learning-based technique and show that it works well for pathology image classification. Further, we extract features from the trained deep model and build a classifier that shows good predictive accuracy for the chosen dataset. Later, we can use these features from histopathology (which are cheap to obtain) to predict the other modality – gene expression (expensive to obtain, but precise).

**Data Preparation**

Here, we use a large dataset Genotype-Tissue Expression (GTEx) that contains both gene expression and histopathological images of various tissues for relatively healthy people. All the data can be downloaded freely from [https://gtexportal.org](https://gtexportal.org/home/histologyPage), including the list of images with the labels, by pressing the button CSV (`GTEx Portal.csv` from <https://gtexportal.org/home/histologyPage>) and gene expression archive file (`GTEx\_Analysis\_2017-06-05\_v8\_RNASeQCv1.1.9\_gene\_tpm.gct` from <https://gtexportal.org/home/downloads/adult-gtex/bulk_tissue_expression>).

These data include bulk RNA sequencing (RNAseq) and standard eosin-hematoxylin (HE-stained) histological images for each sample across 40 types of non-diseased tissues. Gene data are in text, tab-separated format, whereas images are in Aperio scanner binary (SVS) files and require a special package (we used `*slideio*` Python package) or software (e.g. ImageScope) to be imported. Although the GTEx portal hosts 25,713 whole-slide images (WSI), only 13,797 of the samples have corresponding gene expression data (from 17,382). More details on the dataset can be seen on a Power BI dashboard (Fig. 1), which was created for a better understanding of data content.

|  |
| --- |
|  |
| Fig.1 – General information about the GTEx dataset |

We start with a certain number of slides per tissue (50), cut them into smaller images called tiles (or patches) with size 448 x 448, save them as `\*.jpg` files, and exclude empty and bad quality tiles (Fig. 2).

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Fig.2 – Examples of excluded tiles (filtering step): dirt and almost empty tiles | | | |

To do this, we have several stages: 1) filtering by image file size –a dictionary was created with tissue names as keys and its threshold file size as values, and all files with a smaller value were removed; 2) filtering half-empty tiles – every tile was divided into several pieces and checked standard deviation for each of them, and if it was less than the threshold value the tile was removed; 3) filtering dirt tiles: for every tile, the standard deviation for three color channels was checked, and if it was less than the threshold value the tile was removed.

The next step is to create a balanced sampling of tiles – 100 tiles per slide; this should result in 50\*100 = 5,000 tiles per tissue and 50\*40\*100 = 200,000 tiles in total. Some tissue types, however, were presented by only a few samples present (e.g. *kidney\_medulla* – 4, *cervix\_ecto* – 9, etc.). In such cases, the proportionally increased number of tiles was taken (for example, *kidney\_medulla* – 1,250 tiles from each of 4 samples to obtain 5,000 tiles per tissue). We got approximately 200,000 tiles from 1,803 samples across 301 individuals at this step. The GTEx sample class label is the sample tissue type.

**Gene expression data** should be preprocessed as well. In the downloaded file the expression of the genes is given in “transcripts per million” or TPM. This metric is based on the number of short RNA fragments of genes detected in the sample and is automatically normalized for gene length and total RNA abundance in the sample. The original file contains 56,200 gene features in rows (including protein-coding genes, non-coding genes, and non-annotated genes) detected in 17,382 samples in columns. TPM is a highly right-skewed measure, so it should be log-transformed to be used with standard data science tools (e.g. PCA). As recommended in the field, TPMs were log-transformed by the formula *logTPM = log2(1 + TPM)*.

The majority of the genes in the dataset are of limited interest as they are either lowly expressed or non-variable. Therefore, we used 3-step filtration: 1) lowly expressed genes were removed – we kept genes that show reasonable expression (*logTPM > 5*) in at least one sample (Fig. 3*a*); 2) we kept only variable genes – standard deviation above 0.5 (Fig. 2*b*); 3) we excluded genes with non-standard or repeated gene names, as those are either non-coding or unknown genes. This reduced the number of genes from 56,200 to only 17,408 informative ones (Fig.3*c*).

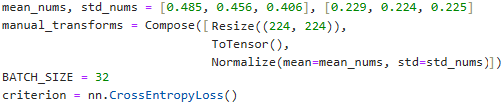
|  |  |  |
| --- | --- | --- |
| *a* | *b* | *c* |
| Fig. 3 – Filtration of RNAseq data: *a* – distribution of maximum gene expression values (the vertical line shows the filtration threshold); *b* – distribution of standard deviation values;  *c* – the distribution of original data (black) and selected data (red). | | |

|  |
| --- |
|  |
| Fig. 4 – Comparison of training time for different CNN |

**Deep Learning Model**

Deep learning stage was realized with PyTorch and base neural network architectures. We trained several torchvision models: EfficientNet\_B0, ResNet50, DenseNet121, and ConvNeXt – to select the optimal for our task. The shortest training time was observed for the model EfficientNet\_B0 (Fig. 4), and it also turned out to be optimal for further research. So, in all cases, if the model is not specified, then ***EfficientNet\_B0*** is used.

*Some details about input train preparation* (Fig. 5)



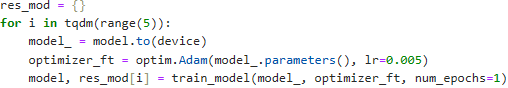
|  |
| --- |
|  |
| Fig. 5 – Standard batch with tissue tiles |

*Some details about dataloaders*

During several trial runs, significant differences were observed in the results obtained: in some cases, the accuracy at the end of the first epoch varied from 3 % to 70%. To avoid this, the generated `*dataloaders*` were saved and downloaded every training cycle to be the same.

Also, some additional tests were executed to check the reproducibility of the results when restarting the training:

1. mode `*restart*`:



1. mode `*at once*`:



|  |
| --- |
|  |
| Fig. 6 – Comparison of different executing modes |

As expected, there were no significant differences observed, though the results for `*restart*` mode were a little better (Fig. 6).

The investigation of the best learning rate for our case also was done (Fig. 7), and it resulted in the value **lr = 0.0001**, which was used for all other training cycles.

|  |
| --- |
|  |
| Fig. 7 – Training results for 5 epochs with different learning rates  (*var* – training with the changeable learning rate) |

The investigation of different architectures showed that all of the above models give similar results, which differ not so much (Fig. 8). Among these, EfficientNet emerged as the superior model, demonstrating optimal performance in the shortest training time.

|  |
| --- |
|  |
| Fig. 8 – Training results for different neural networks (5 epochs) |

The evaluation of the model's efficacy was conducted through the analysis of a confusion matrix, which revealed a high degree of accuracy with only a handful of exceptions in tissue classification (Fig. 9).

|  |
| --- |
|  |
| Fig. 9 – Confusion matrix for EfficientNet\_B0 |

A notable observation from the confusion matrix was the recurrent misclassification between *subcutaneous adipose tissue*and*breast tissue*. This phenomenon can be explained by the inherent histological similarities between these tissues, also due to the inclusion of male patients in the dataset. Male breast tissue, being predominantly adipose, is the same histopathologically as subcutaneous adipose tissue, thereby leading to occasional classification overlaps.

Another pair of tissue types that presented classification challenges were *esophagus\_gastro*and*esophagus\_mucosa*. The difficulty in distinguishing between these types stems from their closely related histological features and spatial continuity within the esophagus. Both tissues are part of the esophageal lining, with subtle differences that can be challenging to notice without the context of their precise anatomical location, making their differentiation challenging.

Furthermore, the classification of *skin* tiles as exposed or non-exposed to the sun revealed a significant number of misclassifications. The primary distinction between these two categories lies in the epidermal layer, which is directly affected by sun exposure. However, the histological features of the subcutaneous layers remain unchanged regardless of sun exposure.

The confusion matrixes for other models demonstrate the same results (Fig. 10).

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **ResNet50** | **DenseNet121** | **ConvNeXt** |
| Fig. 10 – Confusion matrixes for different models | | |

The ability of deep learning models to classify various tissue types highlights their potential to improve diagnostic accuracy and our understanding of tissue-specific characteristics. Future improvements in tissue classification may involve refining models, integrating more contextual information or adding a different modality.

Interestingly, analysis of different models shows that initially, test results outperform training results, but this reverses after 2-4 epochs of training (Fig. 11).

|  |
| --- |
|  |
| Fig. 11 – More detailed results for different models (5 epochs) |

|  |
| --- |
|  |
| Fig. 12 – Results for EfficientNet\_B0 (30 epochs) |

The question arises: can we consider that overfitting starts from the moment when the train results exceed the test and whether it makes sense to continue training if the results are already quite good?

Continued training of the EfficientNet\_B0 shows that the training accuracy is gradually increasing to 98.6 % by the 30th epoch (Fig. 12), and the testing accuracy fluctuates around 88 % after the 5th epoch (has reached a plateau).

**Image Embeddings**

Exploring embeddings from deep learning networks at different training stages shows a complex relationship between model accuracy and embedding quality.

To visualize 1280-element embedding vectors (outputs of the EfficientNet\_B0), we employed dimensionality reduction techniques, including PCA (Principal Component Analysis) retaining 40 components, followed by non-linear data transformations UMAP (Uniform Manifold Approximation and Projection) and t-SNE (t-distributed Stochastic Neighbor Embedding). These techniques transform the high-dimensional embedding vectors into lower-dimensional representations, enabling visualization of the data in a two-dimensional space. UMAP is particularly effective in preserving local and global structures, bringing similar objects close together in the embedding space.

Our original findings suggest that despite training the network for a shorter duration (5 epochs), the accuracy of the model is only marginally lower compared to models trained for longer durations (30 epochs) (Table. 1).

Table. 1 – Results for models trained for different number of epochs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Epochs** | **5** | **10** | **20** | **30** |
| Accuracy, Train (%) | **89.79** | **95.38** | 97.96 | 98.62 |
| Accuracy, Test (%) | 88.12 | 87.76 | 88.39 | 88.32 |
| Loss, Train | 0.30 | 0.13 | 0.06 | 0.04 |
| Loss, Test | 0.37 | 0.44 | 0.54 | 0.58 |

However, the visualizations obtained using UMAP reveal notable differences between early-stage and advanced-stage training (Fig. 13).

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
| **5 epochs** | **10 epochs** | **20 epochs** | **30 epochs** |
| Fig. 13 – Tile embedding representation using UMAP (1st row) and t-SNE (2nd row) | | | |

In early-stage training, the tissue representations appear undifferentiated and lack clear clustering patterns. Conversely, after advanced training, the tissue clusters become much more defined and distinct, indicating that the model has learned more discriminative features and is capable of capturing finer tissue characteristics.

Thus, despite potential overtraining, networks that perform better on the training set can extract more relevant embeddings. Here, we should note that overlaps in tissue representations are reasonable due to common histological structures like adipose tissue, blood vessels, and fibrotic tissue across organs.

Further improvements could be achieved if we switch from tile embeddings to whole slide image embeddings, – this was done by averaging embeddings from all tiles of every single slide (Fig. 14).

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
| **5 epochs** | **10 epochs** | **20 epochs** | **30 epochs** |
| Fig. 14 – Slide embedding representation using UMAP (1st row) and t-SNE (2nd row) | | | |

In these cases, we observe a similar trend, but clusters are formed more quickly and clearly – almost all tissues are separated after 10 epochs, and the complete separation is already after 20 epochs (Fig. 15, 16 – more detailed enlarged view).

|  |
| --- |
|  |
| Fig. 15 – Slide embedding representation after 20 epochs using UMAP |

|  |
| --- |
|  |
| Fig. 16 – Slide embedding representation after 20 epochs using t-SNE |

**Integration with Gene Expression**

In managing the high dimensionality of gene expression data (17,408 genes after filtering), researchers often use two main strategies: dimensional reduction and targeted gene-by-gene analysis. Each approach offers distinct advantages depending on the specific goals of the study.

In the first case, as in the case of image embeddings, UMAP or t-SNE can help visualize principal components (as depicted in Fig. 17, 18), which closely resemble slide embeddings demonstrated above (Fig. 14, 5 epochs).

|  |
| --- |
|  |
| Fig. 17 – UMAP representation of gene expression data |

|  |
| --- |
|  |
| Fig. 18 – t-SNE representation of gene expression data |

These views provide a comprehensive overview of the gene expression landscape, helping to solve problems such as tissue classification and pattern recognition.

In the second case, targeted gene-by-gene analysis offers a more focused approach, allowing researchers to predict the expression levels of certain genes based on the image data. This approach provides a more granular understanding of gene expression dynamics and enables the identification of key molecular players involved in biological processes.

The choice between dimensional reduction for a broad gene expression overview and predicting individual gene expressions depends on the specific research objectives. While dimensional reduction offers a broader perspective suited for macroscopic analyses and tissue classification, targeted gene-by-gene analysis provides deeper insights into the intricate molecular mechanisms underlying tissue-specific functions and pathological conditions.

Now, we can build a united view of the data. To this end, we can concatenate PCA components of slide embeddings and PCA components of gene expression data. The combined vector could then be represented as a unified UMAP (Fig. 19) and t-SNE (Fig. 20).

|  |
| --- |
|  |
| Fig. 19 – Unified UMAP representation of genes and images |

|  |
| --- |
|  |
| Fig. 20 – Unified t-SNE representation of genes and images |

These unified results also show fairly well-distinguished clusters. Thus, one may hope that the combination of molecular and imaging data indeed improves the sensitivity of the analysis (ability to distinguish otherwise similar tissues).

Another application of such integration could be a prediction of gene expression using histopathological images. This could be done, for example, by building a linear or non-linear regression model for each gene. This model would take all image embeddings as independent variables and predict gene expression. Measures like R2 between predicted and real gene expression values could be used to select the top recoverable genes.

The other approach to gene expression prediction would be presenting each cluster of samples on UMAP with a single (averaged) gene expression vector. As soon as a new image is clustered with one of the clusters of the training set, its genomic modality would be estimated by the “average” gene vector. This method may be less precise compared to the previous but it can help when the entire transcriptome must be recovered for the analysis of biological processes.

**Conclusion**

In this study, we leveraged deep learning methods to analyze histopathological images and combine the results with gene expression data from the GTEx dataset. We considered RNA sequencing and HE-stained images across 40 types of non-diseased tissues.

Employing models like EfficientNet\_B0, we achieved significant accuracy in tissue classification, despite challenges like the misclassification between similar tissue types. Our analysis suggests that with further training, models not only improve in accuracy in the training dataset but also in extracting relevant embeddings, as evidenced by clearer tissue clustering in UMAP visualizations. Interestingly, other works show that long training may increase performance not only in training but also in testing sets (so-called grokking of the neural network).

We explored integrating histopathological and gene expression data, employing strategies like dimensional reduction and *gene prediction models*. This integration revealed distinct clusters, enhancing our ability to differentiate tissues and potentially improving diagnostic sensitivity if applied to ill patients. The multimodal approach in the future will improve medical diagnostics and personalized medicine. We emphasize the importance of deep learning in extracting meaningful insights from complex biomedical data.